This work is funded by the ERDF (European Regional Development Fund) through the COMPETE Programme and by the Portuguese Government through the FCT (Portuguese Foundation for Science and Technology) within the doctoral grant SFRH/BD/70248/2010.

Ricardo Manuel Soares Anacleto **All Over the Place Localization System** UMinho | 2015

Universidade do Minho
Escola de Engenharia

Ricardo Manuel Soares Anacleto

All Over the Place Localization System

The MAP-i Doctoral Programme in Informatics, of the Universities of Minho, Aveiro and Porto

universidade de aveiro
Universidade do Minho
U.PORTO

June 2015

Universidade do Minho
Escola de Engenharia

Ricardo Manuel Soares Anacleto

All Over the Place Localization System

The MAP-i Doctoral Programme in Informatics, of the Universities of Minho, Aveiro and Porto

Universidade do Minho

supervisors:
Professor Doutor Paulo Jorge Freitas de Oliveira Novais
Professor Doutor Lino Manuel Baptista Figueiredo
Professora Doutora Ana Maria Neves Almeida Baptista Figueiredo

June 2015

STATEMENT OF INTEGRITY

I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of the thesis elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

University of Minho, 30/06/2015

Full name: Ricardo Manuel Soares Anacleto

Signature: Ricardo Manuel Soares Anacleto

Everything should be made as simple as possible, but not simpler.
— Albert Einstein

ACKNOWLEDGMENTS

The journey of the PhD research was a long process of acquiring new skills and reaching for answers in many different areas. Such a journey could never have lead to the research presented in this thesis without the gracious support and collaboration of several people. Here I would like to acknowledge those who had a significant impact on it.

I would first like to thank Lino Figueiredo, Paulo Novais and Ana Almeida for their support, advice and dedication on helping me reaching this stage of my academic life. To Lino Figueiredo, who suggested the subject of this thesis, for his guidance and encouragement during my pursue of this degree. His constant availability for receiving me, even when his schedule was tight, was priceless to me. Our discussions and his attention to detail were valuable factors for the correct elaboration of this thesis. To Paulo Novais for his constant motivational and comforting words. His constant availability to help me on surpassing each obstacle encountered during this journey, was valuable to me. Also, thank you for your support and for always pushing me to go further. To Ana Almeida who never accepted anything less than my best.

I must also mention ISLab, a research lab of Algoritmi Research Center, and GECAD for providing the equipment and conditions needed for the correct development of this work.

I would also like to thank all the friends and colleagues that were present during this journey. Specially Nuno Luz who is not only a work partner, but a good friend and as given precious suggestions and discussions about my work that lead to good research opportunities. I am also especially grateful to António Meireles and Rui Terra, who, besides sharing this journey with me (or part of it), supported my work with their knowledge on electronics, by designing, assembling and supporting the hardware used in this work. Without them this work would not be possible.

I would be remiss without extending my most sincere thanks to my parents and brother, Manuel, Madalena and Pedro. This work is once again a product of the love and dedication that they have shown to me throughout my years.

Finally, I would like to thank the most important person in the world to me, Daniela Rocha. Although she does not have a deep knowledge in this area, has supported me, bringing to me happiness and companion during this long academic endeavour. Thank you for all that you are and all that you have allowed me to become.

RESUMO

A localização é normalmente obtida utilizando um sistema de navegação baseado num ambiente estruturado. No entanto, estes sistemas não funcionam ou são difíceis de serem implantados em ambientes densos. Assim, considerando que as pessoas se deslocam geralmente a pé, neste trabalho é proposto um Sistema de Navigação Inercial para Pedestres (PINS).

Nesta tese são identificadas as principais vantagens e desvantagens dos PINS, bem como, os algoritmos que estão na base destes sistemas. O objetivo é fornecer uma perspectiva abrangente sobre o que é necessário para desenvolver um PINS e quais os problemas encontrados mais frequentemente durante o seu desenvolvimento. São também identificados e comparados os sistemas e tecnologias mais importantes da literatura.

Duas unidades de medição inercial foram desenvolvidas, sendo que os sensores inerciais foram combinados com sensores de força para melhorar a detecção das diferentes fases (fase de apoio e fase de balanço) da marcha humana, assim como, para ter uma informação mais precisa sobre a força de contacto. É muito importante que a fase de apoio seja devidamente detectada. Assim três diferentes algoritmos, utilizando diferentes sensores e métodos de fusão sensorial, são explicados e avaliados.

A marcha humana representa um padrão que é repetido ao longo do tempo, o qual é aprendido utilizando algoritmos de aprendizagem com base nos dados obtidos pelas diferentes fontes de informação para realizar uma caracterização do passo. Esta caracterização leva a uma melhoria no desempenho do sistema, uma vez que os erros sistemáticos podem ser aprendidos, para depois serem corrigidos em tempo real. Como neste sistema existe mais do que uma fonte de informação, além das técnicas de fusão sensorial, são também aplicadas técnicas de fusão de informação.

Depois dos dados serem obtidos com o equipamento desenvolvido, e do passo ser caracterizado com os dados aprendidos, são aplicados os algoritmos que fazem a estimativa do deslocamento. A arquitetura proposta é avaliada em quatro cenários de utilização real, dentro de um edifício, envolvendo diferentes tipos de caminhadas. Esta arquitectura levou a uma melhoria significativa da precisão da estimativa do deslocamento.

ABSTRACT

Nowadays location information is typically obtained using a navigation system based on a structured environment. However, these systems do not work or are very difficult to be deployed in dense environments. Thus, considering that persons are usually on foot, in this work is proposed a Pedestrian Inertial Navigation System (PINS).

In this thesis are identified the main advantages/disadvantages about PINS, as well as, the algorithms that are the base of this type of systems. It is provided a good insight about what is necessary to create a PINS and the problems that are encountered during its development. To complement these insights the fundamentals about Human Gait are presented, along with the main sensor and information fusion strategies used in this type of system. Also, the most important systems and technologies are identified and compared.

Two inertial measurement units were developed, where the inertial sensors were combined with force sensors to improve the detection of different phases (stance and swing phase) of the human gait, as well as, to have proper information about the contact force. The stance phase is very important to be properly detected, therefore, three different algorithms using different sensors and sensor fusion methods are explained and evaluated.

The human gait cycle represents a pattern that is a repeatable over time. Thus, this pattern is learned using machine learning algorithms, which are applied to the data obtained from the different data sources to perform a step characterization. This characterization leads to an improvement on the system's performance, since the systematic errors can be learned to then be corrected in real-time. Since there is more than one source of information, besides sensor fusion techniques, it was also implemented an information fusion strategy.

After collecting the data with the developed hardware and characterize the step according to the learned data, it is demonstrated the developed displacement estimation architecture. The proposed architecture and algorithms are evaluated through four real use case scenarios in a typical indoor environment involving different types of walking paths. This architecture led to a significant improvement on the displacement estimation accuracy.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI Artificial Intelligence

API Application Programming Interface

BCU Body Central Unit

BSN Body Sensor Network

BSU Body Sensor Unit

CPU Central Processing Unit

DR Dead Reckoning

DTW Dynamic Time Warping

ECG Electrocardiography

EDR Enhanced Data Rate

EKF Extended Kalman Filter

EMG ElectroMyoGraphy

FIFO First In First Out

GNSS Global Navigation Satellite System

GPS Global Positioning System

HDR Heuristic Heading Reduction

HLIS High Level Integration Software

I^2C Inter-Integrated Circuit

IKF Interval Kalman Filter

IMU Inertial Measurement Unit

INS Inertial Navigation System

KF Kalman Filter

LLIS Low Level Integration Software

MEMS Microelectromechanical systems

ODR Output Data Rate

PC Personal Computer

PDR Pedestrian Dead Reckoning

PINS Pedestrian Inertial Navigation System

PLASYS All Over the Place Localization System

PSiS Personalized Sightseeing Tours Recommendation System

RFID Radio-frequency identification

SVM Support Vector Machine

UKF Unscented Kalman Filter

UART Universal Asynchronous Receiver Transmitter

USART Universal Synchronous Asynchronous Receiver Transmitter

ZARU Zero Angular Rate Update

ZUPT Zero Velocity Update

ZVD Zero Velocity Detector

1

INTRODUCTION

Location information is an important source of context for ubiquitous systems. Using this information, applications can provide richer, more productive and more rewarding user experiences [98]. For example, the ability to locate an individual or an object can be exploited, among others, to help or to assist in decision-making [94].

This means that there are some applications where a system that provides location everywhere is essential:

- Emergency teams - most of these teams are controlled by a commander and it is very important for him to know the position of each one of his team members, in order to send them to where they are needed. Some examples are: fire fighters, military forces [35] and medics [84];
- Tourism - tourism recommendation systems [71] [10] [74] can give better recommendations and more efficient tour plans if tourists' indoor location is known;
- Patient monitoring - there is an increasing need for localization systems to track people with special needs (e.g. the elderly [97] [49] [37] [88]);
- Children - nowadays people are always running from one side to another, and sometimes they lost their sons from their point of view (e.g. in shopping malls [5], carnivals). With a localization (tracking) system, parents can easily find their sons;
- Factories - when an accident occurs in a factory or in a building and the first safety teams arrive to the site, an indoor localization system can be used to monitor the persons which are trapped inside [82].

The major limitation of these systems is related to retrieving individual's location, which nowadays is only based on a Global Navigation Satellite System (GNSS), restricting the use of these systems to environments where GNSS signals are available [3]. However, GNSS signal is not available inside buildings, in urban canyons, in the underground, underwater and in dense forests. Consequently location-aware applications sometimes cannot know the user location. Therefore, developing complementary local-

ization technologies for these environments would unleash the use of many applications as presented above.

To suppress this limitation, there are already some proposed systems that retrieve indoor location with good accuracy. One of the first indoor localization systems was based on electromagnetic sensing [96]. After this one, many other approaches have been developed based on smart floor [90], infra-red [117], Wi-Fi [27], ultrasound [15] and many other technologies [51]. Also, important technology companies, like Google [78] and Microsoft [29], are doing research on indoor localization systems.

However, most of these solutions force the existence of a structured environment which make them difficult to be deployed over large buildings (with very expensive implementation costs), because it needs a location-fingerprint approach that is labor-intensive and vulnerable to environmental changes. These can be possible solutions to use when GNSS signals are not available, but only indoors, since in a dense forest this kind of systems does not exist or work.

To suppress structured environment limitations, Pedestrian Inertial Navigation System (PINS) are being studied, which will be discussed in Section 3. Typically, an Inertial Navigation System (INS) uses a computer, motion sensors (accelerometers), rotation sensors (gyroscopes), among other sensors, to continuously calculate via Dead Reckoning (DR) the position and orientation of a moving object.

DR has been important throughout history. Sailors were one of the first to use this technique. It involved combining compass heading with knowledge of sea currents and the speed of the vessel measured by the time taken by an object thrown overboard to travel a fixed length along the side of the ship. This was the technique used by Columbus in his voyages to discover the "New World" [99].

The advantage of an INS is that it does not require, once initialized, external references in order to determine position, orientation and velocity. It is used on vehicles/objects such as ships, aircraft, submarines, guided missiles and spacecraft. Recent advances in the construction of Microelectromechanical systems (MEMS) have made possible to manufacture smaller and lighter INSs. These advances have widened the range of possible applications to include areas such as human and animal position tracking.

However, DR approaches rely on the knowledge of an initial location and suffer from error accumulation, as will be presented on Chapter 2.

With a PINS, individual movements information can be obtained independently of the building infrastructure. Some authors [124] [101] [62] propose PINSs assisted by

Wi-Fi, Radio-frequency identification (RFID), among others. to improve the accuracy of this type of systems.

Other authors propose PINSs assisted by a particle filter [14], which is a probabilistically algorithm that describes, in a recursive way, the evolution of a state variable over time. However, these implementations need to have a map of the building where they are being used.

The main problem of these hybrid approaches is that they still need an implemented infrastructure to work, and do not work on dense environments.

1.1 PROBLEM STATEMENT AND MOTIVATIONS

As persons usually are on foot, an INS system based on Pedestrian Dead Reckoning (PDR) technique is an appropriate solution to be used. These systems can be called as PINS, which are typically composed by accelerometers, gyroscopes and other type of sensors placed in human body to detect the body movements. The accelerometer and the gyroscope data are mathematically integrated to provide changes in distance and orientation. Given an initial location, this change in distance and orientation is used to compute the pedestrian's current location. Typically the acceleration data sensed by the accelerometers and the angular orientation sensed by the gyroscopes are collected at a high rate to include the signal frequencies induced by a walking of a pedestrian.

However, errors in inertial sensors measurements can produce huge navigation errors. To avoid or to limit these errors, some approaches use fixed measurements. One example is the limitation of the maximum velocity that a person can perform, thus limiting the velocity error growth and slowing the rate of position error grow. Typically to limit these errors, huge and power-hungry high-quality accelerometers and gyroscopes, like the ones used in ships or in airplanes, are used, but they are infeasible to a person use/wear in a normal walk.

One way to make a PINS less intrusive is by using MEMS. However, while MEMS offer significant advantages over traditional units in terms of cost, size, weight and power consumption, at present they still present to many errors on the measurements [39]. Due to these errors PINS are not much used, since nowadays there is not a viable solution to retrieve location using MEMS inertial sensors.

This research wants to prove if by complementing inertial sensors with other types of sensors and techniques, namely force sensors on the feet and several sensors spread in the pedestrian's body combined with learning algorithms, that learn the walking

characteristics and respective errors, the PINS accuracy can be improved, in order to make it more viable to use by pedestrians. Creating a localization system capable of provide location everywhere, independently of the environment and based only in sensors that are placed in the human body.

The motivation for this project comes from the Personalized Sightseeing Tours Recommendation System (PSiS) [73] project, developed in GECAD (Knowledge Engineering and Decision Support Group) and in ISLAB (Intelligent Systems Lab).

PSiS [71] is a tour planning support system, that aims to define and adapt a visit plan combining, in a tour, the most adequate tourism products, namely interesting places to visit, attractions, restaurants and accommodations, according to tourist's specific profile, which includes interests, personal values, wishes, constraints and disabilities, and available transportation modes between the selected products [72]. Points of interest and transportation schedules are considered to generate a tour plan when a tourist wants to visit a region.

PSiS Mobile [10], which is integrated into PSiS project, consists on a mobile tool to assist a tourist on the "field". The main objective of PSiS Mobile is to help a tourist after he plan a trip for his vacations on PSiS web application. It presents a list of recommended points of interest to see on a specific location. It is also possible to do a rescheduling of the original planning, according to user's and point of interest context (time, location and weather), but to do this user's current location must be known. If user's indoor and outdoor location can be known the system can track the user more accurately and improve the PSiS capabilities.

Resuming, there is the need to integrate PSiS Mobile with a system that can provide indoor location, since it already can retrieve the outdoor location. With indoor location, it will be also possible to recommend artwork inside a gallery according to tourist's profile.

1.2 HYPOTHESIS AND OBJECTIVES

This project, aims to study and create a PINS that combined with Global Positioning System (GPS) provides location everywhere, including on dense and indoor environments. However, as stated earlier, there are several research issues on developing a PINS, especially on the ones based on low-cost MEMS sensors because of sensor errors.

This is the main problem/challenge of the whole project, how to reduce the impact of the sensors errors on location estimation. It raises several research questions and establishes multiple requirements.

Thus, the overarching goal of this thesis is to investigate if:

The information obtained from inertial sensors, and additional sensors spread along the human body, can be fused and combined with learning algorithms to learn the human gait characteristics, to improve the pedestrian walking path estimation.

RESEARCH QUESTION NO.1: *Can the inclusion of new type of sensors and techniques improve the step detection accuracy?*

Typically a step is detected using an accelerometer placed in the person's foot. However, accelerometers are subject to various sources of errors and, therefore, the step detection results sometimes are not accurate enough. The step detection needs to be very accurate, because to estimate the pedestrian displacement, with the minimum error, the integration of the acceleration values most be only performed during the swing of legs. Meaning that the velocity errors can be reset at each step since true velocity must be zero if the pedestrian foot is stationary.

In this research work force sensors and other techniques will be added/implemented to improve step detection, since the force sensors provide accurate information to determine when the user puts his feet on the ground, as well as, the respective contact force. The contact force is important to classify the type of walking (e.g. running, normal walking, slow walking) that the pedestrian is performing. Thus, combining force sensors with the accelerometer data (to get the step acceleration) a more exact detection of the step can be achieved.

After developing different algorithms, that perform the step detection, they will be compared to the typical PINS approach that uses only an accelerometer to detect steps.

RESEARCH QUESTION NO.2: *Can learning algorithms be used to learn pedestrian step patterns and its errors in order to correct, in real-time, the inertial sensors estimations?*

In order to reduce errors provided by MEMS, learning algorithms will be developed. These algorithms will learn the walking/motion characteristics to correct, in real-time, the data gathered by the sensors. It is intended that the data aggregation from all the

sensors for a specific period of time will be done at the Body Central Unit (BCU)[1], and the data correction and the presentation of the user current location will be done on a mobile device.

The learning algorithms should be trained based on a set of exercises or by using GPS data. In the exercises environment or when the user is in environments where the GPS signal is very good, the system self-learns how the user steps are performed and the gaps (errors) that exist on the estimated step. So, when the pedestrian is indoors or in a dense environment, the learned data will be used to, in real-time, improve the PINS accuracy.

RESEARCH QUESTION NO.3: *Can a low-cost PINS provide good accuracy when used by a pedestrian?*

To make this system really low-cost, the sensor modules will be developed using inertial sensors with a price tag around €10 each. The main objective is that the system will cost less than €100, turning it really low-cost compared to other so-called "low-cost" systems that cost more than a thousand euros.

Two BSU[2] should be developed to be distributed in the lower limbs, legs and hip area, to collect data. The sensors will be developed based on smart sensors philosophy, connected to an integrated circuitry module to pre-process and codify the collected signal to send it to a BCU. The BCU must be capable to interact with all sensors around the body, to handle the calculations to determine, in real-time, the user location and to exchange data with a mobile device.

An important consideration that must be introduced is a noise reduction algorithm. This process will be useful to remove noise from a signal that is given by a sensor. In this case, noise is something that is captured from the sensor, which can influence, negatively, the walking estimation. This algorithm must decompose the signal and extract only the frequencies bands that are important to the PINS, in order to lighten the data processing, thus reducing the necessity for high computational power.

1 BCU is a sensor module that aggregates the data received from several Body Sensor Unit (BSU). Normally it stores and/or forwards this information. It can also act as a BSU.

2 A BSU is able to sample the human body movements through the sensors incorporated on it, and convert it into digital data and transmit it to a central node.

The interconnection of the modules will be done through a Body Sensor Network (BSN)[3], implemented with a ZigBee[4] or a Bluetooth[5] based protocol.

The inclusion of more than one BSU on human body will permit the usage of different algorithms (techniques) in each module to detect the step, and then, combine this information to have a better perception about the type of step that was taken by the user.

However, an optimal BSU position on the person's body must be found in order to obtain more accurate data to properly recognize the actual person's activity. Also, because of people different sizes, the ideal sensors position to one person can be different to another.

A problem that can occur in this project is the delay between the real location and the processed one (that appears to the user on the mobile device). As represented in Figure 1 the process is complex: it starts on the sensors that produce the measurements, which are afterwards sent to the BCU. This process has already communication delays. When the BCU has the sensors values, it must combine them according to the data timestamps to process the sensor and information fusion and, therefore, estimate the walking path, which takes some processing time. Afterwards, the learning algorithms are executed to correct the estimated walking path. All of these algorithms can run for a large amount of time, thus they should be efficient to process the data in a short period of time, in order to minimize this delay.

SUCCESS CRITERIA

The goal of this project is to demonstrate that it can improve the PINS accuracy, so the implemented algorithms will be compared with the traditional PINS approach, which uses a foot BSU composed by an accelerometer and a gyroscope, to detect the steps and, therefore, estimate the pedestrian displacement. Unfortunately, does not exist a standard to compare this type of systems. The scenarios provided by each researcher are very specific and different from each other, thus it is impossible to directly compare the implemented system with others.

There will be two quantifiable success criteria to define the project's success degree. The first criteria is regarding the estimated location accuracy that must be higher than

3 BSN is a wireless network used on wearable computing devices.

4 Zigbee is a specification for a high-level communication protocol used to create personal area networks built from small and low-power digital sensors.

5 Bluetooth is a wireless technology standard used by two (or more) devices for communication over short distances. The main advantages of Bluetooth are that it is inexpensive and easy to use.

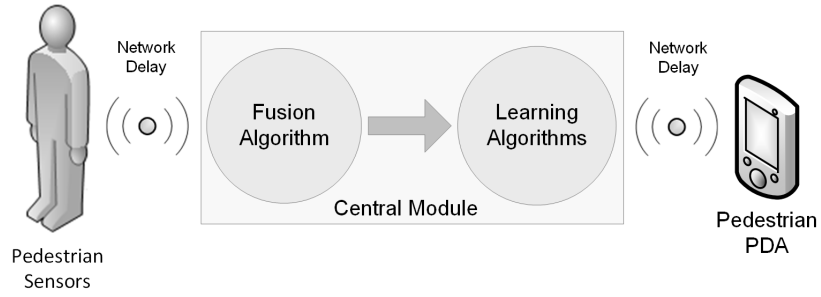


Figure 1: Location estimation process flow

90%[6] or in other words, per each 100 meters walked the system must have an error lower than 10 meters. Also, the accumulated error must be as low as possible.

The second criteria, is regarding the delay between the measurements and the presentation of the current user location, that must be less than 2 seconds[7]. More than that period is not considered real-time, as this system aims to be.

If the project completes these two criteria with success and if it could be integrated with some real ambient assisted applications (e.g. tourism, healthcare) it can be said that the project objectives were achieved.

1.3 OVERVIEW

The aim of this thesis is to develop a PINS based on MEMS and on learning algorithms, which learn the human gait to perform real-time corrections on the estimated walking path. This goal is addressed throughout the document, which is structured as follows:

- Chapter 2 gives an insight about INSs, identifying the most common techniques and technologies, as well as, their strengths and weaknesses;
- Chapter 3 outlines the current state of the art, where it is presented an overview over recent and important PINS available in the literature;
- Chapter 4 describes the architecture of the developed hardware. It presents the Central Processing Unit (CPU), the sensors, the communication modules and the software used to connect all these pieces;

6 As will be seen in the analysis of the state of the art this is the mean accuracy of this type of systems, thus it is expected to have at least this amount of accuracy.

7 The GPS modules of mobile devices work at 1Hz, thus the proposed PINS should present a similar frequency.

- Chapter 5 introduces the step detection algorithms, where the first one is based on the approach typically used by PINS, the second one is the proposed approach that includes force sensors on the feet, and the third one is based on the data given by the pelvic rotation. The implemented approaches are all evaluated in order to verify which have the best results, and the correspondent magnitude of improvement;
- Chapter 6 provides a description of the proposed learning approach. After a step being detected, it is characterized according to the current person walking characteristics. This characterization is done by retrieving some features from the several sources of data, which are then feed into learning algorithms that characterize the step. All the implemented approaches are evaluated, as well as, some different learning algorithms, to verify which is more feasible to be used in PINS;
- Chapter 7 presents the implemented orientation algorithm, as well as, the two implemented displacement estimation algorithms. Then, the implemented approaches are compared and the results analysed;
- Chapter 8 concludes with a summary of the contributions, a discussion about the obtained results and future work directions.

2

FUNDAMENTALS OF INERTIAL NAVIGATION

In this chapter it will be presented the fundamentals about INSs. The main techniques and algorithms used in this area of research will be introduced and discussed, as well as, some insights about how the human gait works (Section 2.1). The techniques and algorithms include how the distance (Section 2.2) and the orientation (Section 2.3) are estimated, and what types of information fusion (Section 2.4) can be used.

A typical scenario of a DR application is a ship travelling at a constant speed in a fixed direction. The ship's current location is on a line starting at the known starting point in the direction of travel, and the distance from the starting point is given by the speed along the direction of travel multiplied by the time since the ship was at that known point. As given by the equations of motion (Equation 1).

$$s = s_0 + vt \tag{1}$$

where s represents the position to be estimated, s_0 the previous position, v the current velocity and t the time interval.

If the ship changes the direction or the speed of travel, the navigator must note the current location estimation and start the process again using this new location as the starting point. Over time, these estimations become less accurate, because they rely on previous estimates, which are imperfect due to errors in speed and heading measurements.

A PINS uses the PDR technique [19] to estimate step length and, therefore, the walking distance, using inertial, or other type, of sensors. Unfortunately, a module working only with PDR is not able to ensure that the geographical location is accurate within a few meters. As result, location and orientation errors tend to grow unbounded. In fact, although these deviations may be small for every millisecond, the location error caused by a sustainable use of the system can exceed one meter in 10 seconds [118]. These estimation errors often occur during starts, stops, sharp curves and walking on inclines.

PDR positioning accuracy normally ranges from 0.5% to 10% of the total walking distance, but these numbers strongly depend on the algorithm implemented, on the inertial sensor technology and on the evaluation environment.

PINS can provide both two-dimension and three-dimension localization. However, errors in Z-direction (altitude) are usually higher. Actually the location errors are strongly coupled with the heading errors via the true (relative) location. A heading error of 0.5° gives a relative position error of 1% of the walked distance. However, if the user walks back the same distance, the location errors are cancelled out.

An INS is composed by one or more Inertial Measurement Unit (IMU). Typically an IMU contains a three axis gyroscope's and accelerometer's, which report angular velocity and acceleration respectively. For portable applications, like the ones already presented on Chapter 1, IMUs with small dimensions are essential. High-performance devices, such as the NG LITEF [46] IMU which is designed for civil and military aviation, are equipped with hermetically sealed cases and fiber-optic gyroscopes that provide low-noise, high-stability and a low-drift factor. However, due to their price, size, weight or power consumption, this type of IMUs are not suitable to be used by mobile applications in small vehicles, clothes or shoes.

MEMS are composed by mechanical devices with related electronics and sometimes with optical and fluid elements. Since they are inexpensive, small and lightweight, they can be easily engineered to fit into existing clothes and shoes. However, they are not yet able to provide an accuracy like the bigger and expensive sensors, since they suffer from significant bias and noise [97].

Besides the high-performance devices mentioned above, smaller IMUs, such as Xsens MTi [125], Sparkfun Electronics Razor IMU [110] and X-IMU [113], are commercially available. However, the smaller size comes with reduced sensor data performance and accuracy. Whereas the Xsens MTi provides orientation estimates from onboard sensor data fusion and compensated raw sensor data, most small-sized IMUs provide only uncompensated raw sensor data. Therefore, the Xsens MTi is the most used, small-sized inertial sensor, in research works [52].

Sensors based on MEMS technology can be accelerometers, magnetometers, gyroscopes or barometers. Accelerometers measure the linear acceleration of the system in the inertial reference frame, meaning that measurements are relative to the moving system. As the accelerometers are usually fixed to the system and rotate with the system, they are not aware of their own orientation [91].

Gyroscopes measure the angular velocity of the system in the inertial reference frame. The system's current orientation is known at all times by using the original orientation of the system in the inertial reference frame as the initial condition and integrating the angular velocity to obtain the follow orientations [91].

Speed and direction of a person can be calculated by complementing the accelerometer and gyroscope data. However, for accurate results the sensors output must have a minimum drift. Especially, the drift of the gyroscope is crucial since it cannot be reset during the measurement phase. Also, errors in the accelerometer and gyroscope measurements are mathematically integrated in the navigation algorithm and produce increasing navigation errors. To avoid or to limit these errors, some approaches use fixed measurements to calibrate/limit the system.

A barometer is typically used to obtain altitude information. It measures atmospheric pressure and the altitude is determined according to the sea level atmospheric pressure. This pressure decreases at altitudes above sea level and increases below sea level.

Magnetometers (magnetic compasses) are used to obtain the north direction and then estimate the moving direction. However, they are not the best choice for urban navigation because they are subject to strong magnetic disturbance such as power lines, computers, and various metal/steel objects and structures. Those disturbances, being of sporadic time/space nature, are hard to detect and are even harder to correct without using external systems. When a magnetic compass is coupled with a gyroscope, the magnetic disturbances can potentially be detected and corrected, but the tuning of such a filter is extremely difficult.

While the quality of inertial sensors is a key factor for the performance of an INS, its placement on the user's body is also a very important factor. Different placement of the sensors, such as on user's waist [26], trunk [64], leg [100], foot [42] [38] or even the head [18] enables different complementary navigation algorithms that reduce system sensitivity to sensor errors, thereby enhancing its navigation capability.

The waist or trunk locations are probably the less intrusive IMU placements and also the most reliable to estimate heading using gyroscopes or magnetometers since it is near the user's center of gravity [56]. However, for the distance estimation the sensors placed on or inside the user's foot usually have the best results since it has some advantages, as the application of Zero Velocity Update (ZUPT) strategy (Section 2.2.3) which reduce drifts after integrating accelerations [42] [44].

An INS usually uses two approaches to estimate the person location. One uses a set of mechanization equations, based on the human gait cycle (Section 2.1), to convert the IMU measurements into useful position, velocity and heading information. The other uses a magnetometer, a gyroscope and an accelerometer to estimate heading and a pedometer to count the number of steps.

The first approach is based on an algorithm that involves three phases [81]:

- Detect a step - once a step is detected, the algorithm defines the type of step, i.e. forward or backward;
- Estimate the step length - the accelerometer and the gyroscope data are mathematically integrated to provide changes, given an initial location, in velocity and orientation (Section 2.2);
- Estimate heading - the data from a magnetometer, a gyroscope and an accelerometer are fused to estimate heading (Section 2.3).

This algorithm is susceptible to several sources of error, including the sensors bias offset, drift, calibration errors and the random walk errors associated with noise integration.

The second approach uses a pedometer which is a device that counts steps by detecting the pedestrian's motion. Typically, it is composed by an accelerometer and the best pedometers are accurate to within 5% error. However, the position where the sensor is placed has very importance to the accuracy of the system.

In this type of systems an average step length is predefined according to the person's height relationship. It is assumed that the pedestrian walking steps have constant length. Since persons can assume several types of walking along the time, this approach introduce large errors depending on the environment where the system is used.

2.1 HUMAN GAIT

Since a PINS attempts to estimate the step length, in this section is presented a theoretical explanation about human gait. Human gait represents a movement pattern that repeats itself indefinitely at each step and enables the human body transportation from one location to another [106].

The goal of human gait is moving the body through bipedalism. This motion occurs via a series of events that are repeated after each new contact of the heel with the ground (initial contact). These cyclical events allow the shift of the body support from one member to the other. Thus, as the body moves, a member serves as a source of support and the other to advance the body until it contacts the ground [121]. This sequence of events is named as gait cycle and is represented in Figure 2.

The human gait cycle is divided into two phases: stance phase and swing phase. During the stance phase, the foot is in contact with the ground, whereas in the swing phase the foot is not in contact with the ground and the leg is moving until the foot enters, again, in contact with the ground. In a normal gait cycle the stance phase represents 60% of the time and swing phase represents 40% of the time.

The stance phase is divided into three separate phases: first double support, in which both feet are in contact with the ground; single limb stance, when one foot is swinging and the other is in contact with the ground; second double support, when both feet are again in contact with the ground.

Whenever there is a double support the body weight is transferred from one member to the other. The two moments of double support, in a gait cycle, coincide with the time where a member is starting and the other is ending the stance phase. It should be noted that, the less time spent in the double support, the higher is the walking speed [121] [67]. When a person increases the step rate, the duration of the double support decreases and it is *null* if the person is running [102].

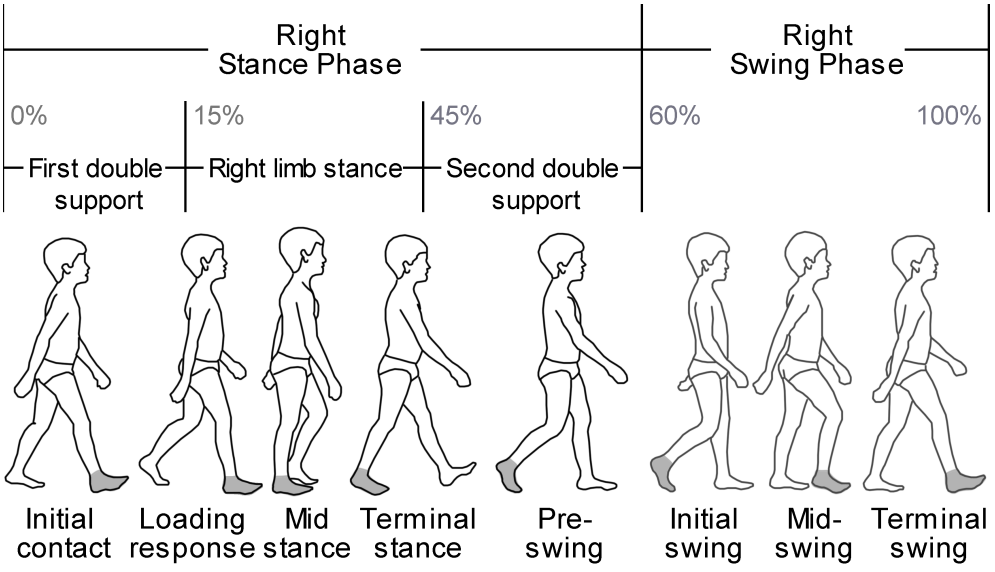


Figure 2: Gait cycle represented for the right foot (adapted from [121])

Analyzing the nomenclature represented in Figure 2, it should be noted that it refers to the right side of the body. The same terminology would be applied to the left side.

The loading response phase represents the interval at which the body weight is transferred to the member that will serve as support. In this interval the foot is completely in contact with the ground. The mid-stance phase begins when the foot, which was previously serving as support, is no longer in contact with the ground and moves forward (body support is only performed by one member). The terminal stance phase begins when the heel is raised and ends when the opposite foot begins its contact with the ground. During this phase the body weight moves to the foot in analysis (right foot).

The pre-swing phase is the terminal phase of the double support that begins when the opposite leg (left in this example) comes in contact with the ground and ends when the foot in question (right) is no longer in contact with the ground. During the pre-swing the body weight is fully transferred to the foot that will serve as support [16].

The initial swing phase starts when the foot is no longer in contact with the ground and continues until the maximum flexion of the knee, i.e. it is the beginning of the lifting motion of the leg. The mid-swing phase begins after the knee maximum flexion and ends when the tibia is vertical. This phase is the end of the leg lifting motion. The terminal swing phase is the preparation for the contact with the ground by the heel [16].

The human stride is defined by the range from which an event occurs on a member until this same event repeats in the same member. Typically, an event like the initial contact of the foot with the ground defines the beginning of the stride. A stride can be defined from the contact of the right leg heel until the same heel contacts, again, with the ground.

A step is a portion of the stride and is defined since an event occurs on a member until this same event occurs on the opposite member. A step can be defined from the lift of the left heel from the ground until it touches again the ground. Thus, two steps are equivalent to a stride, which corresponds to a gait cycle [48]. The step length and the stride are represented in Figure 3.

2.1.1 *Foot Acceleration*

The foot acceleration represents the speed variations over time. This parameter is important because the foot acceleration follows a particular pattern over the gait cycle. It

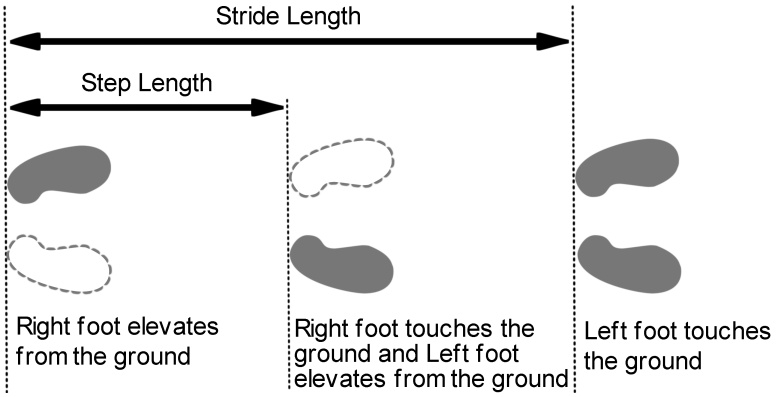


Figure 3: Step and stride length representation

is clear that the acceleration signal is different for each person, for example an elderly walking pace is not like a teenager one [121].

It should be noted that there are some differences between the running and the walking gait cycles. In running it is necessary a better balance since it is characterized not only by a substantial reduction of support but also by the absence of double support and by suspension periods in which both feet are not in contact with the ground [67].

The study of the acceleration is important not only because of its pattern, but also because it can be used to calculate the velocity and the corresponding walking distance. Velocity corresponds to the integral of accelerations against time (Equation 2), and the walking distance results from the integral of the velocity (Equation 3).

$$v = \int_{t_1}^{t_2} a(t)dt \tag{2}$$

$$d = \int_{t_1}^{t_2} v(t)dt \tag{3}$$

where v represents velocity, d is the distance, dt is the derivative of time.

The estimation of the walking distance is calculated based on the acceleration when the foot is not in contact with the ground. When the sole of the foot is in contact with the ground it should be considered that there is no acceleration, and therefore no displacement, because the foot in question is not moving.

2.1.2 *Force Applied on Foot Plant*

To distinguish more accurately each phase of the gait cycle it could be analyzed the force that is applied to the foot plant. By knowing this force the step detection can be significantly improved, as well as, the correspondent estimation of the walking distance. Figure 4 presents the force sequence (the applied force is shown in grey) in the foot plant during a normal gait cycle. It can be seen that the central region of the foot has almost no participation in the step motion.

The initial phase (A), which represents the first contact of the foot with the ground, and the final stage (D and E), which represents the last contact of the foot with the ground, are the stages where the most significant force on the plantar surface is applied. Specially on the heel and in the front area. This force is much more significant than in the stance phase (B and C). Also, on stairs ascent and descent, the force is concentrated mainly in the front area and in the heel the force is negligible [115].

Figure 4 is directly related to the sub-phases that occur during the stance phase, wherein:

(A) Represents the initial contact of the foot with the ground;

(B) Represents the time at which the response to the load occurs;

(C) Represents the middle stance phase where support is being made solely by this foot;

(D) Is the terminal stance phase;

(E) Represents the pre-swing.

After these phases the foot is no longer in contact with the ground, since it had begun the swing phase.

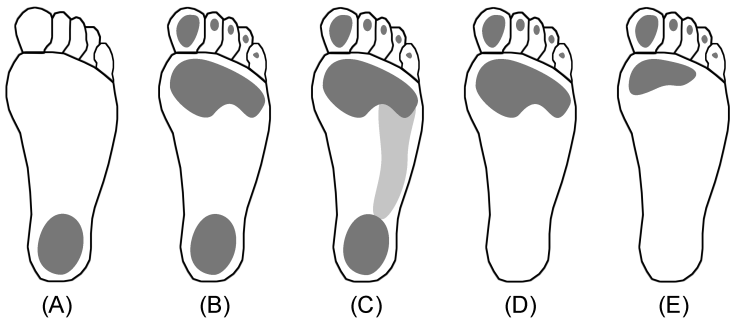


Figure 4: Force applied on the foot plant during the stance phase

2.2 DISTANCE ESTIMATION

Distance is, typically, estimated based on information about the step length. The step length is defined as the distance between the point where the heel of one foot is in contact with the ground and the point where it enters, again, in contact with the ground, as represented in Figure 3. However, step length can change not only from individual to individual, but also according to the slope and texture of the terrain.

Typically, several sensors are used to properly estimate the step length, so a sensor fusion must be performed to integrate all the data. Usually the sensor fusion is performed using a Kalman Filter (KF) (Section 2.2.5).

There are two ways to estimate the pedestrian walking distance based on inertial sensors. The first one is by counting the number of steps and multiplying it by a predefined estimation of the step length (Section 2.2.1). The second one estimates the walking distance based on the double mathematical integration of the acceleration values (Section 2.2.2).

2.2.1 *Step Count*

In a PINS, a pedometer can be used to count the number of steps, that combined with a predefined estimated step length, is used to estimate the walking distance, as presented in Equation 4.

$$d = step\ length \times number\ of\ steps \tag{4}$$

The number of steps is typically obtained by counting the number of acceleration peaks that are detected in the gait cycle, as presented in Figure 5. To correspond to a step, each acceleration negative peak must be followed by a positive acceleration peak. In Figure 5 are represented five acceleration peaks, which represent five steps given by the pedestrian during a walk. The distance per step depends on the user speed, as well as, his height. For example, the step length may be larger if the user is tall and is running.

Zhao [127] has made some experiments and formulated the data presented in Table 1, where the length of each step is a function of the number of steps performed at every two seconds depending on the user's height [127]. For example, the first line shows

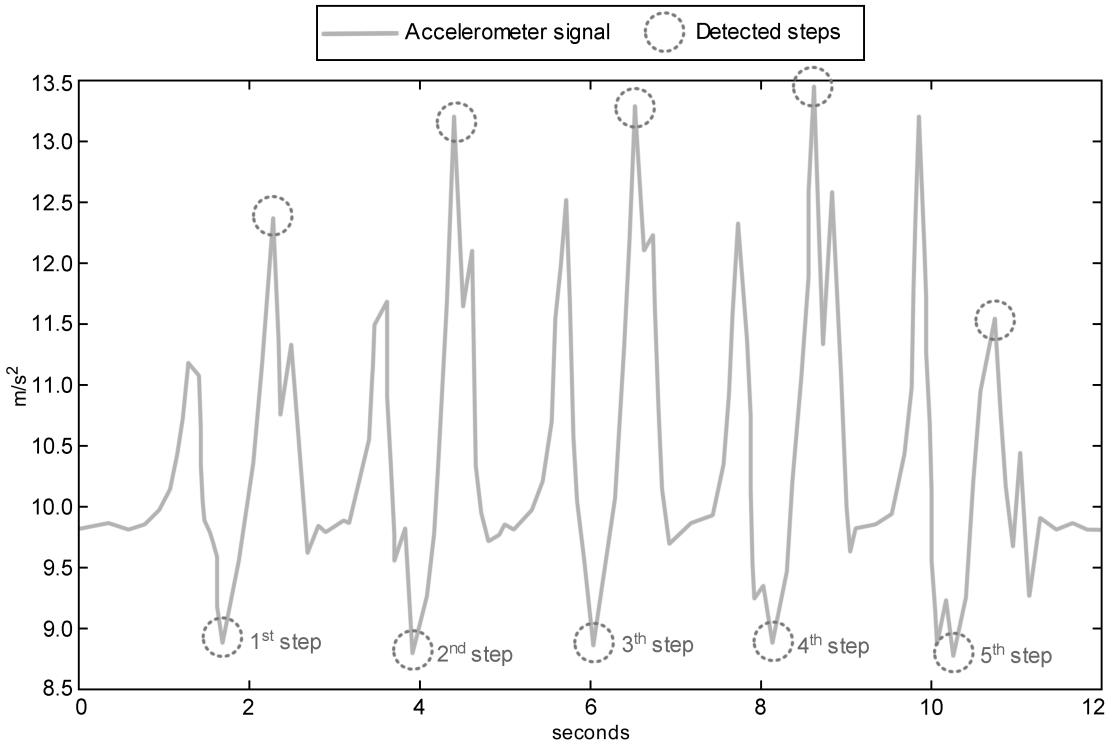


Figure 5: Combined acceleration data for a five step walk

that when a single step occurs in a two seconds period, this step has the length of the user's height divided by 5, and so on.

Table 1: Stride as function of speed (steps per 2 seconds) and pedestrian height [127]

Steps per 2 seconds	**Step length**
$0 \sim 2$	$\mathrm{Height} \div 5$
$2 \sim 3$	$\mathrm{Height} \div 4$
$3 \sim 4$	$\mathrm{Height} \div 3$
$4 \sim 5$	$\mathrm{Height} \div 2$
$5 \sim 6$	$\mathrm{Height} \div 1.2$
$6 \sim 8$	Height
$>= 8$	$\mathrm{Height} \times 1.2$

However, the step length depends on several factors, as the velocity, the step frequency or the current environment. For example, the user's step length is shorter when the user is walking in a crowded area. Hence, the predetermined step length cannot be used effectively for distance estimation. An accuracy improvement that can be done is adjusting the step length as the user changes his pace.

2.2.2 *Strapdown Inertial Navigation System*

A strapdown INS estimates the current velocity and the walking distance using the acceleration data. Acceleration, when mathematically integrated with time, provides a change in velocity, and when doubly mathematically integrated with time, provides a change in location. Thus, an INS, generically, provides the change in location and the correspondent velocity. To avoid errors produced by the inertial sensors, acceleration is only considered on the swing phase and usually the ZUPT technique (Section 2.2.3) is applied to improve the accuracy of the INS.

In order to enhance the accuracy of the detected accelerations, the IMU is usually placed on or inside the pedestrian footwear. These accelerations can be decomposed in two types, vertical and forward. Figure 6 represents the correlation that exists between the human gait cycle and the vertical and forward accelerations.

Analyzing Figure 6 it can be seen that between the pre-swing and mid-swing phases the vertical acceleration is increased, and decreases between mid-swing and terminal swing phases. However, the forward acceleration is nearly the opposite. The only difference is that it happens a little bit after the first one. This means that it decreases between initial swing and after mid-swing, and increases, a lot, from that point until the terminal swing.

Before acceleration can be double integrated, it needs to be projected to the navigation coordinate system using gyroscope data. This happens because sensors are

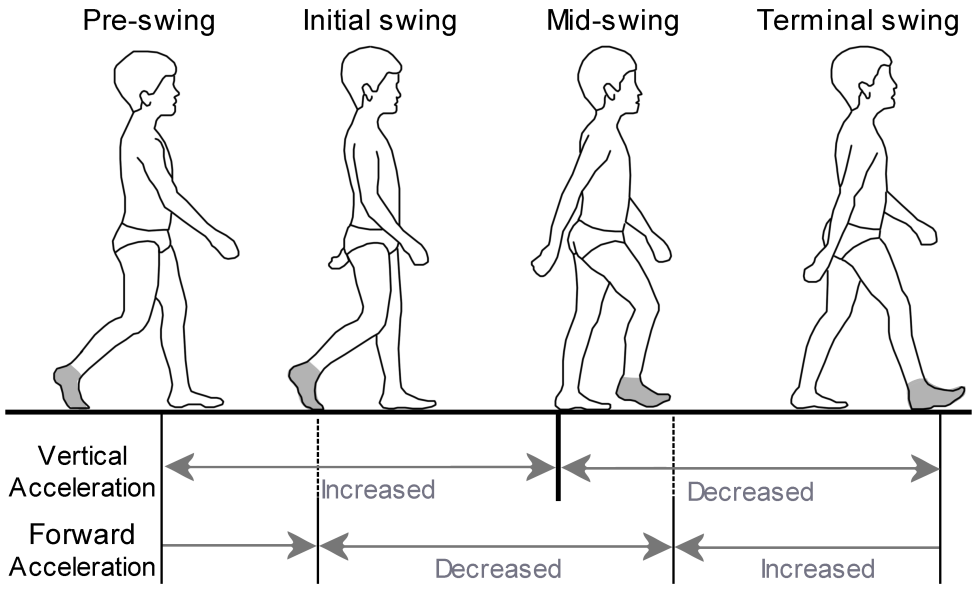


Figure 6: Foot vertical and horizontal accelerations according to the gait cycle

attached to the body that is moving. So, there are two different coordinate systems (frames) to be considered. The sensors coordinate system will be referred to as sensor frame and the navigation coordinate system referred to as navigation frame. The sensor frame is defined by the IMU sensor, that is, the coordinate system attached to the system. The sensor frame rotates relatively to the navigation coordinates by the Roll, Pitch and Yaw angles. Since accelerometers measure accelerations on the sensor (f^s) frame, to obtain the accelerations in the navigation frame (f^n), the vector f^s must be multiplied with the rotation matrix as shown in Equation 5.

$$f^n = R_s^n \times f^s \tag{5}$$

where R_s^n is a 3x3 matrix that defines the attitude of the sensor frame relatively to the navigation frame.

The transformation from one frame to the other can be achieved by three successive rotations, which are expressed as follows:

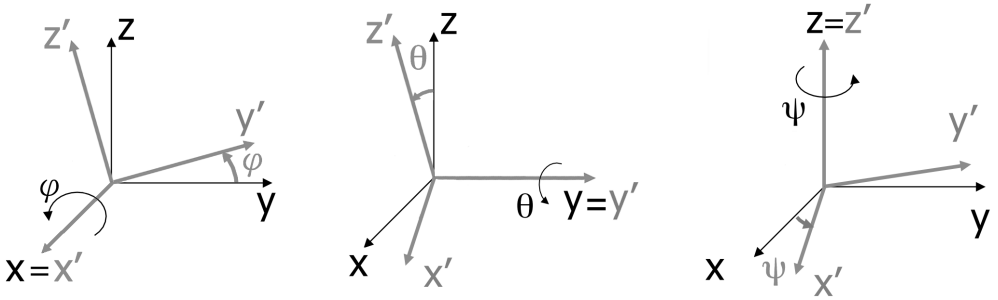
- Rotate through angle φ over the *x-axis* (Matrix 6)
- Rotate through angle θ over the *y-axis* (Matrix 7)
- Rotate through angle ψ over the *z-axis* (Matrix 8)

$$R_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{pmatrix} \tag{6}$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{7}$$

$$R_z(\psi) = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{8}$$

$$R_s^n = R_z(\psi) \times R_y(\theta) \times R_x(\varphi) \tag{9}$$



(a) (b) (c)

Figure 7: Rotations through angles: (a) rotation through angle φ over the *x-axis*; (b) rotation through angle θ over the *y-axis*; (c) rotation through angle ψ over the *z-axis*

These three rotations (Figure 7) can be expressed as three separate matrices as shown in 6, 7 and 8, where ψ, θ and φ are referred as the Euler angles. Thus, the process of transform the acceleration data to the navigation frame can be presented as the product of those matrices, as presented in Equation 9 [120] [93].

After this projection, the effect of gravity is perpendicular to gravity in the navigation frame axis. Then, after correcting the effect of gravity, the position and speed changes can be directly estimated by successive mathematical integrations of the acceleration obtained from the system over time.

An overview of this process is presented in Figure 8. Where the gyroscope data is integrated, in relation with time, to estimate orientation, as well as, to be used to project the acceleration data into the navigation frame. Then the gravity component is removed from the projected acceleration data. This acceleration is integrated, with time and according to an initial velocity, to give the current velocity. Finally, this data is, one more time, integrated with time and according to an initial position, to give the actual pedestrian position.

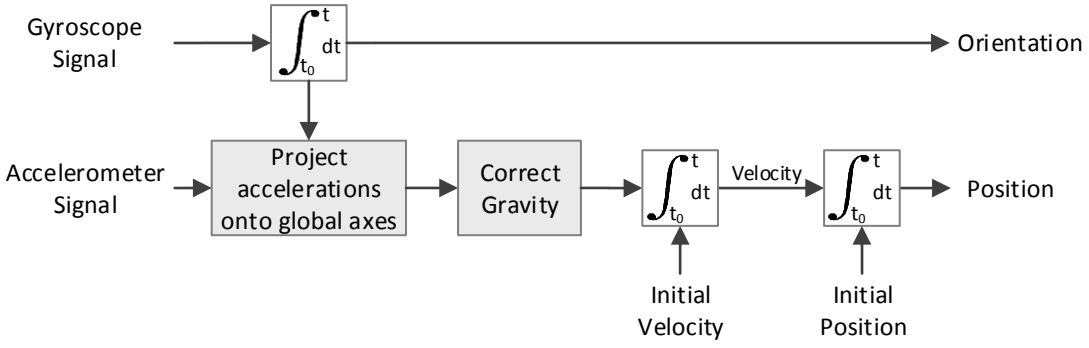


Figure 8: Strapdown INS algorithm

2.2.3 *Zero Velocity Update*

Accelerometers drift can cause huge location errors for walks longer than a few seconds. However, the use of a technique known as ZUPT [44] can reduce these errors. ZUPT is commonly used in underwater navigation [53] and in oil drilling [65].

In 1996, Larry Sher [108] within a Darpa project seems to be the first researcher that have noted that stance phase provides an opportunity to use ZUPT. However, the first published work, in 1999, is from John Elwell [35].

Zero-velocity detection is an essential part of an INS. Without this information, the velocity error would increase linearly with time. Typically, zero-velocity is detected on the foot by using accelerometer data. This has made foot-mounted IMUs a popular choice for PINS, since it is the most accurate body part to detect zero-velocity. The use of ZUPT allows the drift containing, at least for a while, until a new step is made. This algorithm reduces significantly the location errors, thus ensuring more accurate location estimations. However, this accuracy strongly depends on the time intervals between zero-velocity points. In a PINS this update point usually occurs during the stance phase.

During stance phase and unless the sole slides (does not move relative to the ground), it is expected that the accelerometers show measurements of zero. Most stance-based schemes in the literature equate zero-velocity detection to the impact of the heel when it hits the ground. The problem is that the impact shock event only signals the beginning of the stance phase. However, not all the stance sub-phases have zero-velocity. Also, such methods can incorrectly detect zero-velocity if the sensor moves at a constant speed and they might fail to detect a stance phase if the sensors are very noisy.

Zero-velocity only occurs at some point around the mid-stance phase, after a complete contact of the foot with the ground has been reached. This pattern can be seen by analyzing the acceleration data presented in Figure 9, where during the loading response phase, and during the mid-stance phase, the acceleration is not constant. So, if the zero-velocity point is not accurately determined, the resulting ZUPT scheme will have an intrinsic zero-velocity bias that will reduce its effectiveness.

If the pedestrian's foot actually slipped during one step, then the resulting error in velocity is only considered for that step. Subsequent steps are again error-free. The resulting error is just a few centimeters and it remains constant for the rest of the walk, unless new errors occur.

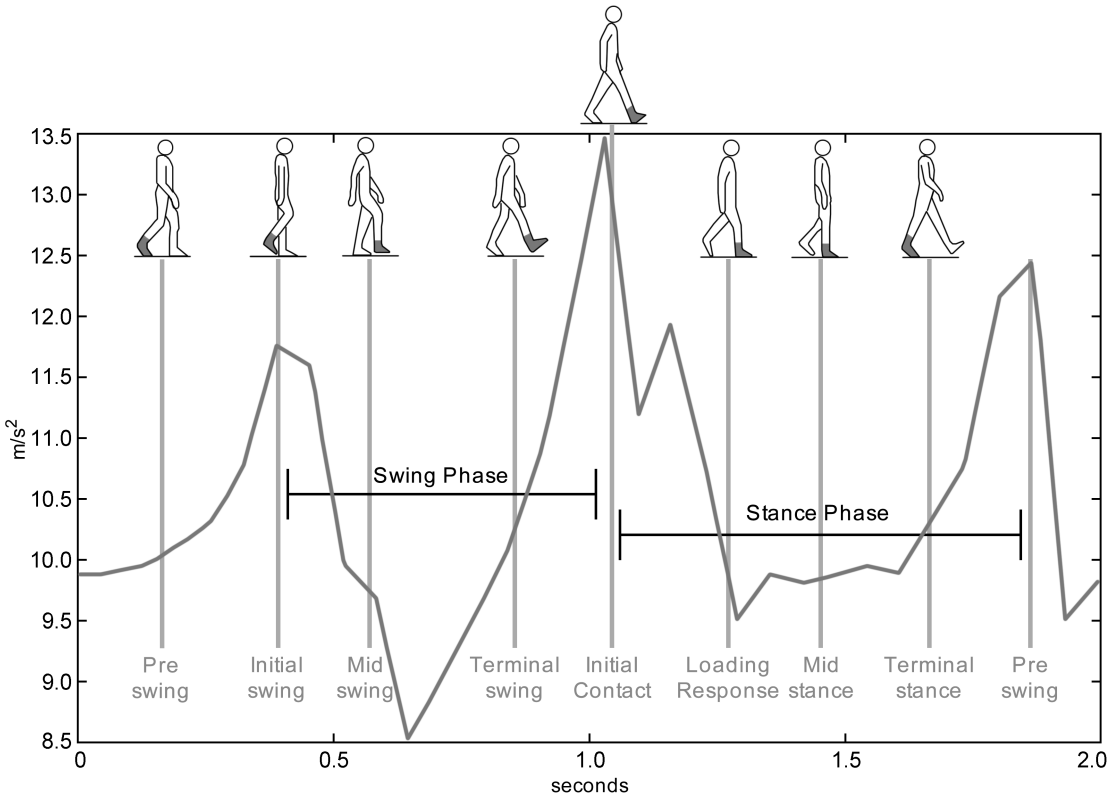


Figure 9: Combined accelerometer measurements in correspondence with gait cycle phases

2.2.4 *Zero Angular Rate Update*

Zero Angular Rate Update (ZARU) uses similar principles as the ZUPT technique, which typically resets the velocity and therefore the estimated displacement. ZARU consists on resetting the angular velocity when the foot is stationary, and it can be also used to reset the angle of the IMU [21].

This technique can be applied in orientation and distance estimation, allowing the estimation of the gyroscope bias and the estimation of the attitude angles. In the case of the orientation, a constant attitude, when the pedestrian is stationary, can be used.

For distance estimation a predefined initial angle, at each step, can be used to apply the rotation matrix to the accelerometer data. For example, since the IMU may have a slope different of zero due to the pedestrian's posture or due to the location of the IMU, this algorithm sets the initial angle, which is the initial condition of the integration process, that can be equal to the one obtained using the acquired accelerometer data during the system setup process.

In order to formalize this algorithm, the attitude's (x_θ) pseudo-measurement model, during a stationary phase, may be written as Equation 10 [26] [21].

$$x_\theta = \theta + n_\theta \tag{10}$$

$$\theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \tag{11}$$

$$n_\theta \sim N(0, Q_\theta) \tag{12}$$

where θ is a vector (Equation 11) containing the true IMU's roll ϕ, pitch θ and yaw ψ angles, and n_θ (Equation 12) is a white Gaussian noise with covariance $Q_\theta = diag(\sigma_\phi^2, \sigma_\theta^2, \sigma_\psi^2)$ [26].

2.2.5 *Kalman Filter*

In the 60's, Rudolf Kalman published a famous article describing a recursive algorithm to discrete-data linear filtering problem, the Kalman filter [60]. Since that time, due to advances in digital computing, this filter has become an essential part of the space and military technology development, but it can also be found in more common applications such as GPS, earthquake prediction, communications, risk assessment, biomedical research, among others.

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error [57]. A Kalman filter performs conditional probability density propagation for problems in which the system can be described through a linear model.

Conceptually, the filter tries to obtain an optimal estimate from data provided by a noisy environment. For example, a GPS can infer the speed of a car. However, this inference has some uncertainty. Since, measurements usually have some noise, the Kalman filter tries to extract valuable information from that noisy signal. A Kalman filter combines all available data (more that one data source), plus prior knowledge about the system and measuring devices, to produce an estimate of the desired variables in such a manner that the error is statistically minimized [58]. A simple scheme of Kalman filter can be seen in Figure 10.

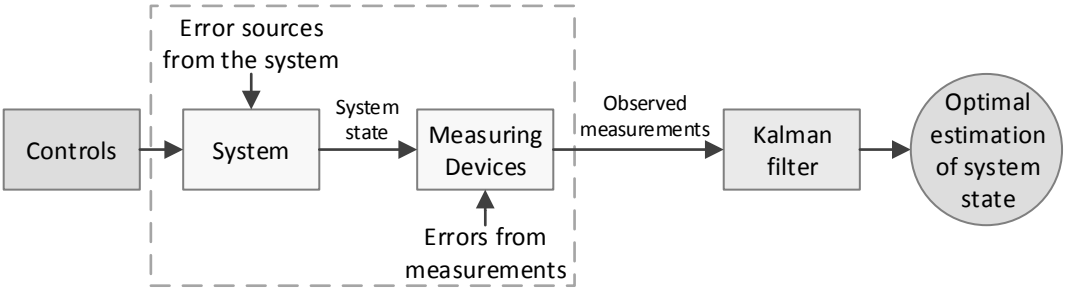


Figure 10: A Kalman filter representation

One aspect of this optimization is that Kalman filter uses all information that can be provided. It processes all available measurements, regardless of their precision, to estimate the current value of the variables of interest, using:

- Knowledge of the system and measurement device dynamics;
- Statistical description of the system noises, measurement errors, and uncertainty in the dynamics models;
- Any available information about initial conditions of the variables of interest.

The Kalman filter predicts a system's state using a measurements weighted average. The purpose of the weights is that values with lower uncertainty are more trusted. The weights are calculated from the covariance, which is a measure of the uncertainty of the prediction for a system's state. The result of the weighted average is a new state estimated that lies in between the predicted and the measured state, and has a better estimated uncertainty than either alone. This process is repeated every time, with the new estimate and its covariance informing the prediction to be used in the next iterations [69].

Passing this to the mathematical model, the estimation of the state x of a process at instant k, is a discrete-time controlled process that is given by Equation 13. This model assumes that the true state at time k is evolved from the previous filtered state x_{k-1}.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \tag{13}$$

where A represents the state transition matrix, which is an $n \times n$ matrix that relates the state at the previous measurement (x_{k-1}) to the state at the current measurement, B is the control-input matrix, which is an $n \times m$ matrix, that relates the control-input u to the state x, and w is the process noise. The state transition matrix A and the control-input matrix B are considered to be constant.

The state observations are performed through a measurement system that can be represented by Equation 14.

$$z_k = Hx_k + v_k \tag{14}$$

where z_k is the observation made at instant k, H is the observation matrix, which is an $n \times m$ matrix that relates the state x to the measured value z, and v represents the observation noise. The observation matrix H is considered to be constant.

The random variables w and v that represent the process and observation noise are defined by Equations 15 and 16, respectively. They are assumed as white noise, independent (of each other), and described as normal probability distributions.

$$p(w) \sim N(0, Q) \tag{15}$$

$$p(v) \sim N(0, R) \tag{16}$$

where Q is the process noise covariance matrix and R is the measurement noise covariance matrix. These matrices might change with each time step or measurement. However, it is assumed that these two matrices are constant.

The Kalman filter estimates a process by using a form of feedback: the filter estimates the process state at a given time and then obtains feedback in the form of measurements (with noise). Thus, the Kalman filter equations can be decomposed in two distinct groups: Time Update (prediction), composed by Equations 17 and 18, and Measurement Update (correction), composed by Equations 19, 20 and 21. Both set of equations are applied at each k^{th} state [123].

The Time Update equations are responsible for projecting forward (in time) the current state (x_k) and an estimated error covariance (P_k). These estimates are also known as prior estimates. P_k is the estimation of the uncertainty associated with the estimate of the current state (x_k). The Measurement Update equations use the new prior estimated measurement, in order to obtain a best posterior estimate. This is a recursive process [123], and is described in Figure 11 with the equations that will be then explained.

Resuming, the Time Update projects the state ahead in time and the Measurement Update adjusts the projected state estimation. The state ahead is projected using Equation 17. Then the error covariance is projected ahead using Equation 18. The state ahead

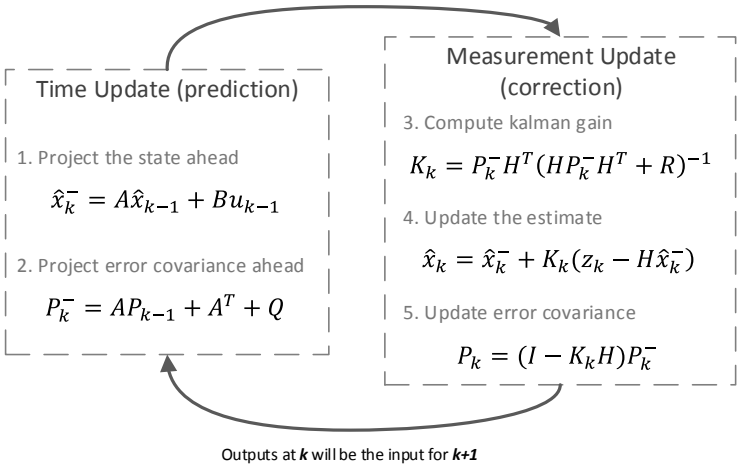


Figure 11: Kalman filter recursiveness with the Time Update and Measurement Update equations

is represented as $\hat{x}_k^-$ and typically referred as the prior estimate, which is the estimate before the measurement update correction. The result of Equation 18 is the P_k^-, which is also referred as the prior error covariance.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{17}$$

$$P_k^- = AP_{k-1} + A^T + Q \tag{18}$$

The Kalman gain K is calculated using Equation 19, which is estimated based on the covariances (degree of trust). This gain minimizes the posterior error covariance P_k. The measurement covariance R, exists because the sensors do not always give correct data. The parametrization of this variable is much dependent on intuition. When R goes to zero, the gain K_k gives a higher weight to the residue. On the other hand, when P_k^- tends to zero, the gain K_k gives less weight to the residue.

Resuming the Kalman gain, when the measurement noise error covariance R tends to zero, the measurement z_k is more trusted, while the predicted state $H\hat{x}_k^-$ is less trusted. On the other hand, when the prior error covariance P_k^- tends to zero the current z_k measurement is less trusted, while the measurement forecast $H\hat{x}_k^-$ is becoming more trusted. This relation is obtained using Equations 19 and 20.

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{19}$$

Equation 20 is the correction performed to the state variable $\hat{x}_k$ from the predicted state variable ($\hat{x}_k^-$) by adding the product of the Kalman gain with the residual from the measured data z_k and the predicted data. The predicted data is composed by the H output equation and the predicted state variable $\hat{x}_k^-$. The difference between the measured data and the predicted data is called the measurement innovation or residue. If the resulting value is zero means that the prediction is similar as the actual measured value.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{20}$$

Equation 21 is responsible to update the posterior error covariance, where I is an identity matrix.

$$P_k = (I - K_k H)P_k^- \tag{21}$$

Analysing all the equations that composes the recursive cycle of the Kalman filter, it can be concluded that, Time Update equations project the state (Equation 17) and the error covariance (Equation 18) ahead in time from $k-1$ to k. Regarding the Measurement Update, the first task is to calculate the Kalman filter gain, K_k (Equation 19). The next step is to obtain the posterior state based on the z_k measurement (Equation 20). Finally, is obtained the posterior error covariance (Equation 21).

Unlike certain data processing concepts, the Kalman filter does not process all the previous data every time a new measurement is taken, requiring only the last data. This filter is very useful to perform drift corrections on an INS. Since, it can be used to correct the error states from combining the acceleration, gyroscope and other sensors uncertainties.

Several variants of this filter appeared over time. In PINS the most common are: Extended Kalman Filter (EKF), which is the non-linear version of the Kalman filter [58] [42] [103] [57] [20] [69] [54]; Interval Kalman Filter (IKF), which includes bounded uncertainty into the model [26]; and Unscented Kalman Filter (UKF) that uses statistical linearization techniques [77].

2.3 ORIENTATION ESTIMATION

The heading information is normally acquired in the user's waist. However, several systems place only one IMU in the person's body, which typically is on the foot in order to estimate both distance and orientation. To determinate the heading a gyroscope [89]

[69] or a magnetometer [42] [38] are used. However, it is difficult to estimate an accurate heading, due to the drift of low-cost MEMS gyroscopes and unpredictable magnetic disturbances.

The most common approach is the use of a magnetometer, which measures the strength of a magnetic field. Typically it is used to sense the earth's magnetic field in order to indicate where the north is. This geomagnetic field strength varies over the earth's surface from a minimum of 22 μT over South America to a maximum of 67 μT in south of Australia [40]. Magnetometers are constantly sensing the magnetic field and the precision tends to be relatively low, because the earth's magnetic field is far from constant and therefore the reliability is reduced. Also, since these sensors are very sensitive and responsive to all other magnetic fields, they are affected by magnetic disturbances caused by power lines, computers, among others.

The combination of a gyroscope with a magnetometer takes the advantage that gyroscopes can offer relatively accurate heading rate during short periods of time and magnetometer provides absolute heading with respect to the magnetic north in an environment without magnetic distortions [28]. The combination of these sensors data can be achieved, for example, by a complementary filter. However, the filter thresholds can be very difficult to define [59]. When these thresholds are too low, gyroscope noise may erroneously lead to identify movements when they do not exist. When these thresholds are too high, either the start of a movement is delayed or its end is anticipated.

Another sensor that is typically combined with the magnetometer is the accelerometer. The accelerometer measures the components of the earth's gravity and the magnetometer measures the components of earth's magnetic field. Thus, heading can be computed from the arctangent of the ratio of the two horizontal magnetic field components. The accelerometers will correct the sensors tilt (roll and pitch) and magnetometer corrects heading (yaw). However, to have acceptable results it is crucial that the accelerometer and magnetometer outputs are aligned with the coordinate system.

Since the orientation estimated from the mathematical integration of the gyroscope data accumulates errors from gyroscope's noise, and the accelerometer and magnetometer are affected by the sensor's bias and magnetic interference, respectively. The three sensors can be combined to improve the estimation of the orientation.

Concluding, gyroscope is more accurate and with short response times (compared to the other sensors). Although the magnetometer and accelerometer have as main disadvantage the noise, the downside of gyroscope is the drift. To avoid both the gyro-

scope drift and the noise of magnetometer and accelerometer, the gyroscope output is applied in a short time basis, while the data from the accelerometer and magnetometer are applied at long intervals of time. This process is equivalent to filtering the data from accelerometer and magnetometer with a low-pass filter, and filtering the data from the gyroscope with a high pass filter, as represented on Figure 12. Therefore, the use of the three sensors becomes essential to suppress each other errors and limitations.

The complementary filter can be described as in Equation 22. Having x and y represented as two signals with noise and assuming that the y noise is mostly of high frequency, and the noise of x is mostly low frequency. Then the function $G(s)$ can stand out as a low-pass filter in order to filter the high frequency noise of y and $[1 - G(s)]$ can stand out as the complement, i.e. a high-pass filter that filters the low noise x frequency. This results in an estimative $\hat{z}$ produced by the filter. If the observations had any noise or no error the result of Equation 22 would be a perfect estimate.

$$\hat{z} = z[1 - G(s)] + zG(s) \tag{22}$$

A recent, and popular, information filter for orientation estimation is the one proposed by Madgwick [76]. It uses a quaternion approach to represent the attitude, based on the information obtained from a gyroscope, an accelerometer and a magnetometer. It uses a quaternion representation, which allows the accelerometer and magnetometer data to be used in an analytically derived and optimized gradient-descent algorithm to compute the direction of the gyroscope measurement error as a quaternion derivative and is not subject to the problematic singularities associated with an Euler angle representation.

This filter is computationally inexpensive, meaning that it requires low scalar arithmetic operations per each filter update. Since it operates at low sampling rates (e.g. 10Hz), it means that the necessary hardware and power for wearable inertial move-

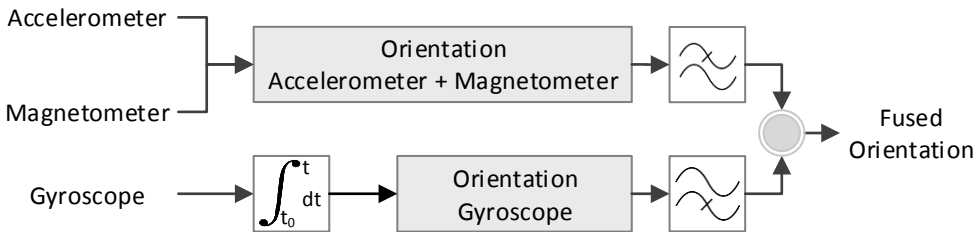


Figure 12: Complementary filter to obtain orientation using an accelerometer, a magnetometer and a gyroscope

ment tracking can be reduced, which enables the creation of lightweight and inexpensive systems, capable of functioning in real-time and for extended periods of time.

This implementation was compared against a Kalman-based algorithm and the results have pointed out that Madgwick filter achieves best accuracy than a Kalman-based algorithm. Also, unlike Kalman-based algorithms it has only one or two filter gains to be defined, which simplifies the filter usage and tuning [76].

2.4 INFORMATION FUSION

Information fusion is a multi-disciplinary research field with a wide range of potential applications in areas such as defense, robotics, automation and pattern recognition. It is so important that during the past two decades, extensive research and development on multiple sensor/information fusion has been performed for the Department of Defense of the United States of America [70]. This subject has been and will continue to be an ever-increasing interest field in research community, where it is intended to develop more advanced information fusion methodologies and architectures.

Information fusion is the process of combing information from a number of different sources to provide a robust and complete description of an environment or process of interest. It has special significance in any application where large amounts of data must be combined, fused and distilled to provide information of appropriate quality and integrity on which decisions can be made. The exploration of huge volume of data requires reliable methods of data reduction to support the understanding and extraction of useful rules, decisions or information from the data.

In the case of PINS, the MEMS sensors presents some limitations and have low accuracy, which does not happen on more expensive sensors like the ones used on aviation and on military applications. To reduce the sensors complexity and thereby its cost, the information from a set of simple and low-cost sensors can be combined to provide a less expensive system, with accurate and reliable information about the pedestrian movements. Moreover, this fusion makes the system more fault tolerant.

Almost all information fusion architectures involve the following process: a set of features are extracted from the sensors/sources that compose the system, then the information is fused and classified, and finally a reasoning about the data is performed.

However, decision-making with multiple sources of information is complex. The essential reason, to be complex, is that it is difficult to provide a consistent measure of utility or loss for each decision maker. In particular, two basic problems exist. First,

how can the utilities of two different decision makers be compared unless there is a common measurement type. Second, should decision makers aim to maximize an utility which expresses only local preferences, or should each decision maker evaluate its actions with respect to some common or group utility function.

Usually, in INSs, sensor fusion is performed using a Kalman filter. However, a new trend is to use a Kalman filter combined with Artificial Intelligence (AI) techniques or only AI techniques.

AI techniques are being used in different INS fields to assist in displacement estimation. In robotics, Faceli et al. [36] use these techniques to improve the accuracy and reliability, in the order of 7%, of distance measurements between a robot and the objects present in the environment.

These techniques are also being used in autonomous driving vehicles. Stanley [119] software relied on machine learning and probabilistic reasoning techniques. Its IMU combined with AI techniques were able to maintain accurate pose of the vehicle during GPS outages of up to 2 minutes. Authors believe that those techniques contributed significantly to Stanley's success in the race.

For land vehicle applications [41], Caron et al. [25] and Noureldin et al. [86] propose machine learning techniques like neural networks, which introduce context variables and errors modelling for each sensor. Authors conclude that with an adequate modelling an accuracy improvement of 20% can be achieved. Recently, Noureldin et al. [87] also consider past samples of INS position and velocity errors, which have improved the previous results. Bhatt et al. [22] propose a hybrid data fusion methodology using Dempster-Shafer theory augmented by a trained Support Vector Machine (SVM), which corrects the INS errors. The proposed methodology has shown an accuracy improvement of 20%.

Since these experiences presented results in each area, in this project similar techniques will be applied to a PINS. With the fusion of the information obtained from several IMU spread in the person's body, and the application of learning algorithms.

In this section some of the most used machine learning techniques will be presented, the Neural Networks (Section 2.4.2.1) and the SVM (Section 2.4.2.2). Also, it will be presented, in Section 2.4.1, the Dynamic Time Warping (DTW) algorithm that measures the similarity between two temporal sequences, which may vary in time or speed.

Based on this exposition, it is clear that research on information fusion systems is becoming more and more common-place. There are a number of areas in the information fusion community that will be highly active in the near future [61]. For instance, the

ever-increasing demand for information fusion on extremely large scales, such as sensor networks and the Web, will drive intense research on highly scalable information fusion algorithms based on distributed architectures [61].

2.4.1 *Dynamic Time Warping*

DTW is a time series alignment algorithm [63] that was largely employed in the early speech recognizers. This technique was used to accommodate differences in timing between sample words and templates. Nowadays it is applied to many other fields like bioinformatics, econometrics, robotics, manufacturing, handwriting recognition, data mining, information retrieval, among others to automatically cope with time deformations and different speeds associated with time-dependent data.

For instance, similarities in walking patterns could be detected using DTW, even if one person walks faster than the other, or if there were accelerations and decelerations during the course of an observation.

This algorithm aims to find the optimal alignment between two sequences of data (e.g. time series). After this alignment it calculates the similarity independently of non-linear variations that can exist in the time dimension. It gives intuitive distance measurements between time series by ignoring both global and local shifts in the time dimension, which enables the determination of a degree of similarity between time series.

The algorithm starts by constructing a warp path of the given two series. An example of how two time series are "warped" is shown in Figure 13. Each vertical line connects a point in one time series to its corresponding similar point in the other time series. If both time series were identical, all the lines would be straight verticals because no warping would be necessary to line up the two time series. The warp path must start at the beginning of each time series and finish at the end of both time series.

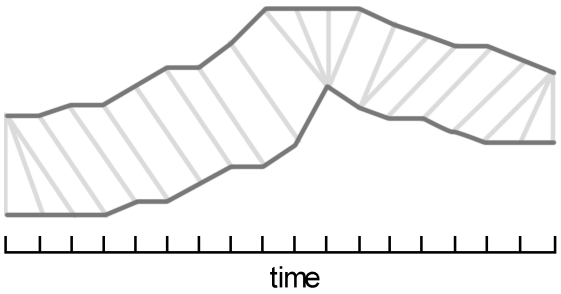


Figure 13: DTW warp path of two time series

The optimal warp path is the one that has the minimum distance. The warp path distance is a measure of the difference between the two time series after they have been warped together. It is measured by the sum of the distances between each pair of points connected by the vertical lines. A lower DTW distance denotes a higher similarity. Thus, two time series that are identical, except for localized stretching of the time axis, will have a warp path distance of zero.

To find the minimum warp path distance a dynamic programming approach is used. A matrix is generated where each *cell*(i, j) represents the distance between the i^{th} element of the first sequence and the j^{th} element of the second sequence. A representation of this matrix can be seen in Figure 14.

Then for each cell, starting at the bottom left cell, which corresponds to the beginning of both signals, and ending at the top right cell, which corresponds to the end of both signals, the cumulative distance is calculated by picking the neighbouring cell in the matrix to the left or beneath with the lowest cumulative distance.

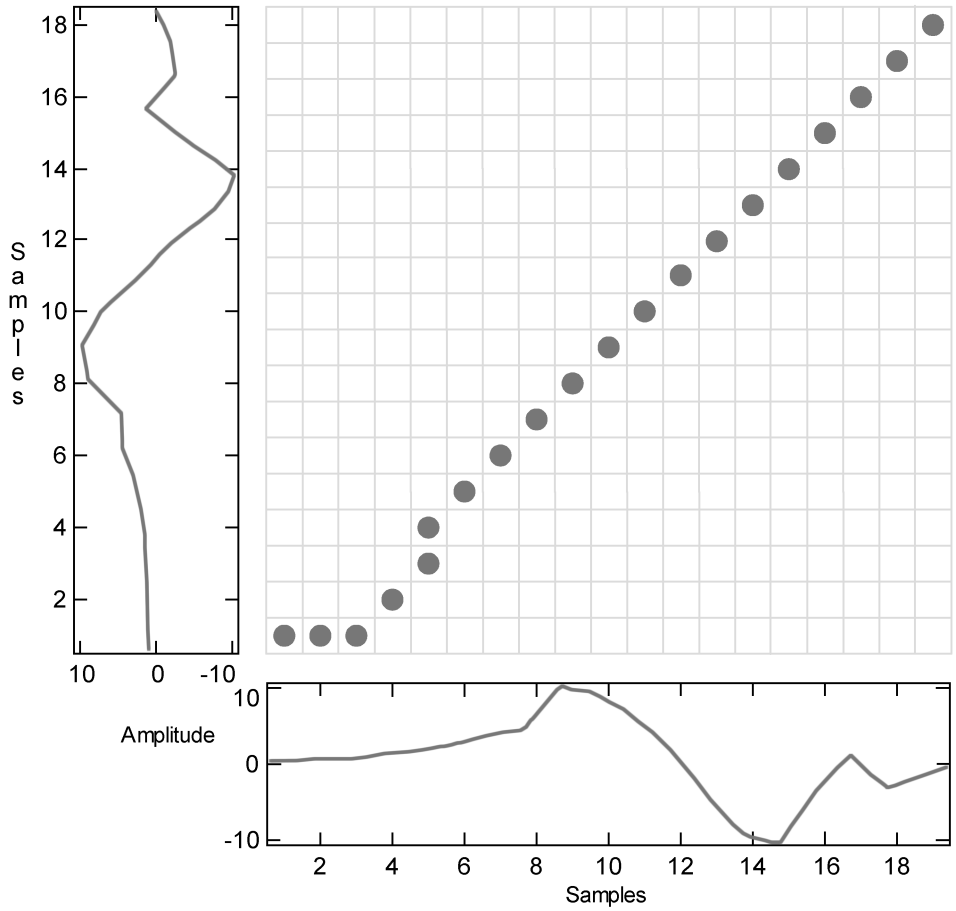


Figure 14: DTW matrix warp for two time series

This results in the shortest path to go from the bottom left cell to the top right cell of the matrix. The length of a path is simply the sum of all the cells that were visited along that path. The further away the optimal path wanders from the diagonal, the more the two sequences need to be warped to match together, and the lower is the similarity between the two paths.

Despite the effectiveness of the DTW algorithm, its main problem is that it has a quadratic time and space complexity, $O(n^2)$, that limits its use to small time series datasets. A faster approach, entitled FastDTW, was proposed by Salvador and Chan [104] which is an approximation of DTW, but with a linear time and space complexity, $O(n)$. The main limitation of the FastDTW algorithm is that it does not guarantee to find the optimal solution. Meaning that the error, compared to the DTW approach is typically higher.

2.4.2 *Machine Learning*

Machine learning is a scientific discipline that explores the construction and study of algorithms that can learn from data without assuming a predetermined equation as a model. The model is created based on a set of examples, that will be used to produce predictions or decisions for a given input, without using strictly static program instructions [75].

The base for machine learning is the data, so with good data the algorithm will perform very well. Also, the performance is better as it is increased the number of samples available for learning.

Machine learning algorithms are typically used in two situations: (i) when it is too complicated to develop a program to find something in a large amount of data, (ii) or to calculate the probabilities of something to happen based on some rules [83]. It can be applied to several areas and can be used to:

- Find patterns - objects in an image, facial expressions, words;
- Find anomalies - anomalous bank transactions, bad sensor measurements;
- Predictions - stock prices, recommendation systems, energy price and load forecasting, Electrocardiography (ECG) signal prediction [80].

The learning can be divided into three categories: supervised, unsupervised and reinforcement learning [83]. In supervised learning, the output predictions are learned based in a given input, meaning that these algorithms learn from experiences. The goal

is to learn a general rule that associates inputs to outputs and it is intended to minimize misclassification. A flowchart for this type of learning can be seen in Figure 15, where the data is first treated, then a model is evaluated based on a set of known inputs and outputs. After the selection of the best model, it is used to predict the outputs for a new dataset of inputs.

Unsupervised learning, tries to discover a representation about the input, meaning that no labels feed into the learning algorithm, leaving it on its own to find a structure in the input.

Reinforcement learning, selects a set or a sequence of actions to predict the output. It can be used in a dynamic environment in which it must perform a certain goal, such as driving a vehicle.

Regarding the algorithms output, it is divided into three different categories [83]: classification, regression and clustering. Classification builds a model to classify the inputs into a class label, which can be two or more. Typically it is used in supervised learning. An example can be an email spam filter, that classifies an email as "spam" or "not spam". Some examples of algorithms that use this type of output are: SVM, k-Nearest Neighbour, Naïve Bayes, Neural Networks, among others.

Regression builds a model to predict continuous data rather than discrete, meaning that it can be a real value or a vector of numbers and is used for weather forecasting or stock prediction. Some examples of algorithms that use this type of output are: Linear Model, Neural Networks, Adaptive Neuro-fuzzy learning, among others.

In clustering, the algorithm tries to find natural groups and patterns in the inputs. Unlike in classification, the groups are previously unknown, making this typically an

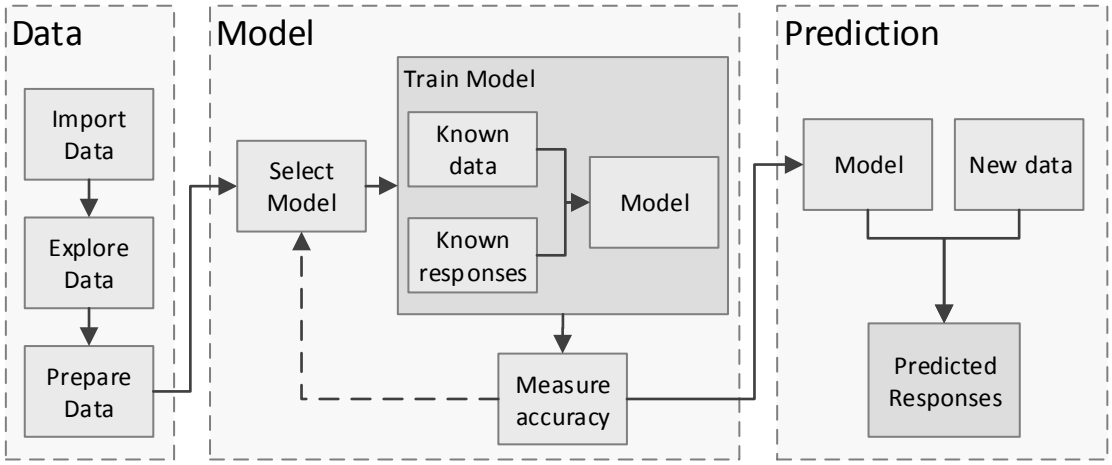


Figure 15: Supervised learning flowchart

unsupervised task. Some examples of algorithms that use this type of output are: k-Means, Hidden Markov Models, Gaussian Mixture Models, among others.

In order to validate a learning model, a dataset must be divided into three sub-sets: (i) training; (ii) validation; (iii) test. It is also important to use a cross validation using these datasets.

The main problem of machine learning algorithms is that they can overfit or underfit. Overfit occurs when the dataset is too large to be properly modelled. Underfit occurs when the dataset is very limited in terms of different case scenarios.

Another problem can be the number of dimensions, which can lead to high variance, meaning that the algorithm does not generalize well. Also, by having too many dimensions (d) the algorithm becomes computationally heavy and slow to answer, because it has a complexity of $O(2^d)$.

2.4.2.1 *Neural Network*

A neural network is a learning algorithm that is inspired by the structure and functional aspects of biological neural networks. A neural network is composed by simple neurons that operate in parallel. As in nature, the network function is largely determined by the connections between the neurons. The network can be trained to perform a particular function by adjusting the weights (values of the connections) between the neurons.

These networks are normally trained to solve complex problems that are difficult for conventional computers or humans. They can be applied in several fields of application, including pattern recognition, identification, classification, speech, vision and control systems.

Commonly, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. They find patterns in data or capture the statistical structure in an unknown joint probability distribution between observed variables. This process is demonstrated in Figure 16, where the network is adjusted, based on a comparison between the neural network output and the known target, until they match each other. These matches are found by adjusting the weights between the connections of the neurons.

A simple linear neuron model is represented in Equation 23.

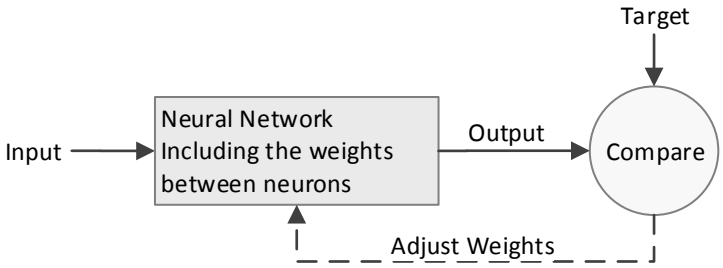


Figure 16: Neural networks architecture

$$y = b + \sum_i x_i w_i \tag{23}$$

where each i^{th} input x is multiplied by a corresponding weight w and these values are summed together. Then the result is summed to a bias b to achieve the output y.

The main idea of a neural network is that weight and bias parameters can be adjusted so that the network exhibits some desired or interesting behaviour. Thus, a network can be trained to do a particular job by adjusting the weight or bias parameters, or perhaps the network itself will adjust these parameters to achieve some desired end.

The amount of change performed to the weight and bias values, during the training, is controlled using the learning rate parameter. It must be reduced when the predictions error is getting worse or if it is oscillating too much. However, if the error is slowly decreasing the learning rate should be increased. At the end of the learning phase the learning rate must be lowered, but the amount of decrease must be done with careful. If it is done too soon a quick win can be achieved or a slower learning can be obtained.

The output is then passed through a transfer function. There are three types of transfer functions: (i) hard-limit; (ii) linear; (iii) log-sigmoid. The hard-limit transfer function, illustrated in Figure 17a, limits the output of the neuron to 0, if the y is less than 0, or 1, if y is greater than or equal to 0. The Linear transfer function, illustrated in Figure 17b, takes the input, which may have any value between plus and minus infinity, and squashes the output into a range between 0 and 1. The Log-Sigmoid transfer function, illustrated in Figure 17c, is commonly used in backpropagation networks, because it is differentiable.

Figure 18 represents an example of a Neural Network notation that will be used throughout this document. The input vector p is a vector of n input elements. These inputs are multiplied by a weight w. The result is then summed with a scalar bias b.

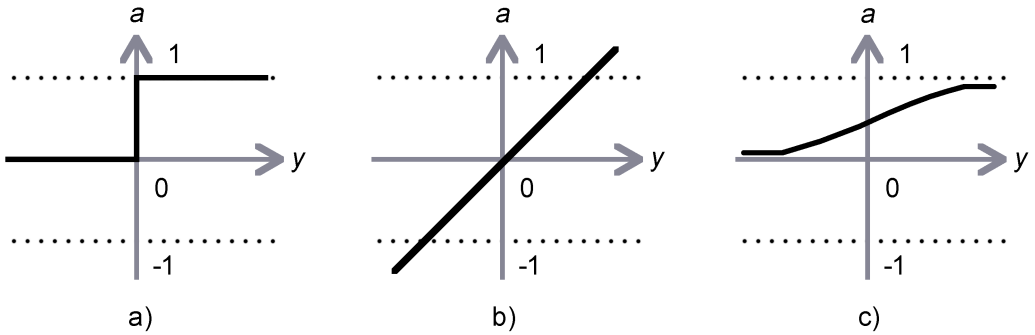


Figure 17: Neural network transfer functions: a) Hard-limit function; b) Linear function; c) Log-sigmoid function

The input of the network to the transfer function f is y. From the transfer function is given the neuron's output a. If the represented notation had more than one neuron, the network output would be a vector.

Typically, to improve the networks performance when presented with new inputs, it is used the backpropagation. It is called backpropagation since it is a gradient descent algorithm in which the network weights are adjusted backward, starting from the output layer and going to the input layer. The term backpropagation refers to the manner in which the gradient is computed for non-linear multilayer networks. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way.

Recently, algorithms based on neural networks have been suggested and applied to different types of INS [86]. For example, it was shown that a neural network could process the INS azimuth and velocity to provide the localization with an accuracy improvement of 20% [34].

In PINS, an accelerometer provides signals with which a neural network can be trained in order to predict step length or to create a step model for a person. With this model the PINS errors can be mitigated, since the system will be adapted to the

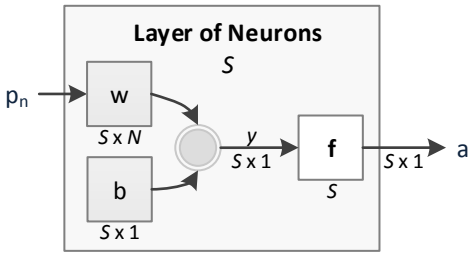


Figure 18: Neural network notation that will be used throughout this document

pedestrian characteristics. However, the inconvenient of this approach is that the device must be worn by the same user, since the step model is trained with the particular individual's walking patterns.

2.4.2.2 *Support Vector Machine*

SVM is a supervised learning algorithm used for classification and regression [30]. This algorithm builds a model that classifies the data into one of two classes, making it a non-probabilistic binary linear classifier. Typically this algorithm does not work well with large datasets or with too noisy data.

The model, created by a SVM, classifies the data by finding the best hyperplane that separates all the data points from one class to the other. The best hyperplane is the one that has the largest margin between the two classes. Margin means, the maximum width of a parallel to the hyperplane that has no interior data points. As the margin is greater, the best is the model. New data is then mapped into the same space and its class is predicted based on which side of the hyperplane it falls on.

Figure 19a illustrates these margins in a linear separable problem. The "$\times$" represent the points of type 1, and the circles represent the data points of type -1. As in the Neural Networks, the w represents the weight vector given to the input x and b is the bias. The result of the product between the weights and inputs with the sum of the bias indicates the category where the point belongs, 1 or -1.

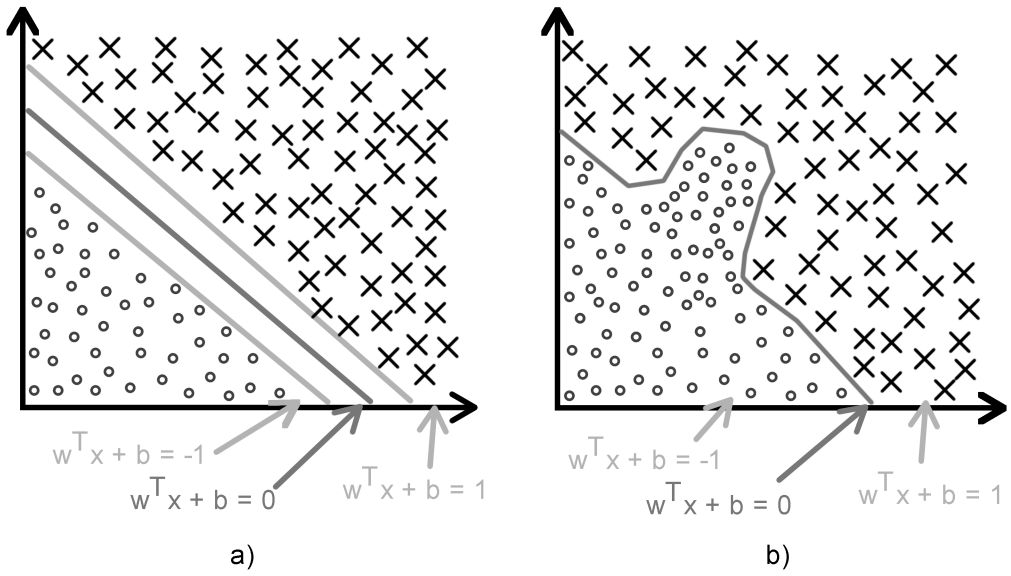


Figure 19: SVM hyperplane separation between two classes: a) Linear separation; b) Non-linear separation

Besides linear classification, like the one presented in Figure 19a where the data is linearly separable by a hyperplane, SVMs can also perform non-linear classifications as illustrated in Figure 19b. To perform the non-linear separation, the so called kernel trick must be used. With the kernel a linear separation can be transformed into other dimensions or plans. Thus, the kernel can be *linear*, *poly* or *rbf*. The *linear* kernel performs a linear separation of the data using a hyperplane. The *rbf* kernel uses a Gaussian radial basis function to separate the data. The *poly* kernel uses a polynomial space to separate the data.

When using the SVM algorithm, it is very important to choose the appropriate kernel to successfully train the SVM classifier. Besides the appropriate selection of the kernel, it is also important to properly select the kernel parameters, namely the *C* and *gamma* (γ) parameters. Typically, each combination of parameter choices is checked using cross validation, and the parameters with best cross validation accuracy are chosen.

The *C* parameter is used during the training phase and indicates how much outliers will be considered when creating the SVM model. Meaning that it trades off misclassification of training examples against a more simple decision surface [2]. If this parameter is configured with a high value, the SVM will classify all the training examples correctly. However, too many support vectors will be created and it will overfit. If the *C* parameter is configured with a low-value, the decision surface will be smoother. However, it will cause the optimizer to look for a larger margin hyperplane, even if that hyperplane incorrectly classifies more points, leading to underfit.

The *rbf* kernel has one more parameter than the others, that is the *gamma*, which represents the radius of the *rbf*. It defines how far the influence of the single training example reaches. A small *gamma* gives a pointed bump in the higher dimensions, and a large *gamma* gives a softer broader bump. So, a small *gamma* will give a low-bias and high-variance while a large *gamma* will give a higher bias and low-variance.

Using SVM for multi-class classification can be possible through the use of several binary classifiers which will distinguish between, one of the labels and the rest, or between every pair of classes. The first approach classifies the new data by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class. The second approach classifies the new data by a voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally, the class with the most votes determines the instance classification.

2.5 SUMMARY

In this chapter was firstly introduced how the human gait works. It is essential to understand how it works, as it is a cyclical event, because it is through the motions of the human body that the distance and orientation are estimated. Human gait is composed by two phases, stance and swing. It is very important to properly distinct these two phases, since the displacement of a human must only be estimated during the swing phase. When a pedestrian is in the stance phase, it is not moving, thus any displacement must be considered.

When in the swing phase, the distance and orientation could be estimated through the presented algorithms. The most used and reliable technique is the *Strapdown Inertial Navigation System*, which estimates the travelled distance based on the acceleration data obtained during each step, and not only based on the number of steps.

Stance phase is also useful to apply some techniques, e.g. ZUPT and ZARU, to reduce the algorithms errors and estimate the new state of the variables.

The fusion between sensors can be performed using a Kalman filter. However, a new trend is to use machine learning algorithms applied to the data obtained from the sensors, to learn some patterns of the system to then correct some deviations that can occur due to the MEMS sensors low accuracy.

Concluding, in this chapter it was presented the fundamentals about INSs, which are, or can be, used when creating a system of this type.

3

PEDESTRIAN INERTIAL NAVIGATION SYSTEMS

In this chapter some unassisted PINS will be presented. Unassisted PINS are systems that do not rely on any external source, or in a map, to correct the location estimation. Also, they were chosen based on their importance and new ideas that they have introduced on this research area.

The analyzes of the PINS are organized based on the algorithms used for sensor fusion. In section 3.1 are presented the systems based on a Kalman filter and its variants. In section 3.2 are presented the systems that do not rely on a Kalman filter to fuse data, but in other techniques.

Figure 20 provides a graphical overview about all the PINS that are discussed in this section. This graphical representation shows where the sensors are mounted, on waist or on foot, the used sensors and the applied fusion techniques. The sensors aggregation is represented by the circle symbol with a plus sign and the ellipse represents an aggregation of some systems. For each flow of data there is a number that indicates a reference to a set of literature works that uses that specific sensor or technique.

3.1 KALMAN FILTER BASED APPROACHES

Most PINS use a Kalman based filter to fuse the sensors data from sensors and thereby to estimate pedestrian location. From the studied systems it was identified that EKF is the most used Kalman variant. This section is divided in the following sub-sections: in Section 3.1.1 are presented the systems that are based on a Kalman filter; in Section 3.1.2 are presented the systems which are based on an EKF; and in Section 3.1.3 are presented the systems based on an IKF or on an UKF.

3.1.1 *Kalman Filter*

Ladetto et al. [64] developed two IMU prototypes to identify the differences between using a gyroscope or a magnetometer to estimate orientation. Both prototypes have

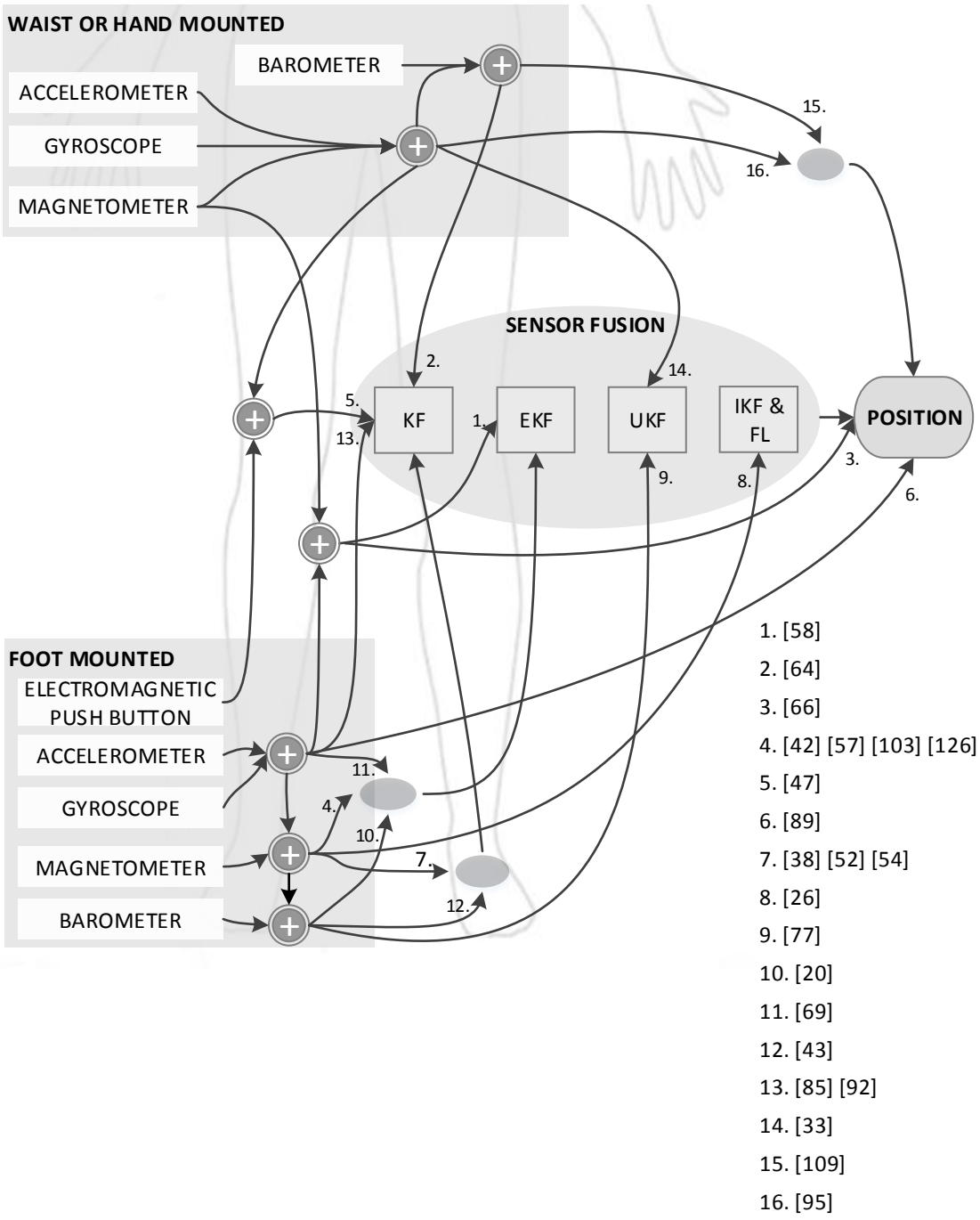


Figure 20: Configuration of the studied PINS

their strengths and weaknesses, but, as stated earlier, these sensors can complement each other. Besides these sensors, both prototypes have an accelerometer, a barometer and a GPS receiver. The IMUs were placed, on the person, at the lower back, which the authors considered to be a relatively stable position while walking.

The walking distance and the orientation, sensed by inertial sensors, are combined with GPS observations through a Kalman filter. When GPS is available, the models for the step length are calibrated, as well as, the magnetometer and gyroscope outputs. The filter is used for both the initialization and the on-line calibration of those parameters.

In the gyroscope prototype the raw signal is firstly processed with a low-pass filter to eliminate random errors. This technique allows the identification of different circumstances during a path (e.g. short curve, long curve, straight line, change in velocity). The goal is to eliminate the noise and the periodic changes while keeping the significant changes of the walking line.

Authors conducted several tests with both prototypes. These tests have shown that the weaknesses of one system are the advantages of the other, since gyroscope can provide useful information to identify magnetic disturbances, while the magnetometer is useful to determine the bias of the gyroscope and the initial heading, even when no GPS signal is available. According to this remark, a more reliable system will be obtained by coupling the gyroscope with the magnetometer.

In the evaluation test, which involved a walking path with a distance of 3000 meters, authors claim an error of 5.2 meters for the magnetometer prototype and 25 meters for the gyroscope prototype.

Hamaguchi et al. [47], in 2006, described a localization method that uses wearable electromagnetic sensors (Figure 21). The proposed method estimates orientation using a magnetometer on the waist, and the geometrical relationship between user's heel through an electromagnetic tracker attached to user's body. When both feet come in contact with the ground, user's position is updated by adding the estimated step length.

The pedestrian is equipped with electromagnetic sensors in the leg, an IMU (composed by an accelerometer, a gyroscope and a magnetometer) at the waist, and push button switches attached to heels. Using these sensors the relative position of the left and right legs, and absolute orientation of user's hip, are estimated continuously. The electromagnetic tracker measures the relationship between right and left foot. However, the authors claim that when the user is running the system does not work properly.

The proposed localization method has been evaluated in a walking path with 200 meters (in a square path) and presented an error of 5.2%.

Feliz et al. [38] developed a method to avoid PINS accumulated errors or at least to reduce them. The IMU was attached to the user foot, and the detection of the swing and stance phases is performed using the values of angular velocity provided by the

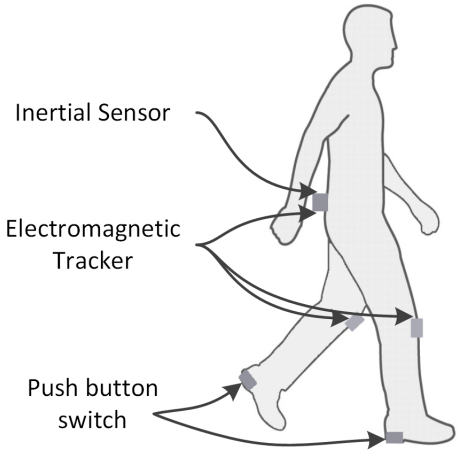


Figure 21: Sensors and their placement in the system proposed by Hamaguchi et al. [47]

gyroscope. This filter takes in consideration the values of the previous and subsequent measurements.

Once a signal is obtained the integration and correction of acceleration measurements is performed. This filter is able to identify the start and the end of the stance phase and consequently the acceleration corrections are performed in the whole period. The proposed method estimates errors at each step ensuring that the estimated distance will be very close to the actual walking distance, producing smaller errors. This means that it will not influence the speed and position estimation for the next step. So, the inertial system's inherent accumulated error is corrected.

Authors tested the system in a straight corridor without any metallic objects around, which involved a curve of π rad and the return to the starting point. The system gave an estimation error of 9% in a walking with a distance of 30 meters.

Gadeke et al. [43] studied a concept to integrate a foot-mounted low-cost IMU, via Bluetooth, with an Android smartphone. Unlike, previous works the authors set high importance on the actual system integration and implementation issues without being limited to a Personal Computer (PC) based concept. The proposed IMU is composed by an accelerometer, a magnetometer, a gyroscope and a barometer. A stochastic filter was used to correct drift based on an underlying system model. Additionally, ZUPT and ZARU mechanisms were introduced to correct the velocity and angular rates in the system model.

Obtaining altitude from atmospheric pressure measurements is a fairly complex procedure as it depends on various unknown parameters in the atmosphere. Meaning that the pressure greatly varies with weather conditions. Obtaining an absolute altitude over sea level is not accurate enough under these circumstances. However, differential

pressure changes are very accurate, so the authors used these variations to estimate the user elevation on the building.

With today's smartphones high processing power and memory capabilities, data collection using mobile devices can be a very handy method. Walking with a smartphone for data collection also allows a more accurate walking pattern since there is not other equipment carried by the person, that can influence the walking characteristics.

An experimental study was performed to compare the accuracy of the proposed IMU with a commercially available IMU (Xsens MTI-G). The evaluation to both IMU was performed inside a typical office building and both IMUs were fixed on the same pedestrian's foot. The trajectory started on a second floor and involved a downstairs to the first floor, a walk around and an upstairs to the starting point. With a walking distance of 82 meters the error was of 2.2 meters for the low-cost IMU and 1.8 meters for the MTi-G.

Nilsson et al. [85] have created an open-source, real-time, embedded implementation of a foot-mounted PINS. The implementation includes software and hardware designs. The IMU is composed by off-the-shelf components, including an accelerometer and a gyroscope, and the respective assembly methods. A Kalman filter, assisted with a ZUPT algorithm, is used to estimate the walking distance.

The system is small enough to be integrated in the sole of a shoe, to get as close as possible to the contact surface between the user and the ground in order to achieve the best performance.

An evaluation was conducted, namely a 100 meters straight-line walk and a 100 meters figure eight walk. These trajectories were taken on a normal walking pace on a solid floor. The error claimed for this system was 1 meter, which corresponds to 1% of the straight-line distance. The figure-of-eight trajectory presented an error of around 0.15 meters.

Hoflinger et al. [52], in 2013, developed a wireless IMU, which eliminates the need for wired data communication. It can send the collected data to a base station with a maximum of 640 samples per second that post-process the acquired data.

This IMU is composed by an accelerometer, a gyroscope and a magnetometer that features automatic temperature compensation technology. It is placed on the pedestrian's shoe and the sensor fusion is based on a modified Kalman filter combined with ZUPT. This modified Kalman filter models the system state, the orientation error and the angular rate error. In the orientation estimation, the gyroscope angular rate data is corrected by subtracting the bias error estimated by the Kalman filter. Then, the mag-

netometer and accelerometer data are fused with the previous one. By processing the magnetic field and acceleration measurement data, authors detect and minimize the magnetic field disturbances in the implemented filter.

The proposed IMU was compared with Xsens MTi. The data from both IMUs was collected over a period of 4 hours. After evaluation, it was observed that noise and drift values of the accelerometers are in the same range, except for the *z-axis* in which the Xsens MTi performs better. The overall performance of the proposed IMU has shown to be comparable to the Xsens MTi. However, to estimate the walking path it relies on heavy post-processed algorithms [52].

To evaluate the performance of both IMUs a pedestrian walked three times on a round trip inside an office building, walking a distance of about 60 meters. The proposed system presented an average error of 1%. The trajectory estimated by the authors IMU was closer to the reference trajectory, than the trajectory estimated by the Xsens MTi IMU.

Prateek et al. [92] used two foot-mounted IMUs, one on each foot, aided with ZUPT. The proposed algorithm is based on the idea that the separation between the two feet at any given instance must always lie within a sphere of radius equal to the maximum possible spatial separation between the two feet (see Figure 22). A Kalman filter is used, which receives one measurement update and two observation updates.

Authors use two OpenShoe [85] units integrated in the sole of each shoe, and an algorithm to unify the trajectories obtained from both foot-mounted IMUs aligned in the same direction using an upper bound in the spatial region. Even though both IMUs were aligned in the same direction, but on different feet, the two trajectories take two

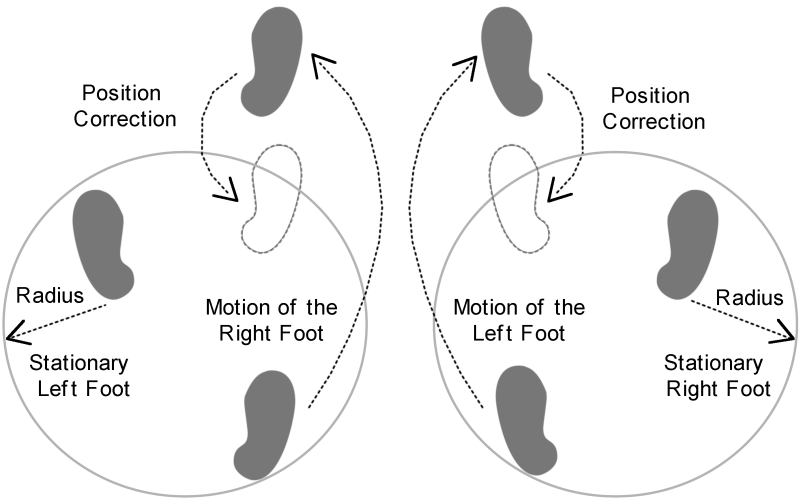


Figure 22: Spatial separation between the two feet algorithm proposed by Prateek et al. [92]

different paths. Also, when studying the motion of the feet on walking or running, authors have notice that the separation between the two feet does not exceed a certain threshold value.

Experiments were conducted in several different paths, such as a 'U' and an 'L' shaped path with sharp $\frac{\pi}{2}$ rad curves. Authors observed that the proposed algorithm with initial heading estimation gives the best trajectory information which is closely similar, in shape, to the actual trajectory.

The proposed system was also compared with the one proposed by Nilsson et al. [85]. This evaluation involved a walk of 110 meters on level ground along a straight line at a normal walking speed. Authors claim that the mean and covariance of final position estimates are significantly reduced by applying the range constraint. The estimation error was 1% of the walking path.

Ibarra-Bonilla et al. [54] presents a PDR algorithm that uses a fuzzy logic Kalman filter for attitude estimation and a fuzzy inference system for step length estimation, which characterizes frequency and step length. Kalman filter carries out a measurement fusion of accelerometer, gyroscope and magnetometer with a fuzzy adaptive tuning.

The general idea behind this technique is that the actual value of the covariance of the residuals matches its theoretical value. When statistical values of the innovation sequence shows discrepancies between theoretical and actual covariance, then a fuzzy inference system adjusts a tuning factor applied on the matrices causing a reduction in the discrepancy. The adaptation process is carried out through the adjustment of the measurement noise covariance matrix. Assuming that the noise covariance matrix is known, an algorithm employing the principles of fuzzy logic is used to adaptively adjust the result matrix.

The results obtained from the comparison between the fuzzy adaptation and the traditional Kalman filter have shown an improvement in the adaptation of the measurement noise covariance matrix. Final results from the fuzzy logic-based PDR system with the described attitude estimation algorithm have presented an error of 6.4% in a walking path with a distance of 100 meters.

3.1.2 *Extended Kalman Filter*

Jirawimut et al. [58], in 2000, presented a system that combines a pedometer to measure the relative distance, and a magnetometer to estimate heading. When GPS signal is

available it provides the position and velocity of the pedestrian, but this measurements are also used to calibrate the DR parameters: step size and magnetometer bias error. It uses an EKF which returns the position and velocity, together with the step size and the magnetometer bias error at each GPS update. When a GPS signal outage occurs, the DR algorithm uses the calibrated parameters to provide relative positions in relation to the last known location. The system has a latency of about 10 seconds, the time period since a step is detected until the estimations are presented to the user.

An evaluation test involving a path with 591 steps was performed but no accuracy values were presented by the authors. Authors claim that the use of an EKF increased the positioning accuracy when all measurements were available. However, when the DR was used solely, the accuracy was decreasing with time. Heading was estimated effectively, but the step size was estimated efficiently only when the GPS signal was available.

Foxlin [42], in 2005, presented NavShoe, which is an IMU placed on the foot that applies the ZUPT technique at each time the foot is stationary. This IMU is composed by an accelerometer, a gyroscope and a magnetometer. Each sensor threshold was tuned to ensure that the immobilization time is never higher than 0.1 seconds during the swing phase, but lasts almost 0.3 seconds during the stance phase. ZUPT is applied as soon as the immobilization time exceeds 0.15 seconds and continues as long as the foot remains stationary. However, the author claimed that the system needs another approach for running motions.

An EKF, with 13 calibration parameters, is used to correct the velocity error after each stride, breaking the error growth, meaning that the error accumulation is linear with the number of steps. ZUPT measurements let the EKF correct position, velocity, accelerometer bias and gyroscope bias.

Before a user starts using the system it needs a calibration procedure of about 30 seconds, where the user must rotate in some orientations while translating as little as possible to maintain the sensor in a constant external magnetic field. Authors have performed an indoor and outdoor experiment where a user walked through a typical wood house during 322 seconds, covering a total distance of 118.5 meters, with an error of 1%. However, this test was performed in an ambient without any magnetic disturbance.

Sabatini [103] presents a shoe-mounted IMU to estimate the walking path in three-dimensions, using ZUPT and a quaternion-based adaptive EKF. The DR algorithm allocates different roles to the IMU sensors. The gyroscope has the task of provide short-

term accurate estimates of the state vector. During each EKF run, the acceleration measurements, taken from a swinging leg, are in fact prevented from influencing the filter behaviour by properly applying a measurement validation test. However, this measurement validation test tends to increase the measurement noise variance of the magnetic sensor when it is exposed to external magnetic disturbances.

In order to make the filtering process robust against these disturbances, which may affect the IMU sensors, several tricks were considered. It adapts the measurement noise co-variance matrix during EKF runs and implements an adaptive alternation between initialization and run of the EKF, which is based on stance/swing phase detection driven by gyroscopic data, and acceleration/magnetic measurement validation. A bias-correction to the magnetic sensor measurements is performed before each EKF run. Finally, an integration of gravity-compensated acceleration components is performed once every step is detected.

The system evaluation consisted in a walking along an outdoor path with a distance of 680 meters, with overall changes in altitude of 40 meters. The system has presented an error of 0.6%. The walk was in a stop-and-go fashion: walk ten consecutive strides before taking a brief rest and then walk another ten consecutive strides, and so on. In conclusion, the author says that it is the foot yaw angle estimation that is critical, which may lead to weird results especially when the gyroscope integration is left free running.

Jimenez el al. [57] proposes a low-cost IMU that integrates a Heuristic Heading Reduction (HDR) algorithm into an EKF, which can represent a valid methodology for operate in spaces with magnetic disturbances. The IMU is placed on the foot and is composed by an accelerometer, a gyroscope and a magnetometer. It uses an EKF to estimate the PINS errors, which is updated with velocity measurements every time the foot is on the floor. Also, it includes a stance phase detection algorithm, ZARU and a HDR. The system architecture can be seen in Figure 23.

The three conditions to declare a foot as stationary (stance phase) are: magnitude of the acceleration must be between two thresholds; the local acceleration variance, which highlights the foot activity, must be above a given threshold; and the magnitude of the gyroscope must be below a given threshold. These three logical conditions must be satisfied simultaneously and the result is filtered out using a median filter with a neighbor window of 11 samples.

HDR makes use of the fact that many paths are straight and the idea of HDR is to detect when a person is walking straight and, in that case, to apply a correction to the

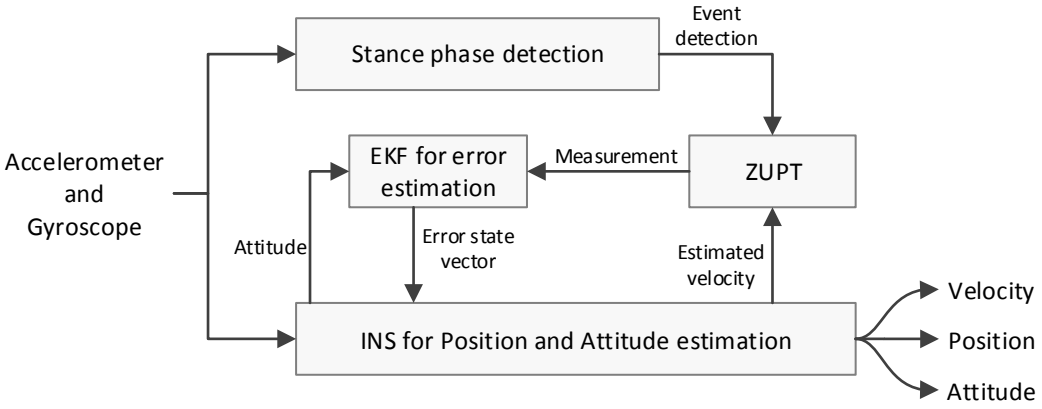


Figure 23: System architecture proposed by Jimenez el al. [57]

gyroscope biases, in order to reduce the heading error. The system was tested in an indoor environment which include counter-clock-wise and clock-wise directions. The path was 125 meters long and authors concluded that the uncorrected heading drift is proportional to the walking distance but not to the elapsed time. The positioning error was 1.5% of the total walking distance.

Bebek et al. [20] added a high-resolution thin flexible tactile sensor to an IMU (composed by an accelerometer, a gyroscope, a magnetometer and an EKF algorithm), which measures zero-velocity to reset the accumulated integration errors from accelerometers and gyroscopes. The tactile sensor data can be used to accurately detect periods of zero-velocity to increase the efficiency of the location resolution.

Compared to the other proposed approaches that detect the step impact with accelerometers, this one uses a more reliable sensor to perform the step detection. The sensors are placed within the heel and the sole of a pedestrian's boot and are wirelessly connected to a handheld unit, that processes the data off-line. The tactile sensor is placed very close to the point of the heel-to-ground contact to provide detailed contact information to the IMU.

The proposed system works in the following way (Figure 24): first, acceleration and orientation information are retrieved from the IMU; next, an EKF is employed to detect the acceleration biases. The bias-compensated acceleration, from the IMU, and zero-velocity points, from the tactile sensor array, are used in the integration via ZUPT to estimate the position of the user; finally, a calibration is applied to correct the drifts in this estimated location.

To evaluate the system a close-loop walk with a total distance of 1215 meters was performed. First, a 5 minutes walk was performed to calibrate the system. Next, the

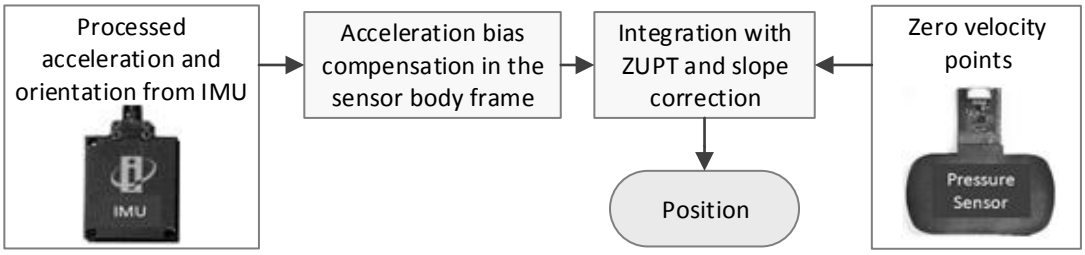


Figure 24: PINS architecture proposed by Bebek et al. [20]

walk test was performed in a sports field with a flat surface, which resulted in an error of 0.4%. Authors concluded that the inclusion of a tactile sensor improves the step detection.

Li and Wang [69], in 2012, studied the PINS behaviour when the pedestrian is running. The running gait cycle is shorter than the walking gait cycle, and the stance phase is less qualified as zero-velocity. In the walking gait, the pre-swing and swing phases are mirrored by the initial contact and stance phases. While one leg is in the pre-swing phase, the other is in the initial contact phase, and while one is in swing phase, the other is in stance phase. This symmetry in the walking gait shows the rhythmical transfer of weight between the two limbs. During the running gait cycle this symmetry does not exist. Instead, both legs can be in the swing phase at the same time and heel strike occurs while the other leg is still in the air.

Two Zero Velocity Detector (ZVD), ZVD1 and ZVD2, were proposed. The ZVD1 detects zero-velocity while walking by analyzing, in real-time, four conditions on both the accelerometer and gyroscope measurements: the magnitude of the total acceleration and the acceleration on the *z-axis*; the magnitude of the gyroscope data and the gyroscope's *y-axis*. When zero-velocity is detected, velocity, position and attitude errors are reset to zero. The accelerometer and gyroscope errors and gravity uncertainty are calculated and feedback into the EKF which corrects the navigation error afterwards. The ZVD2 is designed for the running cycle where only gyroscope measurements are used. The zero-velocity is determined once the norm of gyroscope is less than a predefined threshold. Once a footstep is detected both ZVD1 and ZVD2 are applied in the first half step, while in the last half step the ZVD1 is used to prevent some mistakenly detected zero-crossings in the swing phase.

Combining the detectors this proposal improved the stance phase detection for both walking and running gait cycles. Tests were conducted in a close loop of 171 meters. This experiment was performed with different human gait cycles which includes walking, running and walking alternating with running. Before walking the subject stands

still at the start point for about 4 seconds, in which the ZUPT algorithm sets the parameters and the initial conditions. The system presented an error, for the total walking distance, of 1.22%. Authors concluded that the navigation accuracy of the proposed algorithm for running cases is comparable to the walking only cases.

Yuan et al. [126] carried a set of experiments to test the differences by having an IMU mounted on the person's waist or on the shoe. Also, another set of experiments was performed to analyze the heading estimation by using only a magnetometer, only a gyroscope, and a quaternion-based EKF with a gyroscope and a magnetometer.

An IMU, composed by a accelerometer, a gyroscope and a magnetometer, was fixed through a belt in the waist and another, equal to the first, on the shoe. Authors have concluded that the IMU on the shoe has higher acceleration peaks than the IMU on the waist. Therefore, the step count and step length can be estimated more precisely on the IMU placed on the shoe than on the waist. Also, zero-velocity can be better detected on the foot.

Due to magnetic disturbances and gyroscope drift, authors proposed a quaternion-based EKF for heading estimation by combining the magnetometer and gyroscope data. Because of their complementary characteristics, a better accuracy in the heading solution was obtained. However, authors have not presented any performance metrics in order to compare their proposal with other systems.

3.1.3 *Other Kalman Approaches*

Mathiassen et al. [77], in 2010, focused on the possible gain that a barometer, in addition to the inertial sensors, can have a three-dimensional localization estimation. Authors developed a low-cost IMU that is constituted by a GPS, an accelerometer, a gyroscope, a magnetometer and a barometer. When GPS loses signal, the other sensors are used to keep track of the position and an UKF was used to integrate data.

To remove some of the flicker noise from the gyroscope measurements the authors used a 10^{th} order high pass Infinite Impulse Response Butterworth filter [107]. The magnetometer measurement integrated with the gyroscope data approach uses the fact that rotating the magnetic field vector in the body frame to the navigation frame with quaternions should be equal to the magnetic field vector obtained by the International Geomagnetic Reference Field model (which is a model of the earth magnetic field).

Combining inertial sensors with a barometer the system presented an error of approximately 13 meters after 60 seconds of walking, compared to an error of 58 meters without the use of a barometer.

Castaneda and Lamy-Perbal [26] proposes a shoe-mounted IMU with a fuzzy logic based algorithm to improve stance phase detection. Also, they propose a ZARU algorithm and the associated measurement model to correct drift errors via an IKF. The fuzzy-logic algorithm detects the stance phase based on gyroscope and acceleration measurements. In practice, in order to avoid false step detections, the authors decide that a step is only detected if the proposed algorithm detects no movement for a predefined time period.

Inspired in the ZUPT algorithm, authors propose a ZARU algorithm based on the fact that at the stance phases the IMU's attitude may be partially estimated through coarse alignment methods using acceleration measurements. Furthermore, it is possible to estimate the attitude pseudo-measurement, in a stance phase position, only during the system initialization procedure. The estimation is then used, each time a step is detected, as a reference attitude to feed the IKF, that will correct drift errors.

The system was tested in three walking scenarios. The first one had a forward walk then a left curve and a downstairs, with a total walking distance of 40 meters and presented an error of 30 centimeters. The second experiment was a walk along a square of 6 by 6 meters, with a total walking distance of 24 meters and presented an error of 15 centimeters. The third experiment involved a forward walk with a curve to the left then a curve to the right, giving a total walking distance of 84 meters and presented an error of 27 centimeters.

Diaz et al. [33] propose a PINS based on an IMU located in the trousers' pocket of a person. Authors say that this location was chosen because, for the mass market it is more convenient to integrate a navigation system in a mobile phone. This location is more practical and less obstructive than attaching an extra device to the shoe.

For the orientation estimation an attitude estimator based on an UKF was developed. This estimator receives the data from the accelerometer, gyroscope and the magnetometer. A zero acceleration detector was implemented, since there is not a foot mounted IMU, therefore it is difficult to detect zero-velocity. So, this detector finds locally the periods of zero or close to zero acceleration.

The position estimator is based on an UKF and it is composed by a step detector and a step length estimator. The step detector is based on the detection of consecutive minimum-maximum of the low-pass filtered norm of the acceleration. For the step

length, authors have decided to use the empirical relationship between the maximum and minimum of the acceleration for each step.

The navigation system has been tested by different users with different type of trousers and without any restriction about the tightness of the trousers. The attitude estimator presented a drift of π rad in a walk of approximately 200 seconds, and the navigation system presented an error of about 4 meters in a walking path of 200 meters.

3.2 OTHER FUSION APPROACHES

Lee and Mase [66], in 2002, proposed a system that uses acceleration and angular velocity data to determine user's location, and to recognize and classify sitting, standing and walking behaviours. The proposed recognition method consists in two phases, training and recognition. In the training phase, the system requires a set of data to determine the motion parameters for the three behaviours. The system records motions and heading measurements while the user walks from one location to another, building a location transition table. In the recognition phase, the system continuously tries to find motions and recognize a location transition from a known starting location.

The system is composed by two IMU, one in the leg and another in the waist. The leg module is located in the user's right or left trouser pocket and measures the acceleration and angle of the user's thigh. The other module is attached to the middle of the user's waist and estimates orientation. After some experiments, authors concluded that these are the best positions for activity and location recognition.

The leg module is composed by an accelerometer, to measure the forward and upward accelerations of the user's thigh, and a gyroscope, to measure the angle of the user's thigh movement. The waist module is only composed by a magnetometer.

The proposed recognition algorithm is composed by three modules: a sensing module, a motion recognizer and a location recognizer. The sensing module reads the data from the sensors and executes a set of preprocessing tasks, including filtering and computing statistical properties. The motion recognizer identifies the predefined types of motion. The location recognizer calculates the current displacement vector by dead reckoning and tries to find this location in the location transition table. If it finds a matched location, it changes the user's current location. An evaluation to the system was performed where a user performed 978 steps, and the system presented an error of 3%.

Ojeda and Borenstein [89] developed a small IMU capable of estimate location in the three-dimensions. It was designed with military and emergency responders in mind. The software is composed by three modules: a position estimation module, ZUPT and a step detection module. The IMU is attached to the user's boot and incorporates an accelerometer and a gyroscope that are used to estimate, in real-time, the user location relative to a starting point.

Since the IMU is attached directly to the boot, tilt angles of $\frac{\pi}{2}$ rad or higher are likely to happen. For this reason authors choose the Quaternion representation, which handles tilt angles. In the experiment five different pedestrians walked along a rectangle-shaped path with a total distance of 104 meters. For this path the proposed system presented an estimation error of 0.7%. Authors concluded that a PDR system can be based on a small-sized IMU, despite that IMU's performance limitations.

Qian et al. [95] proposed a system based on smartphone's inertial sensors and camera. Camera, is a common sensor on most current smartphones, so it provides opportunities to improve the inertial sensors errors by using an optical flow algorithm. Optical flow is the pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. Using sequences of ordered images recorded by the camera, the pedestrian motion can be estimated. This optical flow algorithm was used to estimate step length.

However, the use of a camera brings some challenges, like the interferences arising from hands shaking during walking. Also, by having a smartphone in a hand it is impossible to have it parallel to the ground all the time. In this case, the displacement calculated by optical flow algorithm depends on the angle between the smartphone and the ground. Therefore, the smartphone's attitude during walking is acquired to be combined with the optical flow estimation to obtain more data.

Authors made five field tests, three outdoor and two indoor, with different types of ground textures such as asphalt road and lawn. All the trajectories were straight lines. Authors have not considered curves and stairs situations. The mean error for each step was of 1.627 centimeters.

Shin et al. [109] suggested a three-dimension PINS using a smartphone, where the motion of a pedestrian is firstly estimated by a hybrid model classifier. Then, the two-dimension location of pedestrian is estimated by using a PDR algorithm and a barometer is used to estimate the altitude (three-dimension location).

The motion of a pedestrian is detected when a peak exists in the norm value of the accelerometer signal. It is then passed through a hybrid model classifier which

classifies it as a true or a false step. According to the authors a single classifier could not provide accurate recognition rate because the features extracted from sensor data are different for each person. To overcome this limitation three different classifiers were combined.

When a true step is detected a fixed step length is used to estimate the pedestrian displacement. The double integration method was not applied because of the smartphone accelerometer low performance. Tests were performed in a building with four floors and a rectangle shape. Three pedestrians have participated in the experiment and the positioning error was less than 5%.

3.3 DISCUSSION AND ONGOING CHALLENGES

Table 2 presents an overview about the main characteristics of the studied systems namely: the sensor fusion methods; the sensors used and where they are mounted; if they use ZUPT technique; their distinctive features; the evaluation that each author has performed to the system; and the respective positioning error.

PINS try to estimate the person location using the equations of motion through the acceleration information. However, a PINS working by itself cannot keep a good location accuracy over long periods of time. This happens because of the sensors drift and people's different ways of walking.

Therefore the studied approaches consisted on unassisted PINSs that have applied several different techniques to reduce the PINSs typical errors. One approach, to compensate PINS errors, suggested by Jirawimut [58] and, Lee and Mase [66], is to calibrate the PDR parameters, the step size and magnetometer bias error, using the GPS signal when the user is in outdoor environments.

The gyroscope bias causes the most orientation estimation errors so a good practice is to calibrate the gyroscope before each experiment. These errors happen more frequently when a low-cost IMU is used, which can produce results 3 times worse than a high cost one. As concluded by some authors [26] [57] the uncorrected heading drift is proportional to the walking distance but not to the time elapsed.

To mitigate some of the gyroscope errors Jimenez el al. [57] proposes a HDR algorithm. However, one of the best approaches to reduce these errors was proposed by Castaneda and Lamy-Perbal [26], which is a ZARU algorithm. Ladetto et al. [64] tested two prototypes to estimate orientation, one is based on a gyroscope and the other on a magnetometer. Authors concluded that the best approach to estimate orientation is

Table 2: Overview about the features of the presented Pedestrian Inertial Navigation Systems

Authors	Sensors	Sensor fusion	Sensor location	ZUPT	Features	Evaluation	Error
Jirawimut et al. [58]	Accelerometer, Gyroscope, Magnetometer	EKF	Foot and Waist	-	Uses GPS to calibrate PDR parameters	590 steps	-
Ladetto et al. [64]	Accelerometer, Gyroscope, Magnetometer, Barometer	KF	Lower back	-	Compares a gyroscope with a magnetometer to estimate orientation	3000 meters	2%
Lee and Mase [66]	Accelerometer, Gyroscope, Magnetometer	-	Leg and Waist	-	Only detects movements and tries to estimate the location	978 steps	3%
Foxlin [42]	Accelerometer, Gyroscope, Magnetometer	EKF	Foot	Yes	Introduces ZUPT	118 meters	1%
Hamaguchi et al. [47]	Accelerometer, Gyroscope, Magnetometer, Electromagnetic Push button	KF	Waist and feet	-	Measures the relationship between left and right foot	200 meters	5.20%
Ojeda and Borenstein [89]	Accelerometer, Gyroscope	-	Foot	Yes	Introduces three-dimensional walking estimation	500 meters	0.70%
Castañeda and Lamy-Perbal [26]	Accelerometer, Gyroscope, Magnetometer	IKF	Waist	Yes and ZARU	Fuzzy-logic for stance-phase detection and ZARU	40 meters	1%

Jiménez et al. [57]	Accelerometer, Gyroscope, Magnetometer	EKF	Foot	Yes	Stance detection and HDR	125 meters	1.50%
Sabatini [103]	Accelerometer, Gyroscope, Magnetometer	EKF	Foot	Yes	EKF Adaptation	680 meters	10%
Feliz et al. [38]	Accelerometer, Gyroscope, Magnetometer	KF	Foot	Yes	Estimates errors at each step	30 meters	9%
Mathiassen et al. [77]	Accelerometer, Gyroscope, Magnetometer, Barometer	UKF	Foot	-	Includes a barometer	-	58 meters after 60 seconds
Bebek et al. [20]	Accelerometer, Gyroscope, Magnetometer, Barometer	EKF	Foot	Yes	Includes a tactile sensor	1215 meters	0.40%
Li and Wang [69]	Accelerometer, Gyroscope	EKF	Foot	Yes	Distinct algorithms to walking and running gait cycles	171 meters	1.22%
Gadeke et al. [43]	Accelerometer, Gyroscope, Magnetometer, Barometer	KF	Foot	Yes and ZARU	Uses a smartphone	82 meters	2.70%
Nilsson et al. [85]	Accelerometer, Gyroscope	KF	Foot	Yes	OpenSource IMU and algorithms	50 meters	1%

Hoflinger et al. [52]	Accelerometer, Gyroscope, Magnetometer	KF	Foot	Yes	Wireless micro IMU	60 meters	1%
Prateek et al. [92]	Accelerometer, Gyroscope	KF	Foot	Yes	IMU in each foot	110 meters	1%
Ibarra-Bonilla et al. [54]	Accelerometer, Gyroscope, Magnetometer	KF	Foot	-	Fuzzy-logic Kalman filter	100 meters	6.40%
Qian et al. [95]	Accelerometer, Gyroscope, Magnetometer	-	Hand	-	Optical flow using smartphone's camera	400 meters	1.25%
Shin et al. [109]	Accelerometer, Gyroscope, Magnetometer, Barometer	-	Hand	-	Smartphone with heading and motion classifier	133 meters	5%
Yuan et al. [126]	Accelerometer, Gyroscope, Magnetometer	EKF	Foot	Yes	Compares IMU on waist vs foot	-	-
Diaz et al. [33]	Accelerometer, Gyroscope, Magnetometer	UKF	Waist	Yes	Magnetic disturbances detector and a static periods detector	200 meters	2%

to use a combination of those sensors, since each one have their strengths and weaknesses.

A PINS can be composed by several sensors, so it is very important to implement sensor fusion techniques, like the described Kalman filter. Many of the studied works refer the Kalman filter as an optimal estimator, in order to reduce the inertial sensors errors. However, the Kalman filter requires a very good modelling of the system, which can be a very complex task.

Another difficulty encountered when developing a PINS is the stance phase detection. As stated by Beauregard [19], a dynamic threshold detection algorithm and possibly additional sensors, such as a force or proximity switch, can improve PINS stance phase detection. Hamaguchi [47] has introduced wearable electromagnetic sensors and push button switches attached to user's heels. The results demonstrated that this approach can improve the stance detection, so other systems, like the one proposed by Bebek et al. [20] also tended to use similar techniques.

Bebek et al. [20] have introduced a high-resolution thin flexible ground reaction sensor to the IMU, which measures zero-velocity duration to reset the accumulated integration errors from accelerometers and gyroscopes in location estimation. This tactile sensor can be used to accurately detect periods of zero-velocity to increase effective positioning resolution. Compared to the other systems, it can be concluded that the inclusion of a tactile sensor improves the step detection since it can detect, with more accuracy, when the foot is on the ground or not.

Apart from distance and orientation, elevation estimation is also very important. The main application for PINS is in indoor environments and, typically, in this type of locations there are several floors. Ojeda and Borenstein [89], and Sabatini [103] analyzed the performance of a standard IMU, without using a barometer, in a three-dimensional environment. Unlike Mathiassen et al. [77] which uses a barometer, in addition to the normally used inertial sensors. Each one of these approaches has its strengths and weaknesses. However, the barometer approach seems to have more accurate results.

One consideration that, typically, is not considered is the running gait cycle. However, Li and Wang propose a PINS that uses two zero-velocity detectors, one for each type of gait cycle, walking and running. Authors concluded that the accuracy, of the proposed algorithm, for running cases is comparable to walking ones.

Each one of the studied systems has their strengths and weaknesses, and their performance cannot be directly compared since each evaluation scenario is different. In a PINS evaluation several variables influence the final results, as is the case of the different

pedestrians, the sensors quality, the total distance walked, the type of curves and the environment where the test was performed, like the amount of magnetic disturbances, type of floor (i.e. flat or wavy), among others. A variable that has a considerable influence in the PINS results is the step cadence, where the higher errors exist when the user is moving slowly.

Errors in PINS are still considerable, despite all the techniques that are already being implemented, it is believed that more techniques like information fusion and learning algorithms should be applied to improve the systems estimations. For example, more information sources collected from the human body can be used to improve the system's accuracy because one sensor advantage can suppress another sensor disadvantage. Also, techniques that learn the human gait characteristics in several environments can be used to correct, in real-time, the step distance estimation or the user orientation.

Another important feature that must be considered in the future is the system wearability. Due to the current developments on the smartphones performance and characteristics, it is likely that in a near future smartphones will handle strapdown calculations in real-time. This is a good opportunity to make the PINS lighter and more integrated with the human clothes. The developments in the wireless communications will also be very useful to improve the development of lighter and more user-friendly PINS.

From each one of the presented works it can be retrieved the most important techniques and features and combine them to create a precise, comfortable, easy to use PINS capable to work, with good accuracy, in walking and running gait cycles, as well as, in diverse types of environments.

Figure 25 presents an overview for an implementation of a PINS using the most reliable approaches and techniques learned from the previously presented systems. The most common sensors to use are the accelerometer, gyroscope and magnetometer. For each sensor it is removed the correspondent bias, estimated by a Kalman filter approach. The accelerometer data is then tested to verify if the pedestrian is stationary or not, to apply the ZARU and ZUPT techniques, in order to update the Kalman filter data. To have the best orientation estimation the data from these three sensors are combined. Finally, after the projection of the acceleration data to the navigation frame, it is doubly integrated in relation with time in order to obtain the distance. The estimated distance and orientation are then used to estimate the displacement of the pedestrian.

Concluding, the implementation of a localization technology which is capable of providing localization everywhere would unleash the use of many applications [10].

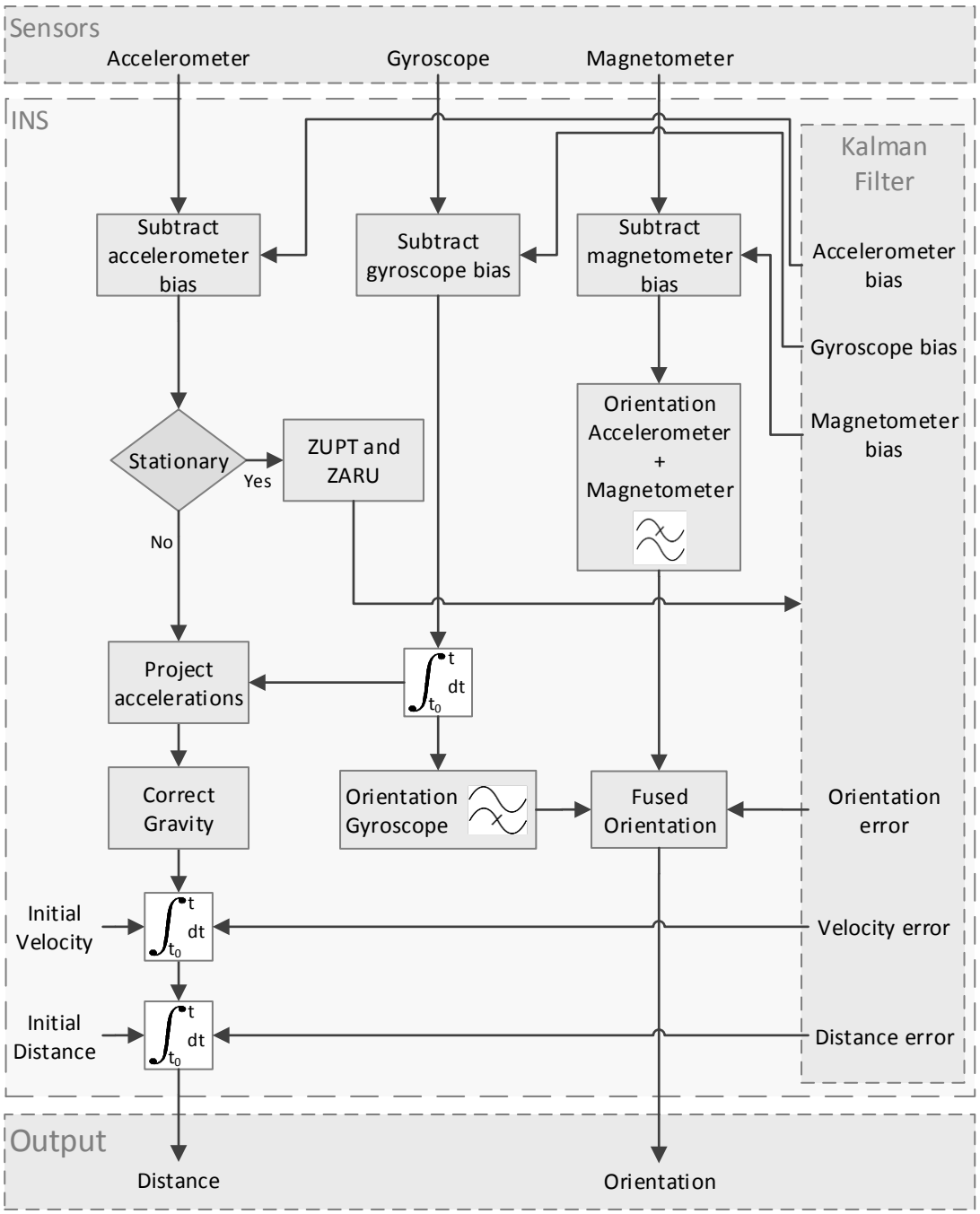


Figure 25: General architecture for the implementation of a pedestrian inertial navigation system

4

THE PROPOSED ARCHITECTURE

The main claim of this thesis proposal is the capability to retrieve location everywhere, independently of the environment and only based in sensors placed in the human body. This chapter describes the developed prototype system focusing on the system architecture and providing the characteristics of the used hardware.

It is also performed an analysis to the data acquired by the sensors, where their data is evaluated.

4.1 ARCHITECTURE

The system is composed by several hardware and software modules, which are represented in Figure 26. The software is divided into four modules: (i) an INS Solution (incorporated on each BSU); (ii) a Low Level Integration Software (LLIS) (incorporated on the BCU); (iii) a *Step Characterization* algorithm (implemented on a mobile device); (iv) and a High Level Integration Software (HLIS) (implemented on a mobile device) [4].

Each *BSU INS Solution* module is composed by three algorithms:

- Data Pre-processing - a noise reduction algorithm used to remove noise from the sensors signals. In this case, noise is something that is captured from the sensor, and can influence, negatively, the walking estimation. This algorithm decomposes the signal and extracts only the frequencies bands that are important to the PINS. The implemented techniques, for each sensor, will be explained throughout this chapter;
- Step Detection - a step detection algorithm, which is one of the most important parts of PINS and of this work proposal. If a step is not properly detected, all the displacement estimations will be faulty. Thus the step detection algorithm must be the most reliable as possible. In this proposal force sensors were added on the foot plant, to improve the step detection and therefore the displacement estimation. The implemented algorithms are presented in Chapter 5;

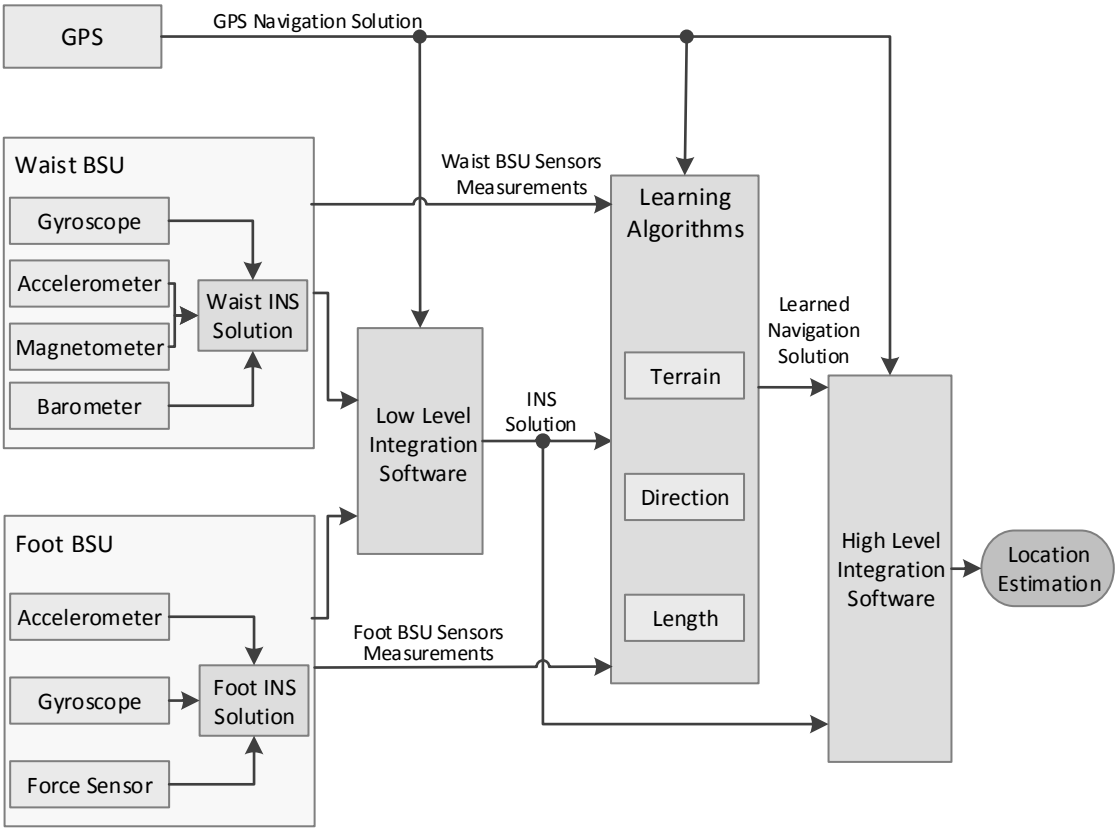


Figure 26: PLASYS architecture

- Navigation Algorithms - algorithm that estimates the distance or orientation of the pedestrian in each corresponding BSU. The *Foot INS Solution* estimates the distance, and its algorithm will be explained in Section 7.2.1. The *Waist INS Solution* estimates the orientation, and the algorithm is explained in Section 7.1.

After the acquisition of the data from the sensors and the estimation of both, distance and orientation, these data is merged, in the LLIS, to estimate the displacement. The displacement is estimated using an absolute position given by the GPS sensor. Then, the estimated displacement will be passed to the HLIS and to the *Step Characterization* (Learning Algorithms) modules.

The *Step Characterization* module is where the learning algorithms are applied. After detecting a step a characterization to the step is performed, which is important to eliminate some of the erroneous measurements that are given by the inertial sensors when integrated. With a proper model of a step a more accurate displacement estimation can be achieved, since some errors can be suppressed using this model. This module receives the data of each BSU (during the step), the displacement estimated by the LLIS

and the absolute position given by the GPS sensor. Then, a characterization for a step is made, which is passed to the HLIS module.

Walking is a cyclic activity, which represents a cyclic pattern of movement that is repeated over and over, step after step [105] [121]. These walking patterns can be extracted in the learning phase and used as a reference model. These patterns are learned during a controlled phase where from a set of exercises the gait analysis is obtained or when GPS is available (with very good signal). Resuming, the system will be always learning the step pattern and will be improving it over time.

This characterization includes: the type of terrain (i.e. normal, descending or ascending stairs) where the pedestrian is walking; the direction (i.e. forward or backward); and the size (i.e. short, normal or long) of the step. Also, it is here that the two sources (BSUs) of data are combined to improve the data accuracy. This algorithm will be presented in Chapter 6.

The HLIS module receives the data given by the two previously presented modules, LLIS and *Step Characterization*, and the GPS data. This data is integrated in order to estimate the pedestrian's displacement. Based on the characterization made by the *Step Characterization* module, different displacement estimation algorithms and rules are used to correct the sensors deviations and the estimations given by the LLIS module. This integration will be presented in Section 7.2.2.

As can be seen in Figure 27 the hardware is composed by two small BSUs distributed in person lower limbs (foot and hip area) to collect information about body movements and are connected through a Bluetooth wireless sensor network. These BSUs were developed based on smart sensors philosophy, connected to an integrated circuitry module to pre-process and codify the collected signal. The waist BSU is capable to interact with the sensory set around the body, and to exchange data between sensors and a mobile device [7].

The first BSU module, waist BSU, is placed in the abdominal area and is represented in Figure 28. It is composed by a gyroscope, an accelerometer, a magnetometer and a barometer. The gyroscope, accelerometer and magnetometer are used to get the body walking direction, since with their complementarity can be used to improve the orientation estimation. The barometer can be used to obtain the user elevation, for example to know in which floor the user is. It is here that part of the algorithms (e.g. the LLIS) are implemented. Also, it is from this module that the information is sent to the mobile device through a Bluetooth connection.

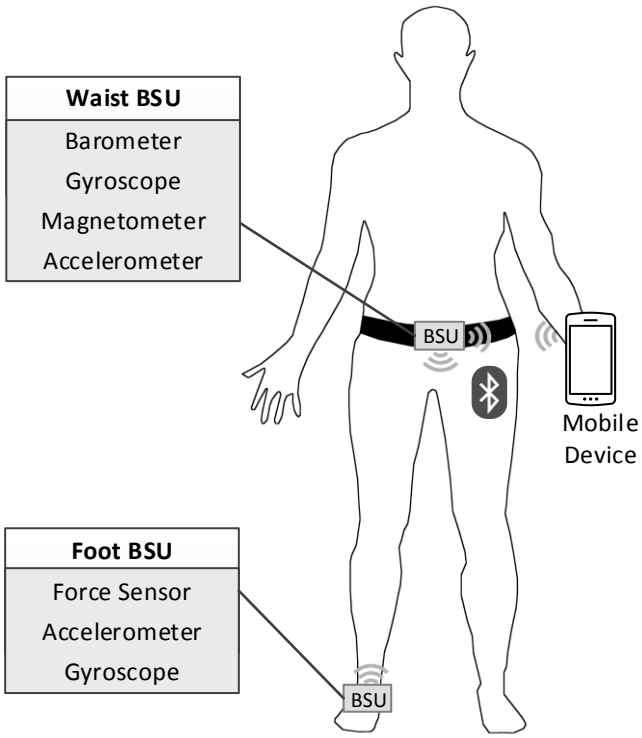


Figure 27: Sensor distribution on the human body and wireless communication

The second BSU module, is placed on the foot and is represented in Figure 29. It is composed by an accelerometer, a gyroscope and two force sensors. The accelerometer is used to detect and quantify the foot movement, and the gyroscope is valuable to transform this acceleration data from the sensor frame to the navigation frame.

Typically the drift is mitigated taking advantage of the ZUPT, meaning that the integration of inertial measurements is only performed during the swing of legs and the velocity errors can be reset at each step, since when the pedestrian is stationary the true velocity must be zero. This technique is used by several systems and typically an accelerometer is used to detect if the pedestrian is stationary or not. However, All Over

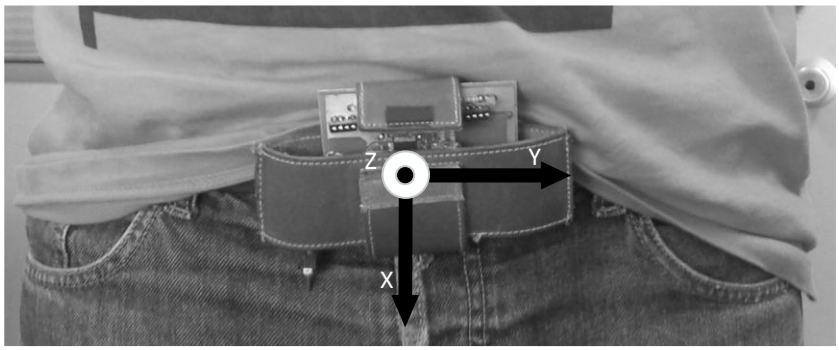


Figure 28: Waist BSU with the corresponding axis

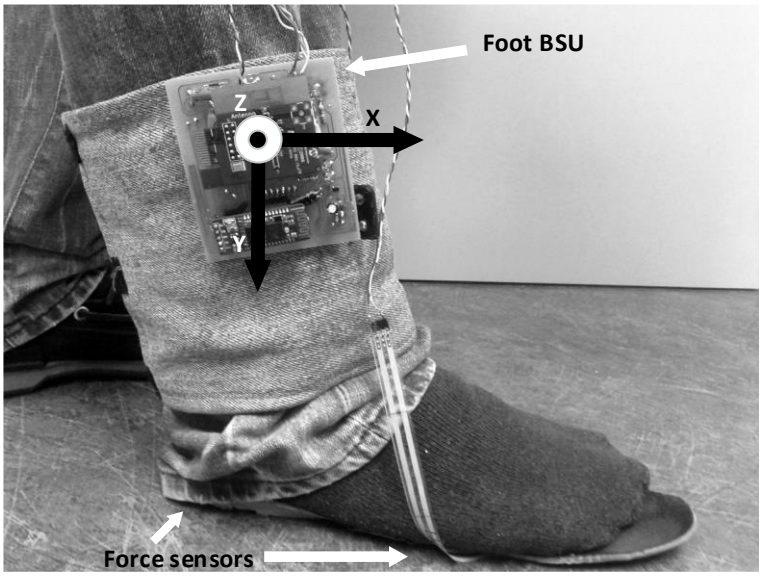


Figure 29: Foot BSU with the corresponding axis

the Place Localization System (PLASYS) uses the force sensors to detect when the user touches his feet on the ground, as well as, the respective contact force, which combined with the accelerometer (used to obtain the step acceleration) provide a more exact step detection. Also, this force information is useful to improve the detection of the activity type that the pedestrian is performing (e.g. running, walking fast).

4.2 FOOT BODY SENSOR UNIT

The foot BSU hardware is composed by three modules: processing unit, sensors (accelerometer, gyroscope and force) and communication module (Bluetooth). The prototype was designed and developed in order to not only fulfil the objectives of the project, but also to allow an easy replacement of modules in the case of a hardware failure.

The force sensors are separated from the main board to easily adjust their placement, for testing proposes, in different places on the foot plant. A photo with the details of the foot BSU can be seen Figure 30, where each module is identified.

According to the objectives of this work the BSU hardware follows the following requirements:

- Small size - smaller boards are more convenient to the person use during the walking path;

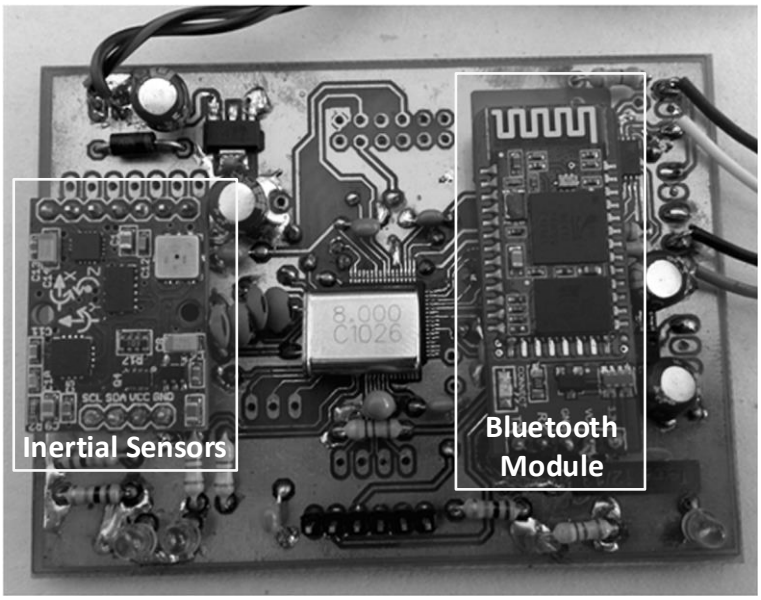


Figure 30: Foot BSU in detail

- Low-energy consumption - in order to have a longer battery life;
- Universal Synchronous Asynchronous Receiver Transmitter (USART)[1] port - to enable the exchange of information between the CPU and the communication module;
- Inter-Integrated Circuit (I^2C)[2] - to enable the communication between the CPU and the sensor's module.

The choice of the processing unit fell into the dsPIC family, in particular the dsPIC33-FJ128GP706A, from Microchip Technology Inc[3]. It was chosen because it has enough processing capabilities to gather the sensor data and pre-process it.

4.2.1 *Sensory Set*

The sensors incorporated in the foot BSU are presented in this section. This BSU is composed by an accelerometer (Section 4.2.1.1), a gyroscope (Section 4.2.1.2) and force sensors (Section 4.2.1.3).

1 USART is an synchronous/asynchronous serial communication port usually included in microcontrollers
2 I^2C is a multi-master, multi-slave, single-ended, serial computer bus which is often used to communicate across circuit-board distances.
3 Microchip Technology Inc. is an American manufacturer of microcontroller, memory and analog semiconductors.

4.2.1.1 *Accelerometer Sensor*

The foot BSU is composed by a ADXL345 accelerometer from Analog Devices[4]. This accelerometer was selected because it fulfils the project requirements, it has low-power consumption, measures the three axis (x, y, z) and has a stable Output Data Rate (ODR)[5] of at least 100 Hz, which is an acceptable rate for this project. Also, the measurement range of ± 16 g is convenient for a PINS, since sometimes a step can produce accelerations of that magnitude. More information about the sensor characteristics can be seen in Table 3 or in the accelerometer data sheet [32].

Table 3: ADXL345 accelerometer specifications [32]

Characteristic	Value
Measurement Range	± 2, ± 4, ± 8, ± 16 g
Operating Temperature	$-40°$C to $+85°$C
Axis	x, y, z
Output data rate	0.1Hz to 3200Hz
Supply current	90μA at 50Hz | 3200μA at 100Hz

The accelerometer was configured to work with the maximum resolution (± 16g, representing 13 bits of information), with the First In First Out (FIFO) option activated and the ODR to 100 Hz.

With this accelerometer and the specified settings, to obtain the acceleration in m/s^2 it is necessary to apply the formula presented in Equation 24. Observing this formula can be seen that, for the accelerometer, one g corresponds to the value of 256.

$$a = \frac{\text{accelerometer measurement}}{256} \times 9.81 \tag{24}$$

The sensors sometimes give incorrect values due to noise or degradation that occurs over time. The most important source of error of an accelerometer is the bias. The bias of an accelerometer is the offset of its output signal from the true value. A constant bias, when double integrated, causes an error in position which grows quadratically with time.

This type of error occurs when the accelerometer output is non-zero when the measured property is zero. To quantify this error the accelerometer must be placed on a flat surface parallel to the ground. Thus it can be measured the average of the accelerations.

4 Analog Devices is an American multinational semiconductor company specialized in data conversion and signal conditioning technology.

5 ODR defines the frequency at which the data is sampled.

If the measurements are not zero in two of the axis and 9.80665 m/s^2 in the other axis, then the accelerometer has an offset. In some cases these deviations may be removed or predicted using some calibration procedures.

To detect and quantify the offset of the used accelerometer, its measurements were recorded during 12 seconds with the device stationary in a flat surface and with the force of gravity applied on the *y-axis*. This period has chosen to have a good indication about the offset over time.

In Figure 31 is shown the noise signal and in Table 4 are presented some statistics about this noise. These statistics are the minimum, maximum and mean value for each one of the accelerometer axis and for the combined acceleration. Also, it was calculated the standard deviation for the measurements of each axis.

This test has shown that the mean of the total acceleration was 9.83m/s^2, which is close to the acceleration of gravity on Earth (9.80665m/s^2). The standard deviation of the total acceleration for all the records is approximately 0.019m/s^2. It was also found that the *z-axis* has the higher offset.

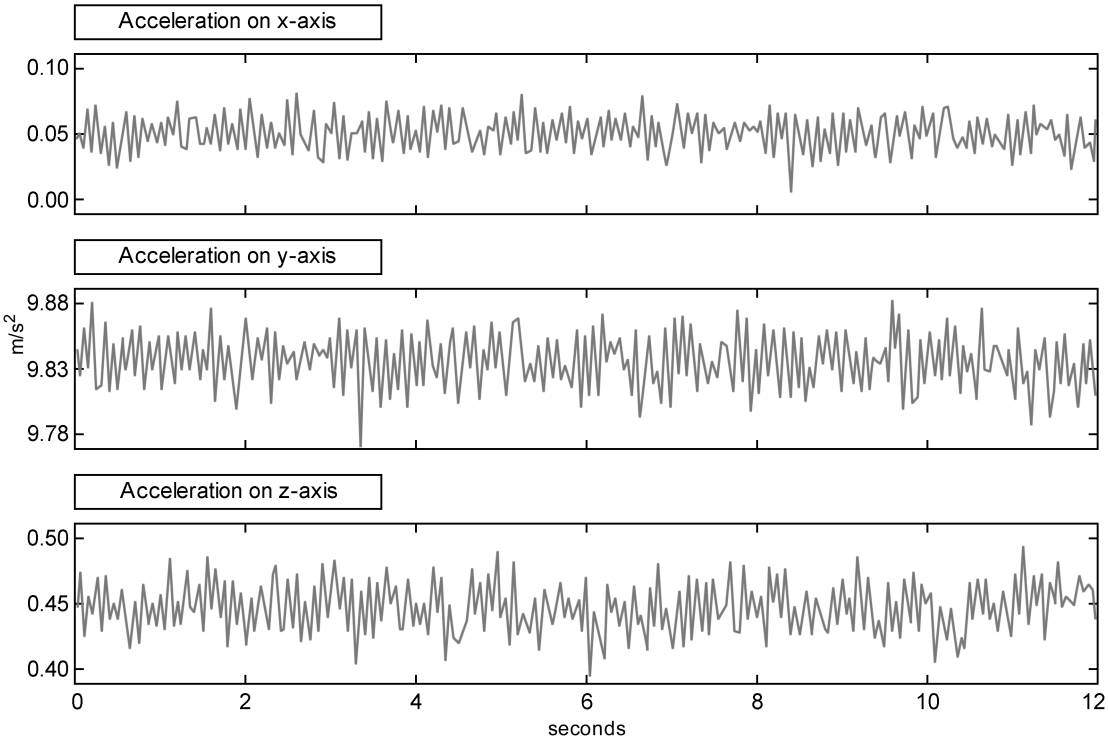


Figure 31: Foot BSU accelerometer output with the device stationary and the gravity force applied on the *y-axis*

Table 4: Foot BSU accelerometer output statistics, in m/s^2, when the device is stationary and the gravity force applied on the *y-axis*

Parameter	Acc_x	Acc_y	Acc_z	Total Acc
Minimum	0.0150	9.7734	0.3832	9.7810
Maximum	0.0916	9.8884	0.4982	9.9014
Mean	0.0414	9.8224	0.4475	9.8327
Deviation	0.0191	0.0193	0.0223	0.0193

4.2.1.2 *Gyroscope Sensor*

The gyroscope is a device that measures the speed of a rotation around an axis, i.e., measures how fast an object rotates relative to an inertial reference system, known as angular velocity. This angular velocity may be integrated over time to obtain the angular position.

The rotation is measured relatively to one of the three axis. The principle behind the MEMS gyroscopes is that it contains vibrating elements to measure the Coriolis Effect. This effect states that a mass m, moving with a linear speed v in a rotating frame with an angular velocity ω suffers the effect of a force called the Coriolis force F_c and the effect of Coriolis acceleration a_c. The Coriolis force and acceleration are given by Equations 25 and 26, respectively.

$$F_c = -2m(\omega \times v) \tag{25}$$

$$a_c = 2\omega \times v \tag{26}$$

The gyroscope gives negative and positive measurements. The signal of the values indicates the direction of the rotation, if the rotation is in a clockwise direction the measurements are negative, if the rotation is in the opposite direction (anticlockwise) the measurements are positive.

The foot BSU has a L3G4200D gyroscope from STMicroelectronics[6]. In Table 5 are described some characteristics of this sensor retrieved from its data sheet [111].

This gyroscope was configured to work with the 2000dps scale, in FIFO and with an ODR of 100Hz. The L3G4200D gyroscope has a set of internal filters disposed as

6 STMicroelectronics is a French-Italian multinational, and is Europe's largest semiconductor chip maker, specially in semiconductors for industrial applications, inkjet printheads, MEMS, smartcard chips, automotive integrated circuits, computer peripherals, and chips for wireless and mobile applications.

Table 5: L3G4200D gyroscope specifications [111]

Characteristic	Value
Measurement Range	±250, ±500, ±2000 dps
Operating Temperature	−40° to +85°
Axis	x,y,z (Yaw, Pitch, Roll)
Output data rate	100Hz, 200Hz, 400Hz, 800Hz
Supply current	6.1mA

Figure 32 shows. These filters were configured with *HPEN* to 0 and the *Out_Sel* to 10. Thus the gyroscope data is obtained after passing through the low-pass filter 1 (LPF1) and the low-pass filter 2 (LPF2) without going through the high-pass filter (HPF). The high-pass filter is not being used because after some tests it was found that it removes useful information from the signal.

The low-pass filter LPF1 does not allow the editing of the cut-off frequency, so the default cut-off frequency was used, which is 32Hz to an ODR of 100Hz. The low-pass filter LPF2 to an ODR of 100Hz only allows the configuration of a cut-off frequency of 12.5Hz or 25Hz. From the tests, the best results were achieved with the frequency set at 12.5Hz. The measurements of the gyroscope are not obtained by using interruptions, so the *INT1_Sel* parameter was not set.

When a gyroscope is immobilized on a flat surface it must give 0 in all the axis. However, due to the noisy measurements and changes in the ambient temperature, gyroscope measurements vary when the sensor is stationary. The constant bias error of a rate gyroscope can be estimated by taking an average of the gyroscope output while it is not performing any rotation.

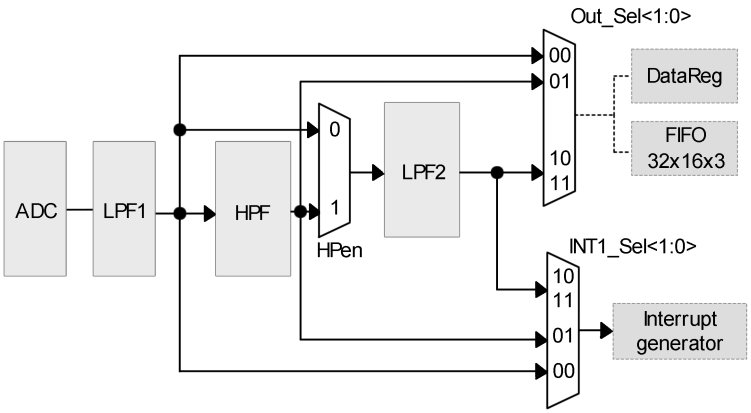


Figure 32: L3G4200D gyroscope diagram with low-pass and high-pass filters (adapted from [111])

In Figure 33 and in Table 6 the gyroscope bias can be observed. As the gyroscope does not show an absolute value, on each axis, of zero while stationary, it can produce a significant error when integrated. Meaning that some drift, over time, will exist.

Table 6: Foot BSU gyroscope output statistics, in rad/s, with the device stationary

Parameter	$Gyro_x$	$Gyro_y$	$Gyro_z$
Minimum	-0.0302	-0.0302	-0.0282
Maximum	0.0218	0.0208	0.0248
Mean	0.0018	-0.0024	-0.0071
Deviation	0.0100	0.0101	0.0107

Once the bias is known it is trivial to compensate it by simply subtracting the bias from the output or to establish a threshold to set to zero the gyroscope measurements. This will eliminate the noise in order to not accumulate angular displacement when the gyroscope is stationary. The threshold value can be defined as three times the standard deviation of the gyroscope measurements, when any angular velocity is applied to any of its axis [112]. The disadvantage of this procedure is that the gyroscope measurements for very slow rotations, the ones that are lower than the threshold value, will be ignored.

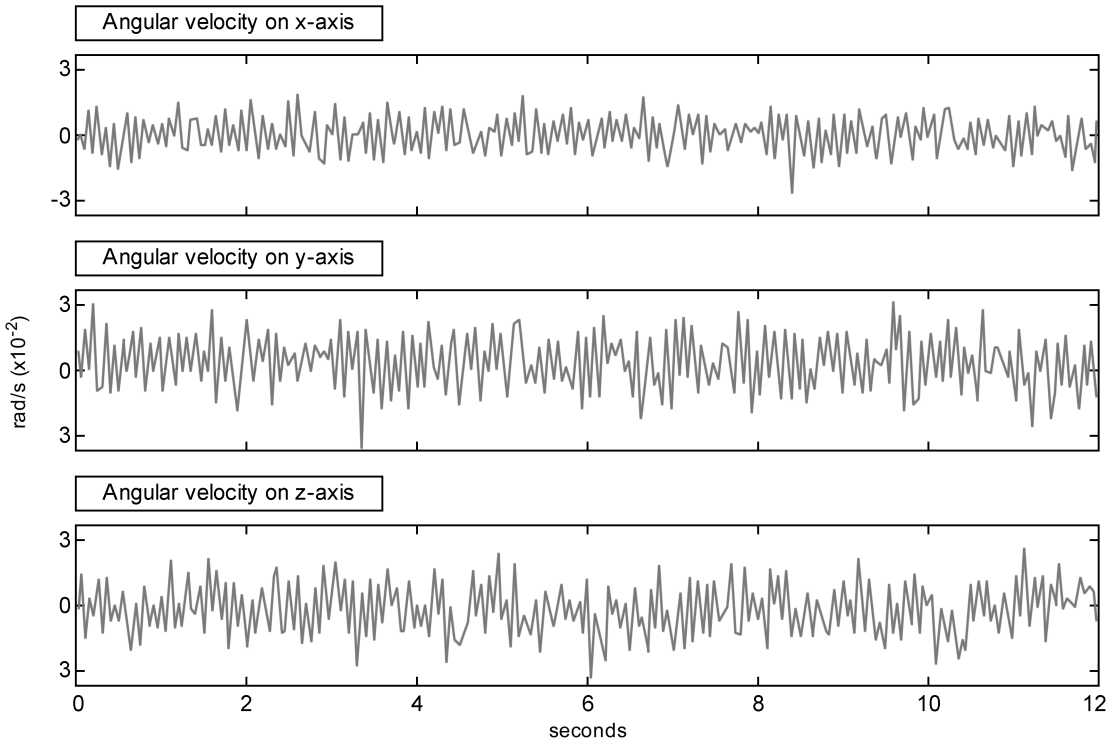


Figure 33: Foot BSU gyroscope output with the device stationary

The gyroscope's drift was also verified by performing one more test, which consists in making a rotation of $\frac{\pi}{2}$ rad in the anticlockwise direction, two rotations of $\frac{\pi}{2}$ rad in the clockwise direction, and finally another rotation of $\frac{\pi}{2}$ rad in the anticlockwise direction. All these rotations were performed over the *z-axis*. The obtained data can be seen in Figure 34.

The drift error is very small, but as it accumulates due to the integration of the angular speed, an adjustment to the gyroscope values needs to be done regularly.

4.2.1.3 *Force Sensor*

A force sensor converts mechanical force into an electrical signal. Most of the force sensors operate on the principle that the electrical resistance will increase when a force is applied. The resistance measurements change in proportion to the applied force. This change in resistance results in a change of the corresponding output voltage. These sensors have been selected mainly because they are soft and comfortable to be placed on the person's foot, as shown in Figure 35.

For this project, the chosen force sensor was the FlexiForce® A201 from Tekscan. It is available in different force ranges and sizes. Its specifications can be seen in Table 7. They are ultra-thin and flexible printed circuits, which can be integrated into multiple force measurement applications, like [114]:

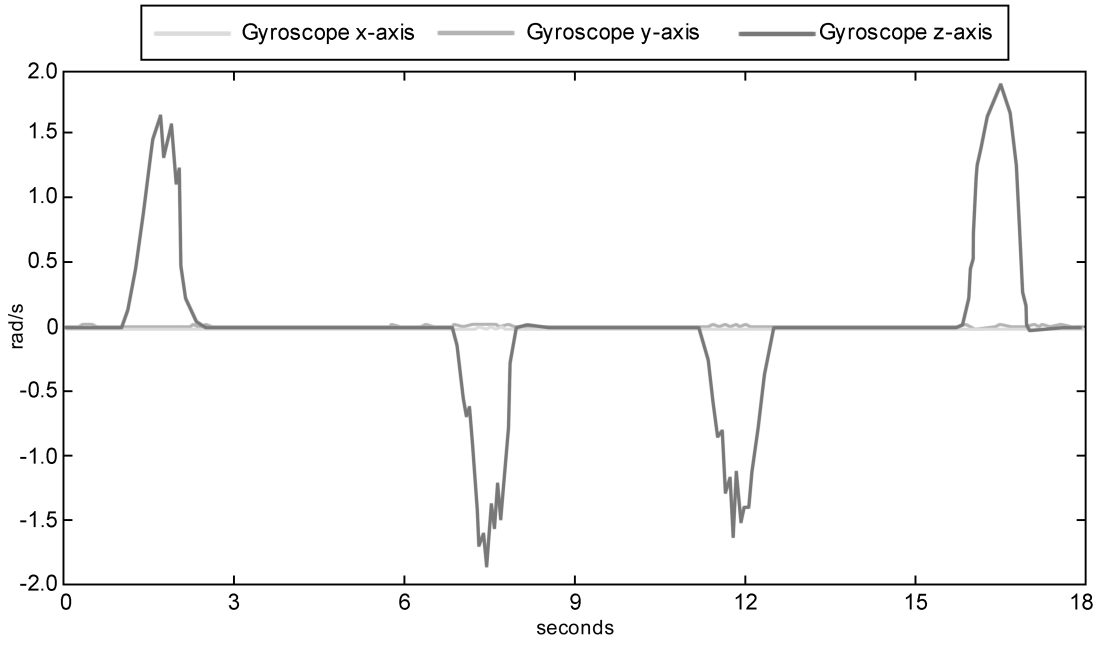


Figure 34: Foot BSU gyroscope output for four $\frac{\pi}{2}$ rad rotations on *z-axis*

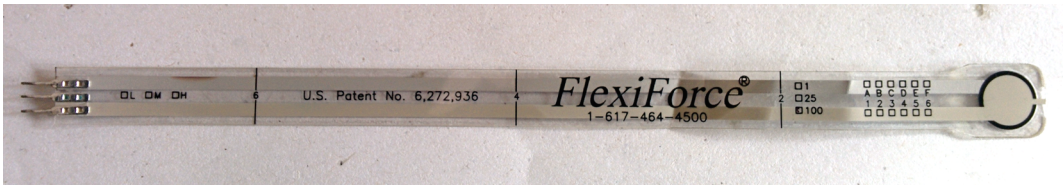


Figure 35: FlexiForce® A201 from Tekscan

- Detect and measure a relative change in force or applied load;
- Detect and measure the rate of change in force;
- Identify force thresholds and trigger appropriate actions;
- Detect contact and/or touch.

Table 7: Tekscan A201 force sensor specifications [114]

Characteristic	Value
Style	Rounded with a sensing area of 9.53mm
Force Range	440 N
Operating Temperature	$-40°C$ to $+60°C$
Accuracy	$\pm 3\%$

To know the force applied on the sensor it is only needed to configure the sensor gain, which was defined to 220kΩ. This value was chosen value since it has shown to be sufficient to enable the detection of any variation on the measurements [114].

The force sensors are placed on the foot plant as shown in Figure 36. This force sensors disposal detects if the foot is in contact with the ground, from the first contact of the heel with the ground (where force sensor A is located) until the front of the foot is no longer in contact with the ground (where force sensor B is located), in order words, during the entire stance phase. Thus, force sensors give information about when the user puts and raises his feet on the ground.

In order to verify the accuracy of the sensor it was performed a test with two different weights, one with 3kg and other with 0.5kg. Both objects were placed in order to concentrate all their mass on the center of the sensing area. It was wanted to verify

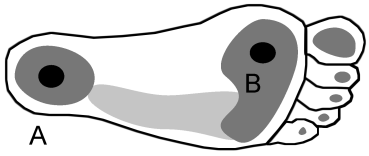


Figure 36: Force sensors location on the foot

if the force sensor maintains the same value in different measurements of the same weight.

Analysing Table 8 it can be seen that the standard deviation for the heaviest object is smaller than to the lightest one. This means that this force sensor is more stable with heavier items than with lighter ones. The higher the weight the lower will be the sensors deviations since the sensors resistance will vary less.

Table 8: Force sensors output statistics, in *MOhm*, for weights of 3kg and 0.5kg concentrated only on the sensor's center

Weight (kg)	Minimum	Maximum	Mean	Deviation
3.0	0.93	1.04	0.97	0.03
0.5	7.20	9.70	8.63	0.87

In Figure 37 is presented the data collected from the two force sensors in an experiment that consisted in a straight line ten step walk. It can be seen that the sensors data are correlated. The dashed line represents the data from force sensor A and the dotted line represents the data from force sensor B. As can be seen the values of force are zero, or almost zero, when the foot is moving (when there is higher variation in acceleration values), and when the foot is on the ground, sensor A goes first to a high value and then sensor B goes to a high value, as expected. This high value occurs when the foot touches the ground. With the fusion of the two force sensors data a more accurate step detection can be achieved.

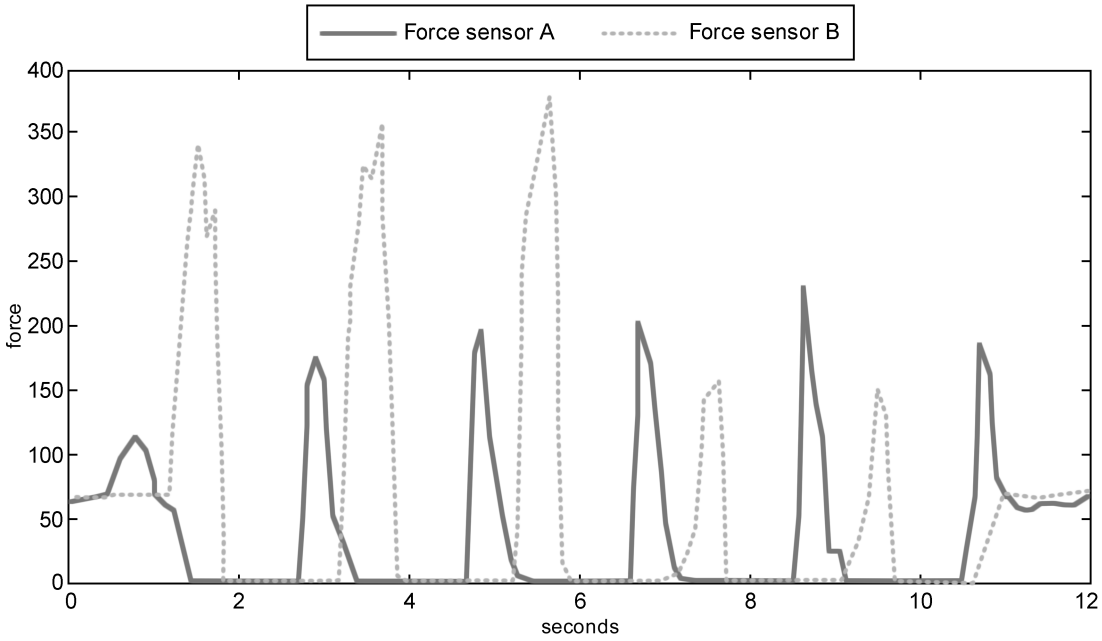


Figure 37: Force sensors data for a 10 step walking

4.2.2 *Communication Module*

This module is based on the Bluetooth technology and was developed to allow the communication between both BSUs.

As one of the project objectives is to transfer the sensor measurements to a mobile device, a Bluetooth module was adopted because it is widely used in mobile devices. The module that was chosen, shown in Figure 38, was the BlueCore4-Ext™ [31] from CSR, that uses the BC417143 chipset and it is a class 2 Bluetooth. More characteristics can be seen in Table 9.

A class 2 Bluetooth is the most common standard on handheld devices and can manage up to 3 Mbps, with a maximum communication distance of up to 10 meters. These specifications are compatible with this project requirements.

Table 9: Broadcom BCM4339 Bluetooth specifications [31]

Characteristic	Value
Bluetooth version	Bluetooth v2.0+Enhanced Data Rate (EDR)
Class	Output power Class II ($\pm$2.5mW range $\pm$10m)
Flash memory	8Mbit
Power supply	3.3v
Interfaces	Universal Asynchronous Receiver Transmitter (UART) and I^2C

This module was chosen due to its price and to be easy to interact. This last advantage is related to the fact that it is not necessary to interpret the Bluetooth protocol communication datagrams, because the information that the module exchanges with the CPU is only the useful information sent by the other device (without the protocol headers).

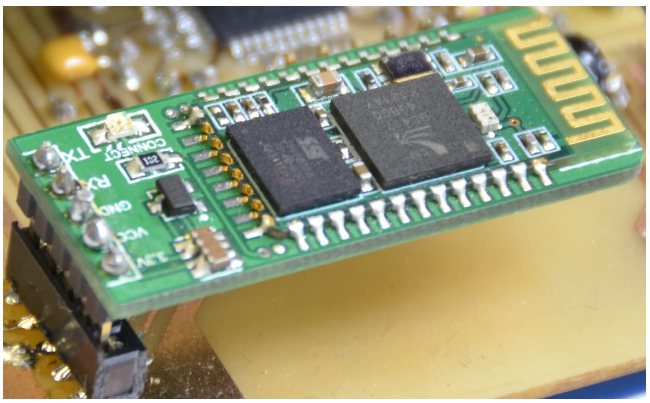


Figure 38: Bluetooth module BlueCore4-Ext™ from CSR

This module was configured as slave, which means that it only accepts connection requests. All the configurations that were used in this module are the following:

- Transmission rate of 115200bps;
- No parity;
- 8 data bits;
- 1 stop bit;
- Without flow control;
- Pairing password: 1234;
- Configured to work as slave.

4.2.3 *Software*

The software on the foot BSU has the data flow presented in Figure 39. It is divided into one interrupt and a main program. The interrupt is used to define the sampling period. This interrupt is defined with the highest priority, thus if new interrupts were added the sampling period maintains the same.

The main program is divided into two parts, the initialization and the data processing. In the initialization phase the sensors and the Bluetooth module are initialized with the settings described in Section 4.2.2. The data processing is performed at each sampling instant.

The sampling period is 50ms (20Hz). This process involves the reading of the sensors measurements and some filtering techniques on those measurements. As seen in Sec-

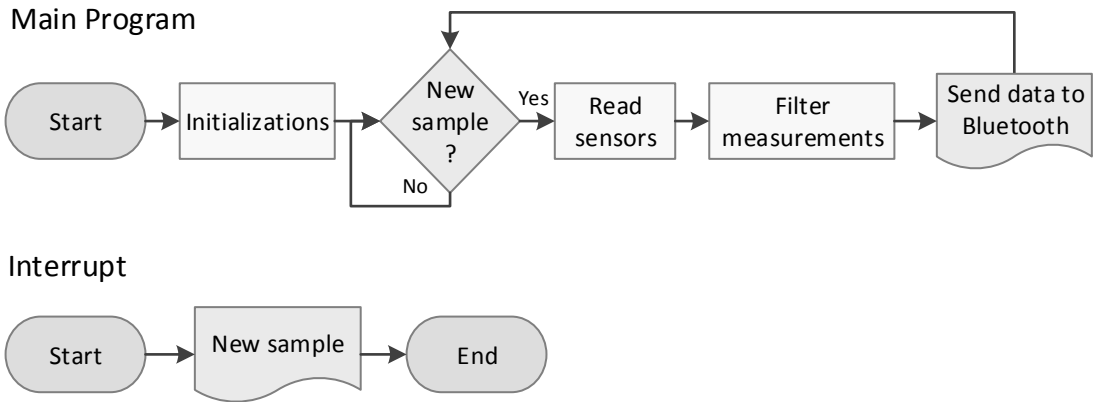


Figure 39: Foot BSU Main Program and Interrupt description

tion 4.2.1.1 the accelerometer ODR was set to 100Hz. This means that for each sampling period the accelerometer has 5 measurements stored in its FIFO. After reading this FIFO an average, of the 5 measurements, is made to obtain a less noisy value. The ODR of the gyroscope is also set to 100Hz, so the mean of the 5 last measurements was also used.

The measurements from each force sensor are obtained by reading the sensor analogue value. For each force sensor, 64 measurements are read at each sampling period and the final value is the average of those measurements.

After these steps all the data is sent to the Bluetooth module, to be sent to the mobile device. This data, in order to the mobile device match a specific value with the corresponding sensor, was formatted with the following protocol:

%	$	sensor_type	x	y	z

where % indicates the beginning of a new set of data (i.e. occurs at each sampling period). Each set of data is composed by the measurements from the accelerometer, the gyroscope and the two force sensors. The *$* indicates the beginning of a sensor measurement, then *sensor_type* defines the sensor for which the measurements belong (Table 10 are presented the possible values), and x, y, z are the measurements for the specified sensor. In the case of the force sensor only the x is sent, which corresponds to the sensed force value.

Table 10: Sensor types encapsulation parity

Sensor Type	Corresponds to
1	Accelerometer
2	Gyroscope
3	Force Sensor A
4	Force Sensor B

4.3 WAIST BODY SENSOR UNIT

After the development of the waist BSU, it was tested to verify its robustness and to validate the platform. However, it has demonstrated some robustness problems regarding the communication protocol, which turned impracticable to use it. This constrain was related to the fact that the ZigBee network, which initially was proposed to be used, carried some implementation problems and it was not so reliable as expected.

Thus, to validate the system, the waist BSU was simulated using an Android smartphone, the Google Nexus 5, which has all the necessary sensory set to work as the developed foot BSU, in the person's waist.

All the sensory data was obtained in its raw form, without any interfering of the mobile operating system sensor fusion or data pre-processing algorithms. To maintain the data consistency between BSUs, the waist BSU ODR was also defined at 20Hz.

4.3.1 *Sensory Set*

As the foot BSU, the waist BSU is composed by an accelerometer (Section 4.3.1.1) and a gyroscope (Section 4.3.1.2). However, it also includes a magnetometer (Section 4.3.1.3) and a barometer (Section 4.3.1.4).

4.3.1.1 *Accelerometer Sensor*

The mobile device used to simulate the waist BSU incorporates an InvenSense[7] MotionTracking™ device (MPU-6515) specially created to be used on smartphones, tablets and wearable sensors.

The MPU-6515 unit integrates a 3-axis gyroscope, a 3-axis accelerometer and a Digital Motion Processor™ to offload complex motion algorithms from the mobile device processor. It has Motion Interrupt, Pedometer, Step Event and Event Batching functions on-chip, only waking the mobile device processor when critical motion events require attention from other parts of the system. Table 11 presents some characteristics of the accelerometer embedded in this unit, which are very similar to the ADXL345 accelerometer used on the foot BSU.

Table 11: MPU-6515 accelerometer specifications [55]

Characteristic	Value
Measurement Range	± 2, ± 4, ± 8, ± 16 g
Operating Temperature	$-40°C$ to $+85°C$
Axis	x, y, z
Output data rate	0.24Hz to 4000Hz
Supply current	450µA

7 InvenSense Inc. is the leading provider of MotionTracking™ solutions for consumer electronic devices, including smartphones, tablets, gaming devices, optical image stabilization and remote controls for Smart TVs.

To obtain the accelerometer data it was used the Android Application Programming Interface (API) and the data comes in m/s^2, not being necessary to make any conversion. To quantify the offset, the accelerometer measurements were recorded during 12 seconds with the mobile device stationary on a flat surface and the force of gravity applied on the *x-axis*.

Figure 40 shows the obtained signal and in Table 12 it is presented some statistics about the obtained signal.

Table 12: Waist BSU accelerometer output statistics, in m/s^2, with the device stationary and the force of gravity applied on the *y-axis*

Parameter	Acc_x	Acc_y	Acc_z	Magnitude
Minimum	-9.9409	0.1431	-0.2538	9.9451
Maximum	-9.8361	0.2730	-0.0981	9.8407
Mean	-9.8869	0.2201	-0.1675	9.8908
Deviation	0.0175	0.0209	0.0323	0.0175

This test has shown that the average total acceleration was $9.89 m/s^2$, which is a little bit higher than the Earth's gravity ($9.80665 m/s^2$), and also has a higher error when compared to the foot accelerometer. The standard deviation of the total acceleration

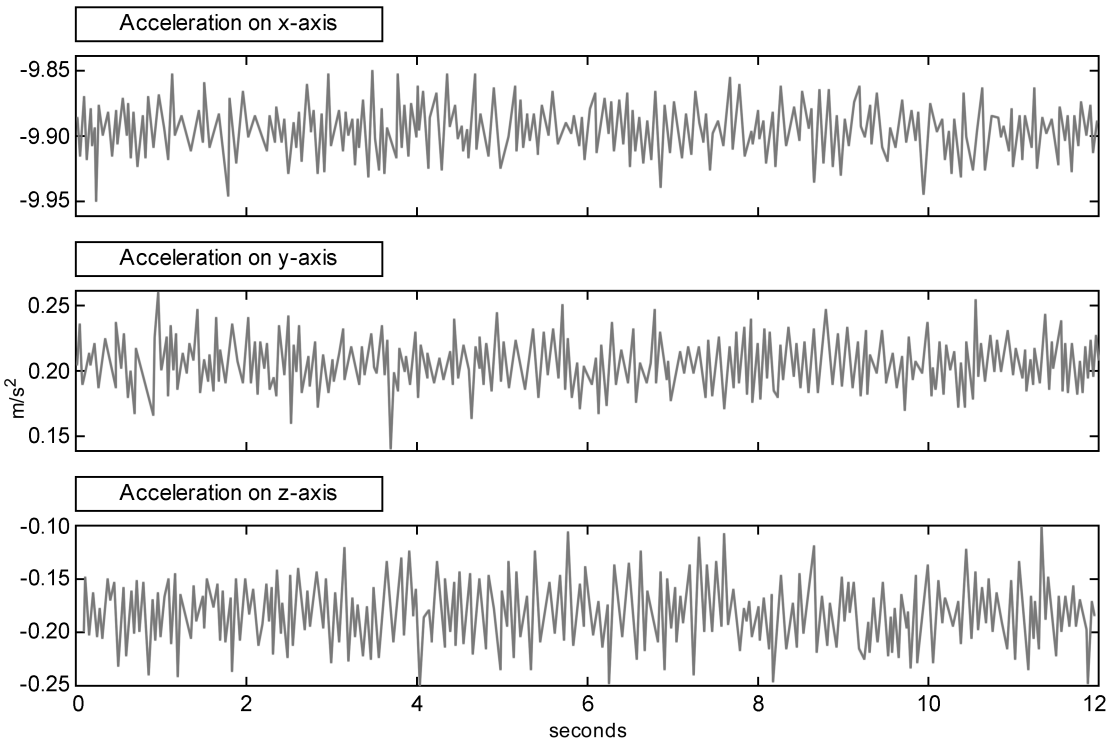


Figure 40: Waist BSU accelerometer output with the device stationary and the force of gravity applied on the *x-axis*

is approximately $0.0175 m/s^2$, which is similar to the foot accelerometer. Like foot accelerometer, this one has also a higher offset on the *z-axis*.

According to these tests it can be concluded that the values given by the accelerometer always follow the same pattern and thereby the same errors. Also, it can be concluded that the accelerometer produces good values and the errors can be surpassed using some algorithms to correct the produced data. According to this results it is suitable to be used in this PINS proposal.

4.3.1.2 *Gyroscope Sensor*

The waist gyroscope is embedded in the MPU-6515 unit presented in Section 4.3.1.1. In Table 13 are described some characteristics of this gyroscope, which are identical to the L3G4200D gyroscope already presented.

Table 13: MPU-6515 gyroscope specifications [55]

Characteristic	Value
Measurement Range	±250, ±500, ±1000, ±2000 dps
Operating Temperature	−40°C to +85°C
Axis	Yaw, Pitch, Roll
Output data rate	4Hz to 800Hz
Supply current	3.2mA

In Figure 41 and in Table 14 can be observed the bias of this gyroscope. Comparing it with the foot gyroscope, this MPU-6515 unit has better results. This happen since this unit is equipped with MotionFusion algorithms, implemented by InvenSense, which filters the signal and makes some corrections to the gyroscope noise using sensor fusion algorithms [55].

Table 14: Waist BSU gyroscope output statistics, in rad/s, with the device stationary

Parameter	$Gyro_x$	$Gyro_y$	$Gyro_z$
Minimum	-0.0044	-0.0042	-0.0024
Maximum	0.0020	0.0022	0.0019
Mean	-0.0014	-0.0005	-0.0004
Deviation	0.0011	0.0009	0.0008

Besides the bias test, another test was performed in order to verify the gyroscope's drift. In Figure 42 can be seen the data obtained from the gyroscope after having a rotation of $\frac{\pi}{2}$ rad in the anticlockwise direction, two rotations of $\frac{\pi}{2}$ rad in the clockwise

direction, and finally another rotation of $\frac{\pi}{2}$ rad in the anticlockwise direction. All these rotations were performed over the *z-axis*.

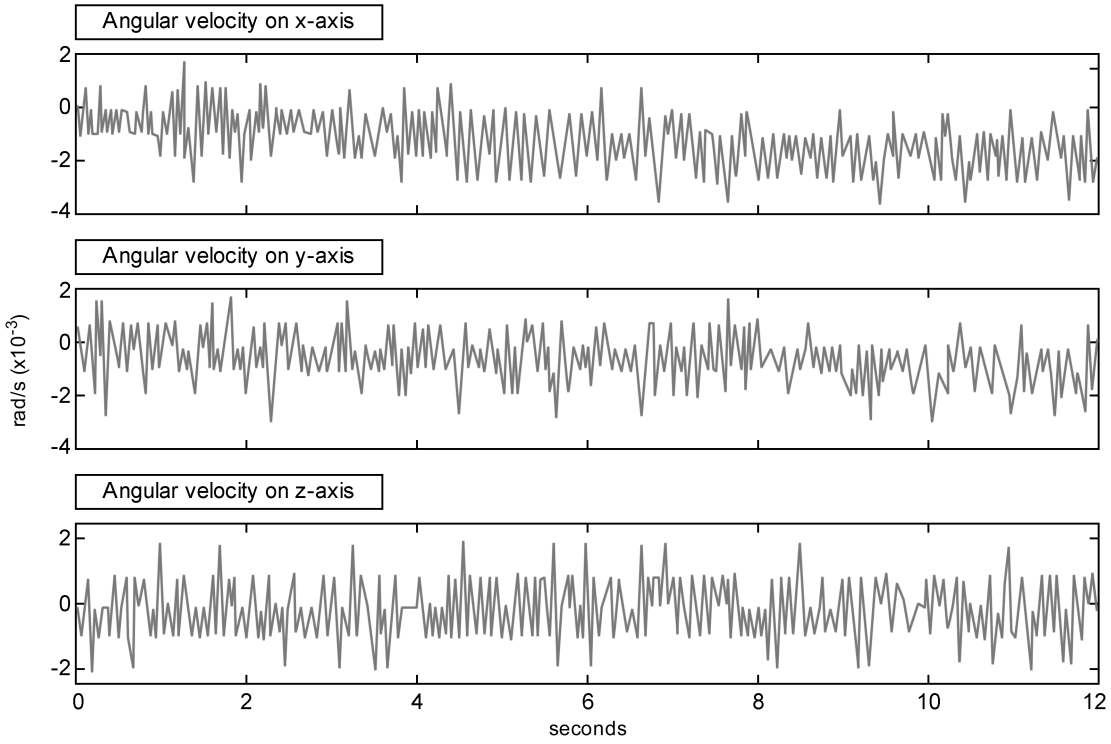


Figure 41: Waist BSU gyroscope output with the device stationary

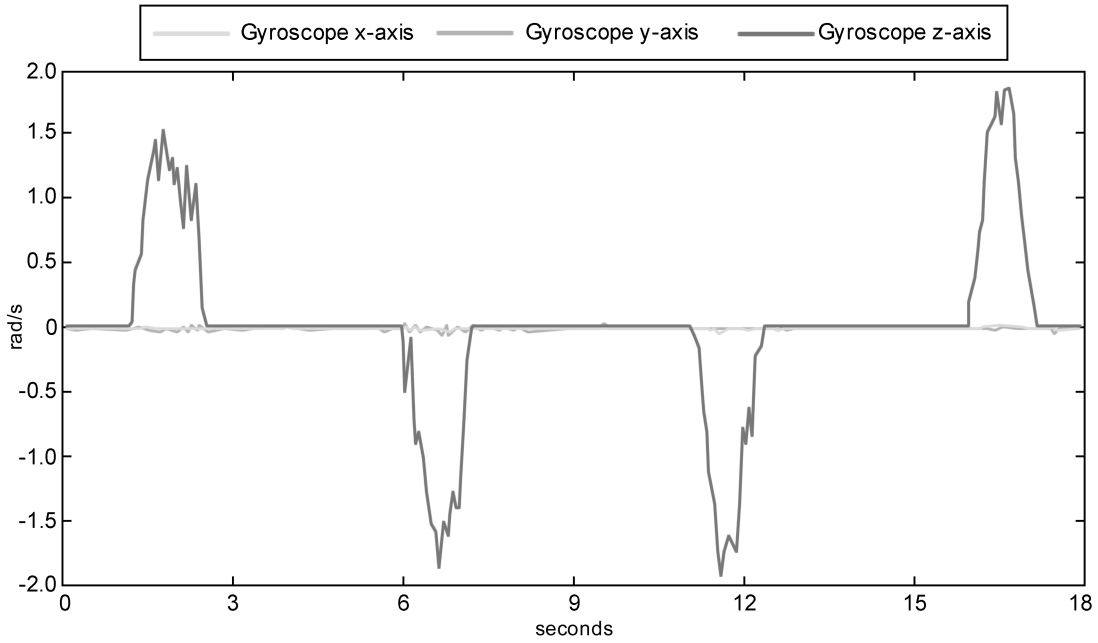


Figure 42: Waist BSU gyroscope output for four $\frac{\pi}{2}$ rad rotations on *z-axis*

Compared to the foot BSU gyroscope unit the drift error is smaller in this MPU-6515 unit. These results can be influenced by the embedded sensor fusion algorithms. However, small adjustments need to be done regularly since the drift accumulates due to the integration of the angular speed.

4.3.1.3 *Magnetometer Sensor*

A magnetometer is used to determine the direction that a device is taken relatively to the Earth's magnetic poles. However, its accuracy is heavily affected by magnetic interferences caused by the approach to other magnetic fields or ferrous materials. Besides this drawback, the magnetometer needs some time, after a sudden movement, to stabilize the measurements again.

Magnetometers can be divided into two basic types: scalar magnetometers that measure the total strength of the magnetic field to which they are subjected, and vector magnetometers (the type of the one used in this project), which have the capability to measure the component of the magnetic field in a particular direction, relative to the spatial orientation of the device.

The sensor used to obtain the magnetic fields was the AK8963 from AKM[8], and its characteristics are presented in Table 15.

Table 15: AKM AK8963 magnetometer specifications [1]

Characteristic	Value
Measurement Range	$\pm 4900\mu T$
Operating Temperature	$-30°C$ to $+85°C$
Axis	x, y, z
Sensitivity	$0.15\mu T$
Supply current	$280\mu A$

This sensor measures the strength of the magnetic field in the environment in μT (microTesla). The measurements range between 30 μT and 60 μT near the north and south poles, respectively.

The magnetometer behaviour can be analysed in Figure 43. This sample was obtained at a frequency of 50Hz during 12 seconds with the device stationary on a flat surface and pointing north. Table 16 helps on having a better understanding about the deviations of this sensor.

8 Asahi Kasei Microdevices Corporation is the world leader in electronic compass sensors

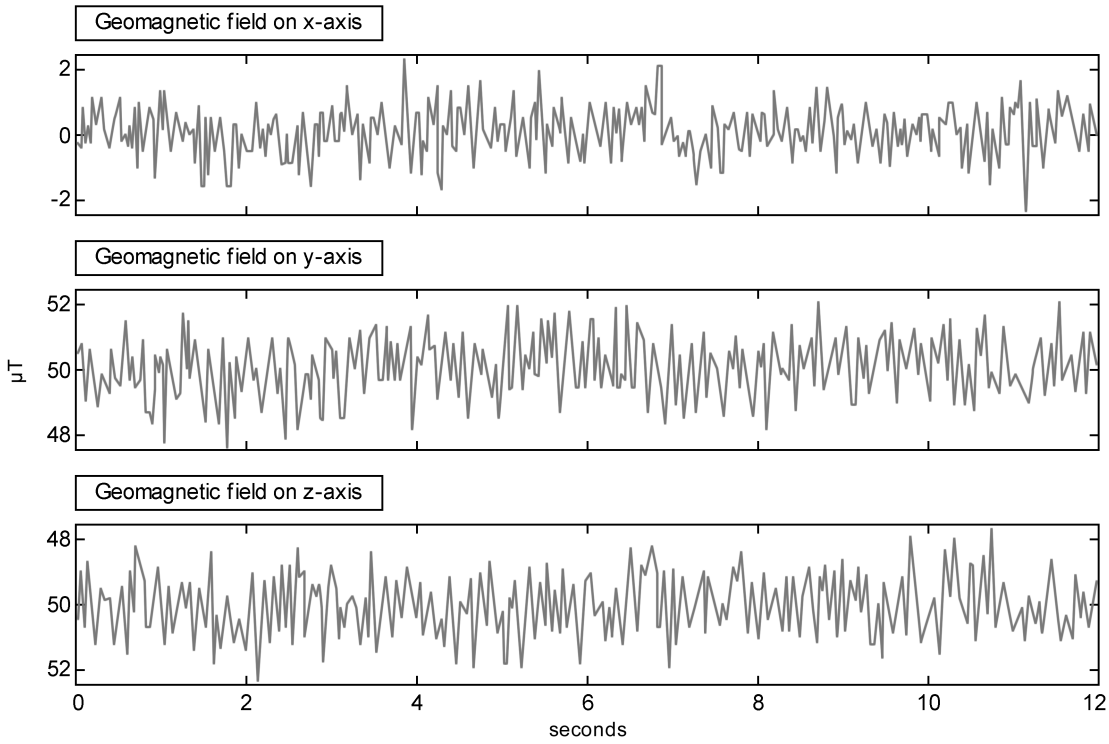


Figure 43: Waist BSU magnetometer output with the device stationary and pointing north

Table 16: Waist BSU magnetometer output statistics, in μT, with the device stationary and pointing north

Parameter	Mag_x	Mag_y	Mag_z
Minimum	-2.4621	47.6118	-52.0200
Maximum	2.2300	51.8018	-47.9184
Mean	0.00002	50.0000	-50.0000
Deviation	0.6835	0.7389	0.7351

One interesting observation is about the sensitivity of the sensor, which results in a rather high standard deviation of approximately 0.70μT for each axis. This sample was obtained with the minimum, as much as possible, magnetic interferences. This deviation can potentially produce significant errors on location estimation. However, the mean value is very close to the real measurements.

4.3.1.4 *Barometer Sensor*

A barometer sensor measures the current raw barometric pressure, which is typically used to correspond it to an altitude. The sensor used was the BMP280 from Bosch[9] and its main operating characteristics are given in Table 17.

Table 17: Bosch BMP280 barometer specifications [23]

Characteristic	Value
Pressure Range	300 to 1100 hPa
Operating Temperature	0C to +65°C
Resolution of data	0.18 Pa
Supply current	2.74μA

There are two factors that affect barometer's performance: supply voltage instability and temperature drift. These issues are somehow similar as the MEMS inertial sensors. Assuming normal values for pressure, supply voltage and temperature errors, the pressure sensor error does not usually exceed 1 to 2 kPa [122].

Typically, this sensor is used to estimate user elevation. However, it does not give altitude value but the amount of atmospheric pressure. Thus, it must be used an empirical model to convert the pressure to an altitude assuming certain initial parameters, such as pressure at sea level and the certain distribution of pressure with altitude. Another approach that can be implemented, is to use the GPS data, when user is outside, to calibrate the system based on the user elevation and the corresponding pressure value. Then when indoors the differences between pressure values can be used to estimate the floor where the user is inside the building.

In regions near sea level the atmosphere is dense, i.e. with a high concentration of molecules, which means that the pressure is higher. In higher altitudes, the atmosphere is thinner, that is, the molecules are more distant from each other. So, the pressure is lower than the regions near the sea level. The value Zero represents the total vacuum.

Figure 44 presents the barometer data in a real usage scenario. This test was performed in a four floor building, which is located at 75 meters above sea level. The first measurement is in the first floor, then the pedestrian went, through the stairs to the second floor and so on. In each floor the pedestrian remained stationary for about 10 seconds, and half way between floors he remained stationary for about 5 seconds. In the figure it is presented the obtained average pressure value at each floor and half-

9 Robert Bosch GmbH, or Bosch, is a German multinational engineering and electronics company, and it is the world's largest supplier of automotive components.

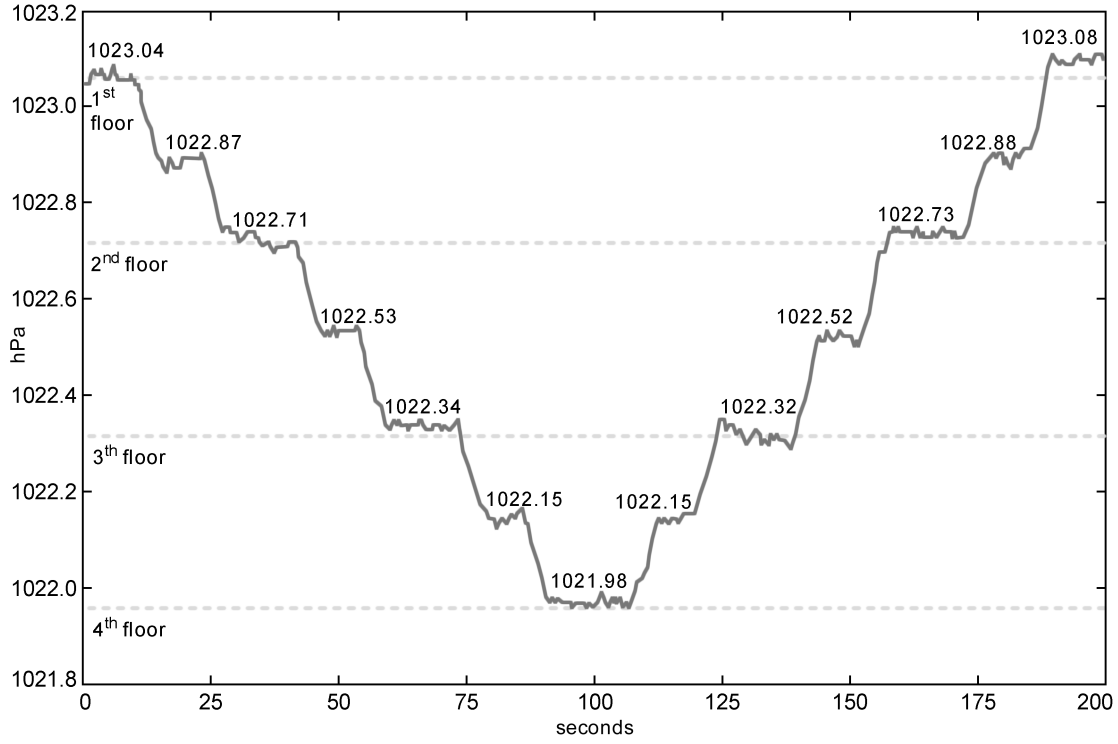


Figure 44: Barometer data in a real usage scenario in a building with four floors

floor. When the pedestrian achieved the last floor he went down to the first floor again. The height difference between floors is 2.5 meters.

From the obtained data it can be concluded that the resolution of the pressure provided by the sensor is quite good, since it varies approximately 32kPa per floor and 16kPa per half floor. Therefore, such sensor can be successfully used in differential mode. Once an altitude becomes known (either through GPS or other aiding means), a differential correction with respect to the initial height can be applied.

To measure the sensor uncertainty a test was performed during a 30-day period, during the entire month of November, 2014, where it involved different climate conditions. At 12:00am for each day of this period the pressure was measured in the same conditions and in the same place for a period of 60 seconds. The results for these tests are presented in Figure 45, which presents a graph with the average pressure value for each day of the 30-day period.

In Table 18 it is shown the minimum, maximum, mean and the standard deviation for all the data from this test. From this data it can be concluded that the values varies too much during a long period. The amount of this variations can assume values that are higher than the height of four floors. Meaning that, due to these variations, a constant value cannot be used to always correlate it to a specific floor. Therefore the

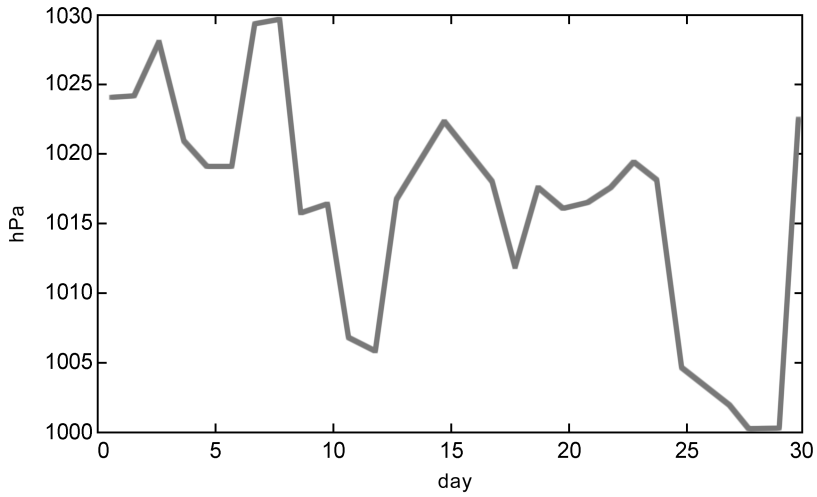


Figure 45: Waist BSU barometer output for a 30-day period

best approach, as already mentioned, is to use the differential of the pressure value starting with a reference point.

Table 18: Waist BSU barometer output statistics, in hPa, for a 30-day period

Minimum	Maximum	Mean	Deviation
996.5423	1028.9545	1015.4992	0.0157

4.3.2 *Communication Module*

The Bluetooth module used on the waist BSU was the BCM4339 from Broadcom Corporation[10].

This module is a single chip device with integrated IEEE 802.11a/b/g/n and single-stream IEEE 802.11ac MAC/baseband/radio, Bluetooth 4.0+EDR, and FM radio receiver. Its characteristics are presented in Table 19. It was used the Bluetooth Android API to send and receive the data from the foot BSU. It receives and delivers the data without any type of encapsulation.

10 Broadcom Corporation is an American fabless semiconductor company in the wireless and broadband communication business

Table 19: Broadcom BCM4339 Bluetooth specifications [45]

Characteristic	Value
Bluetooth version	Bluetooth v4.0+EDR
Class	Output power Class II ($\pm$2.5mW range $\pm$10m)
Flash memory	8Mbit
Power supply	2.3v to 4.8v
Interfaces	UART

4.3.3 *Software*

The software on the waist BSU is mainly composed by a *Data Retrieval Process*, as presented on Figure 46. After all the initializations taken place, it is created one thread per each BSU that is connected to the mobile device and executes the *Data Retrieval Process*. In this case there are two threads: (i) the "Internal Sensors Thread" that initializes the Android sensor listeners to receive the data from the device's hardware; (ii) and the "External Sensors Thread" that communicates with the foot BSU to receive its data. Another threads can be connected to execute the same process.

The *Data Retrieval Process* works in the following way, when a new sensor measurement is received, it is recorded to the corresponding file and the data is shown on

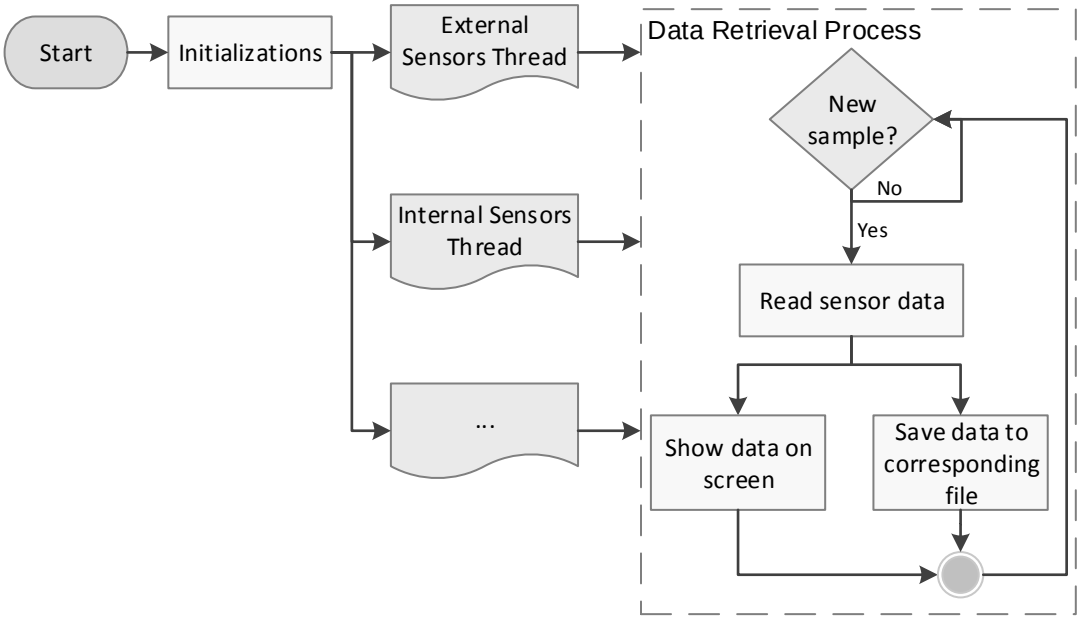


Figure 46: Waist BSU software

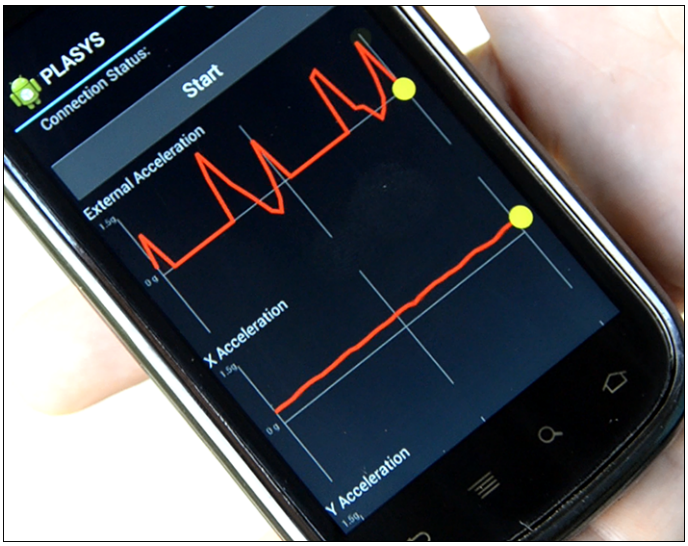


Figure 47: PLASYS Android mobile application

the mobile device's screen, as demonstrated in Figure 47. As already presented on previous sections the sampling periods for both BSUs are 50ms (20Hz).

The data from each sensor is recorded in the corresponding file in the mobile device internal memory. This data is stored to be post processed to calculate the number of steps taken by the user and the displacement.

The files that are created and stored, by this application, with the information from each sensor are the following:

- fAccelerometer - accelerations sensed on the foot;
- fForceA - force sensed by force sensor A;
- fForceB - force sensed by force sensor B;
- fGyroscope - angular velocities sensed on the foot;
- wAccelerometer - accelerations sensed on the waist;
- wBarometer - pressure values sensed on the waist;
- wGyroscope - angular velocities sensed on the waist;
- wMagneticField - magnetic field perceived on the waist.

The foot BSU files are the ones that start with an "f", and the waist BSU files are the ones that start with an "w". Per each 12 seconds are generated, by the two BSU sensors, about 250kb of information. Each file is composed, for each reading, by the timestamp of the measurement and the corresponding measurements for the *x-axis*, *y-axis* and *z-axis*. These values are comma separated, with the following format:

timestamp	x_measurement	y_measurement	z_measurement

For the sensors that have only one measurement (force and barometer), only the *x_measurement* is provided.

An important issue is the time synchronization between the data obtained by the two BSU. The collected data must be synchronized with a sufficient accuracy for this type of application.

If the data, of the two BSU, at a certain time (i.e. absolute time) have different time stamps, the integration results will be useless. The accuracy requirement of the time synchronization for a given application is essentially driven by the speed of the linear and angular motions of the object on which the system is installed. The faster and more manoeuvrable the object, the better must be the timing accuracy.

Therefore, there are different accuracy requirements for time synchronization for aviation, vehicular or pedestrian navigation applications. For aviation GPS/INS integrated systems, where the speeds are reaching a thousand kilometres per hour, a tenth of a second error in timing between the two systems can cause tens of metres of relative positioning errors between the systems. For the pedestrian navigation application with speeds of 1-2 metres per second such a timing error will cause a centimeter level positioning error in the integrated solution. This magnitude of error is tolerable considering the individual accuracy of a PDR [81].

Obviously, the better the time synchronization is, the better is the accuracy of the integrated system. Unfortunately, a more accurate time synchronization requires more hardware and software resources. When aiming for a low-cost solution, this is not always possible.

4.4 SUMMARY

In this chapter was presented the system architecture, as well as, the developed hardware and the components used to create it [4]. Also, it was provided an analysis to the errors of each sensor implemented in the prototype. Due to some robustness problems regarding the ZigBee communication protocol, the waist BSU was simulated using a validated platform. Thus, the waist BSU was simulated using an Android mobile device, the Google Nexus 5, which has similar characteristics as the developed BSU.

When referring to low-cost IMU it implies different things for researchers, since for some a thousand euros IMU is considered low-cost. However, for PINS a low-cost IMU

should cost no more than one hundred euros. This price restriction limits the use of MEMS sensors that are truly low-cost, at a price rounding € 10 or less in current or future mass production. The developed unit has been designed and assembled in order to meet this requirement and has an estimated cost, for the hardware, of approximately € 50.

The resulting BSU is light, with a weight of about 95 grams without the batteries. The prototype uses 4 AA size batteries. However, in a final version more lightweight and small batteries could be used. In terms of the form factor it has the following dimensions 90mm x 65mm x 20mm for length, width and height, respectively. The prototype unit with this dimensions and weight is easy to carry on a belt or on the person's leg. The hardware can achieve a maximum sampling data rate of 100 Hz, but it was decreased to 20 Hz, which is sufficient to include the frequencies in the signals induced by the user walk. Also, it is important to decrease the amount of data, and therefore the necessary computing processing power to handle that data [8].

Since in this work, mobile devices are used to exchange information with other sources [6] and to handle some calculations that can take some time to process, all the algorithms need to be optimized to run in this type of devices [7]. That is why some pre-processing to the data is done in the BSUs.

Every time the system is used the user must be standstill for five seconds, in order to the algorithms calculate and correct the sensors offset and bias. Then, this offset is removed from each new measurement obtained by the sensors.

In the future it is wanted to integrate these sensors into person's clothes and shoes, to make the system more imperceptible to the user.

5

STEP DETECTION

As stated in Chapter 2 a PINS algorithm is divided in three phases, where the first one is the step detection. This is the most important phase, because if the system is not able to properly detect a step then all the displacement estimations will be faulty, since a pedestrian moves through the swing of the legs.

In this chapter are presented three techniques to detect a step with the highest possible accuracy. The first is the typical algorithm that uses only the accelerometer data to detect a step. The second uses the angular velocity given by the gyroscope to detect a step. Finally, the third one is our proposal that combines an accelerometer with a force sensor to detect more accurately a step.

5.1 ACCELEROMETER BASED ALGORITHM

Step detection is the automatic determination of the moments in time at which footsteps occur. The most used technique, to detect these moments, is based on the foot accelerometer measurements associated with one of the following algorithms: peak detection, zero crossing detection and flat zone detection.

The peak detection is based on a threshold applied to the minimum and maximum values of the acceleration magnitude, in order to identify a step. However, it has low accuracy because sometimes the accelerometer gives erroneous measurements, therefore false steps are normally detected by this algorithm. Zero crossing detection suffers from the same problems has the previous algorithm, since the accelerometer has a very unstable signal and varies a lot between positive and negative measurements. Finally, the detection of flat zones in the accelerometer signal has very low accuracy in a high velocity walking, since in these conditions the flat zones are not noticeable.

Figure 48 presents the algorithm implemented to detect the steps given by a pedestrian, which is only based on the accelerometer data as a typical PINS.

For each new measurement given by the accelerometer, it is calculated the acceleration magnitude, then 3^{th} order low-pass filter is applied.

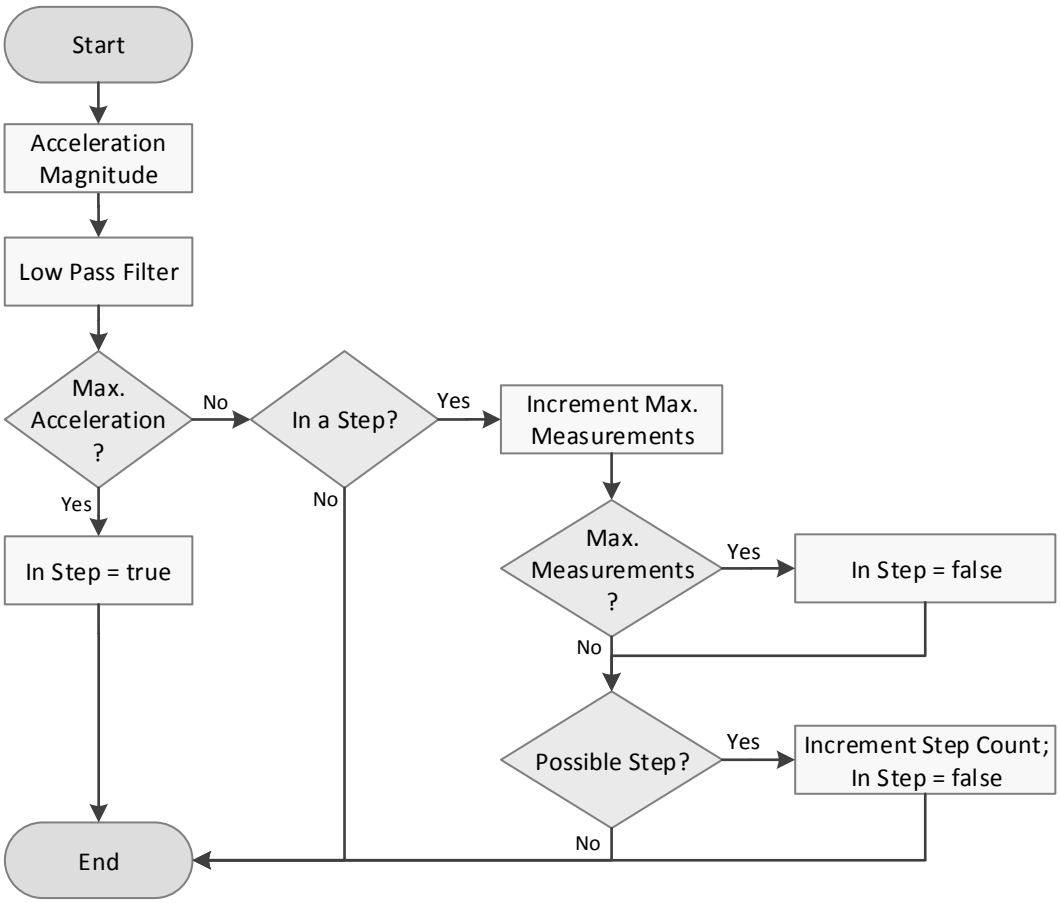


Figure 48: Step count algorithm using only the accelerometer

The obtained acceleration is then compared to the previous higher acceleration value. If the acceleration is higher than this value it will be verified if a step was previously detected for this period, if not the "In Step" variable will be assigned as "true".

If the detected acceleration is lower than the maximum value and if a possible step is not being detected ("In a Step?" condition) the algorithm ends and waits for another measurement. If the algorithm is detecting a possible step the maximum measurements variable, that has as default value 20, which was the value defined as the period, in average, in which a step takes to be made, is decremented. When this value assumes a value of 0 the algorithm identifies that the step must be ended. During this period the maximum and minimum acceleration values are stored to be used in the next condition.

If the algorithm is detecting a possible step ("Possible Step" condition) it is verified if the maximum acceleration value minus the current acceleration value is higher than the threshold. If true, a new step is counted and the variables are initialized. If false, the algorithm ends. The threshold was defined as 1,7 m/s^2.

This algorithm uses the maximum minus the current acceleration values because the maximum peak of acceleration appears at the beginning of a step and the minimum peak of acceleration appears at the end of the step.

Figure 49 shows the foot acceleration signal (dashed line) for a ten step walk, where the acceleration values detected for each step, by the algorithm, are identified by a solid line. In this figure can be verified that there are some accelerations peaks that almost exceed the threshold value and can be detected as a false positive step. This phenomenon can be observed, for example, between the fourth and fifth step, where the difference between the minimum and maximum acceleration value is 1.5 m/s^2, which is very close to the defined threshold. Also, it can be visualized the flat spots that occur at the end of each step. However, from the ninth to the tenth step this zone is almost non-existent, making the step hard to identify when using the flat zones detection technique.

The waist accelerometer was not used to detect a step since it provides to many acceleration erroneous measurements. This mainly happens because of some strange moves that the pedestrian's waist can take. In Figure 50 can be seen the data sensed by the accelerometer on the waist. In this representation four steps were taken by a

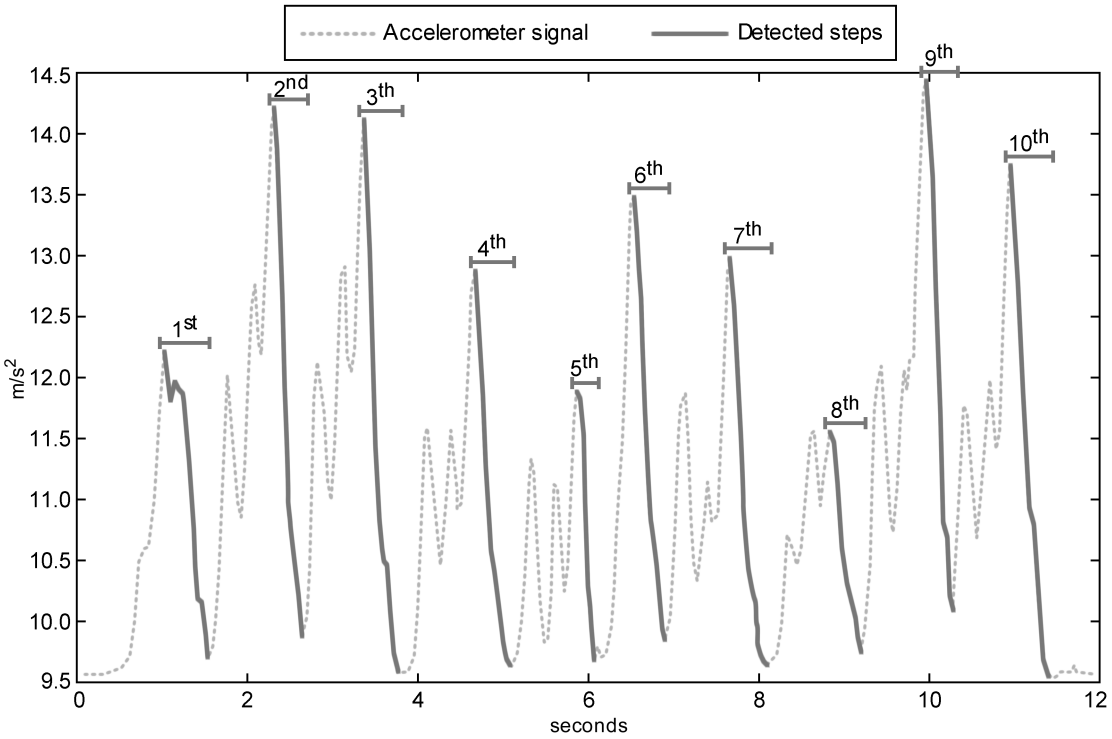


Figure 49: Foot accelerometer signal for a ten steps walking

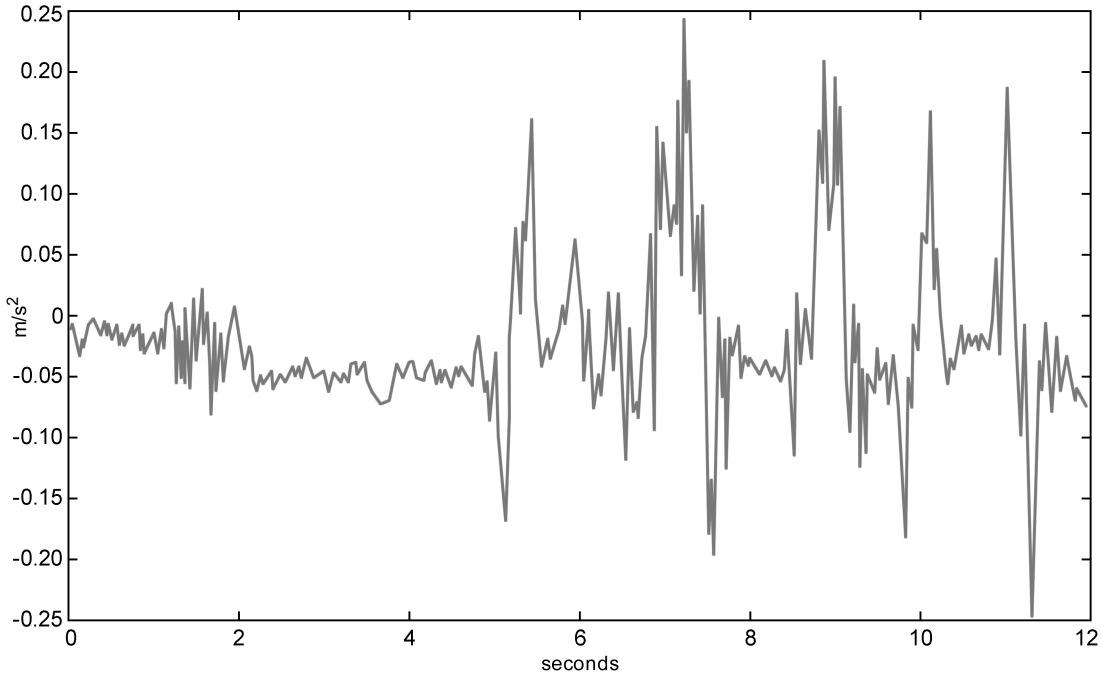


Figure 50: Waist accelerometer signal for a four steps walking

pedestrian. However, using a threshold algorithm more steps, than the four that were really performed, could be detected.

For these reasons the usage of the waist accelerometer was left behind on the step detection algorithm implementation.

5.2 ACCELEROMETER AND FORCE SENSOR BASED ALGORITHM

To improve the step detection, force sensors were included in the pedestrian's foot. Using these sensors the step detection can be improved, since they provide accurate information about the moments when the foot touches or leaves the ground and the respective contact force. This information can be also useful to detect, more accurately, the type of activity that the pedestrian is performing (e.g. running, walking fast).

If the user is in a stationary state, both force sensors must measure the weight that is in contact with them, meaning that the force will be different of zero. The implemented algorithm combines the information from the force sensors with the accelerometer measurements to achieve more useful information about the step in order to eliminate false detections of swing phases when the foot is stationary.

The positioning of the force sensors, shown in Figure 36, is important to provide information about the beginning of a step until its end. The sensor A is placed in a

position where the first contact with the ground occurs and the sensor B gives information about the last contact of the foot with the ground and from this point the foot is in the air until the force sensor A measures a force applied to it.

The aim of this algorithm is to have the smallest counting step error possible, using low computational power as much as possible, since it must detect the steps in real-time. This step count algorithm is presented in Figure 51.

For each iteration the acceleration is projected from the inertial frame to the navigation frame to obtain the acceleration sensed on the *x-axis*. Then it is filtered via a 3^{th} order low-pass filter, and the data from the two force sensors is merged. In this stage

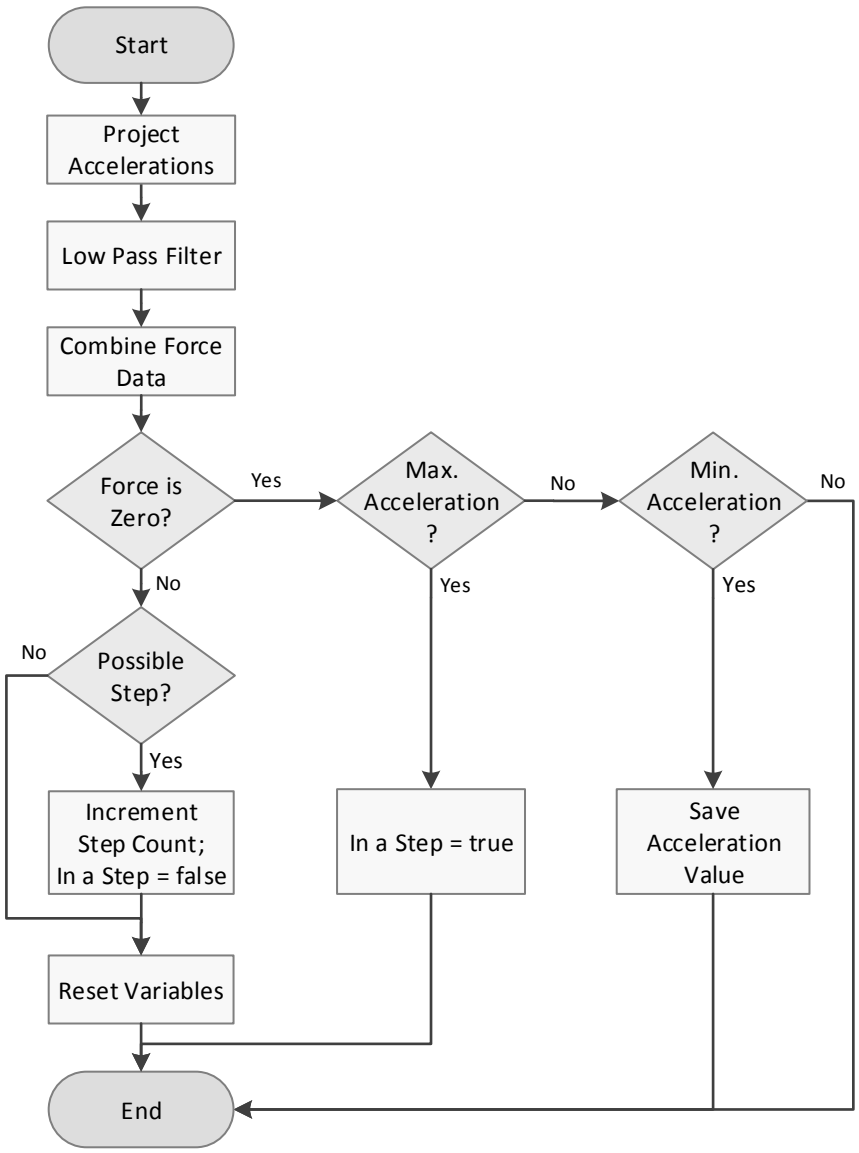


Figure 51: Step count algorithm using the accelerometer and force sensors

the combined force will be different of zero when a force is applied to the heel until no force is applied on the front of the foot (i.e. when any part of the foot is in contact with the ground). When the combined force is different of zero the acceleration is defined to a predefined value of 0, because in these periods the foot is on the floor and is not moving, so there is no displacement to estimate.

If the combined force is zero, the current acceleration value will be compared to the maximum acceleration sensed in this step. If it is higher, then the algorithm will mark it as a possible step. If the acceleration is lower, it will be verified if it is lower than the minimum acceleration sensed in this step. With this process the maximum and minimum accelerations measurements sensed in a step will be saved. It is during this phase that ZUPT and ZARU algorithms can be applied.

When the combined force is different of zero, meaning that the foot is in the stance phase, it is verified, in the "Can be a Step?" condition, if a step was previously detected, and it is also verified if the difference between the minimum and maximum accelerations measurements sensed in the step are higher than a threshold. If both conditions are true then the step counter is incremented.

If false the variables are initialized. The threshold value was defined as 1, which is lower than in the previous algorithm. A lower value was chosen because the force sensors were valuable to improve the step detection and the acceleration threshold is only a second validation, to identify if it was a step and not only a lift of the foot, where a forward (or backward) movement were not performed. Thus, the number of steps is equal to the number of times that the combined force is zero and the acceleration is above the threshold.

This sensor fusion enables the implementation of an algorithm that does not detect false stance phases when the foot is already in the swing phase and permits a computationally lighter algorithm.

In Figure 52 is shown the accelerometer and force sensors data for the same test presented in Section 5.1. This test involved a straight walking path with ten steps. The solid line represents the acceleration sensed in the *x-axis*, only when the force sensors measurements are equal to zero. As referred earlier when the force is different of zero the accelerometer measurements were forced to assume the values of 0. The dotted line represents the data from force sensor A and the dashed line represents the data from force sensor B.

Through this analysis it is possible to notice that the pattern for each step is repeated along the walk and the data from the force sensors take distinct values when the foot is

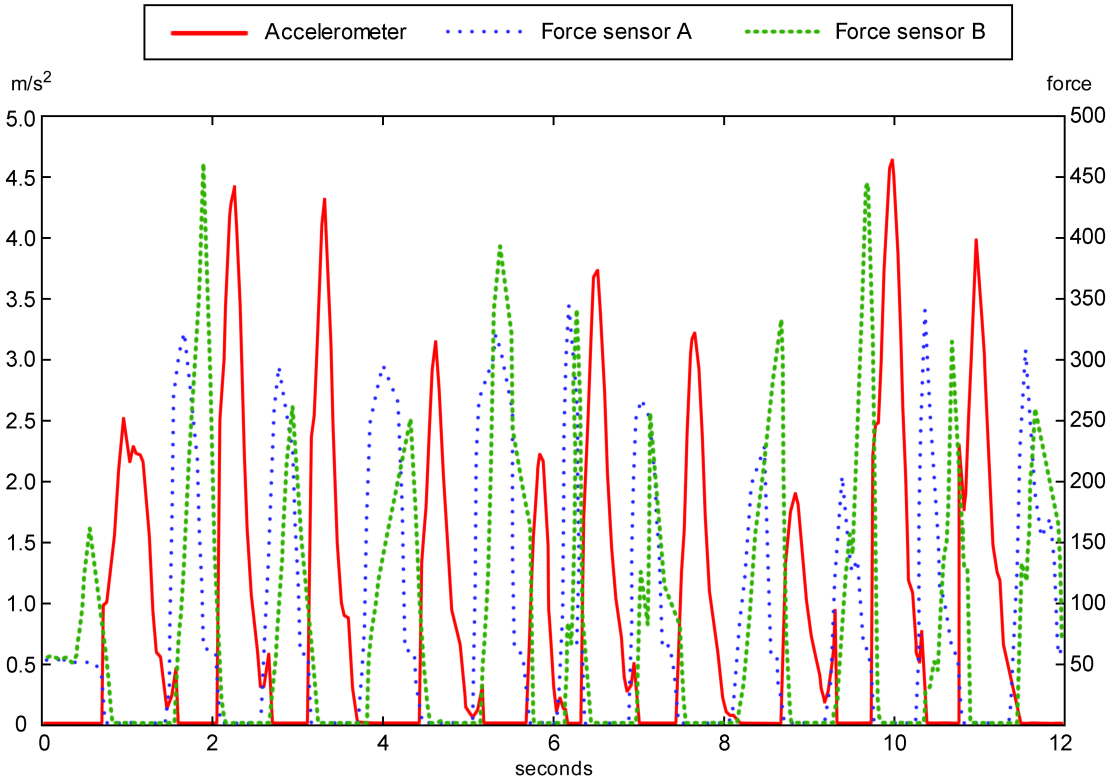


Figure 52: Force sensors and accelerometer *x-axis* data for a ten steps walking path

stationary and when the foot is in motion. As can be seen the values of force are zero, or almost zero, when the foot is moving (when there is higher variations in acceleration values), and when the foot touches the ground, sensor A goes first to a high value and then sensor B goes to a high value, as expected.

5.3 GYROSCOPE BASED ALGORITHM

Using the accelerometer on the waist to perform step detection is very complex and has to many errors, as seen on Section 5.1. However, another sensor can be used to detect a step using the waist BSU.

While a human is walking, the entire lower segment, consisting of the pelvis, femur, tibia and fibula, performs a rotation through the transverse plane[1]. In this case it is interesting to analyse the pelvic rotation. A representation of this rotation can be seen in Figure 53.

1 Transverse plane, which can be also called the horizontal plane, is an imaginary plane that divides the body into superior and inferior parts.

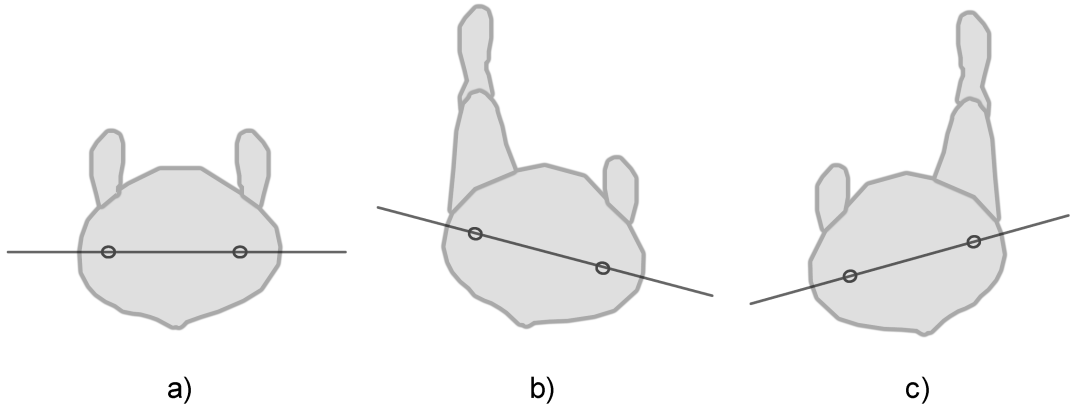


Figure 53: Pelvic rotation: a) pelvic standstill; b) pelvic rotation during a left step; c) pelvic rotation during a right step;

In a normal walking the pelvis rotates alternatively to right and to left in relation to the line of progression in a transverse plane about the vertical axis. During a normal walking path, on a flat surface, the average magnitude of this rotation is approximately four degrees on either side of the central axis. Giving a total rotation of eight degrees. The degree of rotation is higher as the step is longer [121].

This rotation can be measured using the gyroscope included in the waist BSU. Thus, an algorithm to count the steps through the analysis of these rotations was developed. Because of the waist BSU placement (recalling Figure 28) the algorithm input is the gyroscope *x-axis* measurements. Also, according to this placement, a negative rotation occurs when a right step is given and a positive rotation happens when a left step is given.

As represented in Figure 54 for each gyroscope measurement the algorithm starts by subtracting the gyroscope offset. Then a 3^{th} order low-pass filter is applied. The next step verifies if the measurement can indicate a possible step, referred in the figure as "Possible Step". This verification is performed by comparing the measurement with a threshold. The threshold value was defined at $0.11\ rad/s$, since it is the average value that the pelvic rotates during a walk [121].

If the algorithm detects a possible step, the peak value of the signal is obtained. Typically, this peak value is an interval of 50 measurements after passing the threshold value. Thus, when this value is obtained or until the gyroscope measurements changes the signal value, from positive to negative or negative to positive according to the step performed. When this step signal is obtained it is calculated the probability of it being a true step or not.

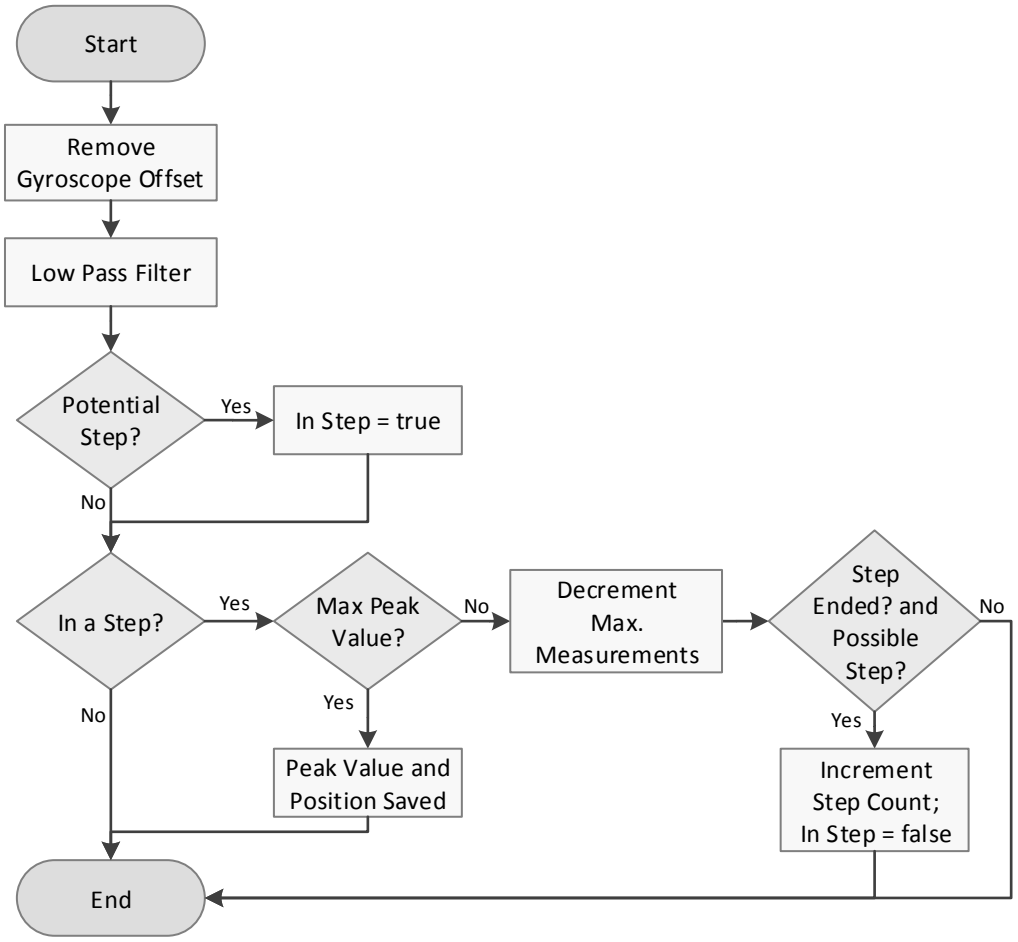


Figure 54: Step count algorithm using gyroscope data

The probability is calculated based on the following information (features) about the signal wave: amplitude, length and two Δt. These features are represented in Figure 55.

The minimum values that each parameter must have to achieve a 100% of probability are: 0.20 rad/s for amplitude; 90 measurements for signal length; for the first delta which is considered at 15 measurements after the peak value is 0.05 rad/s; and for the second delta which is considered at 25 measurements after the peak value and must have at least a difference of 0.10 rad/s. If one or all the parameters are below these values a probability, based on Equation 27, is calculated based on these default values. These deltas were chosen because it is on these moments that higher differences are denoted, to distinguish between a step or an erroneous measurement.

$$P(S) = (P(A) * 0.25) + (P(L) * 0.10) + (P(D1) * 0.40) + (P(D2) * 0.25) \quad (27)$$

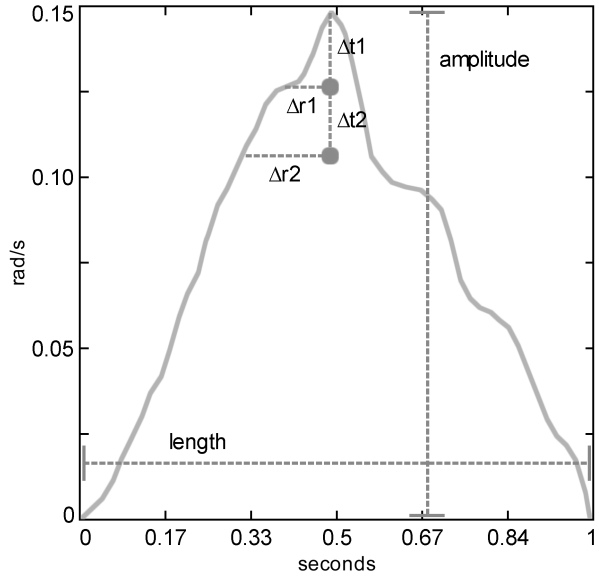


Figure 55: Parameters used to classify the gyroscope signal

where P(S) represents the probability of the signal to be a step based on the sum between, the probability of the signal amplitude P(A) with a weight of 1/4, the probability of the signal length P(L) with a weight of 1/10, the probability of the first delta P(D1) with a weight of 2/5 and the probability of the second delta P(D2) with a weight of 1/4.

These weights were distributed according to the importance that each parameter has in order to declared a signal as a true step. If this probability is higher than 75% then the signal is considered as a step and the step count incremented. This value was chosen since if the probability is lower than that value, it is most likely to be an erroneous value and not a step. From all the collected datasets, a step, has in average, a probability higher than 85%.

In Figure 56 is shown the gyroscope data for the same test presented in Section 5.1. This test involved a straight walking path with ten steps. The solid line represents the gyroscope *x-axis* signal. The dashed circles represent the detected right steps and the solid line circles represents the detected left steps.

5.4 EVALUATION

To compare the three implemented algorithms, two different case study scenarios were created, which are represented in Figure 57. The first one, represented as an "A" in the figure, is a straight walk with a distance of 20 meters, which gives an average of 40

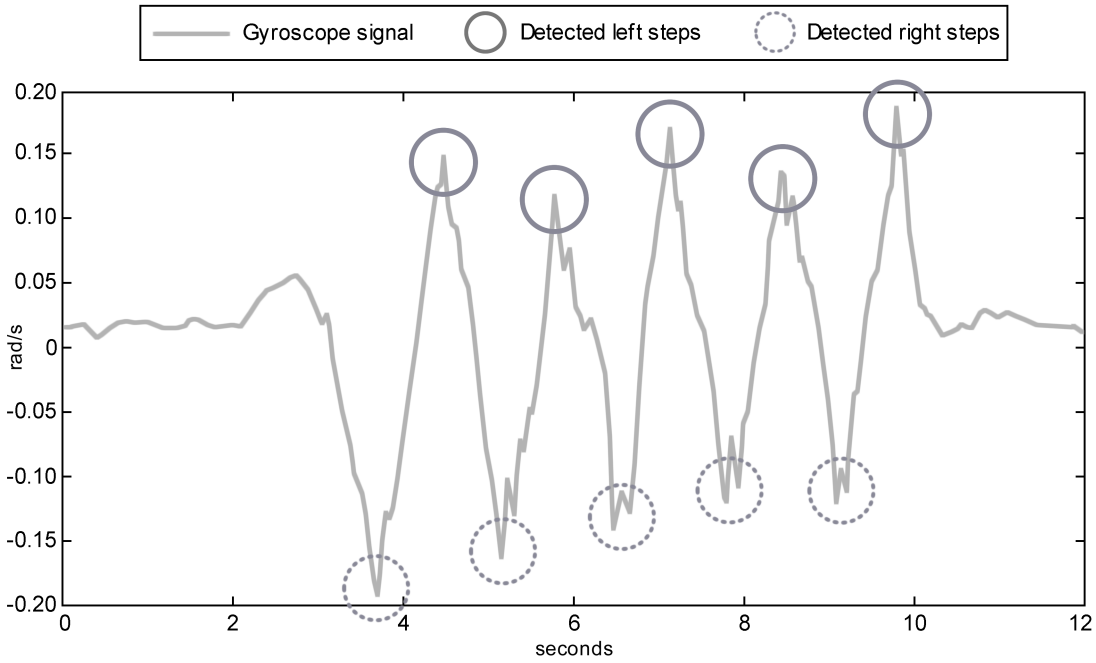


Figure 56: Gyroscope's *x-axis* data for a ten steps walking path

steps, in a normal walking pace. The second one, represented as a "B" in the figure, is a walking path with two curves in the middle with a total walking distance of 25 meters, which gives an average of 50 steps, in a normal walking pace.

The two BSU where placed in the pedestrian's body in all the tests. Thus, the same data was used to test each algorithm.

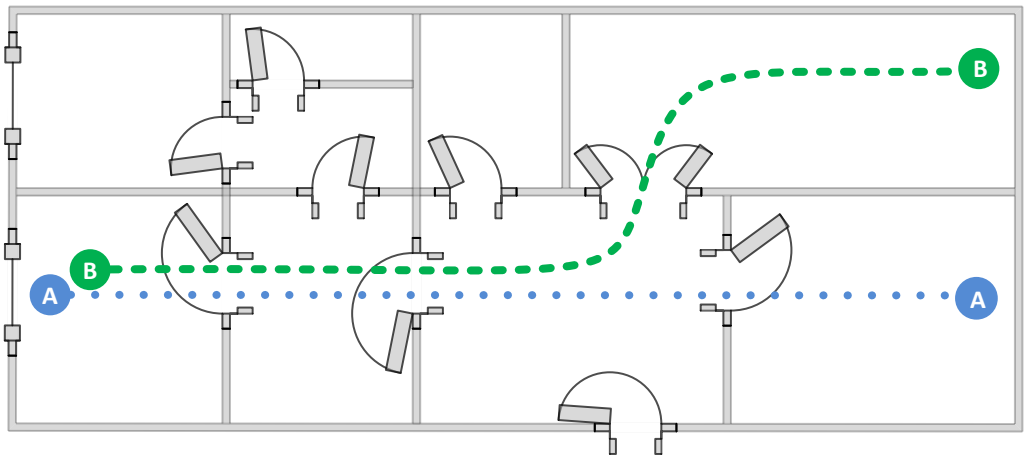


Figure 57: Evaluation Scenarios A and B: A) is a straight walking path; B) is a straight walking with two curves in the middle of the path

Each scenario was performed using three different types of walking: slow, normal and fast. For each type of walking ten tests were performed, which gave a total of 1200 steps for the first scenario and 1400 steps for the second scenario.

The results presented in Table 20 and Table 21 were obtained for the two scenarios, covered in the three types of walking and performed by two pedestrians. The data was firstly collected in the real scenarios and then each algorithm was executed with the same datasets. Both tables show the error percentage on step counting obtained for each algorithm.

In Table 20 are presented the results for the first scenario. The *Accelerometer and Force sensor* approach had the best results in all the walking paces. The results, for this approach, were very similar for each type of walking. The walking pace that has more errors is the fastest one, because it is the most stressful for the system. This type of walking has the hardest step types to be detected by the force sensors, since the force is mainly in the front of the foot and sometimes it is not directly made in the sensor location, thus some steps are not detected accurately.

Table 20: Step count algorithms results for the first scenario (A)

Algorithm	**Slow Walking**	**Normal Walking**	**Fast Walking**
Accelerometer	0.2%	0.1%	1.6%
Accelerometer and Force	0.0%	0.1%	0.4%
Gyroscope	1.4%	2.6%	4.0%

Comparing the *Accelerometer and Force sensor* approach with the *Accelerometer* approach the biggest different is obtained when walking fast. The *Accelerometer* approach, in fast walking, presents too many false positive steps. The other types of walking present minimal differences and these errors tend to be reduced over time, since the hardest steps to detect are in starts/stops and sharp curves. However, the inclusion of the force sensors enables the implementation of a more precise and a more stable system in a long term use, where different types of walking can be achieved.

The *Gyroscope* approach is the one that presents a higher error percentage. For all the walking paces the main problem is always the false positive steps that are detected. This mainly happens because the waist sometimes has unpredicted movements which are not reproduced on the foot movements. Specially in a fast walk the pelvis rotations tend to be more expressive and random, meaning that too much noise exist in the signal.

Table 21 shows the results for the second scenario. Besides the two curves at the middle of the walking path, the results follow the same pattern and are very similar for the *Accelerometer* and the *Accelerometer and Force sensor* approaches. The *Gyroscope* approach has increased the step count errors. The main problem of this approach is that the curves are heavily sensed by the gyroscope, which can be detected as more than one step, giving too much false positive steps.

Table 21: Step count algorithms results for the second scenario (B)

Algorithm	**Slow Walking**	**Normal Walking**	**Fast Walking**
Accelerometer	0.2%	0.2%	1.8%
Accelerometer and Force	0.0%	0.2%	0.5%
Gyroscope	2.2%	3.5%	5.6%

Another problem of the *Accelerometer* approach, apart from the lower accuracy when compared to the *Accelerometer and Force sensor* approach, is that it is computationally heavier [115]. Which can result in a significant delay when compared to the second approach. Resuming, the *Accelerometer and Force sensor* approach is computationally lighter. Its disadvantage is that it depends on more sensors. However, this inconvenient is surpassed by the superior accuracy and lower delays to computationally detect a step.

A more evident advantage of the *Accelerometer and Force sensor* approach is in stressful situations like in fast walking, which continues to have a good accuracy when compared to the other approaches. Also, the results have shown that the *Accelerometer and Force sensor* approach is equal or better in a long term use.

Concluding, the good results obtained in step detection can led to promising results in the displacement estimation. Every time a step is detected, ZUPT and ZARU techniques can be applied which helps mitigate the drift problem that normally exists on PINS. Integration of the inertial measurements only during the swing of legs is important to reduce the displacement errors [9].

6

STEP CHARACTERIZATION

In typical PINS implementations, after detecting a step, the displacement is estimated. However, in PLASYS, after the step detection a step characterization or classification is performed. This characterization is important to eliminate some of the erroneous measurements given by the inertial sensors. This is a very important phase, because it is here that the learning algorithms are applied to correct those errors.

Having a proper step model, it can be provided a more accurate displacement estimation, since some errors can be suppressed using this model. For example, the accelerometer does not capture perfectly the accelerations, meaning that the integration of the acceleration produces several errors. However, it contains the information needed to correctly classify a step by using its patterns.

This step characterization or classification includes: (i) the type of terrain (i.e. normal, descending or ascending stairs) where the pedestrian is walking; (ii) the direction (i.e. forward or backward); (iii) and the size (i.e. short, normal or long).

For each type of classification several algorithms/techniques were implemented and evaluated in order to identify which one provides the most accurate results. The implemented algorithms are based on some heuristics, DTW methods and machine learning algorithms, as Neural Networks and SVM. The results are presented for each BSU without fusing their information, and with their information fused, which are represented with the name of the approach combined with the word "Fusion" (e.g. Heuristic Fusion, Neural Network Fusion and SVM Fusion).

Combining the two sources of data, waist BSU and foot BSU, the quality of the data is improved, since the probability that two sources of data give erroneous measurements patterns at the same time is much reduced. The fusion between all the sensors information improves the number of accurate classifications. Thus, this integration leads to a better characterization of the step.

The implemented algorithms perform some treatment to the obtained signals, before applying the machine learning techniques. Because, giving to a classifier a complete signal can be very heavy and confusing to the machine learning algorithm identify the patterns of the signal and therefore estimate the correct label for that pattern. Usually,

features are extracted from the signal to increase the overall performance of the learning system. Meaning that it generally reduces the dimensionality of a problem domain for the purposes of improving the performance of machine learning algorithms and to decrease the computational load.

Thus, each learning algorithm will use the raw signals, which are first pre-processed to remove some of the noise, to extract some features from the signals, then classify it and finally will reasoning about the obtained data. This process comprises several stages, data acquisition, signal pre-processing, feature extraction and selection, training and classification. The process is described in Figure 58 where the features F_i are passed to a model with n parameters that estimates the characterization classes Y_i with a confidence P_i.

The learning algorithms were implemented based on supervised learning, which learns to estimate an output based on a given input. This type of learning needs a lot of experiments in order to have a proper understanding of the problem. The output of these algorithms is a class label. The walking characteristics are learned from a set of exercises previously elaborated by the pedestrian.

To identify the advantages of the learning approaches, other approaches were implemented: (i) *Heuristics* and (ii) DTW.

The *Heuristics* approaches were implemented to simulate the typical PINS approaches that retrieve some features from the sensors signals which are then compared with some thresholds. After this comparison some rules are applied in order to perform the distinction between each type of step. The thresholds were chosen based on the average values, which were obtained from all the datasets collected during this research work.

The DTW approaches compares the signals obtained from the sensors with a dataset of similar signals, which are labelled with the respective class. The class that has the lowest distance to the obtained signal is the one that is selected by the algorithm.

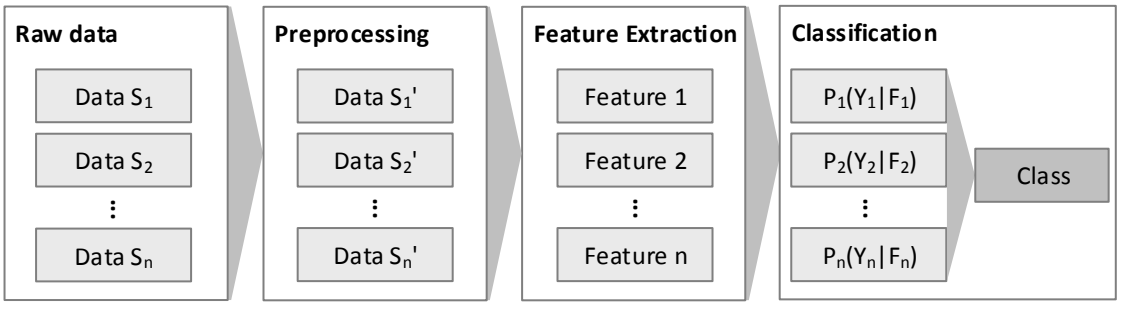


Figure 58: Learning algorithms process chain to classify a step according to the sensors data

This approach is similar to learning algorithms, since it gives a result based on past experiences. However, this approach is slower and does not generalizes as well as the learning algorithms.

Finally, for each type of characterization an evaluation evolving all the implemented algorithms is performed. To perform the evaluations the data was first collected and then post processed using Matlab to obtain the results, meaning that the same dataset was used to test each algorithm. These simulations were performed on a low performance computer, in order to have similar results as nowadays high-end mobile device [6], a Pentium 4 2.8Ghz with 1GB of RAM memory.

6.1 TERRAIN TYPE CLASS

The first characterization that is performed, is about the type of terrain where a step was given. There are three possibilities: (i) in a normal (flat) terrain; (ii) or in ascending; (iii) or descending stairs. The type of terrain was classified with the specified approaches, where the learning algorithms are presented in Section 6.1.1 and the *Heuristic*, *Heuristic Fusion* and DTW approaches are presented in Section 6.1.2. The algorithms evaluation is presented in Section 6.1.3.

For the type of terrain characterization it was used the data from three sensors: (i) foot accelerometer (*y-axis*); (ii) foot gyroscope (*z-axis*); (iii) and waist accelerometer (*x-axis*). In Figures 59, 60 and 61 are represented the signals obtained from these sensors for each class.

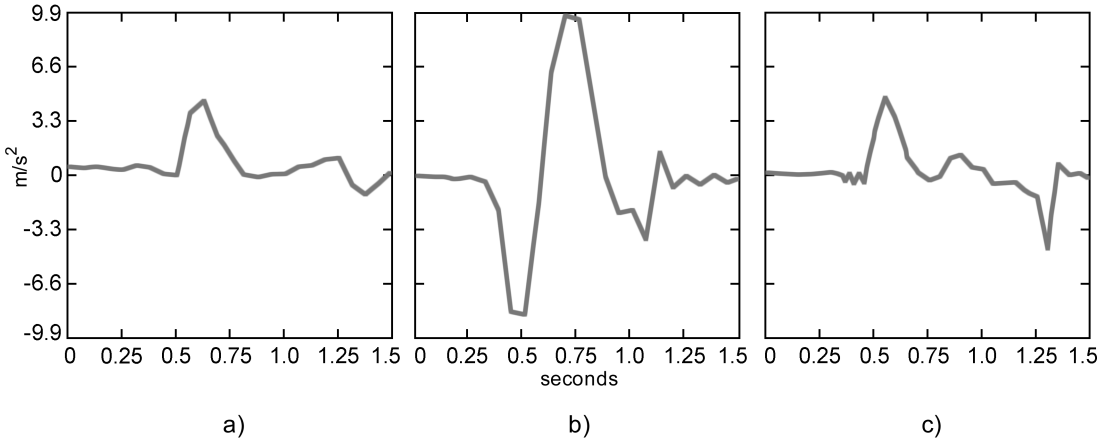


Figure 59: Foot accelerometer (*y-axis*) data for each step terrain characterization: a) Acceleration signal pattern in a flat surface; b) Acceleration signal pattern when ascending stairs; c) Acceleration signal pattern when descending stairs

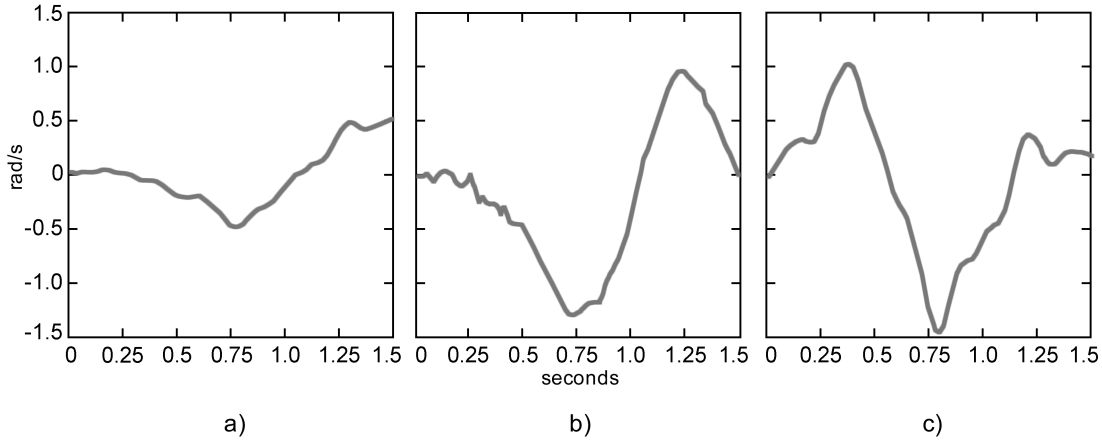


Figure 60: Foot gyroscope (*z-axis*) data for each step terrain characterization: a) Gyroscope signal pattern in a flat surface; b) Gyroscope signal pattern when ascending stairs; c) Gyroscope signal pattern when descending stairs

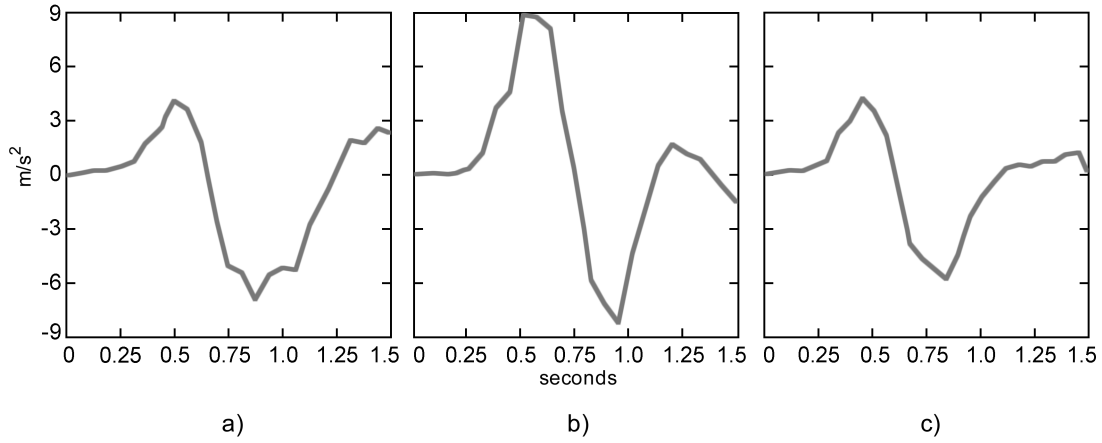


Figure 61: Waist accelerometer (*x-axis*) data for each step terrain characterization: a) Acceleration signal pattern in a flat surface; b) Acceleration signal pattern when ascending stairs; c) Acceleration signal pattern when descending stairs

The *y-axis* of the foot accelerometer gives a good indication about the foot elevation, which is essential to distinguish between ascending or descending stairs, since the forces are the opposite. However, from the several tests performed it was noticed that the main distinction that can be made using this sensor data is between ascending stairs and the other types of terrain. Visualizing Figure 59b it can be seen that on ascending stairs terrain the acceleration achieves higher values than in the other two cases. This happen because when ascending a stair the foot has to perform a higher elevation than in the other cases. Regarding the other two types of terrain, descending stairs and normal, the data obtained from this sensor is very similar, the main difference is at the end of the step that, in the case of descending stairs, a higher acceleration is sensed since the foot touches the ground with a higher force than in the normal terrain type.

The *z-axis* of the foot gyroscope provides information about the foot rotation in each type of terrain. Comparing the different signals presented in Figure 60, the foot rotation is much more noticeable in the ascending and descending stairs terrains. When ascending stairs it has an upward rotation peak and then a downward rotation peak, and it is the opposite when descending stairs. The data from this sensor is very important to make the distinction between these two terrains. Regarding the normal terrain, the pattern is similar to the descending stairs. However, the sensed rotation is much softer. Nonetheless this sensor provides a good accuracy on making the distinction between the three types of terrain.

Finally, the *x-axis* of the waist accelerometer provides similar data as the foot accelerometer. In ascending stairs a higher acceleration is sensed, in both positive and negative scales. When descending stairs this acceleration is much lower than in the other two types of terrain. The acceleration sensed in the normal terrain is within the other two. It provides similar data to distinguish between a flat surface and descending stairs. However, when ascending stairs it provides distinguishable data.

Considering the data provided from these signals it can be established that, combining their data, they are suitable to be used to differentiate each possible characterization terrain. Since the strengths of each signal can be combined to achieve a final consensus.

6.1.1 *Learning Algorithms*

Each learning algorithm was tested with several different inputs. In this characterization each learning algorithm had the best results with two distinct inputs for each other. The inputs used on each learning algorithm will be presented, separately, when presenting it.

The learning algorithms were trained with a total of 970 samples, 358 samples of ascending stairs steps, 358 samples of descending stairs steps and 254 samples of normal terrain steps. To validate the algorithms a total of 170 samples were used (62 ascending, 62 descending and 46 normal). To test the algorithms a total of 540 samples were used (180 ascending, 260 descending and 100 normal), and a 10-fold cross-validation using these datasets was also performed.

6.1.1.1 *SVM*

The SVM algorithm has achieved the best results using as input a set of features retrieved from each sensor signal. From the foot accelerometer are used 6 features: (i) minimum acceleration value; (ii) maximum acceleration value; (iii) difference between the instant moments on which each of the acceleration, minimum and maximum, peak values occur; (iv) difference between the maximum and minimum acceleration peak values; (v) sum of all the negative acceleration measurements; (vi) and the sum of all the positive acceleration measurements.

From the foot gyroscope are used 5 features: (i) minimum rotation value; (ii) maximum rotation value; (iii) difference between the instant moments on which each of the gyroscope, minimum and maximum, peak values occur; (iv) sum of the positive rotations; and (v) sum of the negative rotations.

From the waist accelerometer are used 6 features: (i) maximum acceleration value; (ii) instant moment where the minimum acceleration measurement occurs; (iii) instant moment where the maximum acceleration measurement occurs; (iv) difference between the instant moments on which each of the accelerometer, minimum and maximum, peak values occur; (v) sum of all the negative acceleration measurements; (vi) and the sum of all the positive acceleration measurements.

It gives a total of 17 features that are fed into the SVM algorithm. These features are important since they provide a good indication about the signal pattern. The design of the implemented SVM approach can be seen in Figure 62.

Since in this characterization there are three possible classes (i.e. normal, ascending or descending stairs), and the SVM models can only classify two at each time, three SVM models (SVM Model 1, SVM Model 2 and SVM Model 3) were created. From the realized tests it was verified that the best results were achieved using on each model a "polynomial" kernel, configured as a 3^{th} order polynomial.

The models were trained with the same data, but with different class labels vectors. In this case there are three vectors. The first vector, which is used by the SVM Model 1, indicates that the ascending stairs steps belongs to the positive class and the others to the negative. The second vector, which is used by the SVM Model 2, indicates that the descending stairs steps are the positive entries and the other the negatives. The third vector, which is used by the SVM Model 3, indicates that the normal terrain steps are the positive classifications and the others the negative. Meaning that each classifier has as positive class, ascending stairs, descending stairs and normal terrain, respectively.

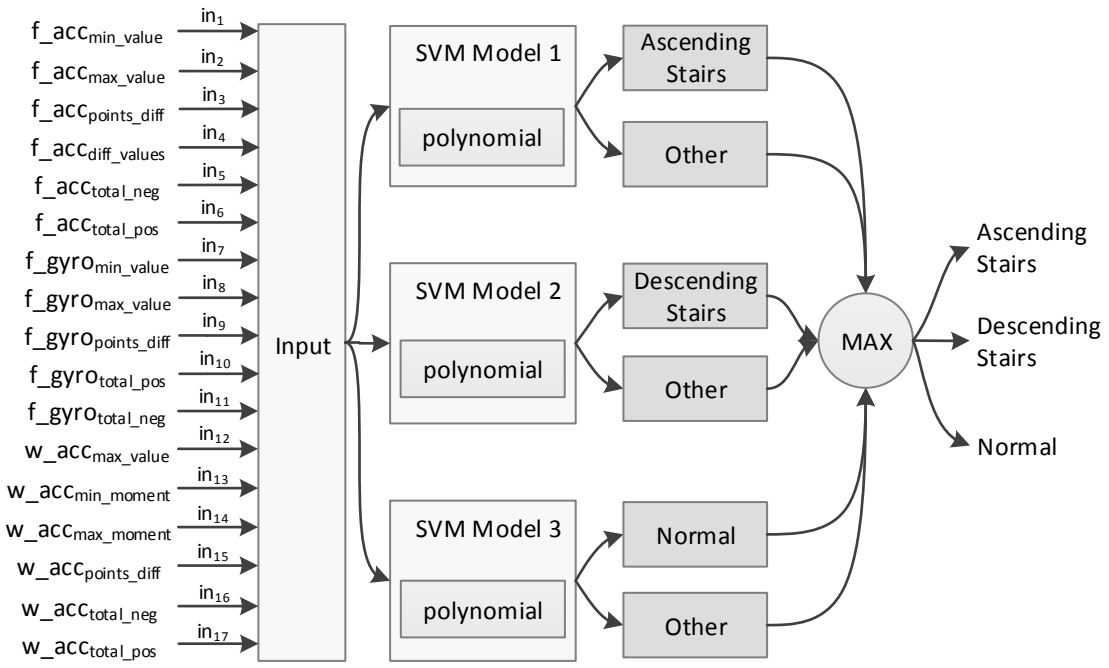


Figure 62: SVM architecture for step terrain characterization

The score of the new observations are then estimated using each classifier. This will create a vector with three scores, one per each classifier. The index of the element with the highest score is the index of the class to which the new observation most likely belong. For example, if the first index has the highest value, then the step is classified as ascending stairs. Thus each new observation is associated with the classifier that gives to it the maximum score.

After the learning phase, a 10-fold cross validation to the model was performed. The SVM Model 1 presented no error, the SVM Model 2 presented an error of 0.8% and the SVM Model 3 presented an error of 2.6%.

6.1.1.2 *Neural Network*

For the neural network algorithm the best results were obtained using 72 inputs (24 inputs per each sensor). Each sensor signal is divided into 6 equal parts, and for each one of these parts the maximum, minimum and mean values were obtained, as well as, the slope. The slope was calculated based on the first and the last measurement of each part. This data gives a total of 24 inputs per each sensor that are fed into the learning algorithm.

The same features that were used as input of the SVM (Section 6.1.1.1) were tested as input of the implemented neural network. However, from the several tests performed,

these features (72 inputs) were the ones that gave the best results on classifying the type of terrain were the step was given.

The division of the signal was made because giving to a classifier a complete signal can be very heavy and confusing to the algorithm to identify the patterns of the signal and therefore estimate the correct label for that pattern. Thus, it is reduced the dimensionality of the problem domain for the purposes of improving the performance of the algorithms and to decrease the computational load.

It was decided to divide the signal in 6 parts, because, during a step, each sensor signal is typically composed by 30 measurements. Thus, in order to have an average of 5 measurements per iteration the signal was divided into 6 equal parts. More parts will divide the signal too much, and less parts will pass insufficient information to the learning algorithm. Thus, the 6 was the number of parts that represented better each one of the signals.

In Figure 63 is represented the design of the implemented neural network that classifies the type of terrain. The neural network receives as input (j) the 72 features previously presented. This input is passed to the *Hidden Layer*, which is composed by 144 neurons. Then, the *Output Layer* returns the final result about the type of terrain where the step was given.

The neural network parameters namely, the number of neurons in the hidden layer, the learning rate and the number of iterations, were tuned by trial and error. The learning rate was defined as 0.01 and the number of iterations as 36.

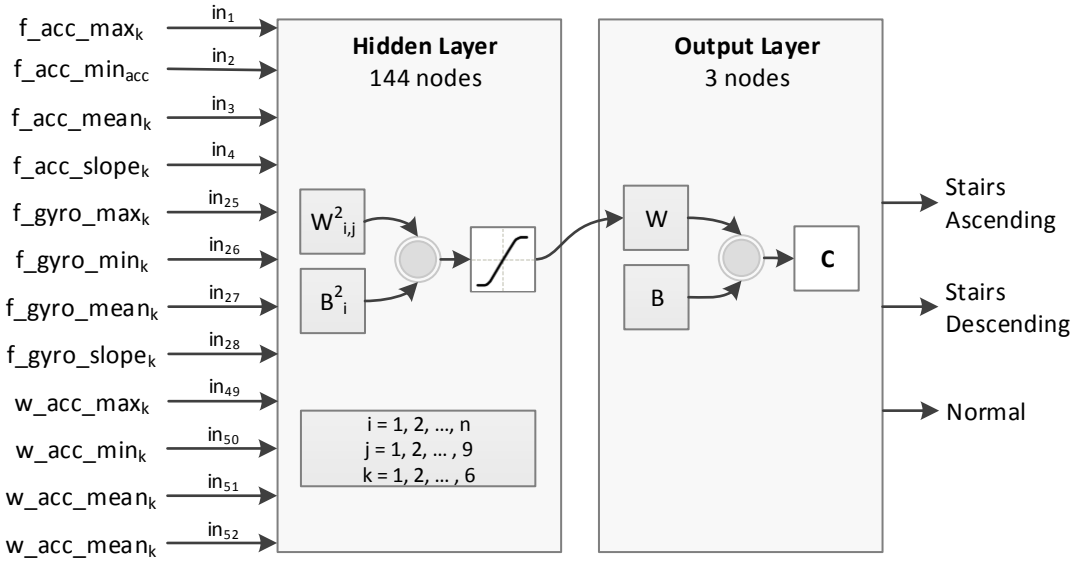


Figure 63: Neural Network architecture for step terrain characterization

The mean squared error of the best validation performance was 7.97×10^{-7} with a gradient of 9.80×10^{-7} at epoch 36.

The error histogram can be seen in Figure 64. From this figure it can be concluded that the error given by the neural network is very low, where more than 98% of the results are very close to zero error. The highest error for an instance, during the network training, was of 1.50×10^{-5}.

6.1.2 *Other Approaches*

After the implementation of the learning approaches, a *Heuristic* (Section 6.1.2.1) and a DTW approach (Section 6.1.2.2) were implemented. Regarding the *Heuristic* it was also implemented one that fuses the data from the two BSU, which will be referred as *Heuristic Fusion* (Section 6.1.2.3).

6.1.2.1 *Heuristic*

The Heuristic algorithm starts by extracting some features from each one of the sensors used to perform this characterization. Then the algorithm uses these features to classify

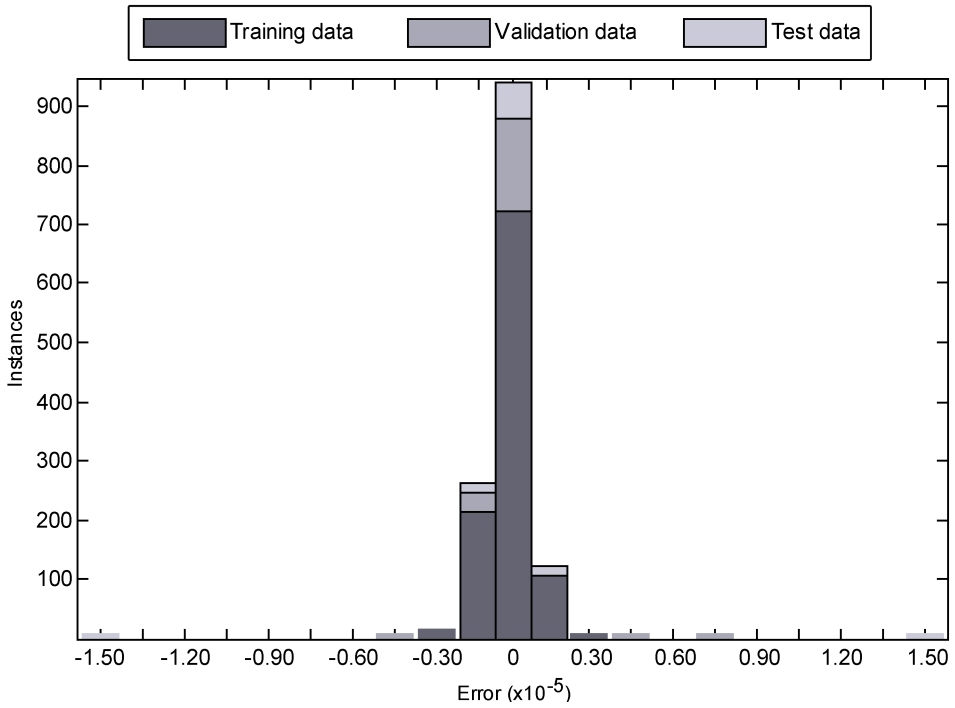


Figure 64: Neural Network training error histogram for step terrain characterization

the step as a normal terrain or stairs (first part). If a step is classified as stairs the algorithm then distinguishes it between being ascending or descending (second part).

The classification is made based on some rules that are applied to each one of the extracted features. The result, which is also the class (*C1* for normal terrain and *C2* for stairs, and *C3* for ascending stairs and *C4* for descending stairs)), of each rule is multiplied with a weight. These weights were given based on the importance that each feature has to make the distinction between each class. The sum of all the results of the features gives the class at which the step belongs. For example, 1 (C1) represents a normal terrain and 2 (C2) a stairs terrain.

The features that were extracted from each sensor are presented in the form of a table, composed by 6 columns:

- Number - identification number for the feature;
- Parameter - feature that was extracted from the sensor signal;
- Condition - condition that is applied to the feature. Different thresholds are applied in order to make the distinction between each class;
- Class - class that will be returned by the feature, based on the result of the condition;
- Weight - weight applied to the result of the feature;
- Note - some notes about what the feature represents and the influence that it has to in making the distinction between each class;

The features extracted from the foot BSU Accelerometer are presented in Table 22, from the foot BSU Gyroscope are presented in Table 23 and, finally, from the waist BSU are presented in Table 24.

For the first part, which makes the distinction between normal and stairs terrain, are used the first three features from the foot BSU accelerometer and gyroscope, and the waist BSU accelerometer. Although features are presented sequentially, only the weights of the foot BSU accelerometer and foot BSU gyroscope complement each other, in other words, the sum of the weights of the features of these two sensors is normalized to give the value of 1. The features extracted from both sensors, of the foot BSU, will represent the final result for the foot BSU. The sum of the waist accelerometer weights is 1, since from the waist BSU it is only used the data from one sensor. Remembering that this algorithm does not perform any fusion of information between the two BSU.

For the second part of the Heuristic algorithm, that makes the distinction between ascending or descending stairs, are used the 4^{th}, 5^{th}, 6^{th}, 7^{th} and 8^{th} feature of the

foot BSU accelerometer. From the foot BSU gyroscope it is used the 4^{th}, 5^{th} and 6^{th} feature, and from the waist BSU accelerometer it is used the 4^{th}, 5^{th} and 6^{th} feature.

Table 22: Features extracted from the foot BSU accelerometer to perform the terrain characterization

#	Parameter	Condition	Class	Weight	Note
1	Minimum acceleration value	< -5	C1	0.20	Acceleration is lower when descending stairs and higher in the other two cases
		$\geqslant -5$	C2		
2	Difference between min and max peak values	$\leqslant 3$	C1	0.20	In normal terrain the positive and negative peak values are more similar than in stairs
		> 3	C2		
3	Difference between min and max peak values moments	$\leqslant 10$	C1	0.20	In ascending stairs and normal terrain these moments are closer, than in descending stairs
		> 10	C2		
4	Maximum acceleration value	> 7	C3	0.10	*y-axis* acceleration is higher when ascending stairs, since foot goes to a upward location
		$\leqslant 7$	C4		
5	Difference between min and max peak values moments	$\leqslant 9$	C3	0.05	Points are closer when ascending stairs
		> 9	C4		
6	Sum of all negative acceleration measurements	> 20	C3	0.20	Lower acceleration (negative) is sensed when ascending stairs.
		$\leqslant 20$	C4		
7	Sum of all positive acceleration measurements	> 25	C3	0.25	Higher acceleration (positive) is sensed when ascending stairs.
		$\leqslant 25$	C4		
8	Difference between max and min acceleration values	> 10	C3	0.05	The difference is higher when ascending stairs
		$\leqslant 10$	C4		

After the extraction of the features, the rules of each feature were applied. The result, which is also the class (*C3* for ascending stairs and *C4* for descending stairs) of each rule is multiplied with the corresponded weight. The sum of all the values gives the class at which the step belongs, where 3 represents ascending stairs and 4 descending stairs. The same conditions about the sum of the weights are applied as in the first part of the algorithm.

After extracting the features and applying the rules, a class is obtained from each one of the rules. Each class represents a number, where *C1* represents the number 1,

Table 23: Features extracted from the foot BSU gyroscope to perform the terrain characterization

#	Parameter	Condition	Class	Weight	Note
1	Minimum rotation value	< -10	C1	0.20	In normal terrain negative rotation is similar to the positive
		$\geqslant -10$	C2		
2	Difference between the min and max peak values moments	< 14	C1	0.10	When descending stairs the difference between both points is higher than in the other two cases
		$\geqslant 14$	C2		
3	Comparison between the sum of the positive (*pos*), and the sum of the negative rotations (*neg*)	$pos \geqslant neg$	C1	0.10	In ascending stairs the positive rotation is higher than in the other two cases, since the foot goes to a higher surface than the original.
		$pos < neg$	C2		
4	Maximum rotation value	> 6	C3	0.15	In ascending stairs positive rotation is higher than when descending stairs, since the foot is going upwards
		$\leqslant 6$	C4		
5	Comparison between the min. rotation value (*min*) and the max rotation (*max*) value	$\lvert max \rvert > \lvert min \rvert$	C3	0.05	In ascending stairs the positive rotation is higher than the negative. In descending stairs it is the opposite.
		$\lvert max \rvert \leqslant \lvert min \rvert$	C4		
6	Sum of all positive rotation measurements	> 35	C3	0.15	The positive rotation is higher when ascending stairs
		$\leqslant 35$	C4		

and so on. Then, for each BSU is made the sum of all the class values multiplied with the respective weight, as represented in Equation 28.

$$R = \sum_{i=1}^{n} class \times weight \tag{28}$$

where R is result of the equation which will be the class, n is the number of features extracted for the BSU, $class$ is the "Resulting Class" value and $weight$ is the weight of each feature.

As example it can be considered the values presented in Table 25. For each extracted feature it contains the value obtained in a step. Also, it presents the *Resulting class*

Table 24: Features extracted from the waist BSU accelerometer to perform the terrain characterization

#	Parameter	Condition	Class	Weight	Note
1	Difference between the min and max peak values instant moments	> 20	C1	0.40	On stairs peaks are very close to each other since a higher force is applied to move the body upward or downward
		$\leqslant 20$	C2		
2	Minimum acceleration instant moment	$\leqslant 15$	C1	0.30	On stairs it occurs later than in the normal terrain
		>15	C2		
3	Comparison between the sum of all positive (*pos*) and the sum of all negative measurements (*neg*)	$pos \geqslant neg$	C1	0.30	In descending stairs exist more negative than positive accelerations. In ascending stairs it is the opposite, and in normal terrain they are similar
		$pos < neg$	C2		
4	Maximum acceleration measurement instant moment	$\geqslant 20$	C3	0.35	In ascending stairs the negative acceleration peak value comes first than the positive, in descending stairs it is the opposite
		<20	C4		
5	Maximum acceleration value	> 0.7	C3	0.30	A higher acceleration value is obtained when ascending stairs
		$\leqslant 0.7$	C4		
6	Comparison between the sum of positive (*pos*) and the sum of negative measurements (*neg*)	$pos > neg$	C3	0.35	In ascending stairs positive acceleration is higher than when descending stairs
		$pos \leqslant neg$	C4		

which is the result given by the rules of each feature. It is also presented the weight that will be applied to each resulting class.

Applying Equation 28 to the data, of the foot BSU, presented in Table 25 it is obtained the data presented in Equation 29.

$$\begin{aligned} R &= (C1 * 0.20) + (C1 * 0.20) + (C1 * 0.20) + (C2 * 0.20) + (C1 * 0.10) + (C1 * 0.10) \\ &= (1 * 0.20) + (1 * 0.20) + (1 * 0.20) + (2 * 0.20) + (1 * 0.10) + (1 * 0.10) \\ &= 1.20 \end{aligned} \tag{29}$$

Table 25: Example of obtained values, during a step, for each feature and the resulting class after applying the specified rules

Sensor	Feature	Value	Weight	Resulting class
Foot accelerometer	1	-6	0.20	C1
	2	3	0.20	C1
	3	12	0.20	C1
Foot gyroscope	1	-10	0.20	C2
	2	10	0.10	C1
	3	$15 > 14$	0.10	C1
Waist accelerometer	1	20	0.40	C2
	2	20	0.30	C2
	3	$20 > 15$	0.30	C1

Rounding the result (R) as an integer it is obtained the value of 1. Thus, the data obtained from the two sensors of the foot BSU is compatible with a step of "C1", meaning that it is a normal terrain.

Making the same reasoning to the waist BSU, it is obtained the data presented in Equation 30. Since the result of this equation is rounded as a integer, it represents that the resulting class is "C2", meaning that the step was taken in a stairs terrain.

$$\begin{aligned} R &= (C2 * 0.40) + (C2 * 0.30) + (C1 * 0.30) \\ &= (2 * 0.40) + (2 * 0.30) + (1 * 0.30) \\ &= 1.70 \end{aligned} \tag{30}$$

For this example each BSU gave a different result for the same step. One way to improve this result is to fuse the data of the two BSUs, in order to achieve a consensus about the step.

6.1.2.2 *DTW*

An implementation based on the DTW algorithm was used to classify the type of terrain. This implementation works as follows, when a step is detected, the foot accelerometer and gyroscope, and the waist accelerometer signals, are compared, using the DTW algorithm, to a dataset of signals previously obtained for that person.

The dataset, for each sensor, was composed by the data of 108 steps. The dataset can be decomposed into three subsets (one per each terrain type), which were composed by 36 signals each. Since the sensors have an identical pattern through time, this amount of data proved to be sufficient to achieve good results. From the realized tests, less data gives worst results, and more data does not affect the accuracy of this algorithm.

When a step is detected it is calculated the distance between the signal of the detected step with each signal of each subset of signals for a specific type of terrain. The subset that has the lowest mean distance is the one that is closer to the detected step. Thus, the class that the subset represents is returned.

6.1.2.3 *Heuristic Fusion*

The two BSUs placed in the person's body enables the combination of the information in order to minimize the complexity of the algorithms and maximize the accuracy and robustness of the navigation solution.

Typically there are three types of fusion: data fusion, feature fusion and decision fusion. The implemented algorithm uses a feature fusion, where each source of data (waist BSU and foot BSU) was combined to achieve a final consensus about the step terrain characterization. In this method the rules, presented in the Heuristic algorithm and applied to both BSU, were combined. Thus, new weights were used since now it will be used the data of both BSU. The new weights are presented in Table 26.

6.1.3 *Evaluation*

The implemented algorithms that characterize the type of terrain where a step was given, were evaluated using a dataset of 800 steps performed by two pedestrians (400 steps for each pedestrian).

The test scenario is the "*D*" path represented in Figure 91, which involves a complex path with a set of straight walks and a set of stairs. The set of stairs was ascended two times and descend one time. Meaning that the pedestrian ascended the stairs, then descend it and finally ascended it again. A total of 200 steps, each time, were performed in this scenario which gives a total walking distance of 70 meters. Two runs in this scenario, for each pedestrian, were performed.

The results obtained for this scenario can be seen in Table 27. This table presents for each algorithm, the categorization accuracy (in percentage) and the execution time

Table 26: Heuristic fusion weights applied to each feature retrieved from the foot accelerometer and gyroscope, and waist accelerometer for step terrain characterization

Sensor	Feature	Weight
Foot accelerometer	1	0.125
	2	0.15
	3	0.125
	4	0.05
	5	0.05
	6	0.15
	7	0.20
	8	0.05
Foot gyroscope	1	0.10
	2	0.05
	3	0.05
	4	0.11
	5	0.03
	6	0.11
Waist accelerometer	1	0.15
	2	0.125
	3	0.125
	4	0.09
	5	0.07
	6	0.09

(in milliseconds). For all the algorithms are presented the results obtained, in separate for each BSU and for the combination of the data of both BSU. This permits to identify which one of the BSU have better accuracy in each characterization type.

Considering the obtained results it can be concluded that the ascending stairs class is the easiest to classify. All the algorithms, except the DTW approach, presented the best results when classifying this class. The normal terrain class is sometimes confused with the descending stairs class, so it is with this misclassification that most errors occur.

Regarding the BSUs, the foot BSU gives more accurate data, since the foot is closer to the ground. The waist BSU can give a good indication about the vertical movement of the body. However, it obtains similar data when descending stairs and in normal terrain. Thus, it presents worst results in these classifications.

Table 27: Accuracy results for the developed algorithms that characterize the step terrain

Method	Ascending		Descending		Normal		Execution time
	Waist BSU	Foot BSU	Waist BSU	Foot BSU	Waist BSU	Foot BSU	
Heuristic	98%	95%	97%	97%	87%	89%	3 ms
DTW	59%	93%	98%	100%	69%	87%	400 ms
SVM	97.5%	99.4%	94.2%	99.4%	85.1%	94.9%	1 ms
Neural Network	98.3%	100%	94.1%	100%	87.2%	94.9%	1 ms
Heuristic Fusion	99%		97%		91%		10 ms
SVM Fusion	99.4%		99.5%		96.2%		2 ms
Neural Network Fusion	100%		100%		98.7%		2 ms

Interpreting the results obtained for each algorithm using each BSU in separate, the Heuristic approach achieves similar results as the learning algorithms, which presented the best results. The DTW approach revealed to have inferior results when compared to the others. This happens because the signals are pretty much similar, varying only the intensity of the peaks. Also, it is the one that takes longer to run.

Analysing the results obtained for each algorithm when considering the fusion of both BSU, the learning algorithms presented the best results. The best of the two was the Neural Network, achieving a mean accuracy of 99.4%, having 100% of accuracy on predicting the ascending and descending stairs classes. The Heuristic Fusion approach had better results than the single Heuristic approach, but it was worse than the learning algorithms. Also, it highly depends on thresholds, which can be different for each different pedestrian (i.e. the maximum acceleration threshold can be different for two pedestrians with different physical characteristics).

Analysing the obtained results, it can be concluded that through the sensors complementarity the type of terrain was categorized with higher accuracy. Also, it can be concluded that the learning of the gait parameters enables a better characterization of a step. From our tests it was identified that a learned dataset 5 times smaller, than the used one, is sufficient to achieve similar results. Making the learning procedure simpler and faster to a pedestrian perform before using PLASYS.

Concluding, the evaluation results show that both BSU give similar results on detecting each type of terrain, but with their integration better results can be achieved.

6.2 STEP DIRECTION CLASS

The second characterization performed to a step is about the direction that it can take. There are two possibilities, a forward step, which is the most natural to perform to a human, or a backward step. The learning algorithms that performs this characterization are presented in Section 6.2.1 and the other implemented approaches are presented in Section 6.2.2.

In Figure 65 is represented an acceleration signal obtained by the foot accelerometer, for a forward (Figure 65a) and a backward (Figure 65b) step. For each direction is represented a simulated acceleration signal, which is the acceleration expected to happen when a forward or backward step is given. As can be seen the accelerometer does not capture the accelerations in a perfect form, but it contains the information needed (acceleration pattern) to be used to infer the step direction.

Although the pattern of the acceleration can be used to classify a step, sometimes the accelerometer produce a signal that does not follow any pattern, which seems to be useless to correctly classify a step. In Figure 66 is represented an acceleration signal,

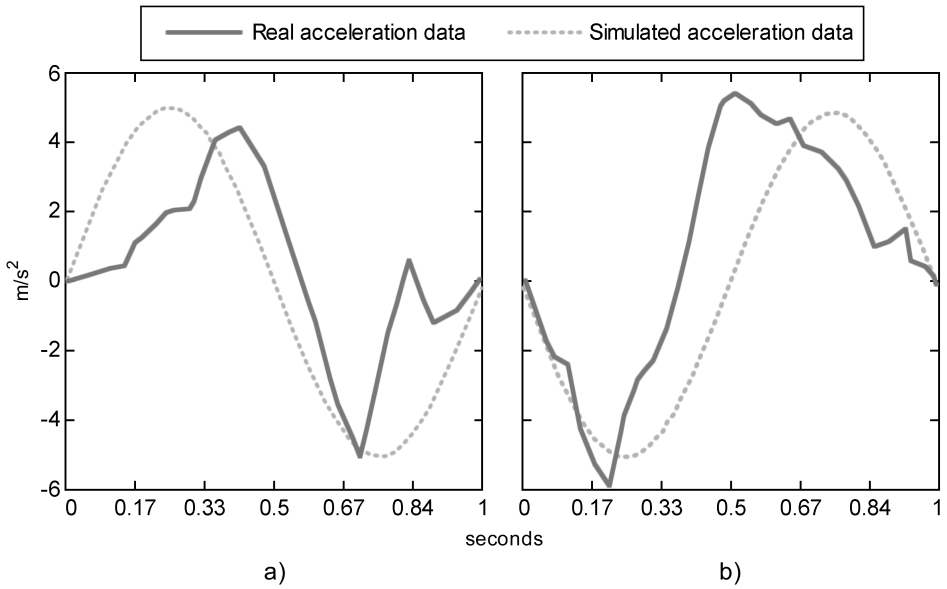


Figure 65: Foot accelerometer (*x-axis*) data for each step direction characterization: a) Acceleration signal pattern for a forward step; b) Acceleration signal pattern for a backward step

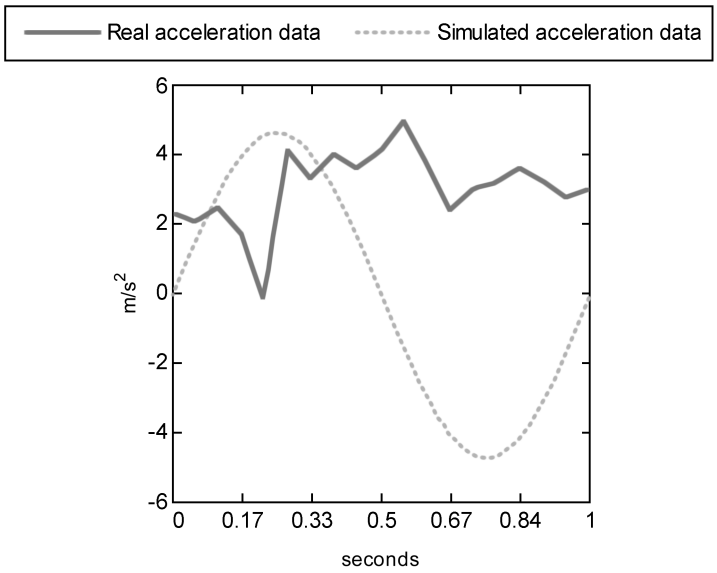


Figure 66: Erroneous accelerometer signal obtained during a forward step

for a forward step, that does not follow any pattern. This acceleration signal cannot be used to correctly classify a step.

6.2.1 *Learning Algorithms*

During this research, by analysing the datasets collected from all the walks, it was verified that step direction can be characterized by combining the data obtained from two sensors placed in different BSU, the foot accelerometer and the waist gyroscope. In the case of the accelerometer, the one placed in the pedestrian's foot gives better results than the one on the waist. However, in the case of the gyroscope, the best results can be achieved with the one placed on the waist, since it give us the pelvic rotation, which combined with the accelerometer data is important to determine the direction of a step.

To classify the step direction 25 features where extracted from the sensors measurements, where 24 are obtained from the accelerometer data and 1 from the gyroscope data. To extract the features from the accelerometer signal, it was divided into 6 equal parts. For each one of these parts the maximum, minimum and mean values were obtained, then it was calculated the slope. The slope was calculated based on the first and last measurement of each part. This gives a total of 24 inputs that are fed into the learning algorithm. The explanation for this division is the same as the one presented in Section 6.1.1.2.

The other input is obtained from the gyroscope signal, which represents the motion of the waist. If the pelvic as a positive rotation, then the value 1 is assigned to the input, if it is a negative rotation the value 0 is assigned to the input. For example, for a forward left step the rotation will be positive and for a backward left step the rotation will be negative, for the right foot it is the opposite.

To train the learning algorithms the following number of samples were used: 450 samples (190 forward and 260 backward) for training, 100 samples for validation (40 forward and 60 backward) and 100 samples (50 forward and 50 backward) for testing. A 10-fold cross-validation using these datasets was also performed.

6.2.1.1 *SVM*

The design of the implemented SVM approach can be seen in Figure 67. It receives as input the 25 features previously presented. This input is passed to the *Hidden Nodes* that estimate the best separating hyperplane between the two classes, which maximizes the margin between the two classes. This division is performed using a "linear" kernel. Then, the *Output Layer* returns the final result about the step direction. A degree of confidence for the two possible results is given by this algorithm, which will be important to the next phase of the PINS.

The score of the algorithm during the cross-validation was 100% for both classes, forward or backward.

6.2.1.2 *Neural Network*

The design of the implemented neural network can be seen in Figure 68. The neural network receives as input (j) the 25 features previously presented. This input is passed

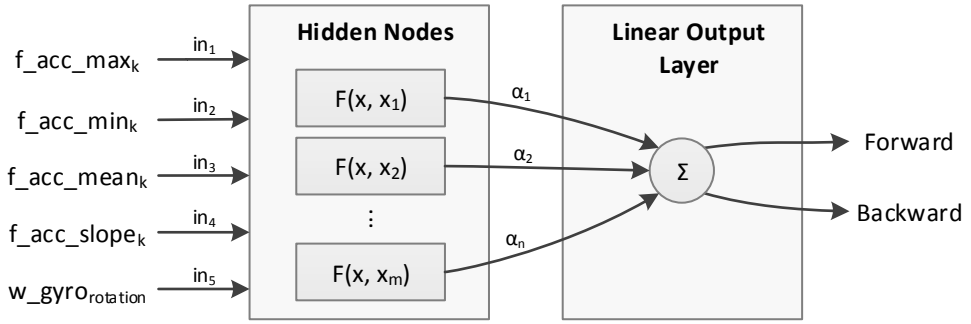


Figure 67: SVM architecture for step direction characterization

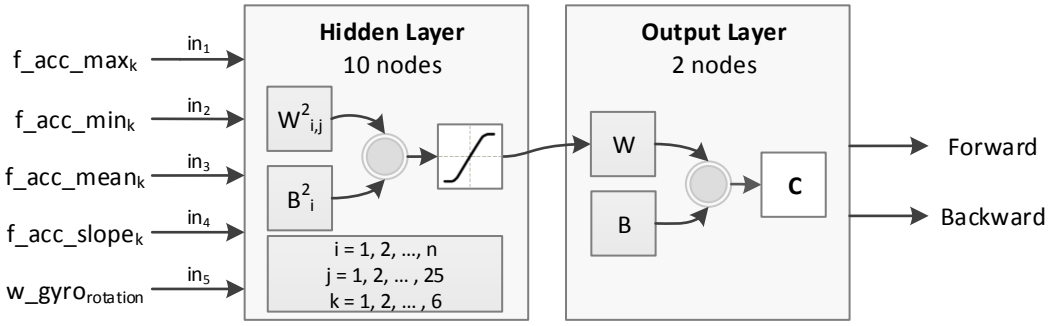


Figure 68: Neural Network architecture for step direction characterization

to the *Hidden Layer*, which is composed by 10 neurons. Then, the *Output Layer* returns the final result about the step direction.

The neural network parameters namely the number of the neurons in the hidden layer, the learning rate and the number of iterations, were tuned by trial and error. The learning rate was defined as 0.01 and the number of iterations was defined as 31.

The mean squared error of the best validation performance is 2.08×10^{-8} with a gradient of 9.58×10^{-7} at epoch 35. The error histogram can be seen in Figure 69. From this figure it can be seen that the error is very low when training the network, but even lower when validating and testing the established neural network. Also, more than 90% of the results are very close to zero error. The highest error for an instance, during the network training, was of 0.15.

6.2.2 *Other Approaches*

As in the previous characterization (Section 6.1), to evaluate the performance of the learning algorithm, it was implemented a *Heuristic* (Section 6.2.2.1), a DTW approach (Section 6.2.2.2) and a *Heuristic Fusion* (Section 6.2.2.3) approach.

6.2.2.1 *Heuristic*

Typically a PINS detects a step by using the accelerometer data and by analysing the forward and upward accelerations during the walking path. But, this sensor signal can also be used to obtain some characteristics to characterize the walking.

To classify the direction of the step, in both BSU, Equation 31 is used.

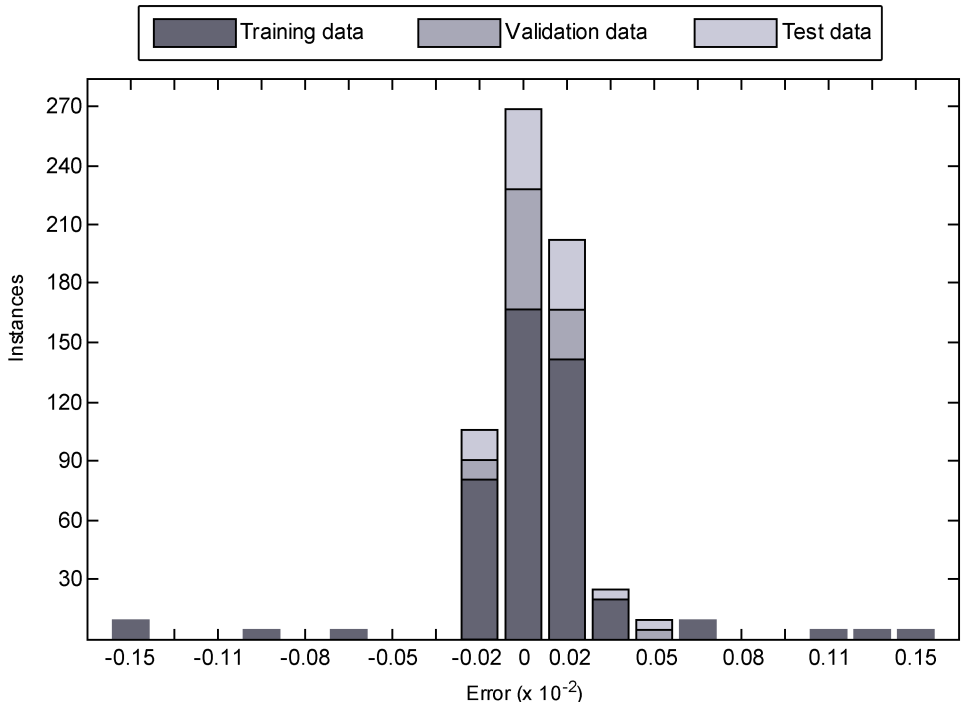


Figure 69: Neural Network training error histogram for step direction characterization

$$\sum_{i=1}^{\frac{n}{2}} acc_i > \sum_{i=\frac{n}{2}}^{n} acc_i \tag{31}$$

where n represents the number of acceleration measurements detected in a step, acc is the acceleration values sensed on the *x-axis*, in the case of the foot accelerometer, and on the *z-axis*, in the case of the waist accelerometer.

This formula sums the accelerations of the first half of the signal and compares it with the sum of the accelerations of the second half. If the first is positive or higher than the second it is a forward step, if not it is a backward step.

6.2.2.2 *DTW*

This implementation is similar to the DTW approach presented in Section 6.1.2.2. In the tests a dataset, for each accelerometer, of 24 (12 forward and 12 backward) signals previously stored and categorized was used. Since the acceleration has an identical pattern through time, this amount of data proved to be sufficient to achieve good results.

In Figure 70a are shown two signals for a forward step, one is one of the stored foot acceleration signals and the other is an acceleration signal for a step that is not on the

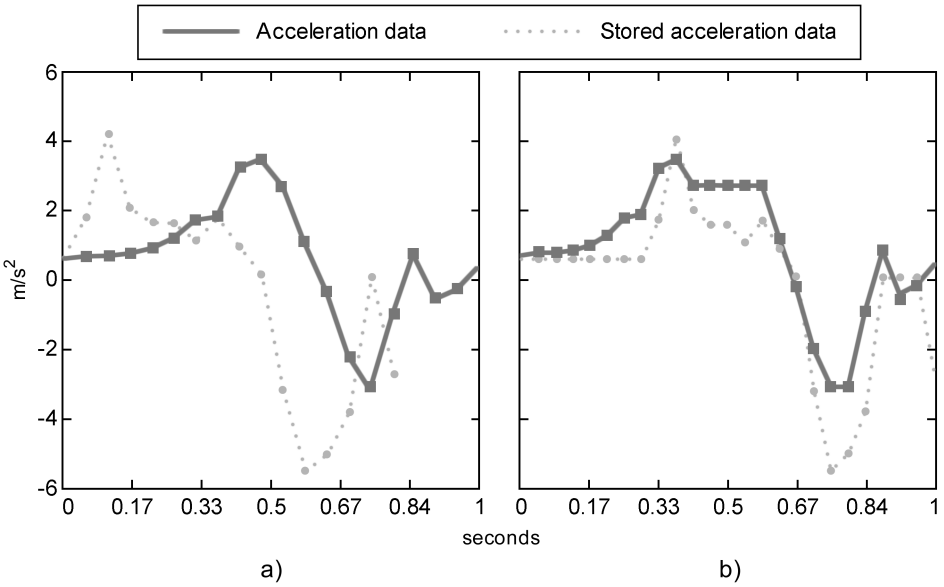


Figure 70: Acceleration signals comparison: a) original acceleration signals; b) warped acceleration signals

stored dataset. In Figure 70b are shown the two signals warped, according to what the DTW algorithm performs. This example gave a distance of 139 with a total of 25 samples. For example, if one of the signals was for a forward step and the other to a backward step the distance will be more than 400 with a total of 40 samples.

6.2.2.3 *Heuristic Fusion*

This algorithm is based on decision fusion, where for each source of data, foot accelerometer and waist accelerometer, it is calculated the probability of the predicted result. This probability ($stepprobability$) is calculated, according to Equation 32, which is based on the fact that a positive acceleration (acc) must be followed by a negative acceleration of the same magnitude (in a forward step), or vice-versa (in a backward step). The value 100 represents the maximum probability, at which will be subtracted the second part of the equation. The value 20 is a penalty factor that is applied to the difference in the acceleration values. The higher this difference is, the worse will be the probability. The $stepprobability$ is then normalized in order to give a value between 0 and 1.

$$stepprobability = 100 - (abs((max(acc) + min(acc))) \times 20) \tag{32}$$

If the acceleration signal does not follow this pattern then a low probability is given to it. For the acceleration signal example shown in Figure 65, the probability that it is

a characterizable step is 100% and for the example shown in Figure 66 the probability is only 20%.

Since each source of data has its degree of accuracy, after calculating this probability different weights are given to each source of data. The foot BSU has a weight of 0.6, since it is the most reliable source of data, and the waist BSU has a weight of 0.4.

6.2.3 *Evaluation*

The step direction characterization algorithms were evaluated using a dataset of 240 steps performed by two pedestrians (120 steps for each pedestrian).

The test scenario is the "*B*" path represented in Figure 57, which involves a straight walk with two curves, in the middle of the path, one to the left and the other to the right. A total of 60 steps (40 forward and 20 backward), each time, were performed in this scenario which gives a total walking distance of 30 meters. Two runs in this scenario, for each pedestrian, were performed.

The results obtained for this scenario can be seen in Table 28. This table presents for each algorithm, the categorization accuracy (in percentage) and the execution time (in milliseconds). For all the algorithms are presented the results obtained, in separate for each BSU and for the combination of the data of both BSU.

Table 28: Accuracy results for the developed algorithms that characterize the step direction

Method	**Forward**		**Backward**		**Execution time**
	Waist BSU	**Foot BSU**	**Waist BSU**	**Foot BSU**	
Heuristic	82.1%	97.4%	74.4%	94.9%	1 ms
DTW	84.6%	100%	79.5%	97.4%	100 ms
SVM	98.6%	100%	96.7%	100%	1 ms
Neural Network	99.4%	100%	95.5%	100%	1 ms
Heuristic Fusion	100%		98%		2 ms
SVM Fusion	100%		100%		1 ms
Neural Network Fusion	100%		100%		1 ms

From the obtained results, it can be concluded that the data obtained from the foot BSU sensors is better, than the data obtained from the waist BSU sensors, to perform this characterization. This mainly happens because when the user is moving the foot is a more stable platform than the waist. A lot of unwanted accelerations are sensed by the waist, which leads to a poor characterization of the step, but there are some features that can be retrieved to help other sources to properly characterize the step.

Regarding the step direction characterization the backward one is more difficult to classify than the forward one. Mainly because for a human a forward step is a more natural movement to perform than a backward one.

Comparing DTW approach with all the other approaches, it can be seen that the DTW has a low error rate, but it has a longer execution time. In order to maintain a lower execution time and an accuracy similar to the DTW approach, the information of both BSUs was fused. Through the sensors complementarity the step was categorized with similar accuracy, but with an execution time 50 times smaller. This is an important help in order to improve the pedestrian displacement estimation. Using BSUs in different locations on pedestrian body, waist and foot, was also important to have these results.

The step direction is a simple characterization to be performed to a step, so the results were the expected. The learning algorithms proved to have the best results with an accuracy of 100%. After the learning phase, the execution time is very fast.

For this characterization it was not detected any difference in behaviour between the two learning algorithms.

6.3 STEP LENGTH CLASS

The third, and final, characterization performed to a step is about the length class. There are three possibilities, a short, a normal or a long step. These intervals must be defined from a set of exercises for a pedestrian in specific. Based on the collected data, and on the average of the collected steps, it was considered that short steps are the ones with a maximum distance of 30cm, the normal steps size ranges between 30cm and 45cm, and the long steps have a size longer than 45cm.

As in the other classifications, the learning algorithms are presented in Section 6.3.1 and the other implemented approaches are presented in Section 6.3.2.

In Figures 71, 72 and 73 are represented the signals obtained from the sensors, for each class, that will be used to perform this characterization. The sensors are: the

foot accelerometer (Figure 71) and force sensors (Figure 72), and the waist gyroscope (Figure 73).

The *x-axis* of the foot accelerometer measures the acceleration that is sensed in the horizontal movement of the foot. The quantification of this acceleration is important, because it is correlated with the displacement performed. As can be seen in Figure 71 as the duration and the peak of the acceleration is higher, the longer is the step.

The force sensor data gives a good indication about the amount of time that the foot is not in contact with the ground and, about the force intensity that is made when touching the ground, as well as, when lifting the foot from the ground. Analysing Figure 72 it can be seen that a higher force indicates that probably the step will be longer. The amount of time that the foot is in the air, can be correlated with the acceleration. A

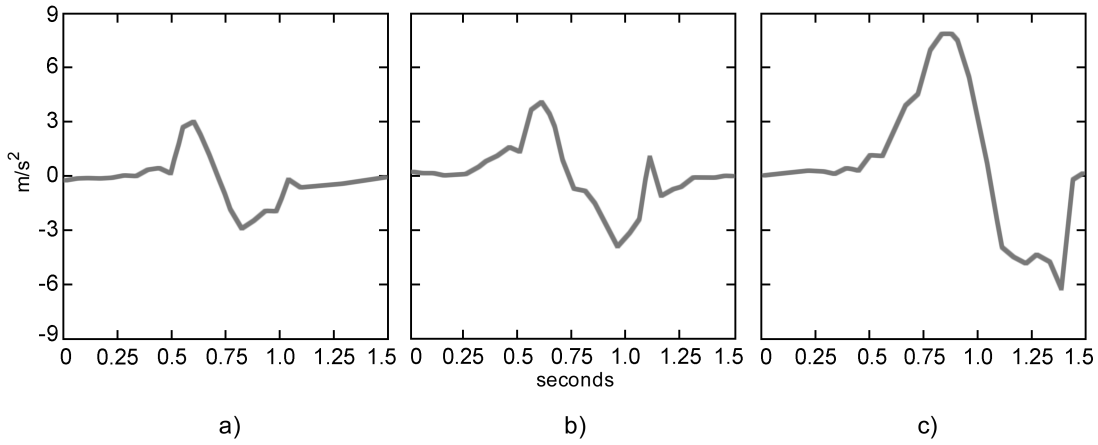


Figure 71: Foot accelerometer (*x-axis*) data for each step length characterization: a) Acceleration signal pattern for a short step; b) Acceleration signal pattern for a normal step; c) Acceleration signal pattern for a long step

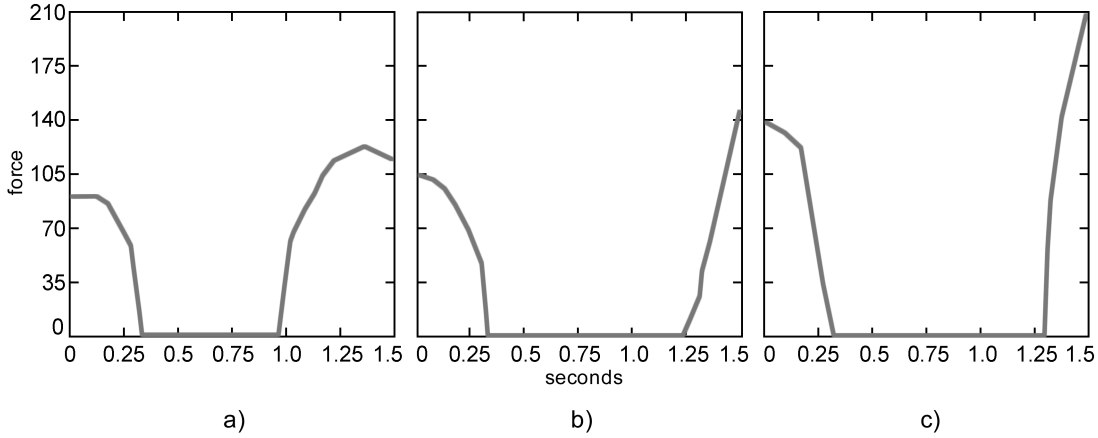


Figure 72: Force sensor data for each step length characterization: a) Force signal pattern for a short step; b) Force signal pattern for a normal step; c) Force signal pattern for a long step

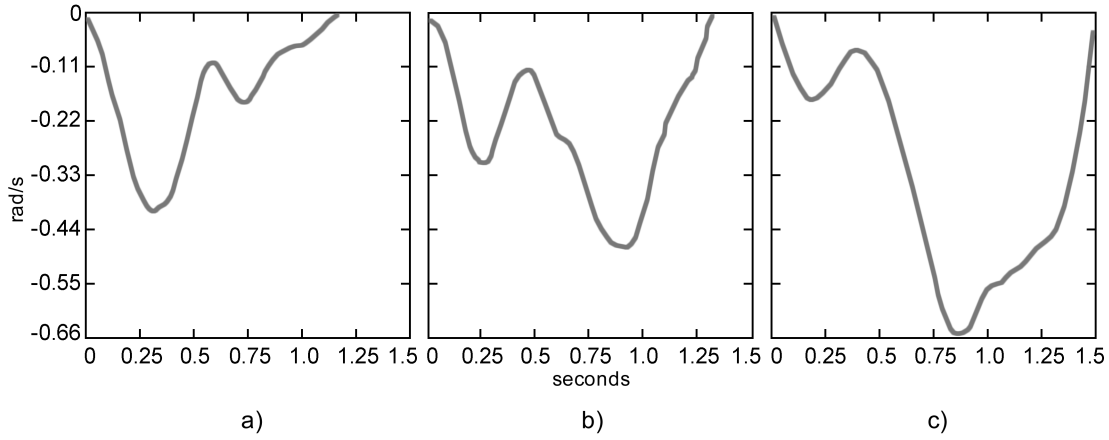


Figure 73: Waist gyroscope data (*x-axis*) for each step length characterization: a) Gyroscope signal pattern for a short step; b) Gyroscope signal pattern for a normal step; c) Gyroscope signal pattern for a long step

high acceleration value combined with a long duration of the foot in the air, indicates that a longer step was made.

The *x-axis* of the gyroscope data gives reliable information about the rotation that was performed by the pelvic, where, as can be seen in Figure 73, a higher rotation corresponds to a longer step. However, in corners this rotation can be higher for the same type of step. Thus, the combination of this data with the data given by the foot BSU sensors is important to achieve better results. Many errors can occur by using the gyroscope data by itself.

After visualizing each signal it can be established that this data is suitable to be used in order to differentiate each possible characterization class.

6.3.1 *Learning Algorithms*

The implemented learning algorithms have as input the foot force sensor and accelerometer data, and the waist gyroscope data. A total of 29 features are fed into the learning algorithms to classify the step length, where 24 are retrieved from the foot accelerometer, 3 from the foot force sensor and 2 from the waist gyroscope.

The foot accelerometer signal was divided into 6 equal parts, as shown in the previous implemented neural networks. This gives a total of 24 inputs that are fed into the learning algorithm. The explanation for this division is the same as the one presented in Section 6.1.1.2.

The next 3 features are obtained from the force sensor signal. The first one is the number of measurements that exist until the maximum force value occur (foot touches the ground). A stronger impact means that the step was longer. The other feature is the force applied when the foot lifts up from the ground. This gives information about the impulse that was performed in the leg in order to do some horizontal movement. Typically, when the impulse is higher the step is longer. The last feature retrieved from the force sensor is the number of measurements with value of zero, which corresponds to the amount of time that the foot is in the air.

From the gyroscope signal two features are extracted. The first one is the amplitude of the signal, which is correlated to the size of a step. Typically, a higher rotation means that the step is longer. The second feature is the length of the signal, which correlated with the amplitude, gives important information about the size of the step.

From the several tests performed, these features were the ones that had the best results in classifying the possible length of a step.

The learning algorithms were trained with a total of 855 samples, 435 samples of a short step, 225 samples of a normal step and 195 samples of a long step. To validate the algorithms a total of 171 samples were used (87 short, 45 normal and 39 long). To test the algorithms a total of 114 samples were used (58 short, 30 normal and 26 long). As in the other characterizations a 10-fold cross-validation using these datasets was also performed.

6.3.1.1 *SVM*

The design of the implemented SVM approach can be seen in Figure 74. This approach receives as input the 29 features previously presented.

In this characterization three SVM models (SVM Model 1, SVM Model 2 and SVM Model 3) were created. After some testing, it was verified that the best results were achieved with the following configuration for each model:

- SVM Model 1 was configured to classify the short steps using a "rbf" (radial basis function or Gaussian) kernel configured with an automatic scale;
- SVM Model 2 was configured to classify the normal steps with a "polynomial" kernel, configured as a 2^{nd} order polynomial;
- SVM Model 3 was configured to classify the long steps using a "linear" kernel.

The models were trained with the same data, but with different class labels vectors. In this case there are three vectors. The first vector, which is used by the SVM Model 1,

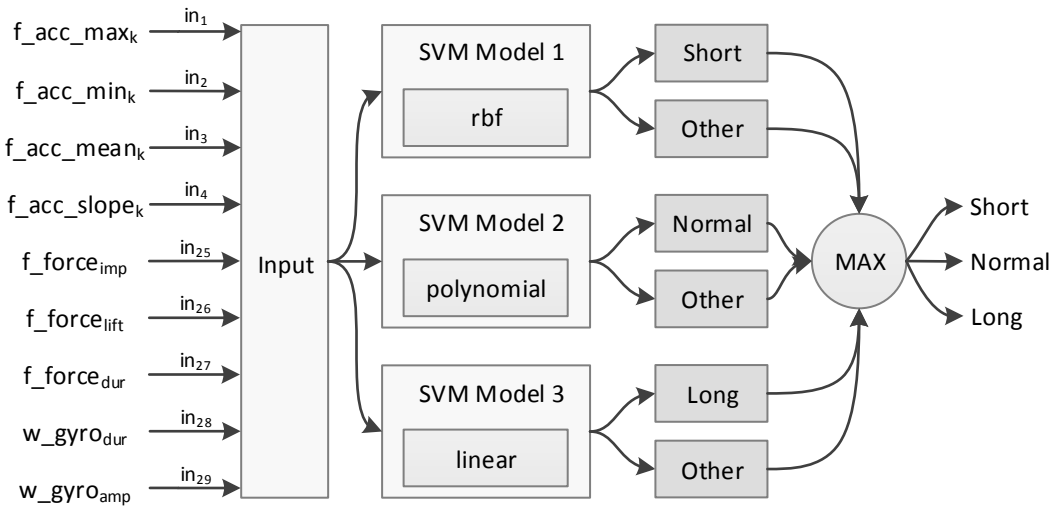


Figure 74: SVM architecture for step length characterization

indicates that the short steps belongs to the positive class and the others to the negative. The second vector, which is used by the SVM Model 2, indicates that the normal steps are the positive entries and the others the negatives. The third vector, which is used by the SVM Model 3, indicates that the long steps are the positive classifications and the others the negatives. Meaning that each classifier has as positive class: short, normal and long, respectively.

The score of the new observations are then estimated using each classifier. This will create a vector with three scores, one per each classifier. The index of the element with the highest score is the index of the class to which the new observation most likely belong. For example, if the first index has the highest value, then the step is characterized as short. Thus, each new observation is associated with the classifier that gives to it the maximum score.

After the learning phase, a 10-fold cross validation to the model was performed. The SVM Model 1 had no error, the SVM Model 2 presented an error of 7% and the SVM Model 3 presented an error of 0.8%.

6.3.1.2 *Neural Network*

The implemented neural network to classify the step length is shown in Figure 75. The neural network receives as input (j) the 29 features previously presented. This input is passed to the *Hidden Layer*, which is composed by 60 neurons. Then, the *Output Layer* returns the final result about the step length.

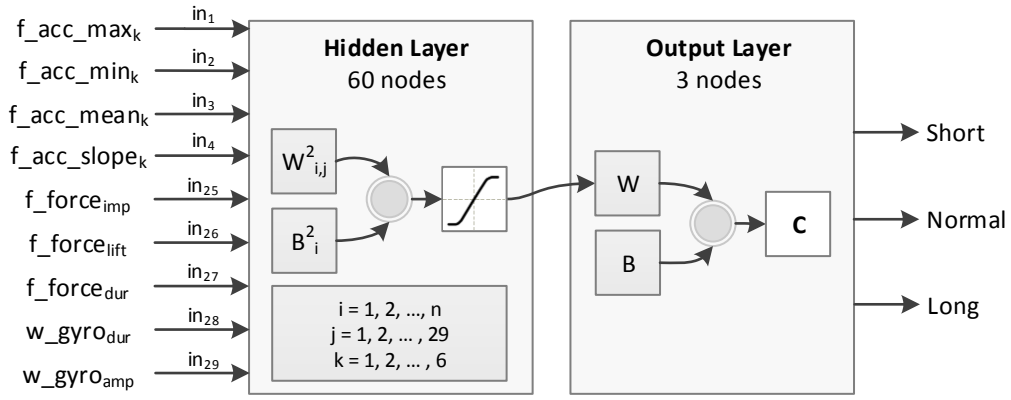


Figure 75: Neural Network architecture for step length characterization

As in the other characterizations the learning rate was defined as 0.01, and the number of iterations as 25.

The mean squared error of the best validation performance was 7.55×10^{-5} with a gradient of 4.59×10^{-4} at epoch 25.

The error histogram can be seen in Figure 76. From this figure it can be concluded that the error given by the neural network is low, where more than 95% of the results are very close to zero error. The highest error for an instance, during the network training, was of 0.20.

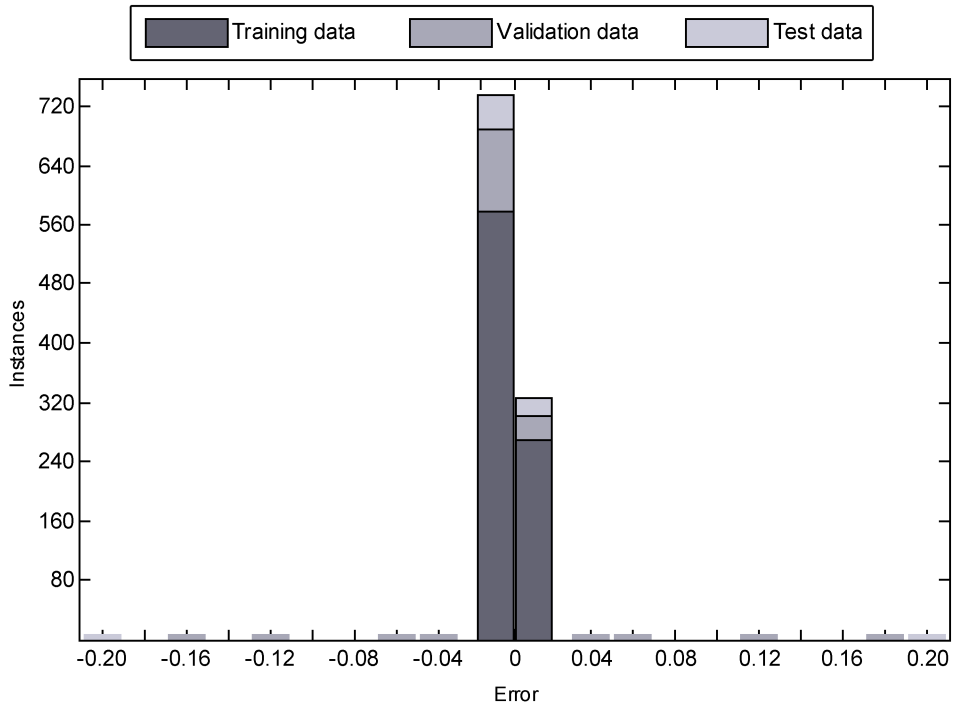


Figure 76: Neural Network training error histogram for step length characterization

6.3.2 *Other Approaches*

For the last implemented characterization made to a step, the *Heuristic* approach is presented in Section 6.3.2.1 and the *Heuristic Fusion* in Section 6.3.2.3. The DTW approach is presented in Section 6.3.2.2.

6.3.2.1 *Heuristic*

Unlike direction characterization, in the step length characterization more than one sensor, in the foot BSU, are used to generate a consensus about the final result. In this heuristic approach, for the foot BSU it was used the accelerometer and the force sensor data, and for the waist BSU was only used the gyroscope data, which gives information about the pelvic rotation. This rotation is important to estimate the step length, since a longer step has a wider rotation.

To perform this characterization it were extracted four features from the foot force sensor, three features from the foot accelerometer and three features from the waist gyroscope, which are presented in Tables 29, 30 and 31 respectively.

Table 29: Features extracted from the foot BSU force sensor to perform the length characterization

#	Parameter	Condition	Class	Weight	Note
1	Number of measurements with value of zero (*number_of_zeros*)	< 17	C1	0.10	Corresponds to the total time that the foot is not in contact with the ground
		$\geqslant 17$ & < 21	C2		
		$\geqslant 21$	C3		
2	Number of measurements until max force when foot touches the ground	$\geqslant 7$	C1	0.07	Information about the intensity of the force that was applied when the foot touches the ground
		< 7 & $\geqslant 4$	C2		
		< 4	C3		
3	Average force values at 4^{th} measurement after and before the max value when the foot touches the ground	< 85	C1	0.10	It is used to verify if it is a short, or a normal/long step
		$\geqslant 85$	C2.5		
4	Force value at the 2^{nd} measurement after the last zero force	< 90	C1	0.10	It is used to verify if it is a short, or a normal/long step
		$\geqslant 90$	C2.5		

Table 30: Features extracted from the foot BSU accelerometer to perform the length characterization

#	Parameter	Condition	Class	Weight	Note
1	Difference between the max and min acceleration peak values (*acc_diff*)	$\leqslant 6.5$	C1	0.05	As the difference is higher, the longer is the step
		> 6.5 & $\leqslant 9$	C2		
		> 9	C3		
2	Difference between the min and max peak values moments	$\leqslant 9$	C1	0.50	As the difference is higher, the longer is the step
		> 9 & < 12	C2		
		$\geqslant 12$	C3		
3	Metric which is the result of the multiplication of the *acc_diff* with the *number_of_zeros*	< 135	C1	0.08	As the metric is higher, the longer is the step
		$\geqslant 135$ & < 190	C2		
		$\geqslant 190$	C3		

After the extraction of these features the rules specified in each feature are applied. The result, which is also the class (*C1* for a short, *C2* for a normal and *C3* for a long), of each rule is multiplied by the corresponding weight. The sum of all the results gives the class at which the step belongs, where 1 represents a short, 2 a normal and 3 a long step. The rules that have as result the value *C2.5*, means that there are not a pattern that clearly distinguishes a step between normal and long classes, so an intermediate value is returned.

Although features are presented sequentially only the weights of the force and accelerometer complement each other, in other words, the sum of the weights of the features of these two sensors is 1. This represents the final result for the foot BSU. The sum of the gyroscope weights is also 1, because it represents the final result for the waist BSU. Recalling that this algorithm does not perform any fusion of information between the two BSU.

After extracting the features and applying the correspondent rules a class is obtained for each feature. Then it is multiplied with the correspondent weight in order to give the class in which the step belongs.

6.3.2.2 *DTW*

As in the other classifications, the DTW based algorithm uses a previously stored dataset to characterize the step according to its length. In the tests, a dataset, for each sensor, of 90 (30 for each characterization type) signals previously stored and categorized was used.

Table 31: Features extracted from the waist BSU gyroscope to perform the length characterization

#	Parameter	Condition	Class	Weight	Note
1	Signal amplitude	$\leqslant 0.45$	C1	0.40	Quantifies the pelvic rotation, where a higher rotation corresponds to a longer step
		> 0.45 & $\leqslant 0.6$	C2		
		> 0.60	C3		
2	Difference between the number of measurements of the left part (*left*) and the right part (*right*) of the signal	*left* > *right*	C1	0.30	The left part are the measurements from the beginning of the signal until the max peak value. The right part are the measurements from the max peak value until the end of the signal
		left $\leqslant$ *right*	C2.5		
3	Number of measurements that exist in the left part of the signal	$\leqslant 75$	C1	0.30	The lower is this number the shorter is the step, as can be seen in Figure 73
		> 75 & $\leqslant 110$	C2		
		> 110	C3		

When a step is detected it is calculated the distance between the signal of the detected step with each signal of each subset of signals for a specific length class. The subset that has the lowest mean distance is the one that is closer to the detected step. Thus, this algorithms returns the class of that subset.

6.3.2.3 *Heuristic Fusion*

Similarly to the algorithm presented in Section 6.1.2.3, this algorithm uses a feature fusion, where each source of data was combined to achieve a final consensus about the step length characterization. In this method the rules, applied to both BSU and presented in the previous method, were combined, and new weights were used. These weights are presented in Table 32.

6.3.3 *Evaluation*

The step length characterization algorithms were evaluated using a dataset of 300 steps performed by two pedestrians (150 steps for each pedestrian).

Table 32: Heuristic fusion weights applied to each feature retrieved from the foot accelerometer and force sensor, and waist gyroscope for step length characterization

Sensor	Feature	Weight
Force	1	0.03
	2	0.02
	3	0.03
	4	0.03
Accelerometer	1	0.02
	2	0.30
	3	0.02
Gyroscope	1	0.25
	2	0.15
	3	0.15

The test scenario is the "*B*" path represented in Figure 57, which involves a straight walk with two curves, in the middle of the path, one to the left and the other to the right. A total of 40 steps, each time, were performed in this scenario which gives a total walking distance of 25 meters. Four runs in this scenario, for each pedestrian, were performed. The obtained results can be seen in Table 33.

Table 33: Accuracy results for the developed algorithms that characterize the step length

Method	**Short**		**Normal**		**Long**		**Execution time**
	Waist BSU	**Foot BSU**	**Waist BSU**	**Foot BSU**	**Waist BSU**	**Foot BSU**	
Heuristic	94%	94%	63%	91%	87%	72%	5 ms
DTW	94%	100%	74%	50%	71%	97%	300 ms
SVM	98.4%	96.1%	80.3%	81.9%	92.6%	90.7%	3 ms
Neural Network	100%	100%	86.7%	83.3%	96.2%	100%	2 ms
Heuristic Fusion	96%		91%		90%		8 ms
SVM Fusion	100%		86.0%		98.1%		3 ms
Neural Network Fusion	100%		90.0%		96.2%		3 ms

Considering the obtained results, it can be concluded that for a short step both BSU present similar results. For this classification the learning algorithms have the best results, with an accuracy of 100%, but the difference is not hugely evident.

In the case of a long step, there is not an evident difference between BSUs, since in some algorithms the waist BSU gives the best results and in the other it is the foot BSU that has better results. Regarding the learning algorithms, when considering the data from each BSU the Neural Network has the best results. However, when combining the data from both BSU, the SVM has the best performance.

The normal step is the most difficult to classify. The main reason for this phenomenon is because it sits between the other two classes. For this classification the combination of both BSU gives better results than using the data from each BSU separately. The Neural Network had the best results for the learning algorithms, but the best result was obtained by the Heuristic Fusion. Nonetheless, none of the misclassifications given by the algorithms was to the opposite class, meaning that a short step was never classified as a long step and vice-versa.

Analysing the obtained results, it can be concluded that through the sensors complementarity the step length was categorized with higher accuracy. Also, it can be concluded that the learning of the gait parameters enables a better characterization of a step. The Neural Network gave the best results, having a mean accuracy very close to 96%.

From our tests it was identified that a learned dataset 10 times smaller is sufficient to achieve similar results. Making the learning procedure simpler and faster to a pedestrian perform before using this system.

The tests and the evaluation results, show that both BSU give similar results on detecting short steps, but in the case of normal steps the foot BSU has a higher accuracy. However, the long steps are better detected by the waist BSU.

Combining the data from both BSU, the weaknesses of one are suppressed by the advantages of the other, thus improving the overall results.

6.4 SUMMARY

From previous experiences step length estimation based only on an accelerometer has low accuracy [116]. Thus, a characterization of each step was performed. With this characterization the localization algorithm has more accurate knowledge about the

type of step that was given, namely in which type of terrain, the direction and its possible length. It is more accurate because it is based on the learned gait characteristics and on the fusion of data, where the measurements of a group of sensors can surpass some erroneous measurements of one sensor [11].

The type of terrain classification is essential to distinguish the terrain where the step was performed. The types of terrain, normal and stairs, are the most common when indoors [13]. After the type of terrain, the direction of the step is detected. Finally, a range for the length of the step is obtained. The possible ranges are short, normal and long steps. With this classification the displacement estimation can be limited according to the bounds of each category. Thus, the localization algorithm can limit the error growth of the PINS, since the errors for a type of step are different to the other, meaning that different estimation algorithms can be used for each state. Making this system suitable to be used by humans in their daily life [9].

Since the kinematics of a walking or a running human are usually described within the context of a step. The human gait is highly repeatable and the step offers a discrete segmentation of a human trajectory. In this chapter was clarified that learning algorithms, based on a previous set of exercises made by a pedestrian, are suitable to be used to perform step characterization with a better accuracy than heuristic and other approaches.

The step characterization presented satisfactory results when using learning algorithms, namely the Neural Network approach. Also, the combination of the information gathered from the two BSU, which are distributed through the human body, was important to achieve better results, since one BSU can complement the other in the different activities that a pedestrian can perform. The implemented neural networks presented better results than the implemented SVMs. Mainly because they tend to generalize more. Thus, smaller differences in the inputs, from what was learned in the learning phase, are better classified by this algorithm.

For each characterization type were extracted, from a set of sensors, several features instead of using all the sensors signal data. This task is important because the exploitation of a huge volume of data requires reliable methods of data reduction to support the understanding and the extraction of useful patterns to be used by the classification algorithms that label the datasets. The sensors were chosen based on their importance to made each characterization. For example, the *x-axis* of the waist accelerometer is not important to detect the walking direction. However, it is very important to detect the type of terrain, in order to quantify the vertical acceleration that is made on ascending or descending stairs.

Using only one feature per sensor demonstrated that too many erroneous characterization errors occurred, thus more than one feature per sensor was retrieved. Then these features were used to label the step according to the characterization classes.

The characterization that presented the best results was the direction of the step. The step direction is the easiest to classify because the sensors data are very different for each one of the possible classes [12]. The characterization that presented the worst results was the length. For this characterization the bounds for each one of the classes are very close, so misclassifications can easily occur [12].

The learning algorithms, when compared to the heuristic approaches, can maintain a similar performance when used by different pedestrians [79]. Which is not possible to be achieved by the Heuristic approach, since it is based on some thresholds. These thresholds are typically correlated to one pedestrian, or group of pedestrians. If the physical characteristics of the pedestrians are too different, new thresholds need to be defined, which is not necessary in the case of the learning algorithms. It only needs to learn the gait characteristics of each pedestrian.

Regarding the algorithms execution performance, the DTW approach is always the one that takes longer to execute. As presented in Section 2.4.1 this algorithm has a quadratic time and space complexity. However, the results are better than the Heuristic approach, but worst than the information fusion algorithms. Despite its accuracy, the execution time is too high to be used on mobile devices, since their performance capabilities are still low [8].

7

LOCALIZATION

In this chapter are described the algorithms developed to estimate the displacement of a pedestrian. This chapter is divided into three parts. In the first part, Section 7.1, is presented the algorithm that estimates the orientation of the pedestrian.

The second part, Section 7.2, describes the implemented algorithms that estimate the displacement. To perform this estimation, the orientation algorithm needs to be combined with an algorithm that estimates the distance walked by the pedestrian. The distance estimation is performed by two algorithms, one estimates the walking distance only based on the sensors of the foot BSU, and the other uses the combination of the two BSU with the implemented learning algorithms.

Finally, an evaluation of the developed system is performed, where this PhD work is compared with the traditional PINS approaches.

7.1 ORIENTATION ESTIMATION

The pedestrian orientation is estimated using the information about the sensed magnetic strength of the north pole. However, as seen from the state of the art (Chapter 2) the magnetometer is too sensible to magnetic disturbances. These disturbances leads to a poor direction estimation.

A combination of sensors is the best way to achieve an accurate orientation estimation. Typically the accelerometer, magnetometer and gyroscope sensors are used to perform this estimation. As previously stated, each sensor has its strengths and weaknesses, thus a best result is achieved by fusing their data.

The implemented orientation algorithm is represented in Figure 77. It starts by obtaining the orientation angles, which are extracted by combining the magnetometer and accelerometer data. The accelerometer provides a three-dimensional vector containing the acceleration for each axis. With the vector constantly pointing to the center of the Earth gravity, the pitch and roll angles can be determined. To determine the yaw angle, the magnetic component, provided by the magnetometer, is used.

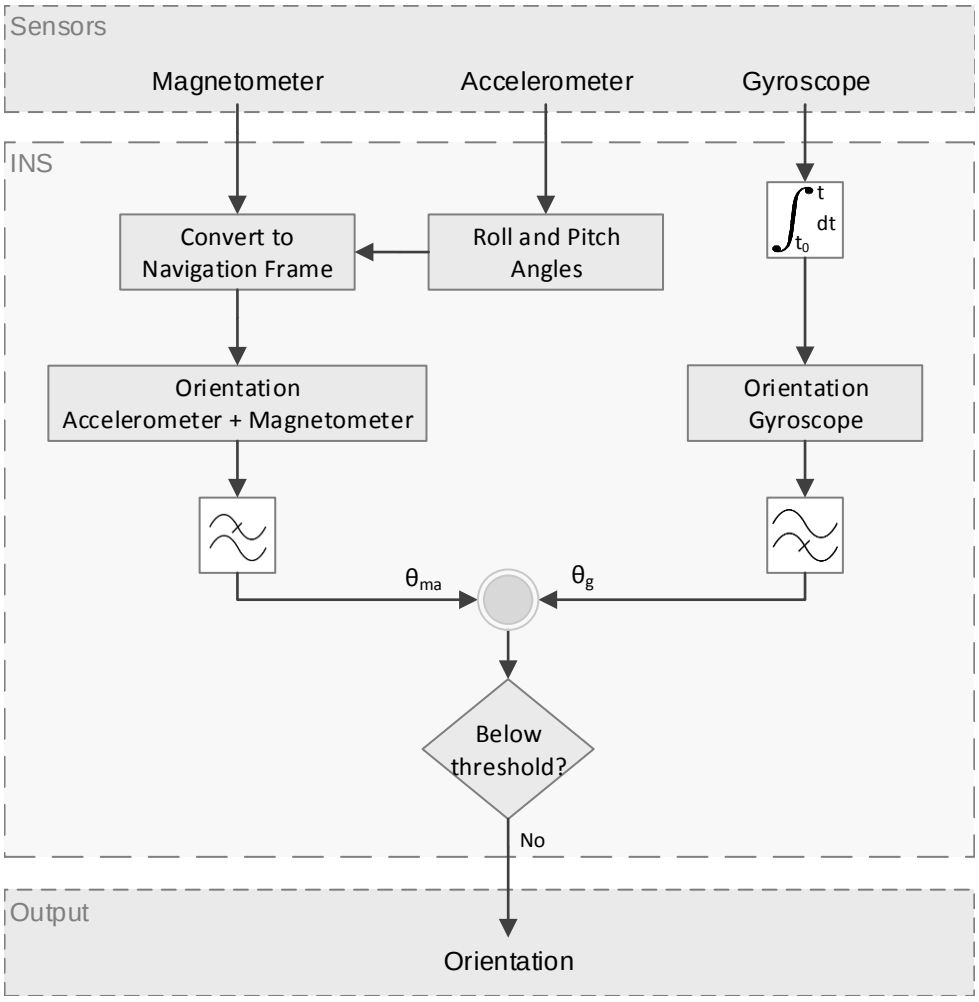


Figure 77: Orientation algorithm overview

Since both, accelerometer and magnetometer, are fixed on the BSU, their measurements change according to the orientation of the BSU. Thus, as the sensors are aligned with the inertial frame their data must be transformed into the navigation frame.

Equations 33 and 34 are used to transform the magnetometer measurements into the navigation frame. The compass heading is then computed from the arctangent of the ratio of the two horizontal magnetic field components, as presented in Equation 35.

$$M_x = m_x \cos\beta + m_y \sin\beta \sin\alpha + m_z \sin\beta \cos\alpha \tag{33}$$

$$M_y = m_y \cos\alpha + m_z \sin\alpha \tag{34}$$

$$\gamma = atan2(\frac{M_x}{M_y}) \tag{35}$$

where m_x, m_y and m_z correspond to the intensity of the magnetic field on the sensor frame's *x-axis*, *y-axis* and *z-axis* respectively, M_x and M_y correspond to the intensity of the magnetic field on the *x-axis* and on the *y-axis*, but in the navigation frame, and α, β and γ correspond to the roll, pitch and yaw angles respectively.

From the obtained results, which are presented in Section 7.3.1, the walking direction is best estimated using the BSU placed on the waist rather than on the foot. Thus, the implemented orientation algorithm runs on the waist BSU.

The walking direction, in this case the azimuth, is obtained from the yaw angle and is defined in rad that ranges from $0\ rad$ to $2\pi\ rad$ and is measured based on the magnetic north. As can be seen in Figure 78 this is the angle that gives information about the direction of the pedestrian.

Although the accelerometer and magnetometer sensors can provide orientation data that does not derive (drift) over time, they contain too much noise. The gyroscopes produce rotational speeds with low noise. However, small errors derived from the integration of the data accumulated over time, can give inaccurate orientation data.

To estimate the walking direction using these three sensors one of the best approaches is to use a complementary filter. This type of filter allows the combination of two sources of data with different frequency bands to form more accurate values in the time domain. There are several advantages in using this method, as the low implementation complexity and the quality of the obtained results.

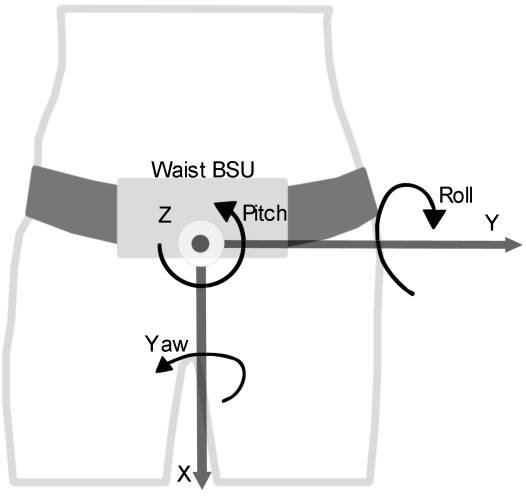


Figure 78: Waist BSU with the corresponding roll, pitch and yaw angles

This type of filter operates in the frequency domain and may be defined by using two or more functions that complement each other. In this system, the accelerometer combined with the magnetometer (θ_{ma}) provide information of high frequency noise, which is filtered through a low-pass filter. The result of the integration of gyroscope data (θ_g) produces low-frequency noise, which is filtered through a high-pass filter. Since these filters are mathematically complementary, then the result of the complementary filter (θ) is the complete reconstruction of each variable of interest without the noise. The implemented filter is defined in Equation 36.

$$\theta = (1 - \phi)\theta_g + (\phi)\theta_{ma} \tag{36}$$

where ϕ is the filter coefficient applied to each part, where a higher coefficient gives a higher importance to one of the parts. The coefficient must be higher than zero and lower than 1 ($0 < \phi < 1$). In this algorithm the coefficient (ϕ) was defined as 0.92.

Since the data sample was defined to 20Hz, then the sampling period (ΔT) is 0.05 seconds. So, the time constant of the complementary filter is 0.58 seconds, which is the result of Equation 37.

$$t = \frac{\phi \Delta T}{1 - \phi} \tag{37}$$

where t is the relative duration of the signal on which the filter will operate.

The angle that is obtained from the orientation algorithm is then compared to a threshold. If the resulting angle is inferior to the threshold any change on the orientation is considered. The threshold was defined as 0.035 rad. As referred in Section 5.3 during a normal walk, on a flat surface, the average magnitude of the pelvis rotation is of 0.070 rad. Therefore the threshold was defined as half this value, thus the error that the algorithm can produce is negligible. Also, the best results were achieved with this threshold.

The complementary filter can be considered as a simple filter that attributes a weight to the combination of the magnetometer and accelerometer data, and to the gyroscope data. When the motion is faster than the 0.58 second time period, the gyroscope integration for angular displacement calculation (θ_g) is more weighted and the magnetometer and accelerometer combination noise is filtered out. When the motion is slower than the 0.58 second time period, the magnetometer and accelerometer tilt measurement (θ_{ma}) has a higher weight than the gyroscope (θ_g), to reduce the gyroscope drift impact.

By merging the data obtained from the magnetometer and accelerometer with the data from the gyroscope, the filter produces a smoother and a more accurate orientation estimation, which is important to achieve the main objectives of this work. Also, it is not sensitive to the linear horizontal acceleration and gyroscope drift.

To verify the responsiveness and performance of the algorithm two experiments were conducted. The first one was performed to identify the algorithm offset when the device is stationary. This test had a duration of 60 seconds and the obtained signal can be seen in Figure 79. Some performance metrics can be visualized in Table 34. It includes the minimum and maximum obtained values, and the mean and standard deviation obtained during the test. The standard deviation of the algorithm is very low, which indicates that it produces stable data. Also, the difference between minimum and maximum values is way lower than the defined threshold.

Table 34: Orientation algorithm statistics, in rad, with the device stationary for 60 seconds

Parameter	Orientation Algorithm
Minimum	0.9340
Maximum	0.9359
Mean	0.9350
Standard Deviation	0.00003

The second experiment involved four rotations of about $\frac{\pi}{2}$ rad each. The obtained data can be seen in Figure 80. For this test the algorithm gave, for each rotation, the results presented in Table 35. The obtained results are very satisfactory, since the estimated rotations were similar to the ones made manually by hand. The sum of all rotations gave a total of 6.28 rad (2π rad). Also, the values obtained at the start and at the end of the test are very similar.

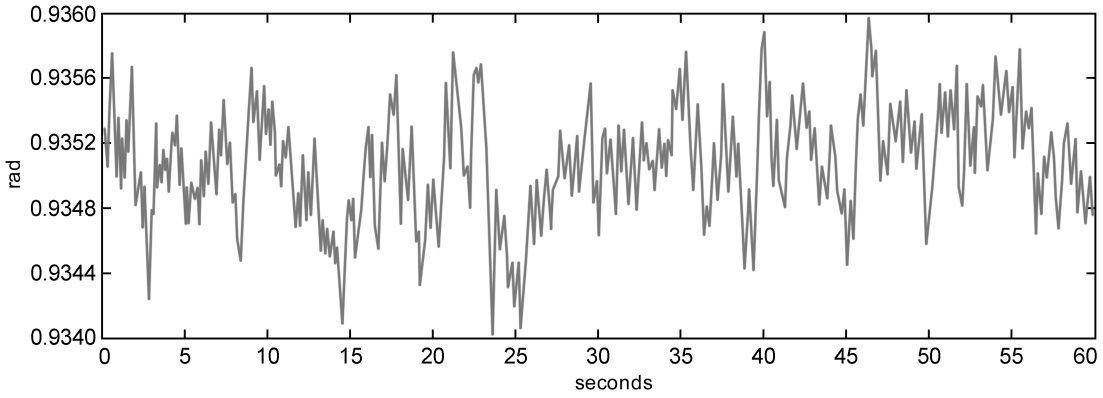


Figure 79: Orientation algorithm data with the device stationary for 60 seconds

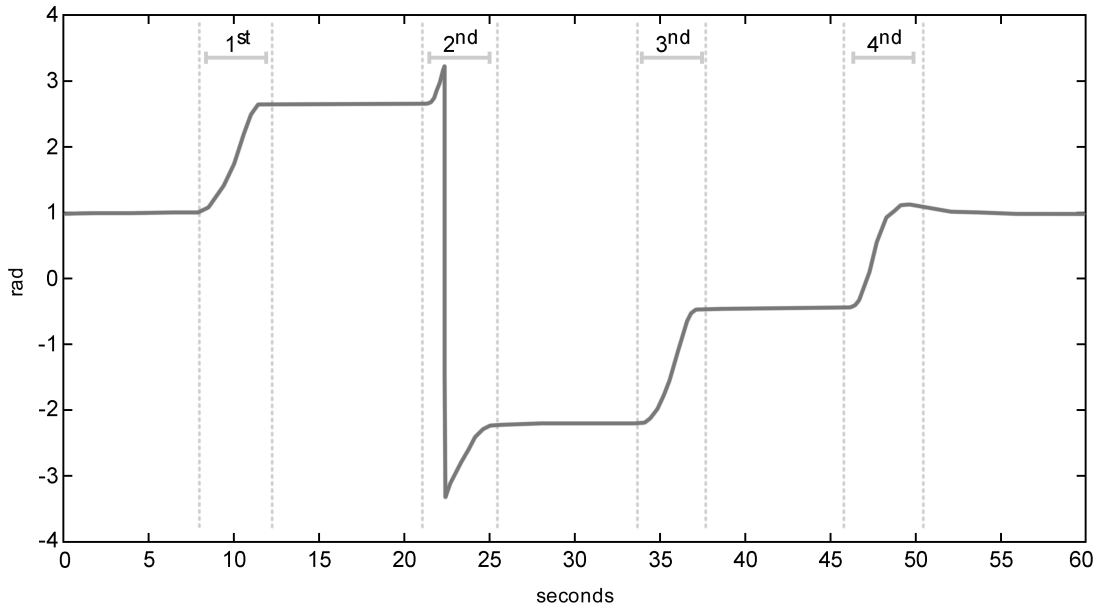


Figure 80: Orientation algorithm output for four $\frac{\pi}{2}$ rad rotations

Table 35: Orientation algorithm results for four $\frac{\pi}{2}$ rad rotations

Rotation	Radians
1^{st}	1.58
2^{nd}	1.70
3^{th}	1.63
4^{th}	1.37
Total	6.28

This filter, when compared to a Kalman approach, is faster to process. This responsiveness is important when mobile devices are being used to handle these calculations.

7.2 DISPLACEMENT ESTIMATION

This section describes the algorithms developed to estimate the displacement taken by a pedestrian through the usage of the developed BSUs. These approaches estimate the distance based on an algorithm that is not dependent on a specific user. Meaning that these approaches do not estimate displacement based on the step count and on a fixed distance value for each performed step.

As seen in the state of the art (Chapter 3) PINS are typically based on DR algorithms that estimates displacement based on walking distance and azimuth information. To estimate distance, the most common approach is based on using a double mathematical

integration of the accelerometer data. This integration is based on the equations of motion. However, this approach, because of the walking dynamics and the sensors configuration, has too many errors. This approach will be referred throughout the document as *M1* and it only relies on the data from two sensors, accelerometer and gyroscope.

One of the objectives of this work was to improve the accuracy of the most common PINS approaches by including new types of sensors, like force sensors. Thus, the first implemented approach, to estimate the displacement, besides using the accelerometer data it also uses the force sensors data to improve the displacement estimation accuracy. Throughout the document this approach will be referred as *Accelerometer and Force Sensor* (*M2*).

Using the same sensory set as *M2*, it is proposed the *PLASYS* (*M3*) approach, which learns the human gait characteristics using the data of the two BSUs to then characterize the step. Thus, the information from both BSUs are fused and used to apply the learning algorithms. Then according to the characterized step, different algorithms and rules are applied to estimate the displacement.

7.2.1 *Accelerometer and Force Sensor*

Displacement is typically calculated using two inertial sensors, an accelerometer and a gyroscope. However, in this algorithm the force sensors were also included, which were an important component to improve the step detection (Chapter 5). Thus, this algorithm, to estimate the displacement, uses the data from three sensors. The implemented algorithm is represented in Figure 81.

Before using this system, a setup procedure must be performed. This procedure has the duration of two seconds, where the devices (BSUs) must be stationary. In this procedure the accelerometer and gyroscope data samples are obtained to estimate their offset. The estimated offset is then removed from each source of data.

Then, the step detection algorithm, presented in Section 5.2, which uses the accelerometer and the force sensor data to verify if the pedestrian is stationary, is applied.

When the pedestrian is stationary, the speed is set to zero and any displacement is estimated, in other words the ZUPT technique (Section 2.2.3) is applied. Besides the speed, the angular displacement and the BSU angles are also set to zero, in other words the ZARU technique (Section 2.2.4), is applied. This technique is important to be applied since, as mentioned previously, the angular displacement, which is calculated by the

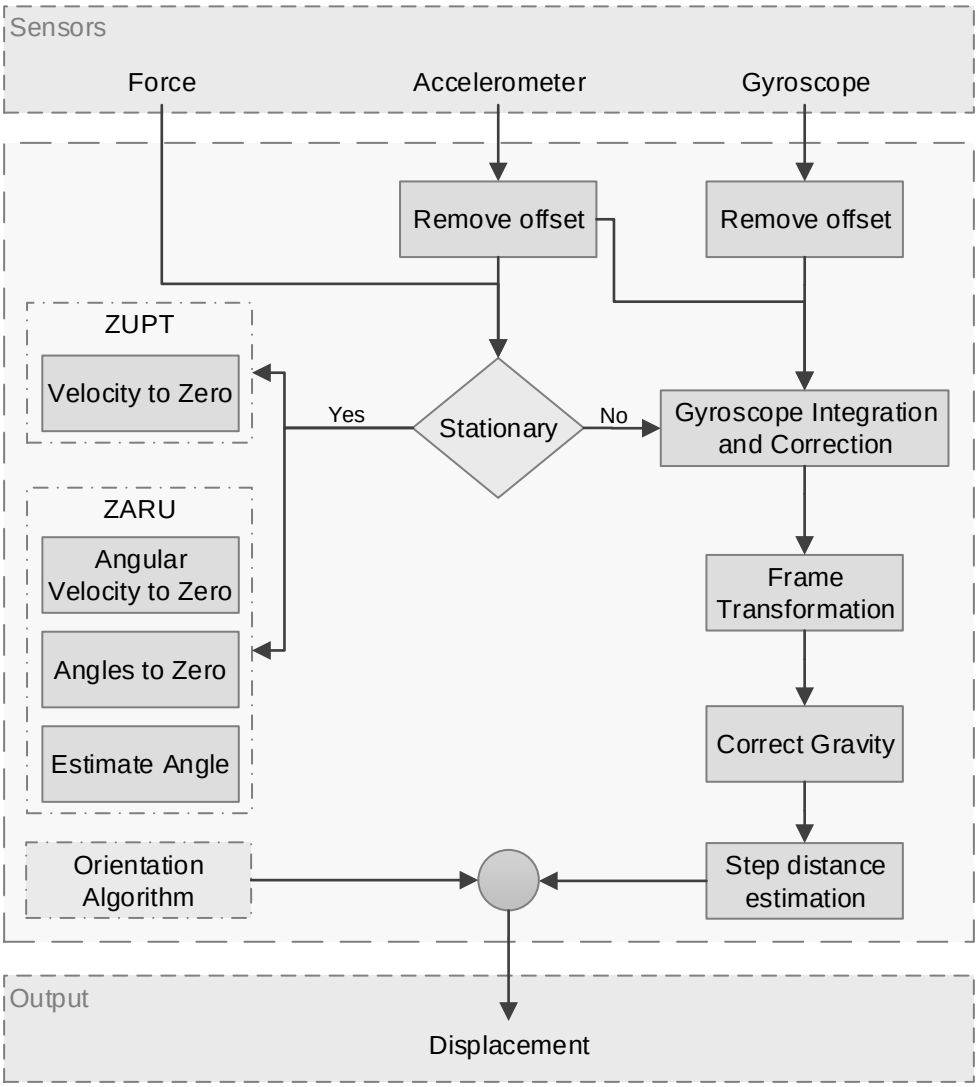


Figure 81: Accelerometer and force sensor displacement estimation algorithm

integration of the gyroscope data, tends to diverge over time (drift). This drift has great impact on the transformation of the acceleration data from the inertial to the navigation frame, which by its turn has a significant impact when used to estimate the walking distance.

Thus, when the pedestrian is stationary it is calculated the angle that the BSU is in correspondence with the ground, which serves as the initial angle to be used by the frame transformation algorithm when the foot is on movement. To calculate the BSU position the information about the roll, pitch and yaw angles is necessary. These angles are estimated based on the accelerometer data and are represented in Figure 82. This information is not obtained directly from the accelerometer data, instead it is calculated using two trigonometric equations (Equations 38 and 39). However, with the

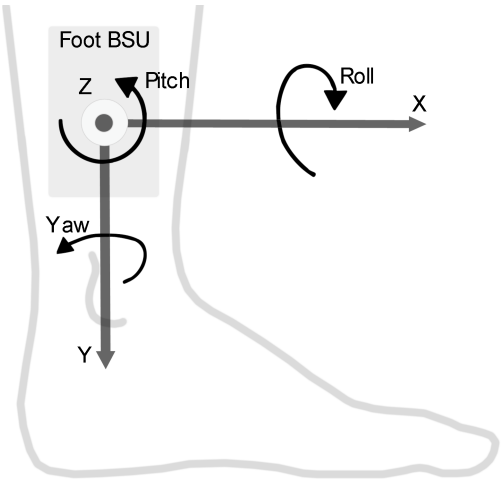


Figure 82: Foot BSU with the corresponding roll, pitch and yaw angles

accelerometer data only the roll and pitch angles are possible to calculate. The third angle (yaw) can not be obtained using the accelerometer data, since the angles only can be obtained when a gravitational force exists on the correspondent accelerometer axis.

The force of gravity always acts perpendicular to the earth's surface. Thus, when the BSU is tilted at an angle, part of that force acts along the *x-axis* of the BSU and the other part acts along the *y-axis*. The pitch and roll angles are estimated using the Equation 38 and Equation 39, respectively.

$$\mathrm{pitch} = \arctan\left(\frac{-x}{\sqrt{y^2 + z^2}}\right) \tag{38}$$

$$\mathrm{roll} = \arctan\left(\frac{z}{\sqrt{x^2 + y^2}}\right) \tag{39}$$

Whenever a step is detected, the gyroscope drift is corrected, continuously during a step, using a sensor fusion algorithm. This fusion is made by combining the gyroscope data with the accelerometer data. Through the complementary of the sensors is intended to obtain a signal without drift, as the accelerometer signal, and without the gyroscope data (after integration) noise.

To implement this fusion, two algorithms were tested, one based on a complementary filter and the other based on a Kalman filter. The implemented Kalman filter [50], which will be described below, achieved the best results. The complementary filter was composed by a low-pass filter applied to the accelerometer data and a high-pass filter applied to the gyroscope data. This complementary filter was simple to implement and

achieved satisfactory results. However, it is not a so good estimator as the Kalman filter, because it does not consider any statistical description about the noise that affects the sensor data [123].

The Kalman filter implementation attempts to estimate the state x of the discrete process that is modelled based on Equation 13 and the state observations are made based on Equation 14, presented in Section 2.2.5. The state transition matrix A, the control-input matrix B and the observation matrix H are constant. The process noise covariance matrix Q and the measurement noise covariance matrix R are also considered constant [50].

The Kalman filter gain K_k is calculated at each iteration of the filter to keep the error covariance matrix P_k as low as possible to maintain accurate state updates. The state vector x_k is equal to $[\theta_k \; b_k]^T$, where θ_k (in rad) is the angle and b_k (in rad/s) is the gyroscope deviation. The input of the system u_k corresponds to the angular velocity obtained by the gyroscope ωg_k.

Based on these premisses the state equation (Equation 13) is represented as in Equation 40.

$$x_k = \begin{bmatrix} 1 & -\Delta T \\ 0 & 1 \end{bmatrix} x_{k-1} + \begin{bmatrix} \Delta T \\ 0 \end{bmatrix} u_{k-1} \tag{40}$$

The measured value is the angle obtained by the accelerometer. Thus, the measurement or observation equation z_k is represented as in Equation 41.

$$z_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta acc_k \\ b_k \end{bmatrix} \tag{41}$$

The process noise covariance matrix Q is given in Equation 42, and the measurement noise covariance matrix R is given in Equation 43.

$$Q = E(x_k x_k^T) = \begin{bmatrix} E(\theta_k \theta_k^T) & 0 \\ 0 & E(\theta_k \theta_k^T) \end{bmatrix} \approx \begin{bmatrix} \sigma(\omega g)\Delta T & 0 \\ 0 & \sigma(b) \end{bmatrix} \tag{42}$$

$$R = E(z_k z_k^T) = \sigma(\theta acc) \tag{43}$$

As already mentioned the Q and R matrices are considered to be constant. Thus, the measurements are obtained with the BSU stationary and the matrices are then defined.

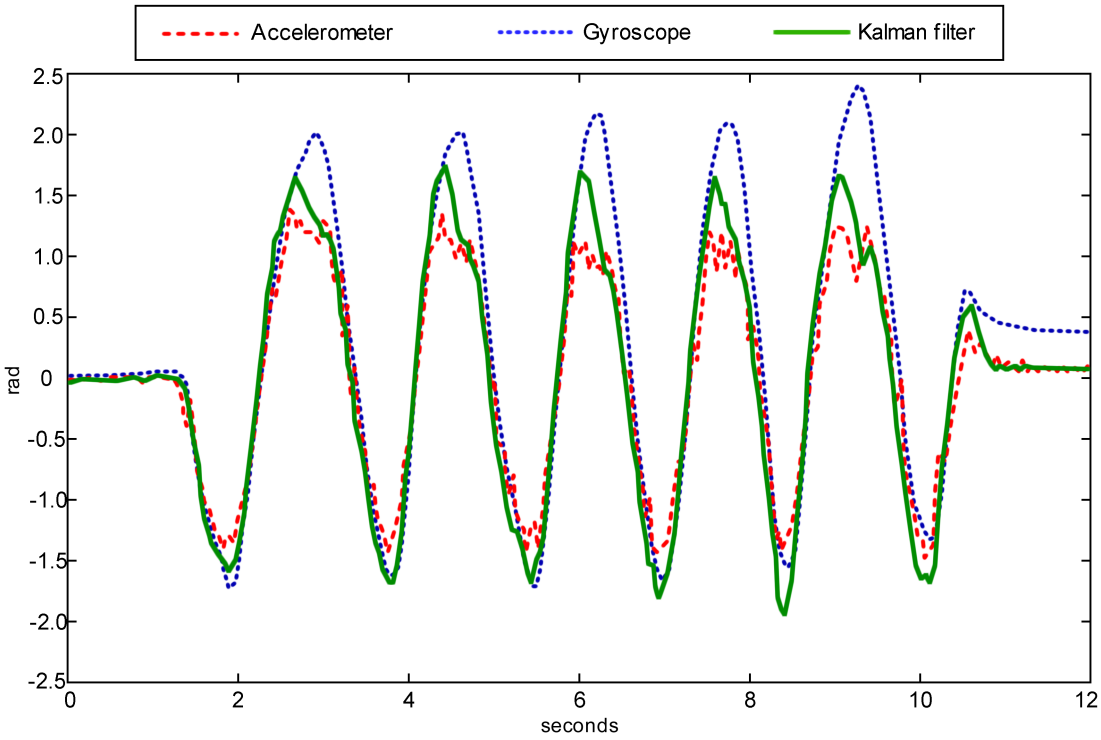


Figure 83: Angular rotation estimated by a Kalman filter, compared with the angular rotation estimated by the accelerometer and the gyroscope separately

An example of the output of the implemented filter is shown in Figure 83. This figure presents the angle calculated around the *z-axis*. The final result of the filter is a gyroscope signal with reduced or without any drift.

When walking, as seen in the state of the art and from the conducted tests, despite the foot normal behaviour that is to move in the horizontal plane, there is the possibility of unintended rotations made by the foot. Which causes the gravity to influence more than one axis, which results in too many errors. Thus, the accelerometer data needs to be transformed from the inertial frame to the navigation frame. This transformation is performed using a rotation matrix, which uses the data from the gyroscope, smoothed with the Kalman filter previously presented, and the data from the accelerometer. The initial state, at each step, is given by the angles calculated whenever the foot is stationary.

In Figure 84 is presented the accelerometer data for the *x-axis*, for a five step walk, after the frame transformation and without any previous data preprocessing. In the same figure it is presented the accelerometer data converted to the navigation frame, where were previously applied the ZUPT and ZARU techniques, and the Kalman filter that smooths the gyroscope data. In other words, with the algorithms explained in this section. As can be seen, during the 5 steps the acceleration pattern is similar. However,

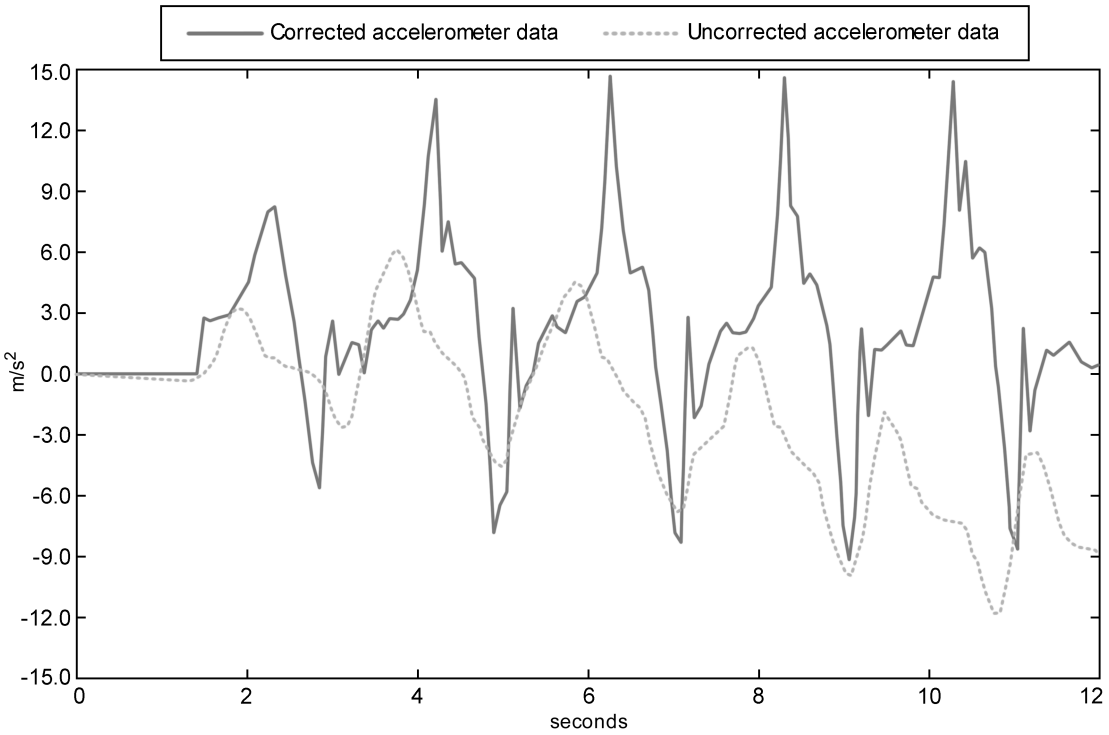


Figure 84: Accelerometer data for the *x-axis*, corrected with the proposed algorithm (Corrected accelerometer data) and without its correction (Uncorrected accelerometer data)

the acceleration values are very different. These differences leads to poor results when using the acceleration data to estimate the walking distance. With the proposed approach there is a more stable acceleration pattern and the correspondent acceleration values are very similar throughout the walking path.

Then, from the resulting acceleration data, it is removed the effect of gravity. The resulting data is then used to estimate the walking distance. To perform this estimation two algorithms were implemented. The first uses the equations of motion, where the acceleration data is integrated twice in order to obtain the walking distance.

The second algorithm is based on a Kalman filter. More specifically this implementation is based on the Wiener-process acceleration model in the *white-noise jerk* version [17]. The model is also referred as the *nearly-constant-acceleration model* or the *constant-acceleration model*. This model assumes that acceleration can be described as a Wiener process, which is a continuous-time process with random increments [68]. This model is typically applied in kinematics to describe the motions of objects.

The implemented *white-noise jerk* version of this model, assumes that since the acceleration is not exactly constant, its derivative $\dot{a}$ (i.e. the "jerk") can be modelled by white noise $w(t)$ (Equation 44).

$$\dot{a}(t) = w(t) \tag{44}$$

The representation of the state vector x is defined in Equation 45. This vector represents the states for the estimated distance (d), velocity (v) and acceleration (a).

$$x(t) = \begin{bmatrix} d(t) & v(t) & a(t) \end{bmatrix} \tag{45}$$

The continuous-time state equation represents a third-order model with three integrations (Equation 46).

$$\dot{x}(t) = Ax(t) + Dw(t) \tag{46}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{47}$$

$$D = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{48}$$

Its discrete equivalent is represented as in Equation 47.

$$x_{k+1} = Fx_k + w_k \tag{49}$$

Since this model is supposed to estimate the acceleration (a), the velocity (v) and the distance (d), the equations that estimate these variables need to be represented. These equations are based on the Kinematic Equations, and are represented in Equations 50, 51 and 52. In this representation, T represents the sampling time, which is the period between two samples.

$$a_k = a_{k-1} \tag{50}$$

$$v_k = v_{k-1} + a_{k-1}T \tag{51}$$

$$d_k = d_{k-1} + v_{k-1}T + \frac{1}{2}a_{k-1}T^2 \tag{52}$$

Then, the discrete Equation 49 can be transformed in Equation 53.

$$x_{k+1} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_k \\ v_k \\ a_k \end{bmatrix} + w_k \tag{53}$$

The process noise covariance matrix Q_k is defined in Equation 54 and the measurement noise covariance matrix R is defined in Equation 55.

$$Q_k = S_w \begin{bmatrix} \frac{1}{20}T^5 & \frac{1}{8}T^4 & \frac{1}{6}T^3 \\ \frac{1}{8}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{6}T^3 & \frac{1}{2}T^2 & T \end{bmatrix} \tag{54}$$

$$R = [1] \times 0.01 \tag{55}$$

It should be noted that S_w corresponds to the power spectral density, not the variance, of the continuous-time white noise. Using the S_w allows the model to match the actual movement of the object. An almost constant acceleration model can be obtained by selecting a very small intensity value. After testing several constants, the best and more reliable results were obtained using the S_w with the value of 1.8.

Having the two components, azimuth (given by the orientation algorithm) and the distance of a step, the new location of the pedestrian can be estimated. This new location is always estimated based on a previous location. The latitude of the new location is estimated using Equation 56 and the longitude using Equation 57. It receives as input the current azimuth, the estimated distance and returns the current location. The initial latitude and longitude must be in *radians*. The distance must be in *km* and the azimuth must be in *radians*.

$$\varphi_2 = \operatorname{asin}(\sin\varphi_1 \times \cos\delta + \cos\varphi_1 \times \sin\delta \times \cos\theta) \tag{56}$$

$$\lambda_2 = \lambda_1 + \operatorname{atan2}(\sin\theta \times \sin\delta \times \cos\varphi_1, \cos\delta - \sin\varphi_1 \times \sin\varphi_2) \tag{57}$$

where φ is latitude, λ is longitude, θ is the azimuth, given in a clockwise order in correspondence of the north pole, δ is the angular distance $\frac{d}{R}$, where d is the walking distance and R is the earth's radius (6371km).

7.2.2 *PLASYS*

After improving the step detection with the inclusion of force sensors, and therefore improving the displacement estimation, another approach was developed. In this approach the step was characterized, based on learning the human gait characteristics. With the step characterization different algorithms and rules can be applied for each characterization. Thus, emerged the algorithm presented in Figure 85.

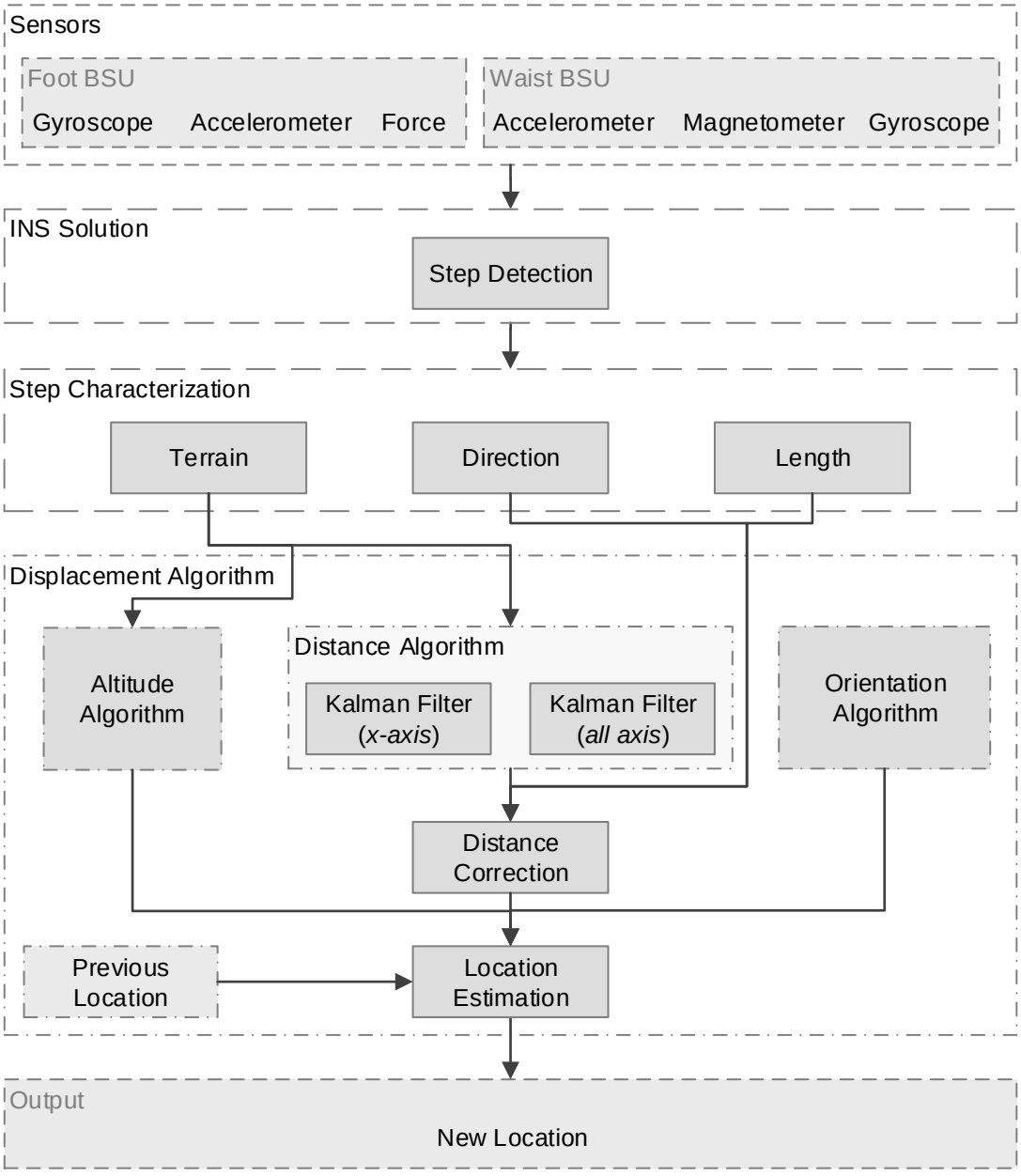


Figure 85: PLASYS displacement algorithm overview

The implemented approach starts by obtaining the sensors data from the two BSUs, foot and waist. These data will be used by each algorithm of this approach, which, since it is based on the PDR technique, is decomposed into three parts: step detection, step characterization and displacement estimation.

The first part (i.e. step detection) is performed using the algorithm presented in Section 5.2, which uses the foot accelerometer and force sensor data to detect a step. However, the algorithm presented in Section 5.3, which uses the gyroscope data to detect a step, is also used to help on combining the data of the two BSUs.

Since two BSUs are used in this work, one important aspect to do is a correct merging of the data. When the data from the foot BSU is received on the waist BSU it is given to it a timestamp. Thus, it is known what is the moment in which the data corresponds. However, due to communication delays, some discrepancies between the data from the two BSU can occur. Therefore the step detection using the gyroscope is used to correct some misalignments that can occur between the data.

When a step is detected by the foot BSU, it is verified for that period of time where the same step was detected by the gyroscope algorithm, and then the step, for the waist BSU data, is considered in a different (or not) period of time. For example, a step was detected by the foot BSU between the time period 5 and 8, then it is verified using the gyroscope algorithm, for this period, the correspondent step. If the time period detected in the waist BSU is different (e.g. time period between 4 and 7, which is a earlier time period when compared to the foot BSU), then it is considered that for the waist BSU the step period is between 4 and 7 and for the foot BSU between 5 and 8. This correction worked well on the tests, however a more complex approach (i.e. less ad-hoc than this approach) must be encountered to be achieved more precise results.

After detecting a step and making the correction of the BSUs timestamps, the second part (i.e. step characterization) of the algorithm is performed. This characterization is performed to the data of the period in which a step is detected. The step is then characterized according to the terrain, direction and length, using the corresponding neural network that were presented in Chapter 6. As the step characterization needs a dataset of steps to correctly classify the steps in real-time, a set of exercises needs to be done prior to using the proposed system. Thus, it was established that the pedestrian must perform, before using the system, 75 steps in the different types of terrain, 50 steps in both directions and 90 steps with the three possible lengths classes. Then, this dataset will be used to train the network. Further developments must be performed to make simpler the system usage.

The third, and last, part of this algorithm gives as output the pedestrian new estimated location. To estimate the displacement two informations are needed: orientation and distance. Apart from estimating the displacement, the algorithm also estimates the altitude. The orientation is estimated using the algorithm presented in Section 7.1.

The implemented distance estimation algorithm is similar to the one presented in Section 7.2.1. Thus, when the foot is stationary ZUPT and ZARU techniques are applied, where the velocity, angular velocity and angles are set to zero, and the foot BSU angle is estimated. When the foot is not stationary the gyroscope data is mathematically integrated and corrected, then it is used to perform the frame transformation. Only then the step distance is estimated using one of two approaches.

These approaches are based on the same Kalman filter implemented in Section 7.2.1, which is an implementation based on the Wiener-process acceleration model in the *white-noise jerk* version. After some testing on calculating the distance for all the datasets that were collected during the development of this work, it was concluded that, for each type of terrain, better results can be achieved using different parameters on the Kalman filter.

Thus, two different approaches, using the same algorithm, were developed. The first one is similar to the one presented in Section 7.2.1, where the Kalman filter estimates, based on the *x-axis* data from the accelerometer, the states for the distance (d), velocity (v) and acceleration (a), according to Equation 46. The second approach estimates the same states, but for the data of each one of the *3-axis* of the accelerometer. Thus, it is applied three Kalman filters where each one estimates one of the following state vectors: x, y and z, which can be represented as in Equation 58, Equation 59 and Equation 60, respectively. After the estimation of the states, the distance is calculated using the norm of the distance (d_c) given by Equation 61.

$$x(t) = \begin{bmatrix} d_x(t) & v_x(t) & a_x(t) \end{bmatrix} \tag{58}$$

$$y(t) = \begin{bmatrix} d_y(t) & v_y(t) & a_y(t) \end{bmatrix} \tag{59}$$

$$z(t) = \begin{bmatrix} d_z(t) & v_z(t) & a_z(t) \end{bmatrix} \tag{60}$$

$$d_c = \sqrt{d_x^2 + d_y^2 + d_z^2} \tag{61}$$

These two approaches are applied according to the type of terrain where the step was given. The first approach is applied to ascending stairs and to normal terrain, and the second approach is applied to descending stairs. This solution was selected based on the results that were obtained for each approach applied to the datasets of each type of terrain. In the case of ascending stairs the first approach achieved a result 33% better than the second approach. For descending stairs the second approach had a result 20% better than the first approach. Finally, for normal terrain the first approach had a result 12% better than the second approach.

Supposedly, since it is made the conversion from the inertial to the navigation frame, it would only be necessary the data of the *x-axis* to estimate the walking distance. However, due to the inherent inertial sensors deviations the results when descending stairs are better using the data of the three axis. With the distinction about the type of terrain, through the step characterization, was important to establish different algorithms and parameters on distance estimation, which lead to better results.

After estimating the walking distance, with the algorithm chosen according to the type of terrain, it is corrected according to the direction and length characterizations made for that step.

In the case of the direction a correction is made to the sign of the estimated value. Therefore, it is verified if the estimated distance is in correspondence with the direction class. In other words, if the distance is positive the direction class must be forward, and when the distance is negative the direction class must be backward. If the distance and the direction class are dissimilar, the sign of the estimated distance is inverted (i.e. if the distance is positive and the direction class is backward, the distance is transformed from positive to negative, and the opposite happens if the distance is negative and the direction class is forward). Meaning that the direction class prevails over the estimated distance.

In the case of the length correction, it is verified if the estimated distance is in correspondence with the class obtained in the length characterization. If the estimated distance is out the boundaries of that class the distance is corrected to the extremity which is more closer to the estimated value. If it is inside the boundaries nothing is done. For example, if the estimated distance is 60cm and the length class is *Small* then the value is corrected to 30cm, which is the maximum value on which a *Small* step fits. Another example, if the estimated distance is 40cm and the length class is *Long*, then the value is corrected to the minimum value on which a *Long* step can fit, which is 46cm.

Thus, the distance limits (i.e. minimum and maximum) are corrected with this algorithm, making the step shorter or longer according to the classification done by the step characterization algorithm. Therefore the estimation errors that can occur from the inertial sensors inaccuracies are corrected and the walking distance is estimated with better accuracy. This correction has a great impact when estimating the distance on stairs terrain. In this type of terrain the distance is always over-estimated. With this correction the estimated distance remains much more closer to the real distance.

Finally, the new location is estimated based on the obtained orientation, corrected distance and the previous location. The latitude of the new location is estimated using Equation 56 and the longitude using Equation 57, which were presented in Section 7.2.1.

Regarding the altitude estimation, it takes as an initial reference the data (i.e. estimated altitude) given by the GPS. Then, based on the type of terrain some corrections to the current altitude of the pedestrian are made. If the type of terrain is normal (flat), any change to the altitude value is made. However, if the type of terrain is stairs (i.e. ascending or descending) changes in the altitude value are made. The altitude is estimated, using the initial reference, and the distance estimated using the same algorithm (presented above) that estimates the horizontal distance made by a pedestrian, but it only uses the data of one of the accelerometer axis (*y-axis*). When descending stairs the altitude value is decreased, and when ascending stairs the altitude value is increased.

Although this is a valid approach, the errors given by the accelerometer data can produce a large deviation on altitude estimation. From the datasets collected throughout this research, when descending stairs this algorithm presented an average error of 18%, and when ascending stairs it presented an average error of 30%.

The barometer is not used to estimate the altitude, because as seen on Section 4.3.1.4, it does not provide a constant reference point. It varies a lot along different days, and also even during the same day. Thus, it does not provide the necessary accuracy.

As seen along this section, the type of terrain characterization is very important to use different distance estimation algorithms according to the current context. Thus the accuracy of this characterization is very important on the final result (i.e. displacement estimation) of the proposed work. Since the altitude is also estimated, the system performs a three-dimension localization.

7.3 EVALUATION

In order to validate the system, and the described features, were performed a set of experiments involving real case scenarios, which represent situations that are commonly encountered when indoors. Also, to validate the proposed system it is compared with the traditional approach, where only a foot IMU is used to estimate the displacement.

This section is divided into two sub-sections. In Section 7.3.1 are presented the results obtained using the implemented orientation algorithm running on the waist and on the foot. The results obtained in both places are compared to identify which is the best place, i.e. the one with the best results, to run this algorithm.

In Section 7.3.2 are presented the results obtained on estimating the displacement using three different methodologies: (i) the typically PINS approach, that detect steps using the accelerometer and therefore estimates the displacement (*M1*); (ii) the algorithm presented in Section 7.2.1 which adds force sensors and other techniques (*M2*); (iii) and the proposed architecture that adds learning algorithms and information fusion between the BSUs (*M3*).

In each section an overview of the execution environment is given along with a detailed description of each use case scenario and the obtained results. These results were collected from a series of experiments conducted in the first three months of 2015, inside a typical building with four floors. The different methodologies are always compared based on the same dataset. The data was firstly obtained on the field then the results were estimated off-line in order to be possible the comparison between the different methodologies.

All the data was collected using the presented BSUs and using two different pedestrians. One is a male with a height of $1.90 meters$ and the other is a female with a height of $1.65 meters$. The presented results are the average values obtained for ten runs for each pedestrian.

7.3.1 *Orientation*

In this proposal two BSUs were spread along the human body to estimate the pedestrian displacement. Thus, several tests were performed to identify the best spots to place those BSUs, but also to test which algorithms work better in each spot. One of these tests was to identify where the orientation algorithm presented the best performance and results.

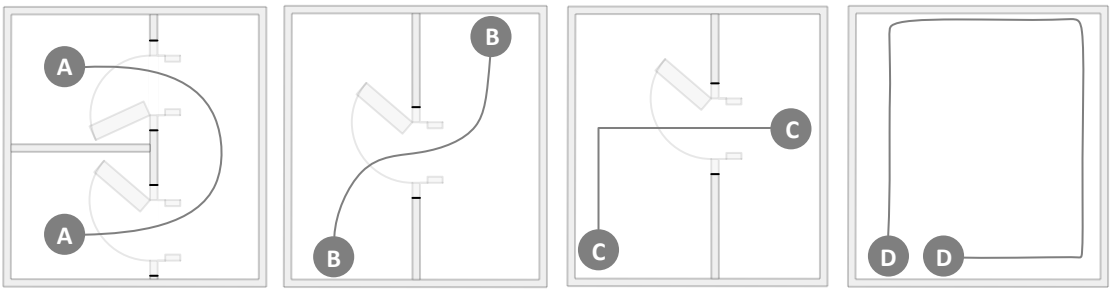
Therefore, in this section the developed orientation algorithm, presented in Section 7.1, is tested running in each BSU, separately. In Section 7.3.1.1 are presented the use case scenarios that were used to evaluate the best place to run the algorithm, and then, in Section 7.3.1.2, are presented the results obtained on each BSU.

7.3.1.1 *Use Case Scenarios*

To evaluate the performance of the developed orientation algorithm running in the different BSUs, four use case scenarios were created. These scenarios are presented in Figure 86.

All scenarios involves a total walking distance of about 6 meters, which gives an average of 8 steps, and are characterized as follow:

- Scenario A - the pedestrian starts in a room and exits that room entering into another, this change on the room is performed by making a curve of about π rad. It is represented as *A* in Figure 86;
- Scenario B - first is performed a moderate curve to the right, followed by another moderate curve, but in this case to the left. This scenario is represented as *B* in Figure 86;
- Scenario C - involves a sharp curve of $\frac{\pi}{2}$ rad to the right and is represented as *C* in Figure 86;
- Scenario D - is a square path made by three right curves of $\frac{\pi}{2}$ rad, and is represented as *D* in Figure 86;



(a) (b) (c) (d)

Figure 86: Use case scenarios to evaluate the orientation algorithm running in the different BSU locations: (a) Scenario A; (b) Scenario B; (c) Scenario C; (d) Scenario D

7.3.1.2 *Experiment Results*

This section presents the results obtained on the comparison between running the orientation algorithm in a BSU placed on the waist versus using the BSU placed on the foot.

As presented in Chapter 5 the foot is a better place to detect the steps. Thus, with these tests it is wanted to identify which spot is more accurate in order to estimate the pedestrian orientation. From the state of the art (Chapter 3), most authors stated that the waist is the best location to place an IMU to estimate the orientation. Thus, these tests will serve to prove if the proposed system has the same behaviour.

For each scenario and BSU, it is presented a table with the mean orientation change, the mean orientation error and the standard deviation. The mean orientation change represents the difference in the rotation value between the beginning and the end of the walking path. The error is calculated based on the difference between the estimated orientation and the real orientation at the final of the path that the orientation algorithm should present. The standard deviation is the deviation that existed over all the performed tests.

Also, it is presented the data collected, in the form of a figure, in each one of the scenarios (7.3.1.1), for each one of the two locations.

The same BSU was used in both locations, in order to the results do not be influenced by the use of different sensors. For these tests was used the developed BSU (Section 4.2).

In Scenario A the pedestrian performs a change in direction of about π rad, thus it is expected that such change must be captured by the algorithm. In Table 36 and in Figure 87 are presented the results and the data, respectively, obtained for each BSU location on this scenario.

Table 36: Results for the orientation algorithm for Scenario A

BSU location	**Mean orientation change** (rad)	**Standard deviation**	**Mean error**
Waist	3.19	0.06	1.73%
Foot	3.42	0.17	8.27%

For the obtained data (Figure 87) it can be seen that the BSU on the foot provides higher inaccuracies. The mean error obtained with the BSU on the foot is 5 times higher than the mean error obtained with the BSU on the waist. Also, since the foot has more

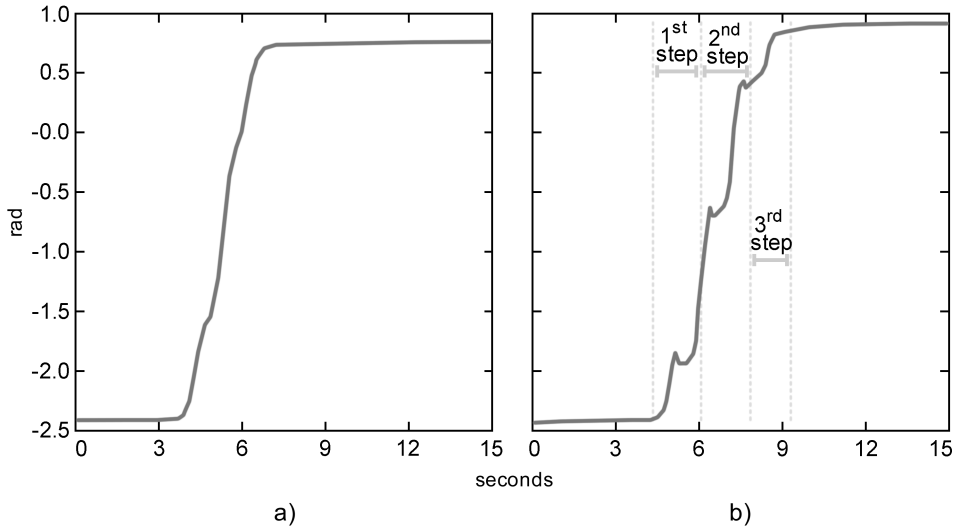


Figure 87: Orientation data obtained in Scenario A: a) data collected on the waist; b) data collected on the foot

directional movement than the waist, which can be seen in Figure 87, the standard deviation tends to be higher. It is higher because the initial and final orientations depends too much on how the foot is placed on the floor, at the beginning/end and during the walking path. This behaviour can be seen in Figure 87b where it is noticed the different steps given through the walking. The signals produced on waist (Figure 87a) are smoother.

In Scenario B the pedestrian starts and finishes the walking path at almost the same heading. The results have shown that both BSU locations provide good data on estimating this heading, where the error was very low. In Table 37 is presented the obtained results and in Figure 88 is presented the collected data.

Table 37: Results for the orientation algorithm for Scenario B

BSU location	**Mean orientation change** (rad)	**Standard deviation**	**Mean error**
Waist	0.02	0.02	0.65%
Foot	0.05	0.05	1.44%

The obtained results followed a similar pattern as the results obtained in Scenario A. However, the error, in both sensors location, is lower. From all the collected data it was noticed that as the walking path is smaller, the error tends to be higher than in longer walking paths. Comparing the Figure 88a with Figure 88b, it can be seen that the path is smoother in the first one and the second one presents too much orientation changes

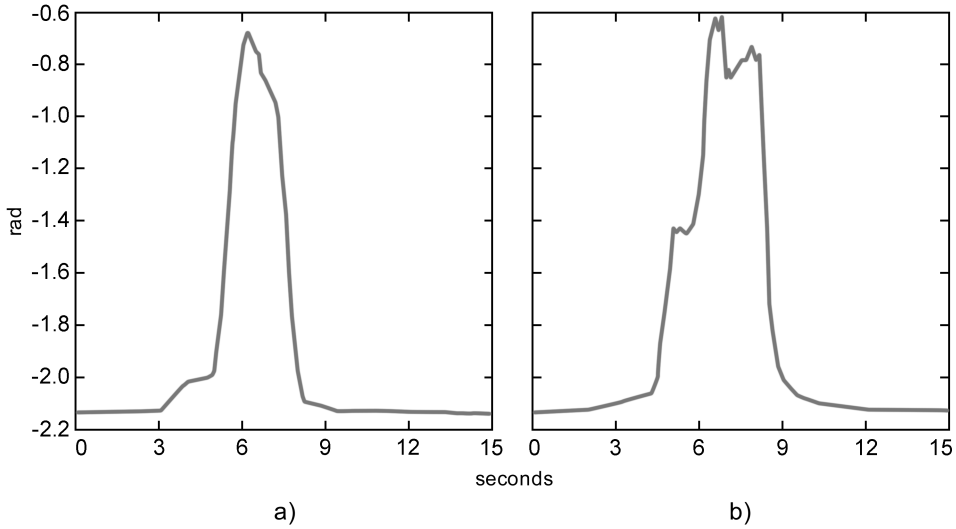


Figure 88: Orientation data obtained in Scenario B: a) data collected on the waist; b) data collected on the foot

during the two curves that exist during the walking path. With the BSU on the waist, exist two similar rotations, one to the left and another to the right.

Regarding the Scenario C, the obtained results are presented in Table 38 and in Figure 89 is presented the collected data in each one of the sensors location. The results are similar as in the last two scenarios.

As in the other scenarios it can be seen that the data obtained in the waist provides a smoother signal throughout the walking path. The data obtained with the BSU on the foot is more unstable. Analysing Figure 89 this phenomenon is clearly evident. At

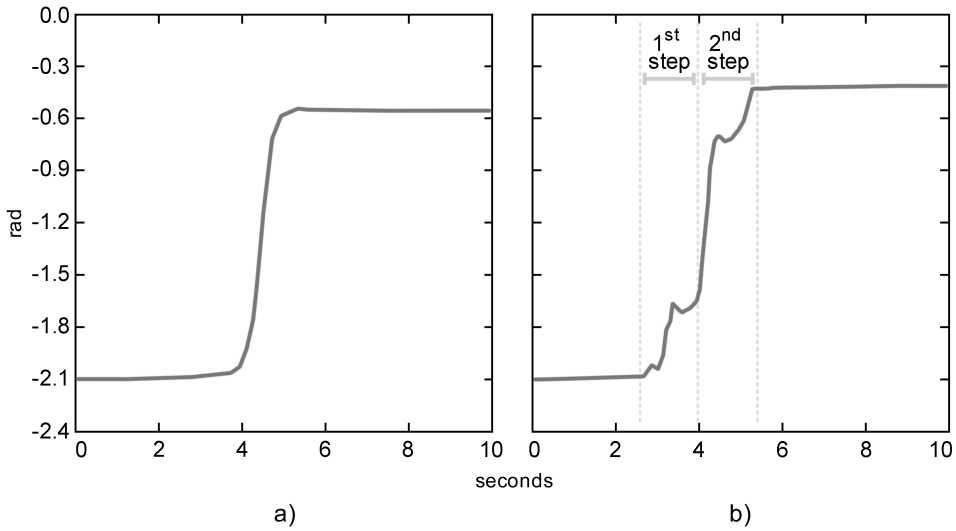


Figure 89: Orientation data obtained in Scenario C: a) data collected on the waist; b) data collected on the foot

Table 38: Results for the orientation algorithm for Scenario C

BSU location	**Mean orientation change** (rad)	**Standard deviation**	**Mean error**
Waist	1.57	0.05	0.30%
Foot	1.66	0.07	5.46%

each step, the orientation algorithm, running on the foot, has an unexpected behaviour because it depends on how the foot was placed on the floor when entering the stance phase. Thus, through the analysis of the signal it can be seen the two steps performed during the curve.

For the final scenario, Scenario D, the collected data is presented in Figure 90, and in Table 39 are presented the results obtained on the two sensors locations. This scenario has a longer walking path than the other two. The results are similar has the ones provided in the other scenarios. However, since this scenario involves more curves and a longer walking path the final error tends to be higher. The foot mounting point provided, almost, twice the error when compared to the waist mounting point.

Table 39: Results for the orientation algorithm for Scenario D

BSU location	**Mean orientation change** (rad)	**Standard deviation**	**Mean error**
Waist	1.66	0.06	5.12%
Foot	1.74	0.16	9.52%

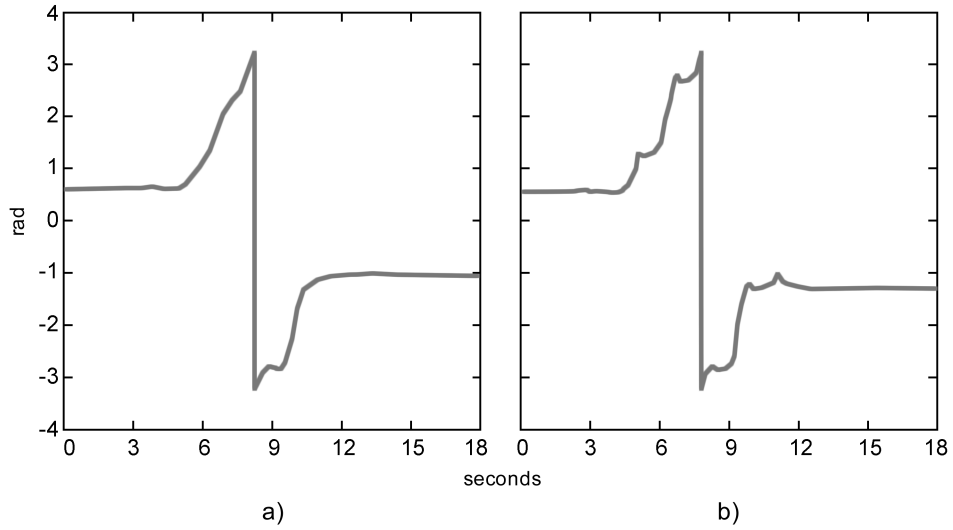


Figure 90: Orientation data obtained in Scenario D: a) data collected on the waist; b) data collected on the foot

Analysing the results obtained in all the scenarios in can be concluded that to estimate the orientation of a pedestrian the waist is a better mounting point than the foot. It is better because of three reasons: (i) the mean error tends to be always lower; (ii) the standard deviation is much better in the case of the waist, since it provides more stable results and does not depends too much on how the foot was placed on the ground when entering the stance phase; (iii) as can be observed from the figures, the data obtained in the waist provides a much more smoother signal, which leads to a better orientation estimation, since the signal is not floating too much during the walking path.

Thus, as could be seen through the obtained results, the waist is a better mounting point to run the orientation algorithm.

7.3.2 *Displacement*

In order to validate any new technology, architecture or methodology into a field of study it must be first compared and contrasted to existing methodologies. Thus a set of scenarios were created to validate and compare this thesis proposal with more traditional approaches. These scenarios are presented in Section 7.3.2.1 and the results obtained for each scenario are presented in Section 7.3.2.2.

The results of the different methodologies are compared where the accuracy is the primary interest.

7.3.2.1 *Use Case Scenarios*

The displacement estimation methodologies were tested and compared in four different use case scenarios, which can be visualized in Figure 57 and Figure 91. The building where the tests were conducted has a length of 20 meters and a width of 12 meters.

The scenarios are the following:

- Scenario A - straight line path on a flat surface, represented as A in Figure 57. It involves a distance of 20 meters, which gives an average of 35 steps in a normal walking pace;
- Scenario B - a straight line path, of about 15 meters, followed by two curves of about $\frac{\pi}{2}$ rad each in opposite directions, that makes the pedestrian to be in the same direction as in the start of the path, then it is performed another straight line path, of about 6 meters. This involves a walking distance of 25 meters, which

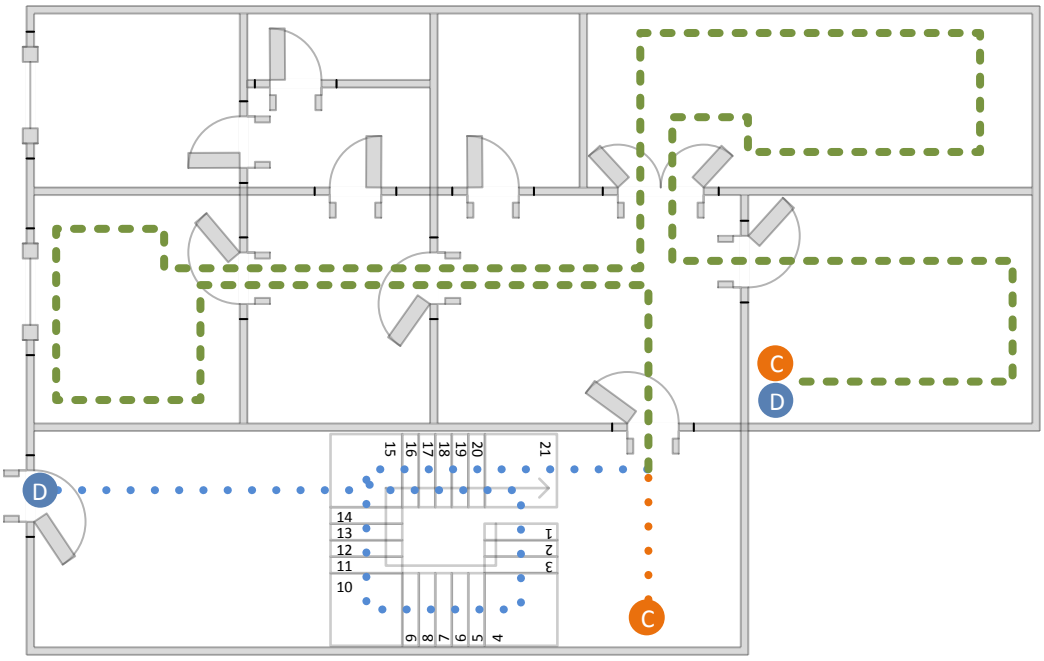


Figure 91: Evaluation Scenarios C and D: C) is a walking path around a building composed by 5 divisions; D) is composed by a straight walk followed by a stair climb and then the same walking path as scenario C

gives an average of 40 steps in a normal walking pace. This walking path is represented as *B* in Figure 57;

- Scenario C - this path is more complex than the previous two, and is represented as *C* in Figure 91. It involves a straight line of 6 meters, then a curve to the left which leads to another straight line of about 10 meters. Then a walk around the room is performed which covers 14 meters including 5 curves. Another straight line, with the length of 10 meters, is covered by the pedestrian and a left curve is made, where the pedestrian enters into another room. In that room the pedestrian walks around covering 22 meters, having 6 curves. Finally, the pedestrian enters in another room where he walks around which covers about 14 meters. This gives a total distance of 76 meters, which gives an average of 120 steps in a normal walking pace;
- Scenario D - in this path, represented as *D* in Figure 91, the pedestrian starts by performing a straight line of about 11 meters, then he climbs 21 stairs which has 3 sharp curves of about $\frac{\pi}{2}$ rad each. Each stair has a size of about 30cm. This upstairs represents an elevation of about 3.8 meters. Then he walks around the floor similarly as scenario 3. This scenario represents a total walking distance of about 90 meters, which gives an average of 160 steps in a normal walking pace.

7.3.2.2 *Experiment Results*

This section presents the results of the comparison between the different methodologies.

The evaluation was performed using three different approaches:

- Accelerometer (*M1*) - uses the foot BSU and the traditional PINS algorithm, which detects the steps only based on the accelerometer data and thereby estimates the distance walked by the pedestrian;
- Accelerometer and Force (*M2*) - is the algorithm presented in Section 7.2.1 which uses the accelerometer and the force sensors of the foot BSU to detect steps and thereby estimates the displacement;
- PLASYS (*M3*) - is the algorithm presented in Section 7.2.2, which uses both BSUs combined with the neural network that characterizes the step.

Having a side-by-side comparison of these three methodologies the goal is to validate the measurements of the proposed methodology (*M3*) against the reliability of the traditional methodology (*M1*) and point out any differences that may exist between the modalities. The algorithms are analyzed and contrasted with respect the performance, i.e. the achieved accuracy.

For each scenario and methodology it is presented the estimated walking path, where the trajectory points are marked with respect the digital map discussed in Section 7.3.2.1. Besides the map it is presented a table that gives information about the mean distance, the mean error and the standard deviation obtained for each methodology. Also, for each methodology it is presented a chart that shows the accumulated error through the walk way. The paths are represented with distinctive polygonal lines, where the blue dashed line represents the path obtained using the (*M1*) approach. The green dotted line represents the path obtained using the (*M2*) approach, and the red solid line represents the path obtained using the (*M3*) approach.

The results obtained for Scenario A are presented in Table 40. From these results it can be concluded that in a straight line the *M3* approach achieves a good accuracy. However, the standard deviation tends to be higher than the one obtained by the *M2* approach. As expected the *M1* approach has a higher error, which is almost twice the error obtained with the *M2* approach.

The accumulated error over the walked distance, for this scenario and for the three methodologies, can be seen in Figure 92. As mentioned in the state of the art, at the starts and stops during the walk is where the INS's accuracy is lower. These inaccuracies

Table 40: Results for the three different methodologies for Scenario A

Method	Mean distance (m)	Mean error	Standard deviation
M1 - Accelerometer	22.329	11.65%	0.53
M2 - Accelerometer and Force	21.312	6.56%	0.28
M3 - PLASYS	20.046	0.23%	0.42

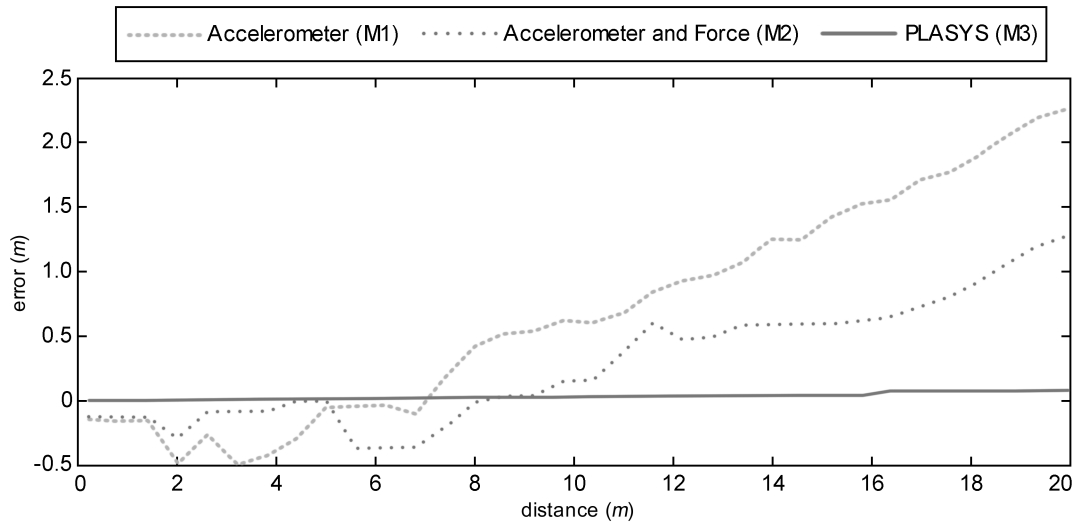


Figure 92: Accumulated error during the walk on scenario A

can be seen in Figure 92 at the beginning of the walk where the algorithms under-estimate the walked distance and the difference between the actual walking path and the estimated one varies a lot. When the walk velocity stabilizes the accumulated error growth rate is identical over time. The resulting walking path is demonstrated in Figure 93.

In Scenario B the three approaches achieved the results presented in Table 41. The mean errors and standard deviations are identical to the ones obtained for Scenario A. As the walking distance increases it is natural, in inertial navigation systems, that the accumulated error also increases.

From Figure 94 it can be seen that for the *M1* approach the distance error stabilizes or diminishes during the two curves. This leads into one of two hypothesis: during curves the walking path is better estimated by this approach, or it estimates incorrectly the distance walked during those curves. After analysing the obtained data, it was verified that *M1* approach under-estimates the distance walked by the pedestrian, thus

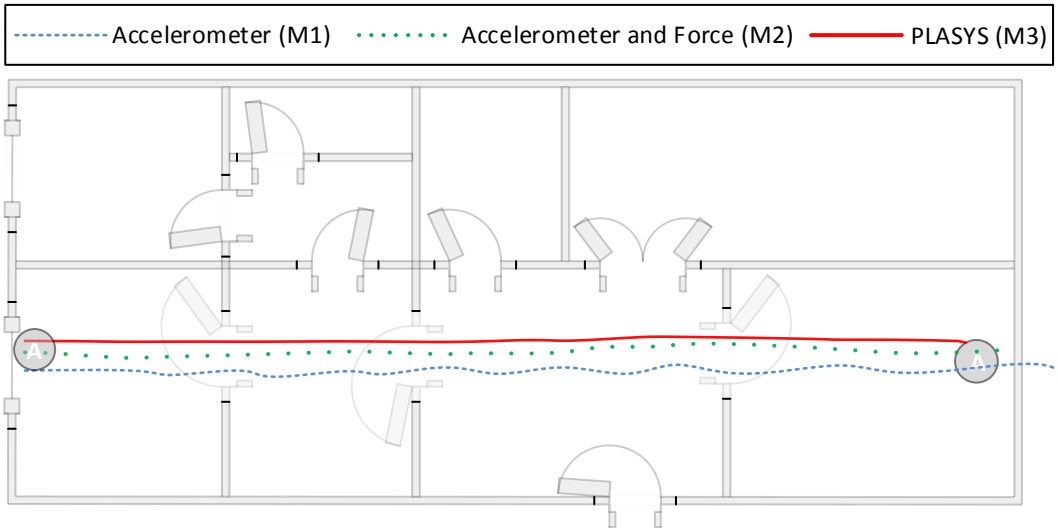


Figure 93: Estimated walking path for scenario A

Table 41: Results for the three different methodologies for Scenario B

Method	Mean distance (m)	Mean error	Standard deviation
M1 - Accelerometer	27.984	11.94%	0.63
M2 - Accelerometer and Force	26.501	6.00%	0.45
M3 - PLASYS	25.169	0.68%	0.32

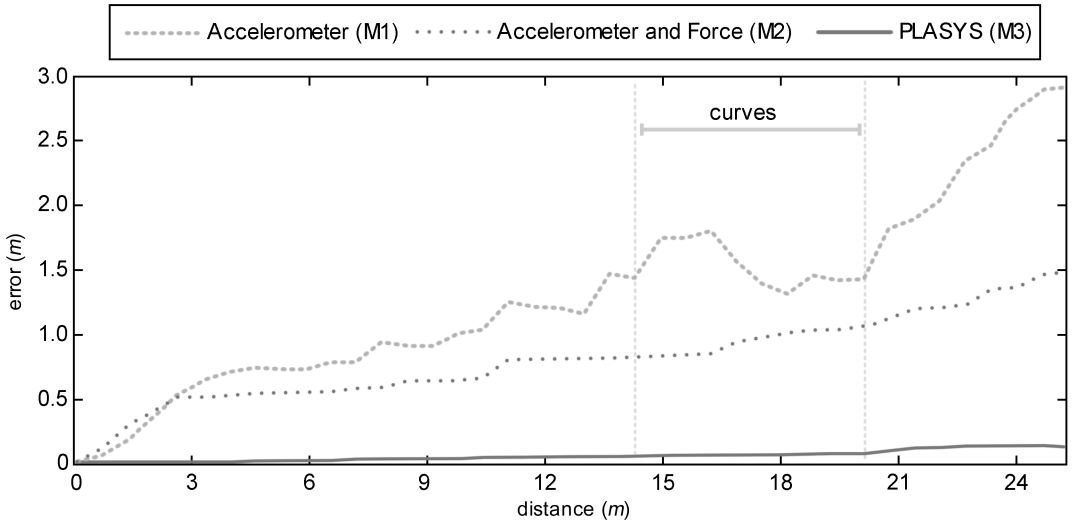


Figure 94: Accumulated error during the walk on scenario B

the error diminishes during those periods. However, these estimations tend to be closer to the real walked distance than when in a straight line.

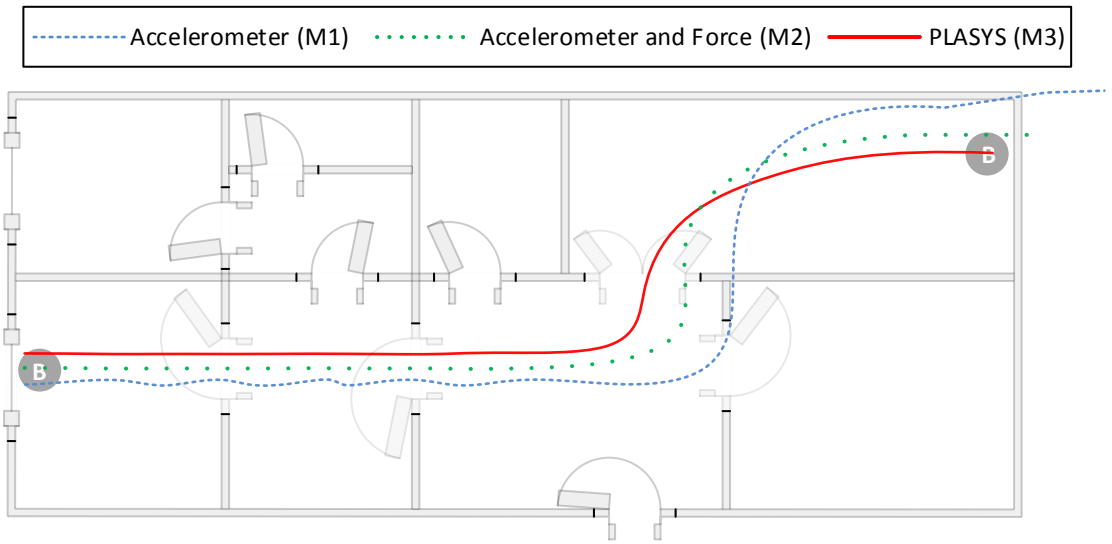


Figure 95: Estimated walking path for scenario B

For this scenario (B) the resulting estimated path is presented in Figure 95. As the learning algorithms correct the estimated distance based on the step characterization, the estimated path tends to be closer to the real one. Therefore the shape of the estimated path tends to be similar to the real one, although some deviations on the distance occur.

The results obtained for Scenario C can be seen in Table 42. This scenario involves a longer path and a lot of sharp curves. Thus, as expected, the mean error is higher than in the other scenarios. Compared to the other scenarios, in this one the error difference between the *M1* and the *M2* approaches is similar, i.e. the *M1* approach presents twice the error of the *M2* approach. However, the difference between the *M2* and the proposed one (*M3*) is lower than in the other scenarios. In the other scenarios this difference was of about 5%, but in this one the difference is only of 2.7%.

Table 42: Results for the three different methodologies for Scenario C

Method	**Mean distance (m)**	**Mean error**	**Standard deviation**
M1 - Accelerometer	87.122	14.63%	0.95
M2 - Accelerometer and Force	82.049	7.96%	1.21
M3 - PLASYS	79.982	5.24%	1.18

After analysing the collected data it was possible to verify that the *M3* approach, during the straights, has more accuracy that when the pedestrian performs some curves. This happens because the algorithm performs less corrections in curves than

on straights. The reason for this is that the distance estimations are under-estimated, thus the algorithm does not correct the data. To improve the results a new characterization to the step needs to be performed, in this case a curve must be detected to other variables be considered when estimating the distance.

To corroborate what has been referred, it can be seen in Figure 96 the distance error growth along the path. When sharp curves appear during the walk, the error tends to grow more than in the other zones of the path.

In this scenario the location error is higher and the difficulties on estimate correctly the walking path can be seen in Figure 97. Analysing the walking path obtained with the *M1* approach, it estimates poorly the pedestrian positioning, where the estimated walking path passes the limits of the room where the pedestrian was, or even the limits of the building. Regarding the orientation estimation, it presents fewer errors than the distance estimation.

Within respect Scenario D, the results that were obtained during the tests are presented in Table 43. Unlike the last scenario the mean error difference between the *M2* and *M3* approaches, has increased significantly, from 2.72% to 4.09%. Comparing these two approaches with *M1* approach, the mean error differences were very similar for all the scenarios, and this scenario was not an exception. However, compared to the other scenarios the standard deviation, in this one, was pretty similar for all the three approaches.

The main difference between Scenario C and Scenario D, is the presence of a set of stairwells at almost the beginning of the path. As seen in previous table the best results

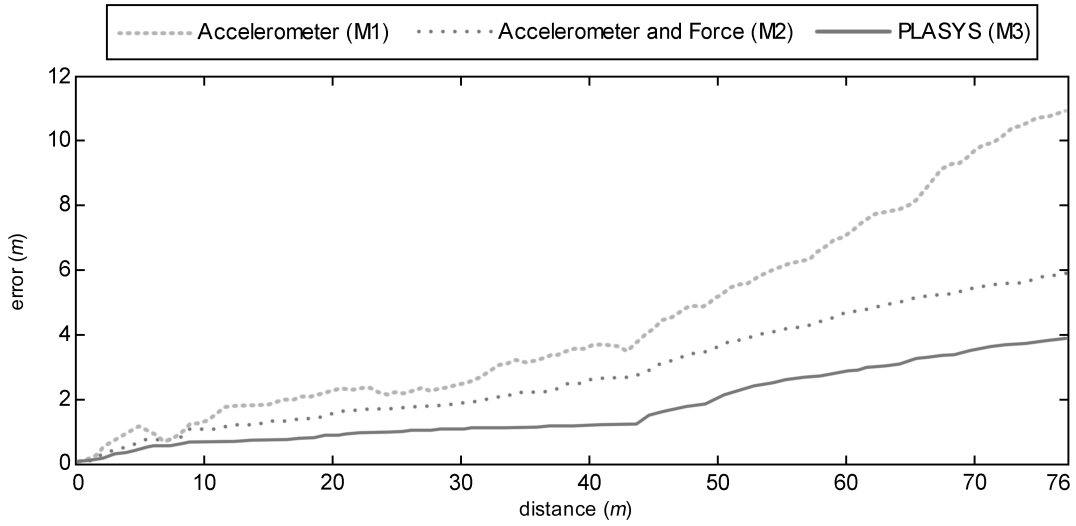


Figure 96: Accumulated error during the walk on scenario C

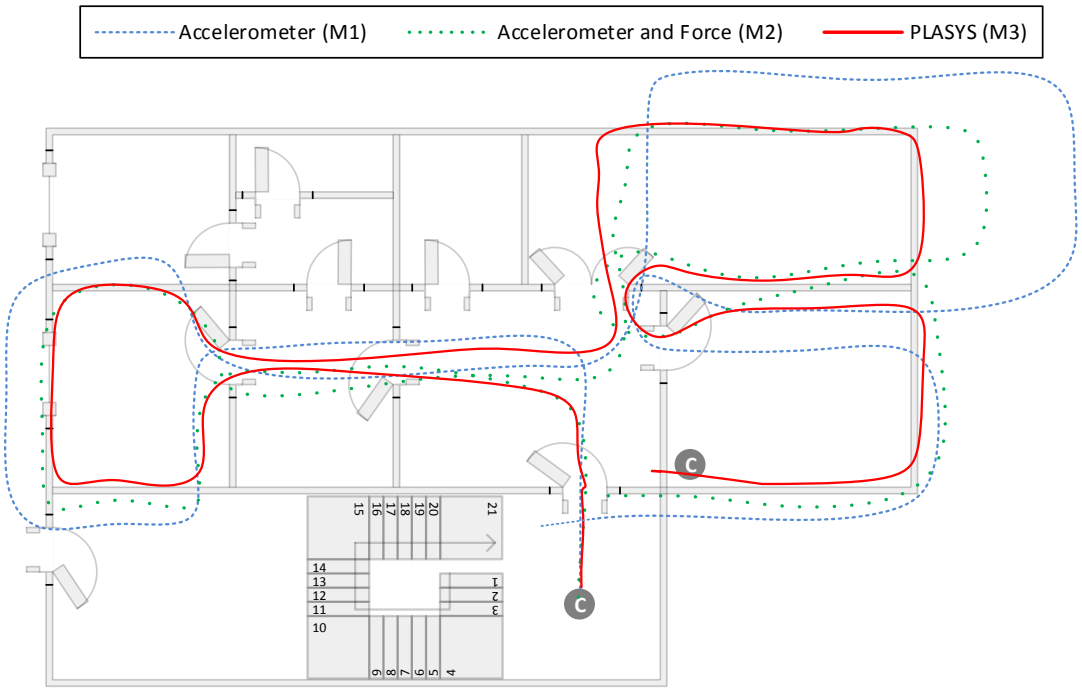


Figure 97: Estimated walking path for scenario C

Table 43: Results for the three different methodologies for Scenario D

Method	Mean distance (m)	Mean error	Standard deviation
M1 - Accelerometer	106.277	18.09%	1.02
M2 - Accelerometer and Force	99.793	10.68%	1.32
M3 - PLASYS	95.931	6.59%	1.19

were achieved with the proposed system (*M3*). For this scenario the best improvement that the *M3* approach introduces when compared to the others is the ability to reduce substantially the error growth during the upstairs.

From Figure 98, which presents the distance error growth during the walking path, it can be seen that when the pedestrian is going upstairs the error growth is almost imperceptible for the proposed system (*M3*). To contextualize, the stairs are in the walking path between the 10 and 17 meters. With the step characterization the distance estimations were very contained to the size of each stair that the pedestrian has climbed. During the rest of the path the results are similar to the ones of Scenario C, since the path is identical.

The resulting walking path is demonstrated in Figure 99. As the learning algorithms correct the estimated distance based on the step characterization, the estimated path,

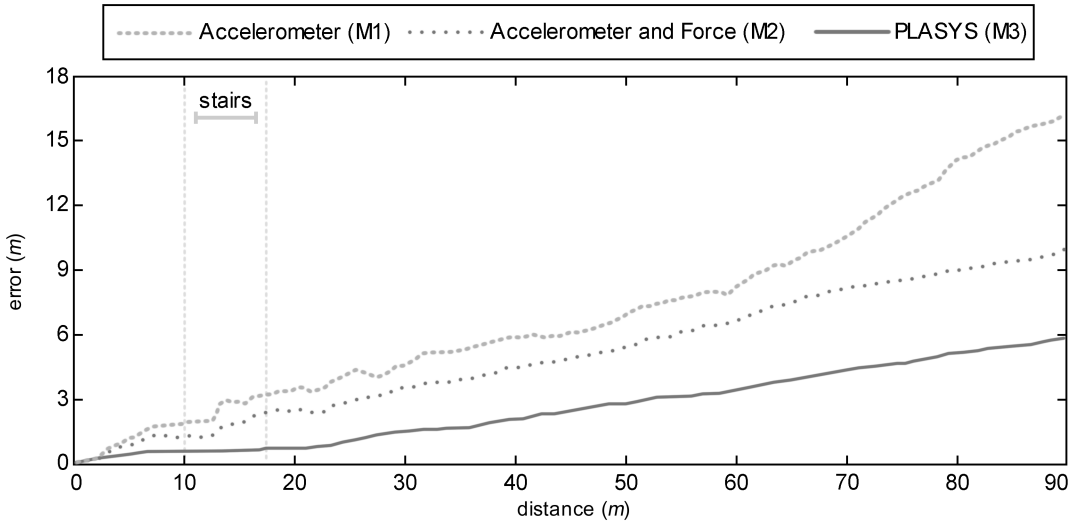


Figure 98: Accumulated error during the walk on scenario D

during the stairs and straight walks tends to be closer to the real one. Therefore the shape of the estimated path tends to be similar to the real one, although some deviations on the distance still exist. The *M1* and *M2* approaches tend to deviate a lot from the original path as the walked distance increases, specially regarding the distance estimation that typically is over-estimated rather than under-estimated. This leads to a higher estimated distance than the covered one.

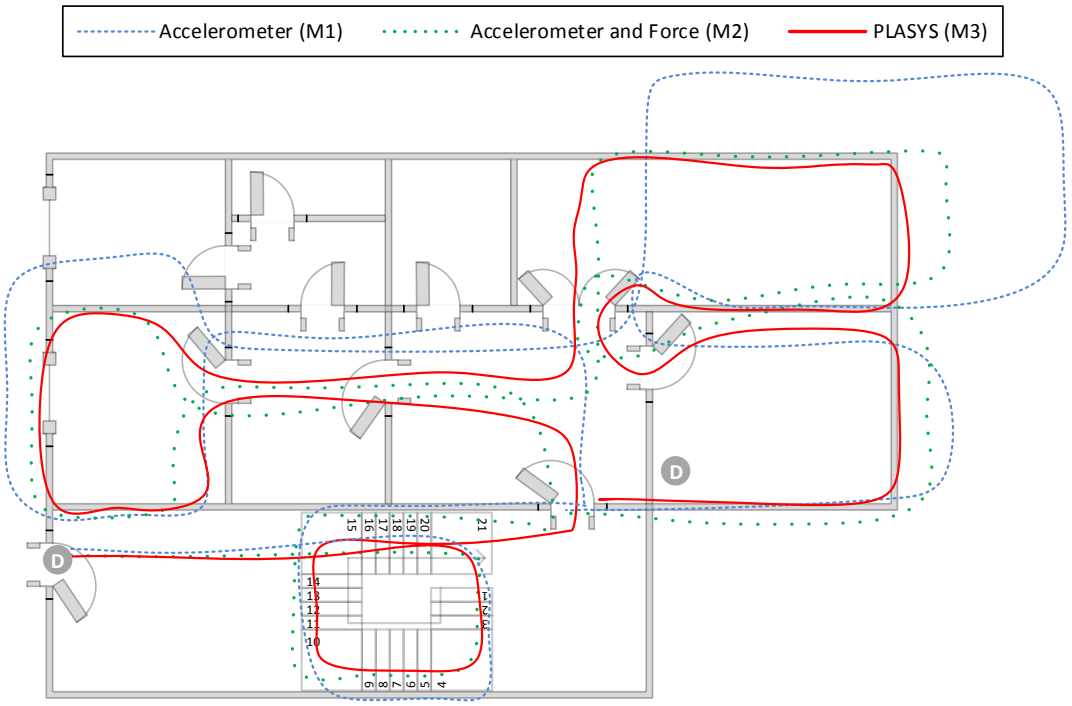


Figure 99: Estimated walking path for scenario D

Regarding the altitude estimation, since this test includes a set of stairwells that corresponded to an altitude change of 3.8 meters. The average final altitude estimation was of 4.719 meters, considering that on the ground floor is 0 meters. Thus, on elevation estimation the system (*M3*) presented an average error of 24%.

7.4 SUMMARY

Considering the knowledge obtained during the development of the state of the art, the orientation estimation is a research field very active which already has achieved good accuracy results in estimating the direction of a pedestrian. Thus, this work focused more on improving the displacement estimation.

To estimate the pedestrian orientation a complementary filter, which uses the information from an accelerometer, a gyroscope and a magnetometer, was implemented. The complementarity between these three sensors is important to have consistent and reliable orientation estimation results. The most important sensor is the magnetometer, since without it the orientation relative to an existent reference point does not exist. With this sensor it is always known the heading in relation with the earth's geomagnetic field. The other sensors are used to mitigate the errors that the magnetometer encounters when it is disturbed by other magnetic fields, which are normal to be encountered inside buildings, because of the existing power lines, computers, among others. Thus, the resulting algorithm proved to be robust and to achieve good results. Also, it tends to be easy to implement, and can run in mobile devices without great need for computational power.

During the system development it was also necessary to establish which is the best BSU to run the orientation algorithm: waist BSU or foot BSU. After some evaluations it was decided, due to the more accurate and consistent results, that the waist provides a better location to execute this algorithm. Thus the system is decomposed into two BSU that complement each other. One is primarily responsible to estimate the direction and the other to estimate the distance.

After estimating orientation, it is estimated the displacement of a pedestrian. This estimation was implemented using two approaches. Both, as already mentioned, uses the waist BSU data to estimate the orientation. The first approach, *Accelerometer and force*, uses the data from the foot BSU to estimate the displacement. The main difference between this methodology and the most commonly used is the inclusion of new sen-

sors (i.e. force sensors), which are important to detect more accurately the human gait phases.

Having a more accurate step detection leads to a more accurate step length estimation. Since the distance must be only considered during the time periods in which the foot is lifted from the ground. This phase is perfectly detected by the implemented algorithm. However, the inertial sensors still present some errors when detecting the acceleration that the foot performs during the step, which leads to some distance estimation inaccuracies. When compared to the *Accelerometer* approach [116], the displacement results were improved by 48% [11].

Although the results were improved with the addition of new sensors, they still lack of some accuracy and were not inside the bounders defined as the quantifiable success criteria for this project. Thus, it was performed the fusion between the two BSUs in order to characterize the step using a neural network. The neural network characterizes a step based on a previous set of steps performed by the pedestrian. Then, according to the given characterization different distance estimation algorithms are used. Besides using different algorithms to estimate the distance, this distance is corrected based on the range of values defined for each characterization type.

The usage of a more adequate algorithm, combined with the definition of the length intervals leads to a mean error of 0.23%, which is lower than the *Accelerometer and force* approach which was of 6.56%. However, when the path includes some curves the mean error of the *PLASYS* approach is 6.59% and the mean error of the *Accelerometer and force* approach is 10.68%, which results on an improvement of 37%. This walking path also involved different situations that are common inside a building. Thus, the obtained error matches the project quantifiable success criteria.

8

CONCLUSION

Throughout this document some conclusions were already being presented, which sustain the development options made in this thesis. Thus, this last chapter is a summary of the key findings, implications and relevance of the work done and future developments.

This thesis presents a PINS approach to estimate the displacement of a pedestrian. This approach combines the typical inertial sensors with force sensors, which were valuable to obtain a more accurate information about the human gait cycle phases. Also, the fusion of the information given by different BSU was important to implement a neural network that predicts the step characteristics in order to apply the most adequate displacement algorithms for each situation.

This approach, entitled PLASYS, is composed by four modules (see Chapter 4): (i) the data acquisition units, (ii) step detection algorithms, (iii) step characterization through machine learning and (iv) displacement estimation.

The first module (data acquisition units) is composed by two inertial units that are spread along the human body to collect the pedestrian movements and sends them to a mobile device through a wireless network (i.e. Bluetooth) (see Chapter 4).

The step detection module is tackled by combining the data given by different sensors (sensor fusion). The accelerometers data are combined with force sensors to detect (see Chapter 5) the different phases of the human gait cycle (see Section 2.1). The distinction between the different phases is important to improve the estimation capabilities of such a system.

Step characterization module introduces a characterization through machine learning algorithms and information fusion from different sources of data. The step is characterized based on three attributes: terrain, direction and length. Each attribute is previously learned from a set of exercises that will serve to train a neural network about the pedestrian walking characteristics when walking in different conditions. The learned data is then used to characterize the step in real-time.

The displacement estimation module, estimates the pedestrian displacement using the inertial units (first module) to detect the steps (second module) and corrects its estimations based on the step characterization (third module). A set of rules are applied to each type of conditions (characterizations) to then select the best estimation algorithm. This characterization is important to improve the results that are obtained with the displacement estimation algorithm.

Four different real use case scenarios were implemented to evaluate the PLASYS approach against the traditional methodologies, and to demonstrate the reliability and the achieved improvements on the displacement estimation results.

In this chapter it will be discuss the results obtained for each research questions presented in Chapter 1. Furthermore, an overview of the contributions (Section 8.1), and its limitations (Section 8.2), is presented along with a discussion about the future directions and work (Section 8.3).

8.1 ANALYSIS OF ACCOMPLISHMENTS

The overarching goal of this thesis is to investigate if:

The information obtained from inertial sensors, and additional sensors spread along the human body, can be fused and combined with learning algorithms to learn the human gait characteristics, to improve the pedestrian walking path estimation.

This goal raises several research questions and establishes multiple requirements.

RESEARCH QUESTION NO.1: *Can the inclusion of new type of sensors and techniques improve the step detection accuracy?*

To improve the step detection accuracy three sensors/techniques were implemented: (i) an accelerometer based step detection algorithm; (ii) a step detection algorithm based on fusing the data of accelerometer and force sensors; (iii) a step detection algorithm based on the pelvic rotation was implemented in the waist BSU, which uses the data provided by the gyroscope.

The step length estimation is highly correlated with the accurate detection of a step. Thus, one of the most important phases, and also the first, of a PINS algorithm is the step detection, which is then followed by stride determination and heading calculation.

The detection of a step enables the distinction between the swing and stance phases of the human gait. The relevance of this distinction is that the distance and the orientation must be only estimated/considered during the stance phase. Meaning that the velocity errors can be reset at each step, since the true velocity must be zero if the pedestrian is stationary.

Typically, a step is detected using an accelerometer placed in the pedestrian's foot. However, this approach presents too many false positive steps. Therefore force sensors where included in the BSU, which combined with the accelerometer data allowed to reduce, significantly, the step detection error [115]. The mean accuracy, given by this approach, was 3 times better than the accelerometer only approach. Besides improving the results, this implementation reduces the computational complexity, thus this objective was successfully achieved [116]. The only disadvantage is that it depends on more sensors.

The force sensors are also very important to perform the step characterization, since they measure the contact force of the foot with the ground. Typically, this force is correlated with the step length. Thus, the inclusion of force sensors enables a more precise and stable step detection algorithm in a long term use where different types of walking take place [11].

The inclusion of more than one BSU on human body enabled the implementation of a step detection algorithm in the waist BSU. The implemented algorithm considers the pelvic rotation in order to detect a step given by the pedestrian. The mean accuracy is lower than the foot approach. However, this algorithm is important to, when estimating the pedestrian location, merge with more precision the data from the two BSU.

The combination of the data from the two BSU is essential since each BSU as an important task in the system. The foot BSU presents better results on estimating the step length, and the waist BSU has better results on estimating the pedestrian heading. These two locations, and BSUs, were very important to achieve the obtained results [9].

After detecting a step, an important task to perform is to reduce the noise of the sensors. Thus, after the setup period, the offset calculated for each sensor is subtracted and the data is passed through a low-pass filter in order to simplify the algorithms that will process this data. Besides noise reduction, the improvements obtained on the step detection are important to apply effectively the ZUPT and ZARU techniques.

RESEARCH QUESTION NO.2: *Can learning algorithms be used to learn pedestrian step patterns and its errors in order to correct, in real-time, the inertial sensors estimations?*

The errors in PINS usually arise from three sources: (i) inertial sensors measurements, (ii) limitations in the navigation equations and (iii) initialization errors. The initialization errors were mitigated as presented in *Research Question no.1*.

The main goal in Machine Learning is to generalize [79], thus the inertial sensors measurements errors were mitigated using a step characterization through the use of a neural network. This neural network learns the walking characteristics in different types of terrain, different directions and different step lengths [13]. This characterization is important in order to limit the sensors measurements errors. This method can consider the non-linear characteristics and is not affected by the accelerometer bias and the acceleration of gravity. From several evaluations performed to different algorithms and approaches (i.e. Heuristics, DTW, SVM, Neural Network), the neural network based algorithm was the one that presented the best results. Mainly because they tend to generalize more. Thus, smaller differences in the inputs, from what was learned in the learning phase, are better classified by this algorithm.

The inherent systematic errors of the low-cost inertial sensors quickly accumulate and can lead to substantial position errors. The presented alternative is to use the sensors signals pattern rather than its value to characterize the step. With this characterization it is possible to adapt the behaviour of the displacement estimation algorithm according to the current context of the step. Thus, different parameters and algorithms can be used according to each characterization, as well as, different range of values that are used to limit the error growth given by the integration of the accelerometer data. This implementation mitigates the navigation equations limitations.

The neural network must be trained specifically with the data for each pedestrian. This data is collected from a set of exercises that the pedestrian must perform before using the system. Then, the system is able to perform real-time corrections with the learned data. The idea, in the beginning of this work, was to use the GPS data to create the dataset to train the learning algorithms. However, it was verified that the GPS chips included in the mobile devices are not accurate enough for this type of implementation. The errors of this hardware, that can achieve 5 meters, are too high to be used to train the algorithms, since a step length ranges between 25cm and 1 meter. Thus, a more accurate auxiliary system is necessary.

To improve the step characterization results, the data from the two BSU are used as input to the neural network. Thus, an information fusion architecture is implemented. With the data from two different sources, the probability that an error occur on the two units at the same time is very reduced, thus better results are achieved.

RESEARCH QUESTION NO.3: *Can a low-cost PINS provide good accuracy when used by a pedestrian?*

To respond to this question an IMU was developed to have more control on the sensors behaviours and to easily incorporate more sensors in the platform. Also, it was defined an INS architecture that combines the data from two IMU with learning algorithms to adapt the displacement algorithms according to the pedestrian walking characteristics.

A low-cost IMU has different definitions for each researcher, for some a thousand euros IMU is considered low-cost and for others a low-cost unit most cost a maximum of €500, it hugely depends on the economic situation of each person. In this work it was developed an IMU that uses MEMS sensors that are truly low-cost, with a price rounding €10 or less. Thus, the developed unit has an estimated cost of approximately €50, for the hardware. As the price of the MEMS sensors is lower, they also tend to give more inaccuracies than the more expensive sensors available in the market.

The development of a single unit can be very difficult, and since the system is not running on a stable platform, it sometimes presents some errors that are derived from it being an early stage prototype. Thus, only the foot BSU was produced. The waist BSU was simulated using an Android mobile device.

Both BSUs are simple to attach to the pedestrian's body and to carry around due to the lightweight and the small form factor. The prototype unit is easy to carry on a belt or in the person's leg. The communication between them is easy to configure, since this communication is based on the Bluetooth wireless network, which compared to the Zigbee is largely available on mobile devices [4].

The sensors of each BSU can give data rates of more than 200Hz. However, the sampling ODR was decreased to 20Hz because of two reasons: battery life and performance. The first one is important, since if the system consumes too much energy, it will be only usable for a couple of hours, which makes it impractical to be used by a pedestrian, for example, in a museum visit. The second reason has a huge importance, since, if more data is obtained more computational power is required to process it and therefore more energy is needed [7]. In order to maintain the low-price, a smartphone is used to handle the displacement estimations. If more computing power is required more difficult will be to the mobile device run the algorithms, due to its inherent characteristics, being unable to provide a result in an acceptable execution time [8]. One advantage of using this type of device is that it is normally used by the pedestrians, thus it is not needed a new hardware.

The developed BSU, which is a prototype and not a final version, is relatively light and small to be used in the two places, foot and waist. Thus, during a normal walking around a building it does not influences the walking behaviour of the pedestrian.

Every time the system is used, it is needed a calibration phase where the user must be standstill for five seconds in order to calculate the sensors offset.

Displacement is estimated based on distance and orientation. Thus, an orientation algorithm was implemented. This algorithm is based on a complementary filter that fuses the data of an accelerometer, a gyroscope and a magnetometer. These sensors complement each other in order to improve the quality of the data. This approach proved to be robust and achieved good results. Also, it tends to be easy to implement and run in mobile devices without high computational power requirements.

MAIN CONTRIBUTIONS

This PhD work resulted in several contributions to the State of the Art in the field of Pedestrian Inertial Navigation Systems. Several Artificial Intelligence methods were used to mitigate the most significant errors of MEMS sensors. This, when combined with multiple sources of information, results in a more accurate PINS. Thus, the main contributions of this work can be compiled as follows:

- State of the art analysis, including the main techniques used in this type of systems (Chapter 2), as well as, the most prominent systems in this research field (Chapter 3);
- Advantage of using force sensors combined with inertial sensors to detect, with more precision, the steps given by a pedestrian (Chapter 5). To identify the advantages of using different techniques and locations to run the algorithms, different step detection algorithms were proposed and evaluated. Also, it was proven that several sensors spread in the pedestrian's body provide more reliable information since the errors of one sensor can be mitigated with the data of other sensor (Chapter 4);
- Applicability of machine learning in PINS. The step characterization using a neural network is important to identify the different walking patterns. Thus, different INS techniques can be used according to the current gait behaviour, which leads to a better displacement estimation (Chapter 6);
- A low-cost inertial navigation system prototype to be used by pedestrians, which can be carried around to estimate with good accuracy the pedestrian displace-

ment. It were also proposed and developed different distance estimation algorithms to be used according to the walking behaviour. To complement the distance estimation, an orientation algorithm based on a complementary filter was implemented (Chapter 7);

- Case studies, in contribution with other researchers, that demonstrate the capacity of using such a system in several different ambient intelligence areas, such as ambient assisted living [98] [97], recommendation systems [74], mobile tourism applications [10], shopping [5], among others;
- Several evaluations were made to identify the best practices to be used on mobile devices development [8], specially regarding their performance [7], which can be very low [6].

SUCCESS CRITERIA

Regarding the results obtained from the implemented architecture, it must be said that having a more accurate step detection leads to a more accurate step length estimation, since the distance must be only considered during the time moments in which the foot is lifted from the ground (swing phase). When compared to the typically used PINS methodology [116], the displacement results were improved by about 48% [11], without considering the step characterization.

With the step characterization, which enables the usage of different distance estimation algorithms for each situation, the mean error on straights decreased from 6.56%, for the *Accelerometer and Force* approach, to 0.23% for the *PLASYS* approach. When the walking path includes some curves the error has decreased from 10.68%, for the *Accelerometer and Force* approach, to 6.59% for the *PLASYS* approach, which results on an improvement of 37%.

The mean error of 6.59% was achieved on the experiment with the longest path, which also involved different situations that are commonly encountered inside a building [12]. Thus, with the obtained error, the proposed system has an accuracy of 93%, which matches the project accuracy quantifiable success criteria that was defined as a value higher than 90%.

During the evaluations it was found that the magnitude of the error is not dependent on time since when the user is not moving, because of the step detection algorithm, the distance remains constant, as expected.

The results of the proposed thesis are not compared with other system presented in the state of the art, because does not exist a standard test to be done to this type of systems. Thus, the comparison that was performed was to the typically used algorithms that uses only a foot IMU and the accelerometer data to detect a step and, therefore, to estimate the distance. A gyroscope is also needed in this approach to transform the acceleration data from the inertial frame to the navigation frame. By conducting this comparison it was insured that the same hardware and scenarios were used, and therefore the results could be compared.

Regarding the delay between the real location and the processed one, the implemented algorithms, besides their performance, were also chosen because of their lightness. Thus, after testing the implemented algorithms it was concluded that a high-end mobile device takes about 2 seconds to handle the necessary estimations. This time period is similar to the one obtained when using the GPS module incorporated on mobile devices, which typically has a frequency of 1 second.

The implemented system also shows that it is a valid approach to be used in the ambient assisted applications that were presented in Chapter 1, like tourism recommendation applications [10], in shopping centers [5] or by the elderly [98].

The results of the proposed system are very encouraging for this line of research and had shown the viability of this methodology by achieving, with success, the two success criteria (i.e. location accuracy and location estimation delay) defined in the beginning of this work [12].

8.2 RESEARCH LIMITATIONS

There are some limitations in the current PLASYS approach, which are mainly related to the hardware, learning algorithms and displacement estimation.

Regarding the hardware there are two limitations: (i) the quality of the MEMS inertial sensors used is, still, conspicuously low, (ii) and the battery used in the developed prototype sometimes influences the data collected from the sensors.

With better sensors more promising results would be achieved. The sensors that had presented the most problems are the accelerometer and the force sensors. The force sensors that were used have a small sensing area, which makes their placement on the pedestrian foot very tricky. Currently, this correct placement can take some time to be found. Having a larger contact area with the foot would, also, provide more consistent results on detecting the different gait cycle phases.

The prototype battery, which weights too much, sometimes influences the data obtained from the sensors, since it moves from its position and touches the BSU. By touching the BSU some accelerations are detected, which has some impact on the obtained results.

Concerning the learning algorithms it were identified two limitations: (i) needs a specific dataset for each pedestrian; (ii) some human movements are not considered and need to be modelled.

Currently, the neural network needs a specific dataset for the pedestrian that will use the system. The system only gives good results if the same user dataset is used. If the dataset of another pedestrian is used the quality of the results decreases. Also, the step length characterization classes are being defined manually, where each range is being set specifically for each pedestrian.

There is the need to model some human movements that are not consider, like forward/backward jumps, among other situations. The step characterization needs further improvements in order to consider these not so natural movements when walking. This work focused on the more typical movements that are performed during a walk.

With respect to the displacement estimation, there are two limitations: (i) the system does not consider automatic transportation facilities; (ii) synchronization between the BSUs lacks sometimes of some accuracy.

The system does not work in elevators, escalators and other automatic transportation facilities. Thus, besides implementing the step characterization, another characterization must be performed like an environment characterization which detects these types of situations.

Synchronization between the BSUs needs to be further improved with more accurate synchronization methods. Since, sometimes the data is not properly aligned, which leads to poor results.

Analysing these limitations it can be concluded that the base for the proposed localization system is made, but some refinements need to be done in the current working prototype.

8.3 FUTURE WORK

Throughout this document, it has been shown that PINS provides a great advantage for people navigating on foot. Thus, it was proposed an original approach that charac-

terizes the step based on a neural network, which is then used to correct the inertial sensors deviations. Besides the more accurate results obtained with this methodology, deviations from the actual walking path still exist. Therefore some efforts need to be done to improve the system results. This section is divided into two parts: (i) it will be presented additional implementations that can be done to improve the current system, (ii) and it will be suggested some new approaches that can be followed to improve the PINS accuracy.

Additionally to the evolution of the proposed method, with the correction of the presented *Research Limitations*, other methodologies could be implemented in order to increase the system performance, such as:

- More characterizations about the pedestrian movements must be implemented. From the collected data it was possible to identify that it is needed to characterize the step when a curve exists during the path. Certainly this modulation will improve the system's accuracy as the other characterizations did. Thus, at the moment the most important characterizations are: starts and stops, curves and environment (e.g. elevator, escalator). However, there are other characterizations that are important to perform such as: terrain inclination, crowed area or not, among others. Also, it will be interesting to divide the step length characterization into more classes to verify if it improves the displacement estimation accuracy;
- To discard the necessity of having a predefined set of exercises to learn the gait characteristics, it could exist a set of predefined datasets to be used by each pedestrian. Thus, the same dataset could be used by similar persons. The profile could be based on the person's characteristics (e.g. height, weight, gender, age) or going even further by making a gait profile (i.e. with identical patterns), since two similar, in age, height and weight, pedestrians could have different gait patterns. Then the dataset that is closer to the current user would be selected;
- If a similar neural network to the one that corrects the data given by the distance algorithm, is applied to the data estimated by the orientation algorithm, the estimated data could be more precise. Having better results on estimating the orientation will lead to better results when estimating the displacement;
- Altitude estimation needs further research. Right now it is estimated but it lacks some accuracy, which is also influenced by the non-existence of a map. A fusion algorithm between the vertical acceleration movement made by the pedestrian and the barometer data needs to be implemented.

It was clear from the beginning of this PhD work that it would be a multidisciplinary one, drawing mostly from Computer Science but also from Human Kinematics. From the technological point of view, it also merged many different fields, including Data Acquisition, Wireless Sensor Networks, Sensor Fusion, Learning Algorithms, Information Fusion, among others. In that sense, it is only natural that at the closure of this endeavour, a multitude of possible future paths exists, each one following a different line of research under the different fields addressed. Several interesting research problems that could be followed after the conclusion of this thesis can be pointed out:

- The inclusion of a heart rate sensor in the chest area will help to known what is the type of activity that the user is performing (e.g. walking, running). This could be one more source of data to be used, in combination with the other sensors data, to improve the step characterization. As seen during this work, having more sources of information is important to achieve better results since the errors from some sensors could be surpassed with the data from the other sensors. The heart rate sensor can improve the system precision, since when a person is walking faster the heartbeat is higher than when a person is only standing-up. Of course that these conditions only work, or are trusted, if at least two sources of data give the same information;
- Another approach that could be used to improve the step length estimation is using a muscle sensor kit on the pedestrian legs to measure the muscle activity by detecting its electric potential, referred to as ElectroMyoGraphy (EMG). These sensors are small, however they are much more intrusive than inertial sensors. The muscle activity could bring important information about the type of force that will be applied on the leg to perform the movement, thus it could be another important source of data to help on the estimation of the step length;
- The step characterization could be pushed even further by using a social architecture. This social architecture could associate a context to a map. The map, by gathering the data from the several pedestrians that could pass on a specific region, can be contextualized (e.g. type of terrain, if it is a crowded area or not) with the information obtained from the developed system. This social context map can be elaborated with the data obtained from all users in the system, which can be categorized based on their steps and on the similarities between them. With this contextual map, the correct learning algorithm could be chosen by the system. Thus, the system would only use small portions of the dataset, instead of using the whole dataset, improving the system responsiveness and results.

- In this research area, and also in other INS areas, emerges the necessity of the existence of a standard to evaluate the systems. Having a standard is valuable to compare the different methodologies proposed by the researchers. This could be a difficult task to execute. However, it will be important to identify the real differences between systems. Without a standard it is almost impossible to compare the different systems since each case study scenario is different, and the results could not be compared;
- To perform the step characterization other machine learning approaches can be considered (e.g. a decision tree). A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, each leaf node can "test" an attribute (e.g. if it is a forward step), and each branch represents the outcome of the test which will lead to a class label (decision taken after computing all attributes);
- Complement the system proposed by Carneiro et al. [24], that measure the levels of acute stress in humans by analysing their behavioral patterns when interacting with technological devices. With the proposed system more features (e.g. walking behaviour) can be used as input to the Carneiro system to determine which behaviours show statistically significant differences, for each user, when under stress or fatigue. Moreover, with the PLASYS system the process of estimating stress is still taken in a non-intrusive way. The main objective is to overcome some of the main drawbacks of communicating online, in dispute resolution, namely the lack of contextual information such as body language or gestures.

Additional research lines could eventually be put forward. The ones listed here have in common the fact that they are already known topics, although they are still open research problems. This may make them valid approaches with potential for success.

Certainly, many other questions were raised and left open for further research. The contribution of this thesis is a little bump in future directions that can be followed by the Pedestrian Inertial Navigation Systems. It opens the door to the combination of different sources of information using machine learning techniques that characterize the type of environment to correct the errors given by each source of data which could lead to further developments with high social and economical impact.

BIBLIOGRAPHY

[1] AKM. AK8963 3-axis electronic compass IC, 2014. URL http://www.akm.com/akm/en/product/datasheet1/?partno=AK8963.

[2] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, 2014. ISBN 9780262028189.

[3] Ricardo Anacleto, Lino Figueiredo, Nuno Luz, Ana Almeida, and Paulo Novais. Recommendation and planning through mobile devices in tourism context. In *Ambient Intelligence - Software and Applications*, volume 92 of *Advances in Intelligent and Soft Computing*, pages 133–140. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-19936-3. doi: 10.1007/978-3-642-19937-0_17.

[4] Ricardo Anacleto, Lino Figueiredo, Paulo Novais, and Ana Almeida. Providing location everywhere. In *Progress in Artificial Intelligence*, volume 7026 of *Lecture Notes in Computer Science*, pages 15–28. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-24768-2. doi: 10.1007/978-3-642-24769-9_2.

[5] Ricardo Anacleto, Nuno Luz, Ana Almeida, Lino Figueiredo, and Paulo Novais. Shopping center tracking and recommendation systems. In *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*, number 87 in Advances in Intelligent and Soft Computing, pages 299 – 308. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-19643-0. doi: 10.1007/978-3-642-19644-7_32.

[6] Ricardo Anacleto, Lino Figueiredo, Ana Almeida, and Paulo Novais. Server to mobile device communication: A case study. In *Ambient Intelligence - Software and Applications*, volume 219 of *Advances in Intelligent Systems and Computing*, pages 79–86. Springer International Publishing, 2013. ISBN 978-3-319-00565-2. doi: 10.1007/978-3-319-00566-9_11.

[7] Ricardo Anacleto, Lino Figueiredo, Ana Almeida, and Paulo Novais. Transfering data from a server to an android mobile application: A case study. *Jurnal Teknologi*, 63(3):85–91, 2013. doi: 10.11113/jt.v63.1959.

[8] Ricardo Anacleto, Nuno Luz, Ana Almeida, Lino Figueiredo, and Paulo Novais. Creating and optimizing client-server applications on mobile devices. In *Ambient*

Intelligence and Smart Environments, volume 17, pages 318–329. IOS Press, 2013. ISBN 978-1-61499-285-1. doi: 10.3233/978-1-61499-286-8-318.

[9] Ricardo Anacleto, Lino Figueiredo, Ana Almeida, and Paulo Novais. Localization system for pedestrians based on sensor and information fusion. In *17th International Conference on Information Fusion (FUSION'14)*, pages 1–8, July 2014.

[10] Ricardo Anacleto, Lino Figueiredo, Ana Almeida, and Paulo Novais. Mobile application to provide personalized sightseeing tours. *Journal of Network and Computer Applications*, 41:56 – 64, 2014. ISSN 1084-8045. doi: 10.1016/j.jnca.2013.10.005.

[11] Ricardo Anacleto, Lino Figueiredo, Ana Almeida, and Paulo Novais. Person localization using sensor information fusion. In *Ambient Intelligence - Software and Applications*, volume 291 of *Advances in Intelligent Systems and Computing*, pages 53–61. Springer International Publishing, 2014. ISBN 978-3-319-07595-2. doi: 10.1007/978-3-319-07596-9_6.

[12] Ricardo Anacleto, Lino Figueiredo, Ana Almeida, Paulo Novais, and António Meireles. Step count and classification using sensor information fusion. In *Ambient Intelligence - Software and Applications*, volume 376 of *Advances in Intelligent Systems and Computing*, pages 85–93. Springer International Publishing, 2015. ISBN 978-3-319-19694-7. doi: 10.1007/978-3-319-19695-4_9.

[13] Ricardo Anacleto, Lino Figueiredo, Ana Almeida, Paulo Novais, and António Meireles. Pedestrian inertial navigation system with terrain characterization using information fusion. In *Proceedings of the 6th International Conference on Security-enriched Urban Computing and Smart Grid*, SUComS'15, 2015.

[14] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. ISSN 1053-587X. doi: 10.1109/78.978374.

[15] Christian Ascher, Lukasz Zwirello, Thomas Zwick, and Gert Trommer. Integrity monitoring for UWB/INS tightly coupled pedestrian indoor scenarios. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1 – 6, 2011. doi: 10.1109/IPIN.2011.6071948.

[16] Ed Ayyappa. Normal human locomotion, part 1: Basic concepts and terminology. *Journal of Prosthetics and Orthotics*, 9(1):10 – 17, 1997.

[17] Yaakov Bar-Shalom, Xiao Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, Inc., New York, USA, 2001. ISBN 9780471416555. doi: 10.1002/0471221279.

[18] Stéphane Beauregard. A helmet-mounted pedestrian dead reckoning system. In *3rd International Forum on Applied Wearable Computing (IFAWC)*, pages 1 – 11, 2006. ISBN 978-3-8007-2954-8.

[19] Stéphane Beauregard and Harald Haas. Pedestrian dead reckoning: A basis for personal positioning. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, pages 27 – 35, 2006.

[20] Özkan Bebek, Michael A. Suster, Srihari Rajgopal, Michael J. Fu, Xuemei Huang, Murat C. Çavuşoğlu, Darrin J. Young, Mehran Mehregany, Antonie J. van den Bogert, and Carlos H. Mastrangelo. Personal navigation via high-resolution gait-corrected inertial measurement units. *IEEE Transactions on Instrumentation and Measurement*, 59(11):3018 – 3027, 2010. ISSN 0018-9456.

[21] Hamza Benzerrouk, Alexander Nebylov, and Pau Closas. MEMS IMU/ZUPT Based Cubature Kalman Filter Applied to Pedestrian Navigation System. page 7. MDPI, 2014. doi: 10.3390/ecsa-1-e002.

[22] Deepak Bhatt, Priyanka Aggarwal, Vijay Devabhaktuni, and Prabir Bhattacharya. A novel hybrid fusion algorithm to bridge the period of GPS outages using low-cost INS. *Expert Systems with Applications*, 41(5):2166–2173, 2014. ISSN 0957-4174. doi: 10.1016/j.eswa.2013.09.015.

[23] Bosch. Bosch sensortec bmp 280 barometric pressure sensor, 2014. URL `https://www.bosch-sensortec.com/en/homepage/products_3/environmental_sensors_1/bmp280/bmp280`.

[24] Davide Carneiro, José Castillo, Paulo Novais, Antonio Fernández-Caballero, and José Neves. Multimodal behavioral analysis for non-invasive stress detection. *Expert Systems with Applications*, 39(18):13376–13389, 2012. ISSN 0957-4174. doi: 10.1016/j.eswa.2012.05.065.

[25] Francois Caron, Emmanuel Duflos, Denis Pomorski, and Philippe Vanheeghe. GPS/IMU Data Fusion Using Multisensor Kalman Filtering: Introduction of Contextual Aspects. *Information Fusion*, 7(2):221–230, 2006. ISSN 1566-2535. doi: 10.1016/j.inffus.2004.07.002.

[26] Nadir Castaneda and Sylvie Lamy-Perbal. An improved shoe-mounted inertial navigation system. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1 – 6, 2010. doi: 10.1109/IPIN.2010.5646858.

[27] Ning Chang, Rashid Rashidzadeh, and Majid Ahmadi. Robust indoor positioning using differential wi-fi access points. *IEEE Transactions on Consumer Electronics*, 56(3):1860 – 1867, 2010. ISSN 0098-3063. doi: 10.1109/TCE.2010.5606338.

[28] Wei Chen, Zhongqian Fu, Ruizhi Chen, Yuwei Chen, Octavian Andrei, Tuomo Kröger, and Jianyu Wang. An integrated GPS and multi-sensor pedestrian positioning system for 3D urban navigation. In *Urban Remote Sensing Event, 2009 Joint*, pages 1 – 6, 2009. doi: 10.1109/URS.2009.5137690.

[29] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata Padmanabhan. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 173 – 184, New York, USA, 2010. ACM. ISBN 978-1-4503-0181-7. doi: 10.1145/1859995.1860016.

[30] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-78019-5.

[31] CSR. Bluecore4-ext, 2014. URL `http://www.csr.com/products/29/bluecore4-ext`.

[32] Analog Devices. ADXL345 digital accelerometer, 2014. URL `http://www.analog.com/static/imported-files/data_sheets/ADXL345.pdf`.

[33] Estefania Munoz Diaz, Ana Luz Mendiguchia Gonzalez, and Fabian de Ponte Müller. Standalone inertial pocket navigation system. In *IEEE/ION Position, Location and Navigation Symposium - PLANS*, pages 241 – 251, 2014. doi: 10.1109/PLANS.2014.6851382.

[34] Naser El-Sheimy, Kai-wei Chiang, and Aboelmagd Noureldin. The Utilization of Artificial Neural Networks for Multisensor System Integration in Navigation and Positioning Instruments. *IEEE Transactions on Instrumentation and Measurement*, 55 (5):1606–1615, 2006. ISSN 0018-9456. doi: 10.1109/TIM.2006.881033.

[35] John Elwell. Inertial navigation for the urban warrior. In *Proceedings of SPIE*, volume 3709, pages 196 – 204, 1999. doi: 10.1117/12.351609.

[36] Katti Faceli, André de Carvalho, and Solange O. Rezende. Combining Intelligent Techniques for Sensor Fusion. *Applied Intelligence*, 20(3):199–213, 2004. ISSN 0924-669X, 1573-7497. doi: 10.1023/B:APIN.0000021413.05467.20.

[37] Filipe Felisberto, Nuno Moreira, Isabel Marcelino, Florentino Riverola, and António Pereira. Elder care's fall detection system. *Ambient Intelligence - Software and Applications (2nd International Symposium on Ambient Intelligence (ISAmI 2011)), Springer - Series Advances in Intelligent and Soft Computing*, pages 85 – 92, 2011. doi: 10.1007/978-3-642-19937-0_11.

[38] Raul Feliz, Eduardo Zalama, and Jaime Gómez. Pedestrian tracking using inertial sensors. *Journal of Physical Agents*, 3(1):35 – 42, 2009. ISSN 1888-0258.

[39] Hélder Ferreira, Lino Figueiredo, and António Meireles. INPERLYS - independent personal location system. *Ambient Intelligence and Smart Environments, volume 5, Ambient Intelligence Perspectives II, Edited by P. Cech, V. Bures, L. Nerudova*, pages 93 – 100, 2009. doi: 10.3233/978-1-60750-481-8-93.

[40] Maria Feychting, Anders Ahlbom, and Leeka Kheifets. EMF and health. *Annual Review of Public Health*, 26:165 – 189, 2005. doi: 10.1146/annurev.publhealth.26.021304.144445.

[41] Lino Figueiredo, Isabel Jesus, José Machado, José Ferreira, and Jorge Carvalho. Towards the development of intelligent transportation systems. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 1206–1211, 2001. doi: 10.1109/ITSC.2001.948835.

[42] Eric Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38 – 46, 2005. ISSN 0272-1716.

[43] Tobias Gädeke, Johannes Schmid, Marc Zahnlecker, Wilhelm Stork, and Klaus D. Müller-Glaser. Smartphone pedestrian navigation by foot-IMU sensor fusion. In *Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), 2012*, pages 1 – 8, 2012. ISBN 978-1-4673-1908-9. doi: 10.1109/UPINLBS.2012.6409787.

[44] Dorota A. Grejner-Brzezinska, Yudan Yi, and Charles K. Toth. Bridging GPS gaps in urban canyons: The benefits of ZUPTs. *Navigation: Journal of the Institute of Navigation*, 48(4):217 – 226, 2001. doi: 10.1002/j.2161-4296.2001.tb00246.x.

[45] Bluetooth Special Interest Group. Broadcom corporation bcm4339, 2014. URL `https://www.bluetooth.org/tpg/QLI_viewQDL.cfm?qid=19987`.

[46] Northrop Grumman. Litef, 2014. URL http://www.northropgrumman.litef.com/en/products-services/land-vehicles/product-overview/.

[47] Akihiro Hamaguchi, Masayuki Kanbara, and Naokazu Yokoya. User localization using wearable electromagnetic tracker and orientation sensor. In *10th IEEE International Symposium on Wearable Computers*, pages 55 – 58, 2006. ISBN 1-4244-0597-1. doi: 10.1109/ISWC.2006.286343.

[48] Joseph Hamill and Kathleen M. Knutzen. *Biomechanical basis of human movement*. Lippincott Williams & Wilkins, 2008. ISBN 978-0781791281.

[49] Hideki Hashimoto, Kazushige Magatani, and Kenji Yanashima. The development of the navigation system for visually impaired persons. In *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 1481 – 1483, 2001. ISBN 0780372115. doi: 10.1109/IEMBS.2001.1020485.

[50] Walter T. Higgins. A comparison of complementary and kalman filtering. *IEEE Transactions on Aerospace and Electronic Systems*, AES-11(3):321–325, May 1975. ISSN 0018-9251. doi: 10.1109/TAES.1975.308081.

[51] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57 – 66, 2001. ISSN 0018-9162. doi: 10.1109/2.940014.

[52] Fabian Höflinger, Jörg Müller, Rui Zhang, Leonhard M. Reindl, and Wolfram Burgard. A wireless micro inertial measurement unit (IMU). *IEEE Transactions on Instrumentation and Measurement*, 62(9):2583 – 2595, September 2013. ISSN 0018-9456. doi: 10.1109/TIM.2013.2255977.

[53] James R. Huddle. Trends in inertial systems technology for high accuracy AUV navigation. In *Proceedings Of The 1998 Workshop on Autonomous Underwater Vehicles AUV'98.*, pages 63 – 73, 1998. doi: 10.1109/AUV.1998.744442.

[54] Mariana N. Ibarra-Bonilla, Ponciano J. Escamilla-Ambrosio, Juan M. Ramirez-Cortes, and Carlos Vianchada. Pedestrian dead reckoning with attitude estimation using a fuzzy logic tuned adaptive kalman filter. In *2013 IEEE Fourth Latin American Symposium on Circuits and Systems (LASCAS)*, pages 1 – 4, February 2013. doi: 10.1109/LASCAS.2013.6519054.

[55] InvenSense. MPU-6500 six-axis (gyroscope + accelerometer) MEMS, 2014. URL http://www.invensense.com/mems/gyro/mpu6500.html.

[56] Antonio R. Jimenez, Fernando Seco, José C. Prieto, and Jorge Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. In *IEEE International Symposium on Intelligent Signal Processing*, pages 37 – 42, 2009. doi: 10.1109/WISP.2009.5286542.

[57] Antonio R. Jimenez, Fernando Seco, José C. Prieto, and Jorge Guevara. Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In *7th Workshop on Positioning Navigation and Communication (WPNC)*, pages 135 – 143, March 2010. doi: 10.1109/WPNC.2010.5649300.

[58] Rommanee Jirawimut, Madad Ali Shah, Piotr Ptasinski, Franjo Cecelja, and Wamadeva Balachandran. Integrated DGPS and dead reckoning for a pedestrian navigation system in signal blocked environments. In *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)*, pages 1741 – 1747, 2000.

[59] Gustav Johansson, Johan Selinder, and Kalevi Hyyppä. DE-Link, an antenna pointing system for stratospheric balloons. In *12th IEEE International Conference on Electronics, Circuits and Systems ICECS*, pages 1 – 4, December 2005. doi: 10.1109/ICECS.2005.4633446.

[60] Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35 – 45, 1960. doi: 10.1115/1.3662552.

[61] Bahador Khaleghi, Alaa Khamis, Fakhreddine O. Karray, and Saiedeh N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14 (1):28–44, 2013. ISSN 1566-2535. doi: 10.1016/j.inffus.2011.08.001.

[62] Masakatsu Kourogi, Tomoya Ishikawa, Yoshinari Kameda, Jun Ishikawa, Kyota Aoki, and Takeshi Kurata. Pedestrian dead reckoning and its applications. In *Proceedings of "Let's Go Out" Workshop in conjunction with ISMAR*, volume 9, 2009.

[63] Joseph B. Kruskall and Mark Y. Liberman. The symmetric time-warping problem: from continuous to discrete. *In Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 125–161, 1983.

[64] Quentin Ladetto, Vincent Gabaglio, and Bertrand Merminod. Two different approaches for augmented GPS pedestrian navigation. In *International Symposium on Location Based Services for Cellular Users, Locellus*, 2001.

[65] Adrián G. Ledroz, Efraim Pecht, David Cramer, and Martin P. Mintchev. FOG-based navigation in downhole environment during horizontal drilling utilizing

a complete inertial measurement unit: directional measurement-while-drilling surveying. *IEEE Transactions on Instrumentation and Measurement*, 54(5):1997 – 2006, 2005. doi: 10.1109/TIM.2005.853562.

[66] Seon-Woo Lee and Kenji Mase. Activity and location recognition using wearable sensors. *IEEE Pervasive Computing*, 1(3):24 – 32, 2002. ISSN 1536-1268. doi: 10.1109/MPRV.2002.1037719.

[67] Pamela K. Levangie and Cynthia C. Norkin. *Joint structure and function a comprehensive analysis*. F.A. Davis Co., Philadelphia, 2011. ISBN 9780803626348.

[68] Xiao Rong Li and Vesselin P. Jilkov. Survey of maneuvering target tracking. Part i. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4): 1333–1364, Oct 2003. ISSN 0018-9251. doi: 10.1109/TAES.2003.1261132.

[69] Yan Li and Jianguo J. Wang. A robust pedestrian navigation algorithm with low cost IMU. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1 – 7, November 2012. doi: 10.1109/IPIN.2012.6418861.

[70] Martin Liggins, David Hall, and James Llinas. *Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition*. CRC Press, 2nd edition, 1997. ISBN 9781420053081.

[71] Joel P. Lucas, Nuno Luz, María N. Moreno, Ricardo Anacleto, Ana Almeida Figueiredo, and Constantino Martins. A hybrid recommendation approach for a tourism system. *Expert Systems with Applications*, 40(9):3532 – 3550, July 2013. ISSN 0957-4174. doi: doi:10.1016/j.eswa.2012.12.061.

[72] Nuno Luz, Ricardo Anacleto, Constantino Martins, and Ana Almeida. Lightweight user modeling: A case study. In *Highlights in Practical Applications of Agents and Multiagent Systems*, volume 89 of *Advances in Intelligent and Soft Computing*, pages 77–84. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-19916-5. doi: 10.1007/978-3-642-19917-2_10.

[73] Nuno Luz, Ana Almeida, Ricardo Anacleto, and Nuno Silva. Collective intelligence in toursplan: An online tourism social network with planning and recommendation services. In *Proceedings of the International C* Conference on Computer Science and Software Engineering*, C3S2E '13, pages 42–48, New York, USA, 2013. ACM. ISBN 978-1-4503-1976-8. doi: 10.1145/2494444.2494449.

[74] Nuno Luz, Ricardo Anacleto, Constantino Martins, Ana Almeida, and Joel P. Lucas. Lightweight tourism recommendation. *International Journal of Web Engi-*

neering and Technology, 8(2):106 – 123, 2013. ISSN 1476-1289. doi: 10.1504/IJWET.2013.055712.

[75] David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, October 2003. ISBN 9780521642989.

[76] Sebastian O. H. Madgwick, Andrew J. L. Harrison, and Ravi Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *IEEE International Conference on Rehabilitation Robotics (ICORR)*, pages 1–7, June 2011. doi: 10.1109/ICORR.2011.5975346.

[77] Kim Mathiassen, Leif Hanssen, and Oddvar Hallingstad. A low cost navigation unit for positioning of personnel after loss of GPS position. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1 – 10, September 2010. doi: 10.1109/IPIN.2010.5646709.

[78] Brian McClendon. A new frontier for google maps: mapping the indoors, 2011. URL `http://googleblog.blogspot.com/2011/11/new-frontier-for-google-maps-mapping.html`.

[79] António Meireles, Lino Figueiredo, Luís Lopes, Ana Almeida, and Ricardo Anacleto. Ecg denoising with adaptive filter and singular value decomposition techniques. In *Proceedings of the Eighth International C* Conference on Computer Science & Software Engineering*, C3S2E'15, New York, USA, 2015. ACM.

[80] António Meireles, Lino Figueiredo, Luís Lopes, and Ricardo Anacleto. Ecg signal prediction for destructive motion artefacts. In *Ambient Intelligence - Software and Applications*, volume 376 of *Advances in Intelligent Systems and Computing*, pages 95–103. Springer International Publishing, 2015. ISBN 978-3-319-19694-7. doi: 10.1007/978-3-319-19695-4_10.

[81] Oleg A. Mezentsev. *Sensor aiding of HSGPS pedestrian navigation*. PhD thesis, University of Calgary, Canada, 2005.

[82] Nader Moayeri, Jalal Mapar, Stefanie Tompkins, and Kaveh Pahlavan. Emerging opportunities for localization and tracking. *IEEE Wireless Communications*, 18(2): 8 – 9, April 2011. ISSN 1536 - 1284. doi: 10.1109/MWC.2011.5751290.

[83] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, Cambridge, MA, August 2012. ISBN 9780262018258.

[84] John Murray. Wearable computers in battle: recent advances in the land warrior system. In *The Fourth International Symposium on Wearable Computers*, pages 169 – 170, 2000. ISBN 0769507956. doi: 10.1109/ISWC.2000.888485.

[85] John-Olof Nilsson, Isaac Skog, Peter Händel, and Kuchibhotla V. S. Hari. Foot-mounted INS for everybody - an open-source embedded implementation. In *IEEE/ION Position Location and Navigation Symposium (PLANS)*, pages 140 – 145, April 2012. doi: 10.1109/PLANS.2012.6236875.

[86] Aboelmagd Noureldin, Tashfeen B. Karamat, Mark D. Eberts, and Ahmed El-Shafie. Performance Enhancement of MEMS-Based INS/GPS Integration for Low-Cost Navigation Applications. *IEEE Transactions on Vehicular Technology*, 58 (3):1077–1096, March 2009. ISSN 0018-9545. doi: 10.1109/TVT.2008.926076.

[87] Aboelmagd Noureldin, Ahmed El-Shafie, and Mohamed Bayoumi. GPS/INS integration utilizing dynamic neural networks for vehicular navigation. *Information Fusion*, 12(1):48–57, January 2011. ISSN 1566-2535. doi: 10.1016/j.inffus.2010.01.003.

[88] Paulo Novais, Ricardo Costa, Davide Carneiro, and José Neves. Inter-organization cooperation for ambient assisted living. *Journal of Ambient Intelligence and Smart Environments*, 2(2):179 – 195, January 2010. doi: 10.3233/AIS-2010-0059.

[89] Lauro Ojeda and Johann Borenstein. Personal dead-reckoning system for GPS-denied environments. In *IEEE International Workshop on Safety, Security and Rescue Robotics SSRR*, pages 1 – 6, September 2007. doi: 10.1109/SSRR.2007.4381271.

[90] Robert J. Orr and Gregory D. Abowd. The smart floor: a mechanism for natural user identification and tracking. In *CHI'oo extended abstracts on Human factors in computing systems*, pages 275 – 276, 2000. ISBN 1581132484. doi: 10.1145/633292.633453.

[91] Meir Pachter and Guner Mutlu. The navigation potential of ground feature tracking for aircraft navigation. In *American Control Conference (ACC)*, pages 3975 – 3979, June 2010. doi: 10.1109/ACC.2010.5531219.

[92] Gundannavar V. Prateek, Shetty R. Girisha, Kuchibhotla V. S. Hari, and Peter Händel. Data fusion of dual foot-mounted INS to reduce the systematic heading drift. In *4th International Conference on Intelligent Systems Modelling Simulation (ISMS)*, pages 208 – 213, January 2013. doi: 10.1109/ISMS.2013.46.

[93] William Premerlani and Paul Bizard. Direction cosine matrix IMU: Theory. *DIY DRONE USA*, pages 13 – 15, 2009.

[94] Davy Preuveneers and Paulo Novais. A survey of software engineering best practices for the development of smart applications in ambient intelligence. *Journal of Ambient Intelligence and Smart Environments*, 4(3):149–162, August 2012. ISSN 1876-1364. doi: 10.3233/AIS-2012-0150.

[95] Jiuchao Qian, Ling Pei, Danping Zou, Kai Qian, and Peilin Liu. Optical flow based step length estimation for indoor pedestrian navigation on a smartphone. In *IEEE/ION Position, Location and Navigation Symposium - PLANS*, pages 205 – 211, May 2014. doi: 10.1109/PLANS.2014.6851377.

[96] Frederick Raab, Ernest Blood, Terry Steiner, and Herbert Jones. Magnetic position and orientation tracking system. *IEEE Transactions on Aerospace and Electronic Systems*, (5):709 – 718, 1979. ISSN 0018-9251. doi: 10.1109/TAES.1979.308860.

[97] João Ramos, Ricardo Anacleto, Ângelo Costa, Paulo Novais, Lino Figueiredo, and Ana Almeida. Orientation system for people with cognitive disabilities. In *Ambient Intelligence - Software and Applications*, number 153 in Advances in Intelligent and Soft Computing, pages 43 – 50. Springer Berlin Heidelberg, January 2012. ISBN 978-3-642-28782-4. doi: 10.1007/978-3-642-28783-1_6.

[98] João Ramos, Ricardo Anacleto, Paulo Novais, Lino Figueiredo, Ana Almeida, and José Neves. Geo-localization system for people with cognitive disabilities. In *Trends in Practical Applications of Agents and Multiagent Systems*, volume 221 of *Advances in Intelligent Systems and Computing*, pages 59–66. Springer International Publishing, 2013. ISBN 978-3-319-00562-1. doi: 10.1007/978-3-319-00563-8_8.

[99] Cliff Randell, Chris Djiallis, and Henk Muller. Personal position measurement using dead reckoning. In *Proceedings of the Seventh IEEE International Symposium on Wearable Computers*, pages 166 – 173, October 2003. doi: 10.1109/ISWC.2003.1241408.

[100] Valérie Renaudin, Okan Yalak, and Phillip Tomé. Hybridization of MEMS and assisted GPS for pedestrian navigation. Technical report, 2007.

[101] Valérie Renaudin, Okan Yalak, Phillip Tomé, and Bertrand Merminod. Indoor navigation of emergency agents. *European Journal of Navigation*, 5(3):36 – 45, 2007.

[102] Jan Rueterbories, Erika G. Spaich, Birgit Larsen, and Ole K. Andersen. Methods for gait event detection and analysis in ambulatory systems. *Medical engineering & physics*, 32(6):545 – 552, 2010. doi: 10.1016/j.medengphy.2010.03.007.

[103] Angelo M. Sabatini. Dead-reckoning method for personal navigation systems using kalman filtering techniques to augment inertial/magnetic sensing. *Kalman Filter: Recent Advances and Applications*, pages 251 – 268, 2009. doi: 10.5772/6809.

[104] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[105] John Saunders, Verne Inman, and Charles H. Eberhart. The major determinants in normal and pathological gait. *The Journal of Bone & Joint Surgery*, 35(3):543 – 558, 1953.

[106] Herman M. Schepers. *Ambulatory assessment of human body kinematics and kinetics*. PhD thesis, University of Twente, Netherlands, 2009.

[107] Ivan W. Selesnick and Charles S. Burrus. Generalized digital butterworth filter design. *IEEE Transactions on Signal Processing*, 46(6):1688 – 1694, June 1998. doi: 10.1109/ICASSP.1996.543681.

[108] Larry Sher. Personal inertial navigation system (PINS), 1996. URL `http://www.darpa.mil/`.

[109] Beomju Shin, Seok Lee, Chulki Kim, Jaehun Kim, Taikjin Lee, Changdon Kee, Sujeong Heo, and Heonsoo Rhee. Implementation and performance analysis of smartphone-based 3D PDR system with hybrid motion and heading classifier. In *IEEE/ION Position, Location and Navigation Symposium - PLANS*, pages 201 – 204, May 2014. doi: 10.1109/PLANS.2014.6851376.

[110] Sparkfun. Razor IMU, 2014. URL `https://www.sparkfun.com/products/10736`.

[111] STMicroelectronics. L3G4200D: Three axis digital output gyroscope, 2014. URL `http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf`.

[112] STMicroelectronics. TA0343 technical article - everything about STMicroelectronics 3 axis digital MEMS gyroscopes, 2014. URL `http://www.st.com/web/en/resource/technical/document/technical_article/DM00034730.pdf`.

[113] X-IO Technologies. x-IMU, 2014. URL `http://www.x-io.co.uk/products/x-imu/`.

[114] Tekscan. Flexiforce sensors for force measurement, 2014. URL http://www.tekscan.com/flexible-force-sensors.

[115] Rui Terra, Lino Figueiredo, Ramiro Barbosa, and Ricardo Anacleto. Step count algorithm adapted to indoor localization. In *Proceedings of the International C* Conference on Computer Science and Software Engineering*, C3S2E '13, pages 128 – 129, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1976-8. doi: 10.1145/2494444.2494457.

[116] Rui Terra, Lino Figueiredo, Ramiro Barbosa, and Ricardo Anacleto. Traveled distance estimation algorithm for indoor localization. *Procedia Technology*, 17: 248–255, 2014. ISSN 2212-0173. doi: 10.1016/j.protcy.2014.10.235.

[117] Ricardo Tesoriero, Jose A. Gallud, María D. Lozano, and Victor Penichet. Using active and passive RFID technology to support indoor location-aware systems. *IEEE Transactions on Consumer Electronics*, 54(2):578 – 583, 2008. ISSN 0098-3063. doi: 10.1109/TCE.2008.4560133.

[118] Yee K. Thong, Malcolm S. Woolfson, John A. Crowe, Barrie R. Hayes-Gill, and Richard Challis. Dependence of inertial measurements of distance on accelerometer noise. *Measurement Science and Technology*, 13:1163 – 1172, 2002. doi: 10.1088/0957-0233/13/8/301.

[119] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. Stanley: The robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, 23(9):661–692, September 2006. ISSN 1556-4967. doi: 10.1002/rob.20147.

[120] David Titterton and John Weston. Strapdown inertial navigation technology, 2nd edition. *The Institution of Electronical Engineers, USA*, 2004.

[121] Christopher Vaughan, Brian Davis, and Jeremy O'connor. *Dynamics of human gait*. Human Kinetics Publishers Champaign, Illinois, 1992. ISBN 978-0873223683.

[122] John G. Webster. *The Measurement, Instrumentation and Sensors Handbook*. CRC Press, 1st edition, December 1998. ISBN 9780849383472.

[123] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[124] Oliver Woodman and Robert Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 114 – 123, 2008. doi: 10.1145/1409635.1409651.

[125] Xsens. MTi series, 2014. URL `https://www.xsens.com/products/mti-10-series/`.

[126] Xuebing Yuan, Chaojun Liu, Shengzhi Zhang, Shuai Yu, and Sheng Liu. Indoor pedestrian navigation using miniaturized low-cost MEMS inertial measurement units. In *IEEE/ION Position, Location and Navigation Symposium - PLANS*, pages 487 – 492, May 2014. doi: 10.1109/PLANS.2014.6851407.

[127] Neil Zhao. Full-featured pedometer design realized with 3-axis digital accelerometer. *Analog Dialogue*, 44(6):1 – 5, 2010.