UNIVERSITÀ DEGLI STUDI DI PARMA

Dottorato di Ricerca in Tecnologie dell'Informazione

XXXI Ciclo

Pedestrian Inertial Navigation

Coordinatore:

Chiar.mo Prof. Marco Locatelli

Tutor:

Chiar.mo Prof. Gianluigi Ferrari

Dottorando: *Nicolò Strozzi*

Anni 2015/2018

To who always believed in me,
to my family.

Abstract

Even though technology-aided pedestrian navigation is an extensively studied research topic, a commercial system is not yet available. However, the ability to locate a person in “any” environment is a key feature in several applications. In outdoor environments, the position can be obtained through satellite-based systems or by triangulating radio signals (e.g., from cellular base stations). All these approaches rely on the use of transmitting devices placed in known positions in order to determine by triangulation, the position of the user’s receiving device. However, the costs and complexity, associated with a pre-deployed infrastructure and its maintenance, together with an often limited position accuracy, represent the mail limitation to an extensive adoption of these technologies, in particular in indoor environments.

Inertial navigation-based solutions can overcome these limitations because of their independence from external infrastructures. Inertial Navigation Systems (INSs) exploit data collected through Inertial Measurement Units (IMUs) to continuously estimate the position, orientation, and velocity of a moving user. Leveraging the sensor’s cost reductions, the massive diffusion, in commercial devices, of cheap IMUs can be exploited to perform pedestrian inertial navigation.

In this dissertation, an extensive overview over the possible approaches already existing in the literature is first provided (Chapter 1). In particular, this work mainly focuses on purely inertial navigation systems. A preliminary analysis of the algorithms able to relate inertial signals to the body displacement is given. Then, two innovative algorithms for pedestrian navigation are proposed (Chapter 2). These solutions exploit different body dynamics because of the different sensor on-body placement. After that, a thorough analysis relative to the impact of the sensors' placement over the human body is provided. Different approaches are studied, and, then, the fusion of the signals collected through different IMUs is proposed in order to increase the accuracy with respect to the systems based on single sensor (Chapter 3 and Chapter 4). The evaluation of the possible approaches to inertial navigation has also led to the use of hand-held commercial devices (Chapter 5). In particular, an innovative algorithm to perform inertial navigation based on the data collected through the IMU embedded in a smartphone is proposed. The considered algorithms are validated through an extensive experimental campaign with multiple smartphones. The obtained results show the feasibility of commercial pedestrian INSs, highlighting their strengths and weaknesses.

Contents

List of Acronyms

INS Inertial Navigation System 2, 21, 39, 53, 66, 85, 103

LF Low Frequency 11

LOS Line Of Sight 15

LP Low Pass 33

MARG Magnetic, Angular Rate, and Gravity 32

MEMS Micro Electro Mechanical System 23, 85

MF Medium Frequency 11

NLOS Non Line Of Sight 19

PDR Pedestrian Dead Reckoning 32

Radar RAdio Detection And Ranging 11

RSSI Received Signal Strength Indicator 17, 104

TDOA Time Difference Of Arrival 14

TO Toe Off 33, 41, 91

TOA Time Of Arrival 14

TOF Time Of Flight 14

UAV Unmanned Autonomous Vehicle 1

UHF Ultra High Frequency 11

USDOD United States Department of Defense 15

UWB Ultra Wide Band 2, 19

WLAN Wireless Local Area Network 17

ZUPT Zero velocity UPdaTe 34, 41, 53, 68

List of Figures

List of Tables

Introduction

If you want to find the secrets of the universe,
think in terms of energy, frequency and vibration.
- Nikola Tesla

The ability to navigate in an unknown environment, without any information about the surroundings, has been a challenge which mankind has faced forever. First navigators started applying new strategies in order to guide themselves over "endless" seas and lands. Nowadays, position informations represents an even more important aspect because of the technological claims.

Substantial investments have been carried out to develop tools able to provide more and more accurate position information. Even though the well known satellite-based localization is only one of these various technologies, it has had, through the widespread use of Global Positioning System (GPS), a disruptive impact over the everyday life habits. Modern lifestyles lead to an "indoor life," in which persons live, work, and even perform sports. In these scenarios, in which satellite signals cannot provide a position estimation, persons/objects location still represents a key information: in extreme scenarios (in which, for example, firefighters or military operate); in malls where the user's position knowledge allows an accurate and effective buying experience; in warehouses where Unmanned Autonomous Vehicles (UAVs) cohabit with human operator to improve safety and productivity; and so on.

The conventional way to estimate location info is represented by radio-based approaches, where external signals are collected by a receiver to estimate its position. Some of these technologies, such as WiFi, exploit preexisting infrastructures which are deployed for other purposes (i.e., to provide internet connection). Bluetooth devices, which have been developed for communication reasons, can be deployed in the environment in order to act as anchors. Moreover, ad hoc technologies, such as Ultra Wide Band (UWB), have been specifically developed to perform very accurate positioning in indoor scenarios. The main drawback of all these technologies is their infrastructure dependency and high sensitivity to noise and environmental changes.

Inertial navigation is characterized by an efficient trade-off between costs, complexity, and reliability. Inertial navigation means estimating the current position by knowing only where one was and the “energy amount” one has put in his/her movement to reach the current position. This approach can be applied to several fields. This thesis focus on one of the most interesting application, namely the *pedestrian inertial navigation*. Multiple Inertial Navigation Systems (INSs) have been investigated, and several prototypes have been built in order to verify the feasibility of an effective pedestrian INS.

This thesis in organized as follows. In *Chapter 1*, inertial navigations basics are provided. A historical overview of the INSs evolution is given, together with a survey about the state of the art and, in particular, pedestrian INSs. In *Chapter 2*, two novel algorithms for pedestrian navigation, based on signals collected by a single wearable Inertial Measurement Units (IMUs), are compared. These two navigation algorithms require the placement of the IMU on the foot or on the chest of the test subject, respectively. Different methods for gait characterization are compared, evaluating navigation dynamics by using data collected through an extensive experimental campaign. The main goal of this chapter is to investigate the peculiarities of different inertial navigation algorithms, in order to highlight the impact of the sensor’s placement, together with inertial sensor issues. *Chapter 3* describes the development of a multi-floor hybrid inertial/barometric navigation system. The prototype integrates several IMUs and a barometer. The inertial sub-system, by properly processing the signals collected by the IMUs placed on the test subject’s feet, reconstructs the two-dimensional navigation pattern. Three different sensors’ configurations are investigated in order to find the best performing set-up. A simpler configuration with a single IMU is also considered to derive a reference performance benchmark without multiple IMU fusion. The barometer is used to detect the floor change. The fusion of inertial and barometric signals enable the complete reconstruction of the movement of a person in both indoor and outdoor environments. In *Chapter 4*, a comparison between different inertial systems is presented, investigating the impact of on-body placement of IMUs and, consequently, of different algorithms for the estimation of the traveled path on the navigation accuracy. In particular, the system performance is investigated considering two IMU placements: (i) on the feet and (ii) on the lower back. Sensor fusion is then considered in order to take advantage of the strengths of each placement. The results are validated through an extensive data collection in indoor and outdoor environments. Finally, in *Chapter 5* an innovative INS based on the data collected through the IMU embedded in a commercial smartphone is presented. An innovative step detection algorithm, which is independent of the holding mode, is proposed, the only assumption behind its derivation being that the device is hand-held (i.e., the user is texting/navigating or phoning) and its movement is related to the upper body displacement during walking. A new approach able to automatically calibrate the step length estimation formula according to the smartphone positioning is also presented. The developed algorithms is validated through a test campaign carried out by considering three different smartphone models and different path lengths.

Chapter 1

Navigation

Following the light of the sun, we left the Old World.
- Christopher Columbus

Navigation can be defined as finding one's way everywhere over the world, through seas, deserts, mountains, and cities. Without a definite path, the navigator has to rely on every information the environment provides him, such as reference points, celestial bodies or electronic marks. The word navigate comes from the Latin words *navis* (ship) and *agere* (to drive or guide).

Navigation involves many scientific fields and has always represented a strategic capability, based on the understanding of earth, seas, and skies. Even though changes in navigation science and technology over the last five hundred years have altered the navigator's work and methods, the navigator's basic task remains constant: keep track of where he/she has been and where he/she is now in order to plan where to go next.

Navigation relies on astronomy, physics, oceanography, meteorology, earth sciences, aerodynamics, and hydrodynamics. Mathematics can include arithmetic, algebra, trigonometry, logarithms, geometry, and analysis. Due to the huge amount of complex data and the continuous environmental variations, the navigator needs to face several challenges in order to make good decisions.

While today's electronic have helped automate navigation, it also provides much more information to be processed by the navigator. The work of navigation requires care, but it is fascinating as it combines many disciplines and requires accurate forethought and planning.

1.1 History of Navigation

The navigation fundamentals arise from the first *Austronesians* which began sailing the Pacific ocean between about 3000 and 1000 B.C. Their descendants, hundreds of years later, started to populate oceanic islands, such as Tonga and Samoa, up to New Zeland, Easter Island and, perhaps, South America. At this time, the main strategies to navigation across the ocean included the observation of birds, star navigation, and the use of waves and swells.

The evolution of navigation come from the Mediterranean sea, where the living populations began looking more accurately to stars. The *Minoans* started building sanctuaries aligned with the rising of the sun on the equinoxes, but also with references to the stars, demonstrating a good astronomic knowledge. In particular, their trips up to Thera and Egypt led them to the study of the constellations, such as the *Ursa Major*, the *Ursa Minor*, and the β *Draconis*. A map of these constellations map is shown in Figure 1.1.

Together with the constellations adoption as reference for navigation overnight, the Greeks became accurate geographers, creating more and more accurate maps. The famous astronomer, geographer and, clearly, navigator *Pytheas of Massalia* was able to discover the polar ice, the German tribes, and, possibly, Stonehenge. He was also the first known person to describe the *Midnight Sun*. The Greeks were also one of the first population able to build mechanical instruments specifically for navigation purposes, such as the *Antikythera mechanism*, shown in Figure 1.2(a). This tool was used to predict astronomical positions and eclipses and the oldest one has been dated around 100 B.C..

The Mediterranean sea was also the cradle of two more civilizations, the *Phoenicians* and the *Carthaginians*, which, around the 500 B.C., became famous for their

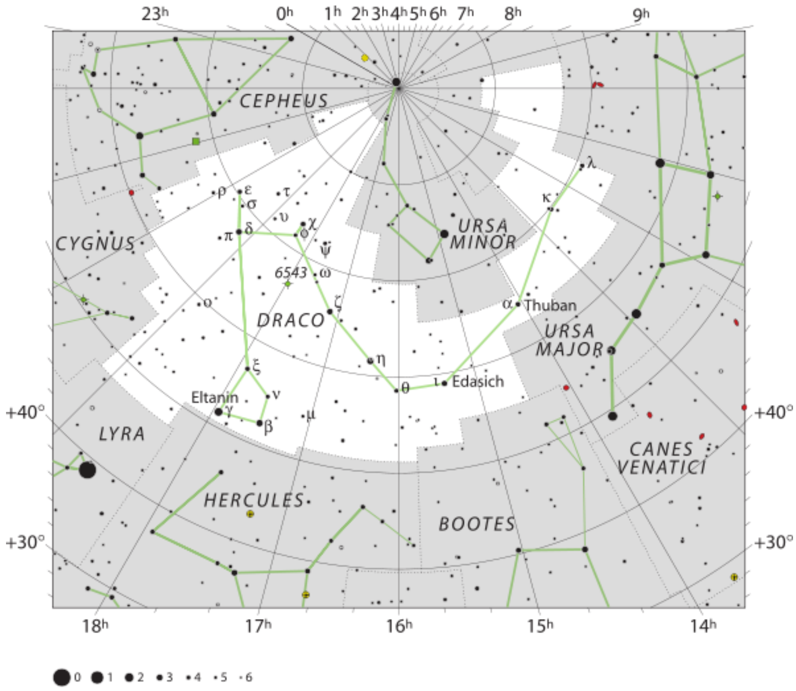


Figure 1.1: Constellations used by the ancients to navigate during nights: the Ursa Major, the Ursa Minor, and the Draco.

ability in sailing. They are credited for introducing new methods and instruments, such as the first *sounding line*, shown in Figure 1.2(b), to estimate how far they were from the land and sediments from the bottom of the sea to determine their relative positions with respect to the sediments.

At the beginning of the 9th century, the *Arab empire* significantly contributed to the development of innovative methods for navigation. Given the small number of navigable rivers in the region, sea navigation became fundamental for their survival. This necessity led to the discovery of the magnetic compass and the *Kamal*, a very basic tool for latitude estimation based on the relative position of the polar star. The magnetic compass invention, however, dates back to the 200 B.C. in China, where it was used for divination. The magnetic compass has been adopted as navigation tool

(a) (b)

Figure 1.2: Illustrative navigation tools: (a) Antikythera mechanism found in 1902 off the coast of the Greek island Antikythera (≈100 B.C.); (b) a modern version of the sounding line, still used to determine the depth of the water.

in Asia starting form the 11th century.

In the 15th century, the increasing commercial activities all over the world provided a boost to navigation technique development. In particular, the Portuguese navigators, in addition to the discovery of new lands (such as Azoreos, Cape Verde, and Sierra Leone), introduced new instruments to support sailors to estimate more and more accurately their positions over the sea. One of these tools is the *Mariner's Astrolabe*, an inclinometer used to determine the latitude by measuring the sun's altitude at midday and comparing it with pre-computed tables. The introduction of these new tools supported some of the most famous navigators of all times, such as: Infante Enrique, Vasco de Gama, Pedro Álvares Cabral, and, obviously, the italian Christopher Columbus to navigate over longer paths with respect to their recent past. This led to the discovery of the American continent (by Christopher Columbus in 1492) and the world circumnavigation (performed by Ferdinand Magellan in 1521). These navigators likely achieved these historical results with just a compass, an astrolabe, and early nautical charts. This epoch is considered the begin of the modern

Tratado da sphera
com a Theorica do Sol ⁊ da
Lua. E ho primeiro liuro da
Geographia de Claudio Pto
lomeo Alexãdrino. Tirados nouamen
te de Latim em lingoagem pello Doutor
Pero Nunez Cosmographo del Rey dõ
João ho terceyro deste nome nosso Se
nhor. E acrecẽtados de muitas annota
ções ⁊ figuras per que mays facilmente
se podem entender.

Item dous tratados q̃ o mesmo Dou
tor fez sobre a carta de marear. Em os
quaes se decrarão todas as principaes
duuidas da nauegação. Cõ as tauoas do
mouimento do sol: ⁊ sua declinação. E o
Regimẽto da altura assi ao meyo dia: co
mo nos outros tempos.

COM PREVILEGIO REAL.

Figure 1.3: *Tratado de sphera*, Pedro Nunes, 1521.

era for navigation. In 1537 Pedro Nunes published "*Tratado da Sphera*," shown in Figure 1.3, which represents the first mathematical approach to navigation. Basically, he laid the basis for a new discipline, the "scientific navigation." Another basic tool, first mentioned in writings around the 1577, was the *chip log*, shown in Figure 1.4, which is a line with knots placed at uniform intervals attached to a wooden plate. Together with an instrument to measure the time, such as a *Sand-glass* or a clock, the chip log enabled the ship speed estimation by measuring the number of knots coming out from the reel over a certain period. The word *knot*, as measurement unit, comes form this tool and is used to express the nautical speed of 1.1508 miles/hour.

Thanks to these new instruments, a new method still used in several "modern

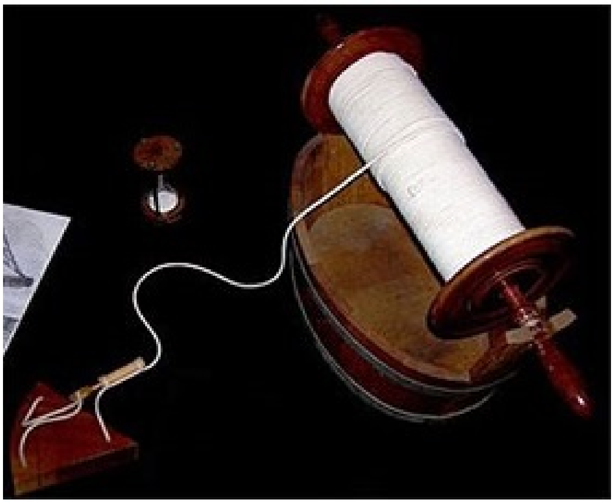


Figure 1.4: A chip log, also called ship log, used to estimate the speed oh the ship in *knots*.

applications," denoted as *Dead Reckoning*, has been invented. Dead reckoning consists in the estimation of the position from the speed (measured through the chip log) and the direction (measured through the astrolabe or the compass). The term *Dead Reckoning* is, possibly, an alteration of the word *Deduced Reckoning*, referring to a position deduced by the last known (or estimated) position. The last relevant instrument discovered before the 19th century radio-based approaches, which have revolutionized the navigation, have been the *Sextant* in 1757. This tool provided a new method to estimate the lunar distance, enabling a more accurate estimation of the longitude.

The methods introduced in this section have lead to the modern concept of navigation. These methods, inherited by ancient navigators who explored the globe, still survive in the current navigation tools which nowadays dominate the every day life. By the end of the 19th century, the radio transmission discovery, first, and the computer era, then, have taken navigation to the next level. The next sections point to all the new methods which are currently used to locate a vehicle or a person everywhere in the world.

1.2 Radio-based Navigation

The transmission of radio signal through free space have been mathematically theorized for the first time from James Clark Maxwell in 1864: his equations, namely the *Maxwell's equations*, have sparked the understanding of this whole new world. In 1886, Heinrich Rudolf Hertz demonstrated the existence of these waves in his laboratory, triggering an extensive scientific research in this field. In 1894, Guglielmo Marconi's wireless telegraphy opened the door to radio transmission. In 1897, Marconi established the first communication with a 20 km far ship, and in 1901 he sent the first message across the Atlantic ocean.

In navigation and localization, radio transmission opened the route to a complete new set of instruments [1, 2]. The first attempt to determine a position through radio signals, was based on the use of *Radio Beacons* transmitting in Low Frequency (LF) and Medium Frequency (MF) bands. Maritime radio beacons were non-directional (as they were typically transmitted from isotropic antennas), and their directions were computed from equipment aboard the ship. The evolution of these beacons led to *Radar Beacons*, which are beacons transmitting in the Ultra High Frequency (UHF) domain. Since then, the RAdio Detection And Ranging (Radar) has become a cutting edge technology, developed simultaneously in many nations during World War I and World War II [3]. The Radar was able to compute the direction between itself and the target by rotating its antenna and looking at the orientation of the latter on the basis of the received reflected signal. The distance from the target, instead, was computed through the comparison between the signal reception epoch and the signal sending epoch (measured through an oscilloscope). The distance was then computed as follows:

$$d = \frac{c_0 \cdot t}{2} \tag{1.1}$$

where d is the estimated distance, c_0 is the speed of light (around $3 \cdot 10^8$ m/s), and t is the time interval between signal transmission and reception instants. The first Radars, given their complexity and sizes, were typically mounted on ground stations. For this reasons the "friendly target" position had to be transmitted to the target itself through a frequency band different from the one used for localization. The development of innovative Radar technology has led to the development of mobile Radars, suitable to aircraft, ships and, nowadays, cars [4, 5, 6] and robots [7, 8]. Modern Radars are equipped with antennas arrays or they exploit smart algorithms, making rotation mechanisms not necessary. The new hardware led to more precise localization, with the millimeter precision, which is necessary in application such as autonomous driving or robot navigation. The new Radars, indeed, are used for several other purposes, such as surface or obstruction detection, map matching and environment recognition, and they do not longer require ground station as reference.

However, a Radar technology still represents a complex system to handle and is not suitable to all navigation applications. Another radio-based approach suitable to localize targets with size, cost, or power constraints — such as a pedestrian — is the radio trilateration. This approach exploits the signals transmitted by multiple base stations, placed in known positions, to *triangulate* the target position. In fact, the powers of the signals received from the target can be used to estimate the distance between a single base station and the target itself. If we assume an ideal transmission, with no reflection or refraction, the distance can be estimated through the Friis formula [9] as follows:

$$r^2 = \frac{P_t A_t A_r \lambda^2}{P_r (4\pi)^2} \tag{1.2}$$

where: r is the distance between the radio station and the target; λ is the wavelength of the radio signal; A_t and A_r are the transmitter and the receiver antenna gains, respectively; P_t is the transmit power; and P_r is the received power. When the target detects the signal from a single antenna, its coordinates could lay on a circumference with center the base station coordinates and radius r as in Figure 1.5(a) (image credit [10]). If the signals from two base stations are available, the possible target coordinates are the intersection between the two circumferences estimated as shown in Figure 1.5(b). Theoretically, by receiving the signal from three base stations one can estimate uniquely the target position as in Figure 1.5(c). More precisely, the target

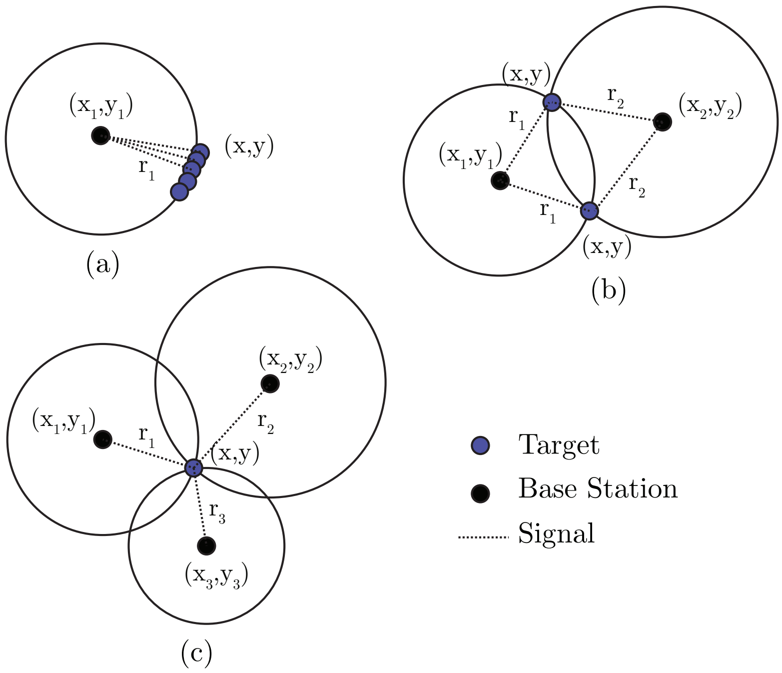


Figure 1.5: Geometric approach to trilateration: (a) Single base station available. The target could be on a circumference with center the base station and ray d. (b) Two base station available. The target could be in the two intersections between the circumferences. (c) Three base stations available. The target could be detected uniquely.

coordinates can be estimated by solving the following system:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = r_1 \\ (x - x_2)^2 + (y - y_2)^2 = r_2 \\ (x - x_3)^2 + (y - y_3)^2 = r_3 \end{cases} \tag{1.3}$$

where: (x, y) are the target coordinates; (x_1, y_1), (x_2, y_2), and (x_3, y_3) are the coordinates of the three base stations; and r_1, r_2, and r_3 are the corresponding distances between the target and the base stations. Signal trilateration represents a basic method which has been widely implemented in real applications. However, distance estimation remains a challenging problem to solve, in particular for applications which need a high accuracy and where the free space propagation assumption is not applicable. Moreover, the anchors evolution, from radio beacons to modern cellular base stations and WiFi access points, have lead to the development of innovative localization algorithms.

The Angle Of Arrival (AOA) approach [11, 12, 13], for example, estimates the target position by measuring the angle of incidence of the received signal. It requires additional hardware in order to estimate accurately the arrival angles with respect to the different base stations. The Time Of Arrival (TOA) [14, 15, 16, 17, 18, 19], also known ad Time Of Flight (TOF) method, estimates the target position by measuring the time delay between transmission and reception epochs. The distance between the target and the base stations can then be computed using equation (1.1) multiplied by two, given that the delay refers to a single path between the station and the receiver. Time Difference Of Arrival (TDOA) [20, 21, 22, 23, 24, 25, 26], which represents an evolution of the previous methods, is based on the pairwise differences among the arrival times at the base stations of the signal transmitted by the subject/object which has to be tracked. This method requires very accurate synchronization between the base stations, in order to compute the time difference without biases. A similar approach which can be used if the target has to be in relative motion with respect to the base station is the Frequency Difference Of Arrival (FDOA) [27, 28, 29]. FDOA is based on the differences between the frequencies in the received signal caused by the Doppler effect and is often used together with the TDOA to increase the localization accuracy.

All the concepts introduced in this section can be applied to several navigation systems based on different technologies. Subsection 1.2.1 and Subsection 1.2.2 provide overviews on the systems which are based on different technologies, namely: satellite-based; WiFi-based; and Bluetooth Low Energy (BLE)-based.

1.2.1 Satellite-based Navigation

The first artificial satellite, the *Sputnik 1*, was launched by the Soviet Union on 4th October 1957. The first satellite for localization purposes, *Transit*, was deployed from the United States military in the 1960s. This first single satellite-based system was exploiting the Doppler effect to estimate the frequency shift caused by the speed difference between the satellite and the ground station. Multiple observation and the knowledge about the satellite orbit was leading to a position estimation. The straightforward evolution of *Transit* has been the Global Positioning System (GPS) [30], deployed by the United States Department of Defense (USDOD) and completely operational on 17th July 1995. GPS relies on a constellation of more than 24 satellites which are placed over six circular orbits, approximately 20000 kilometers above the ground. The idea is to provide a GPS ground receiver with the possibility to be in Line Of Sight (LOS) with at least four satellites, which is the minimum number of "satellite stations" which is needed to estimate the receiver three-dimensional position. The GPS satellites and their orbits are shown in Figure 1.2.1.

In recent times, as reported in [31], Global Navigation Satellite Systems (GNSSs) have also been adopted for multiple civilian applications, such as: airplanes localization; automotive and naval applications; topography; localization-based data management; safety-related applications (autonomous driving, save and rescue application, etc.); and others. The European Commission reported that in 2001 the satellite-based navigation market and all the related activities had a turnover of 15 billion Euros. Many international organizations, together with private companies, are deploying their own GNSSs in order to reach higher accuracies by adopting new generation technologies. The new safety-critical applications have raised also concerns on the reliability, the availability and the (localization information) consistency. The deployment of more then four satellites leads to the performance improvement in all the aforementioned fields.

The GNSS deployed by Russia, namely *GLONASS* [32], became operational in February 2016. The next generation GLONASS constellation, based on newer platforms, is planned to be launched after 2020. The satellites which compose this constellation have an improved clock stability and a new control and command system.

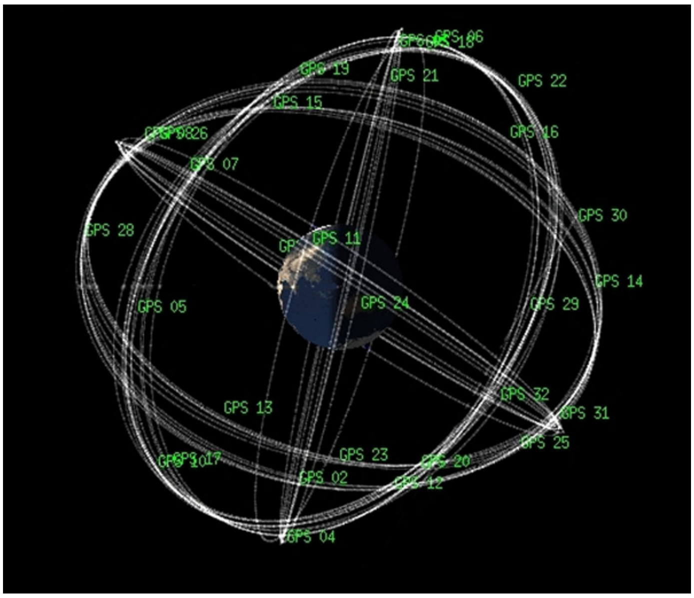


Figure 1.6: GPS satellites constellation: 24 satellites in 6 orbital planes, 4 satellites each plane, 20.200 Km altitude, 55 deg inclination.

BeiDou is the chinese GNSS. The third generation is currently being deployed with the goal to complete the 35 satellite constellation until 2020. The European GNSS, called *Galileo* [33], became operational on December 2016. At the end of August 2018 the Galileo constellation included 26 satellites in orbit, 17 of which fully operational. Finally, two other GNSSs are *QZSS* [34] and *NavIC* [35], which are, respectively, a four-satellite system for the Asia-Oceania region (operational since November 2018) and a seven-satellite system covering the Indian region and its surroundings (expected to be operational in 2019). More details about GNSS constellations can be found in [36].

The GNSSs are based on improved versions of the algorithms introduced in the previous sections (TOF, TDOA, FDOA, etc.), but the main idea behind all these different systems is still based on the constellations capability to maintain the synchronization between the anchors, namely the satellites, in order to estimate the ground receiver position by exploiting the timestamps transmitted by the satellites through broadcast messages.

1.2.2 WiFi-, BLE-, and UWB-based Navigation

The technologies introduced in the previous sections are mainly suitable for outdoor applications, owing to the dimension of the base stations or the propagation channel characteristics. Other radio-based technologies are often adopted in indoor scenarios, where (i) smaller devices can be deployed in order to provide an ad-hoc infrastructure or (ii) devices already placed in the environment for different purposes (i.e., WiFi Access Points (APs) used to implement a Wireless Local Area Network (WLAN)) can be exploited also for positioning purposes. In this section, an overview of the three most adopted radio-technologies for indoor positioning currently available on the market is provided.

- **WiFi:** nowadays WiFi infrastructures are present in almost all the public buildings to provide many services (i.e., in warehouses to provide network access to operators and machines, in shopping malls to offer internet connection and collect data from the customers, etc.). These infrastructures are composed by APs usually wall-mounted or ceiling-mounted, with one or multiple routers connected to local servers or to the internet. These APs can be exploited as base stations in order to perform localization [37, 38]. There are three main approaches which can be used in these scenarios [39]: (i) the use of Received Signal Strength Indicators (RSSIs) for triangulation purposes;, (ii) fingerprinting; and (iii) crowd-sourcing. “Classic approaches,” such as the TOF, AOA, and TDOA, are difficult to implement over the commercial devices because of limited synchronization among the APs and for the specific hardware requirements. For this reason, one of the most used techniques is the RSSI-based method, which relies on the trilateration of the radio signals at the target, by adopting a statistical model which relates the distance from the AP to the RSSI at the target position. An evolution of the RSSI-based localization is the fingerprinting, which includes two phases. In the first phase, one must collect the

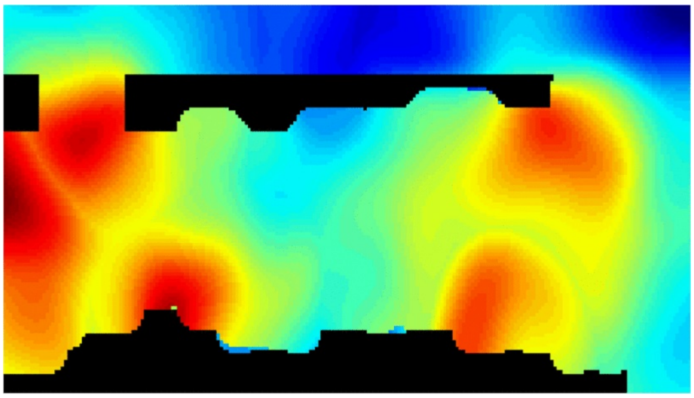


Figure 1.7: Heat map representing the WiFi coverage in terms of maximum signal level from every access point in each point.

RSSIs at each location in order to build a database which relates the signals' strengths with a specific position. The result of this preliminary phase leads to an heat map such as the one shown in Figure 1.7 introduced in [40]. During the second phase, the target devices need to compare the real-time measured RSSI with the map, in order to obtain the estimated position. By using commercial devices, the fingerprint-based localization can provide meter-level accuracy, but an accurate procedure is needed. The crown-sourcing method, instead, relies on the users' devices in order to collect real-time information related to the WiFi signals [41, 42, 43]. This allows to exploit the localization information in a "recursive way," which means that the position estimation is provided to the user and, at the same time, the user information enables the localization of the next users.

- **Bluetooth Low Energy (BLE):** bluetooth technology has been widely used in commercial and industrial applications [44, 45], mainly for data transmission and for monitoring. This has lead to a large diffusion of the bluetooth interfaces in many platforms, such as smartphones and Internet of Things (IoT) devices. Given that usually bluetooth anchors are not present in buildings, bluetooth beacons have been specifically developed in order to be placed in known positions

Figure 1.8: Examples of BLE beacons suitable for indoor positioning.

inside buildings to enable indoor localization. The BLE beacons, such as the ones shown in Figure 1.8, represent an attractive approach to indoor localization and navigation. The algorithms which can be applied in combination with this technology are similar to the ones adopted for the WiFi technology. The very basic information which they provide is the RSSI but, as for the WiFi APs, it is possible to apply other advanced techniques [46, 47, 48, 49, 50, 51, 52].

- **Ultra Wide Band (UWB):** UWB is a radio technology which allows very accurate positioning [53]. Its name comes from the characteristics of the radio signal, which is based on high power short impulses with very large frequency bandwidth. The first commercial devices based on UWB became available in the early 90s. The application of this radio technology is limited to small areas, due to the channel characteristics. This high bandwidth provides high data rate for communication and it also allows applications in Non Line Of Sight (NLOS) environments. An interesting application of UWB technology is localization. Since this technology is not still diffused as WiFi or bluetooth, in order to localize a target one needs to rely on a specific infrastructure composed of anchors equipped with antennas and chips able to manage the short impulses. Furthermore, the same hardware has to be installed on the target or carried by the user.

  The approaches which can be applied in UWB localization are mainly TDOA [54, 55, 56], TOF [57, 58], Two Way Ranging (TWR) [59, 60] and AOA

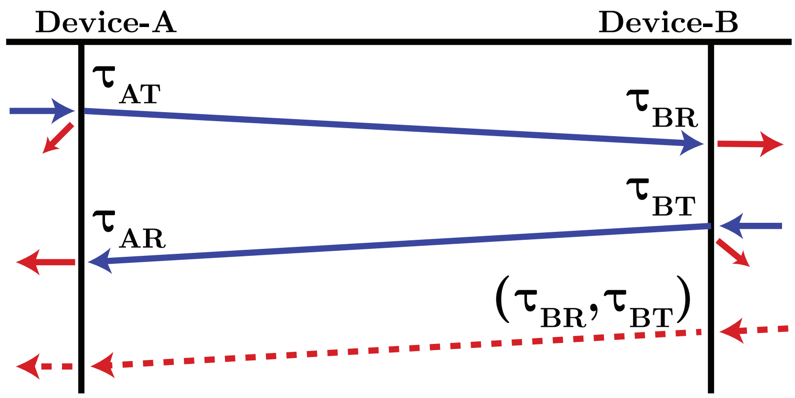


Figure 1.9: Ultra Wide Band (UWB) two way ranging: τ_{AT} is Device A transmission epoch; τ_{BR} is the Device B reception epoch; τ_{BT} is the Device B retransmission epoch; τ_{AR} is the Device A reception epoch; and (τ_{BR},τ_{BR}) is the message, sent from Device A to Device B, which represents the delay between reception and retransmission instants in order to estimate the flight time.

[61, 62]. The TWR algorithm shows a fundamental advantage with respect to the "classic" approaches which need the synchronization between the anchors. In fact, in order to obtain the distance estimate between a anchor and a target, an impulse is sent to the target which, in turn, sends back an impulse to the anchor. By measuring the time between the sending instant and the receiving one, and adding the delay due to the transmission, an anchor can estimate the range with respect to the target. In [60] the two way ranging message exchange shown in Figure 1.9 is considered.

1.2.3 Radio-based Navigation Limitations

All the radio-based localization technologies present issues related to both physical constraints and hardware. An overview of these problems is reported in the following.

- **NLOS:** the radio transmissions needs a LOS between the transmitter and the receiver in order to estimate accurately the distance between the two, for example by using the TOF approach. In the presence of obstacles in the channel, the estimation accuracy decreases. In the presence of heavy obstructions, such as irons or thick walls, the signals could be blocked, making the position estimate unavailable.

- **Infrastructure dependency:** all the radio technologies are based on anchors placed in known positions. In some case, these infrastructure have to be deployed specifically for localization purposes. These preliminary setup could be complex, such as for GNSSs.

- **Signals multi-path:** given the presence of surfaces between the base stations and the target in almost every scenario, the receiver captures multiple signals, namely the one which is traveling on a straight line between the transmitter and the receiver and the ones reflected by the ground, walls, etc. This leads to multi-path and, typically, to a signal's traveled path overestimation.

- **Costs:** the infrastructures which have to be deployed could have high costs, in particular for the ad-hoc systems such as UWB or GNSS.

1.3 Inertial Navigation

1.3.1 Principles

Inertial navigation is a dead reckoning navigation technique in which measurements provided by accelerometers and gyroscopes are used to track the position and orientation of an object relative to a known starting point, orientation and velocity. An Inertial Measurement Unit (IMU) usually embeds a three-axial accelerometer and a three-axial gyroscope, which measure the linear acceleration (i.e., the acceleration along a specific axis) and the angular rate (i.e., the angular rotation rate around a specific axis), respectively. By exploiting the signals collected from these sensors one can compute the orientation and the position of the device equipped with the IMU.

The Inertial Navigation Systems (INSs) were firstly developed for rocket guidance in the US, but in a short time they became popular thanks to the application in

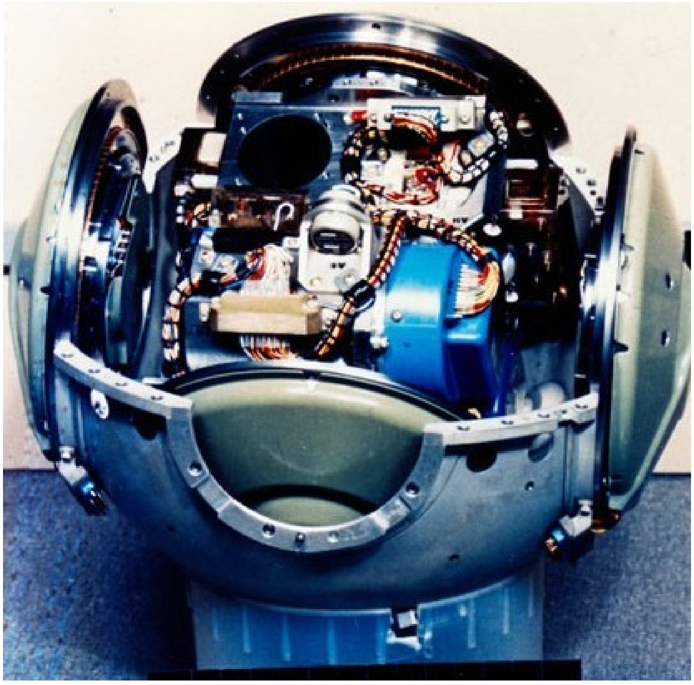


Figure 1.10: The Apollo IMU with the top removed. It embeds: three gimbals, a an electronic platform, and the accelerometer and gyroscope package.

the aeronautical field. They have been adopted in the second world war in the V2 rockets guidance system, by combining two mono-axis gyroscope and a mono-axis accelerometer in order to control the rocket attitude during the flight. The INS military application, in particular in rocket science, have lead to a fast development of this technology. In the early '60s they also became a fundamental technology for space exploration, when NASA commissioned a preliminary study to the MIT laboratories for the Apollo mission. In Figure 1.10 the IMU mounted on the Apollo mission spacecraft is shown.

The IMU development has been carried forward by the next Space Shuttle missions for (i) maintaining the rocket balance during the lift-off and (ii) for the shuttle guidance navigation and control system. The first commercial application based on

INSs has been the aircrafts and ships guidance system. In fact, as can be seen from Figure 1.10, the size of the early IMUs was enabling only application in which a stable mechanical platform was available in order to fix the IMU. The recent evolution of the IMU construction, largely based on the development of innovative Micro Electro Mechanical Systems (MEMSs), have enabled the possibility to build small and light INSs. This new manufacturing process have lead to the application of the inertial-based systems in many other areas, such as human and animal motion capture, IoT applications, medical applications, etc. An overview about inertial sensors is given in Section 1.4.

In general, an INS can measure modification in its speed (i.e., detecting acceleration or deceleration and the direction of these variations) and in its orientation (i.e., by applying an algorithm able to compute this information through the fusion of the raw signals from the sensors). They are used to track different subject/object and their cost highly depends on the application and on the accuracy needed for the specific application.

The present thesis mainly investigates pedestrian inertial navigation systems, i.e., INSs specifically developed to track humans in Three-dimensional space in order to provide localization and navigation information. Details about this specific application are given in Subsection 1.5.

1.3.2 Inertial Navigation Pros and Cons

Inertial navigation represents a different approach with respect to the ones presented above because it relies on measurement directly provided by the sensor, namely the IMU. This leads to the following three main advantages:

- **Infrastructure independent:** by definition, inertial navigation is based on inertial signals, which are exploited to compute the current position together with the knowledge of the starting point. Therefore it is not necessary to deploy APs, beacons, or other anchors in the environment.

- **No external interferences:** all the inertial signals, such as acceleration and angular velocity, are directly collected by the IMU which is carried by the target/user and, because of that, there is no need for an of external signal to compute the position. This means that external — intentional or not — disturbances cannot be introduced in the measurement, making the position estimation reliable even in challenging environments such as forests, mountains or war zones.

- **Low cost:** the early INS were based over large scale complex mechanical inertial sensors. The MEMS technology has lead to the development of low cost sensors which can be embedded in commercial devices, making the INSs cost-effective.

Nevertheless, an INS is affected by a fundamental error, namely the *drift*, which is the error, introduced by multiple noise sources, which leads to a lack of accuracy in the orientation and distance estimation. The drift is the greatest unsolved issue for INSs and it represents the accumulation of microscopic errors in the accelerometer, magnetometer, and gyroscope measurements, which gradually cause the INS position estimation to become more and more inaccurate. A wrong angular velocity integration, for instance, causes an overestimation or an underestimation of the angle actually done by the IMU, which lead to a wrong orientation estimation and, consequently, to a wrong position estimate. In order to tackle the drift, one could exploit an external reliable information (e.g., a GPS estimation when available) to "reset" the position. Unfortunately, this method cannot be applied in every scenario and, above all, it causes the loss of one of the advantages of INSs, namely the infrastructure independence. Therefore, the INS algorithms have to take into account this issue in order to minimize its effect on the position and orientation estimations. Deeper insights on this errors and on their causes are provided in Section 1.4.

Moreover, a further noise source for a INS is represented by the *different users characteristics*. In other words, an inertial system which has to locate a person, highly depends on the user itself (i.e., his height, leg length, walk pace, etc.), so it needs a *per-subject calibration*. This leads to the identification of constants, such as the minimum acceleration used to declare a stationary phase, or the angular rate magnitude used to segment the gait. In this thesis, multiple INSs are introduced but, in order to provide a fair comparison, all the calibration-dependent constants have been tuned on a single test subject.

Another noise source in INSs is represented by the *incorrect sensors positioning*, that is the wrong placement of the IMUs on the user body. For instance, a foot-mounted IMU-based INS requires that the sensor is placed as close as possible to the flat part of the foot. If the sensor is placed on a different position, such as next to the ankle, the foot stationary phases will not be correctly detected, leading to a wrong position estimate. A similar problem afflicts the INSs which are based on waist-mounted sensors. If the IMUs have to be mounted on the breastbone in order to exploit the body symmetry, the sensor wrong positioning, far from the body frontal plane, causes an inaccurate step detection.

1.4 Inertial Sensors

As introduced in Section 1.3, the MEMS technology has led to the miniaturization, the cost reduction, and the accuracy increment of inertial sensors with respect to the first inertial platforms adopted in ship and aircraft guidance systems. This has enabled the application of this technology in many fields, such as in: military systems; biomedical systems; in robotics; control systems; and sport performance monitoring. Inertial sensors, as indicated from the word "inertial," identify all the sensors which exploit the *inertia principle* in order to measure a physical quantity. The inertia principle refers to *Newton's First Law* (also called *Galileo's Principle*) which says that "An object that is at rest will stay at rest unless a force acts upon it." In the following, the characteristics of accelerometers and gyroscopes are introduced. Moreover, an overview about magnetometers (which is not an inertial sensor by definition) is provided, given that usually a IMU also embeds this sensor in order to perform the IMU global orientation with respect to the earth coordinate system.

1.4.1 Accelerometer

An accelerometer, despite its name, is not measuring an acceleration. It measures a force per unit of mass. In other words, it measures the force that the accelerometer generates to react to the gravity acceleration in addition to any other acceleration it

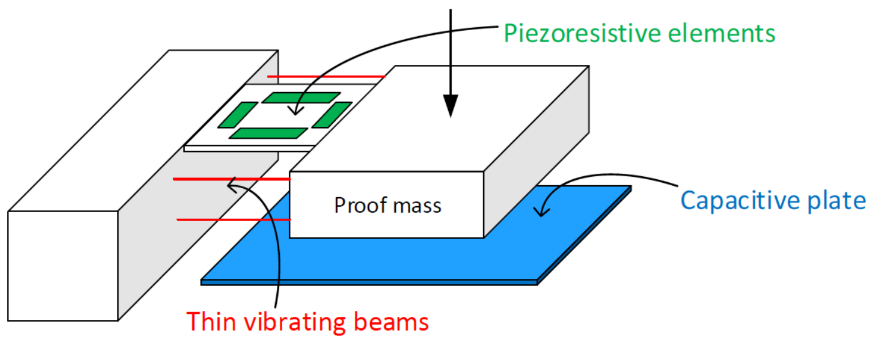


Figure 1.11: Three different mechanical ways to measure the acceleration: (i) a piezo-electric system, (ii) vibrating beams and (iii) a capacitive plate.

is subject to. In order to make the accelerometer operations more understandable, one can imagine the force as applied to a proof mass m (dimension: [kg]) which is free to move over one direction but is bounded to a spring. The acceleration a (in SI dimension: [m/s/s]; in imperial system dimension: [g], where 1 g = 9.8 m/s/s) is related to the force f (dimension: [N]) according with the following equation:

$$f = ma. \tag{1.4}$$

Given this setup, the force can be measured according with the Hooke's law

$$f = kd \tag{1.5}$$

where d (dimension: [m]) represents the mass displacement and k (dimension: [N/m]) is a constant related to the spring physical characteristics.

As shown in Figure 1.11, there are many systems which can be applied to detect the force on this mass. The transducer used in these accelerometers to measure the mass displacement can be a piezoelectric resistance, a capacitive plate or a vibrating beam under tension which changes its natural frequency if exposed to a force. There are also the *Hall-effect accelerometer*, which exploits the Hall effect (i.e., when an electrical current passes through a sample placed in a magnetic field, a potential proportional to the current and to the magnetic field is developed across the material

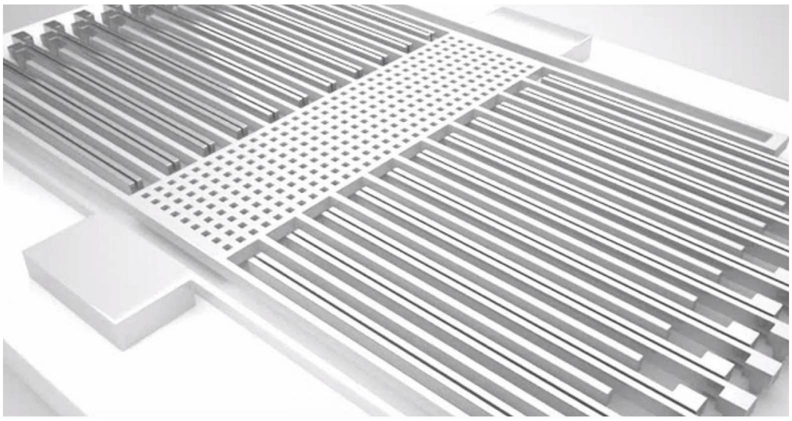


Figure 1.12: A MEMS accelerometer microscopic geometry.

in a direction perpendicular to both the current and to the magnetic field). Finally, the MEMS accelerometer exploits different microelectronic fabrication techniques. These techniques create mechanical sensing structures of microscopic size, typically on silicon. When coupled with microelectronic circuits, MEMS sensors can be used to measure physical parameters, such as acceleration. One possible MEMS building geometry is shown in Figure 1.12.

1.4.2 Gyroscope

The physical quantity measured by a gyroscope is the angular rate. From an historical point of view, in 20th century the first mechanical gyroscopes were including a spinning wheel or disc suspended within at least three gimbals (i.e., supporting rings) connected through joints placed orthogonally to each other. In this system, the gimbals, which are nearly frictionless, isolate the spinning mass which tends to keep its attitude according with the inertia principle. When an external torque is applied, the rotating mass tries to maintain its orientation, so the gimbals adapt their configuration. This allows the attitude estimation by measuring the inclination angles taken by the gimbals. However, these gyroscopes have several disadvantages such as the need for expensive moving parts with high accuracy and considerable size: this makes them unsuitable

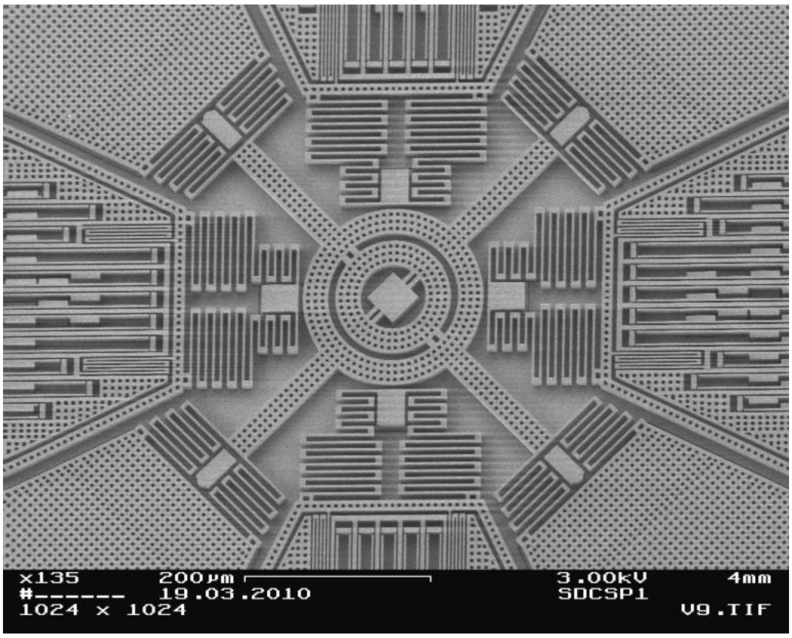


Figure 1.13: A MEMS gyroscope microscopic geometry.

for many applications. Moreover, the gimballed gyroscopes are subject to the *gimbal lock* problem, namely the loss of one degree of freedom when two of the gimbals are on the same line. Thanks to innovations in electronics and mechanical design, new types of gyroscopes have been introduced in the market such as: (i) Ring Laser Gyroscope (RLG); (ii), Fiber Optic Gyroscope (FOG); (iii) Hemispherical Resonator Gyroscope (HRG); (iv) Gyrostat; (v) Magneto-Hydro-Dynamics (MHD) effect-based gyroscope; and (vi) Dynamically Tuned Gyroscope (DTG). All these gyroscopes are very reliable and accurate but, for this reason, their cost remains high.

As for MEMS accelerometers, MEMS gyroscopes guarantee a good trade-off between accuracy, small size, energy efficiency and cost-effectiveness. These gyroscopes usually exploit a vibrating proof mass or the tuning fork principle. In both these designs, a vibrating element is used to measure the *Coriolis* force. The idea is to determine the angular velocity exploiting the physical principle for which a vibrating element tends to maintain its (vibrating) status even when its supporting structure is rotating. The Coriolis apparent force f_c is generated over the support structure and, by measuring it, one can estimate the angular rate ω (dimension: [rad/s]) according with the following equation:

$$f_c = -2m(\omega \times v) \tag{1.6}$$

where m is the proof mass (dimension: [kg]) and v is its velocity (dimension: [m/s]). MEMS gyroscopes can be designed by adopting several microscopic geometries in order to detect the Coriolis force. In Figure 1.13, one of these structures is shown as an illustrative example.

1.4.3 Magnetometer

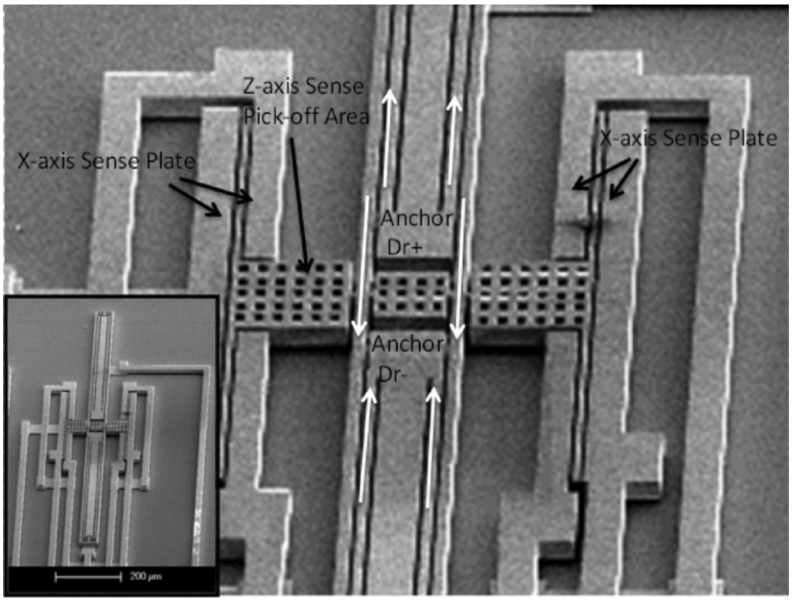


Figure 1.14: A MEMS magnetometer microscopic geometry.

A magnetometer or magnetic sensor is an instrument that measures a magnetic field which can be generated by the earth (i.e., the magnetic north pole) or by a local magnetic source, such as a ferromagnetic element or an electric circuit. The compass is considered the first scalar magnetometer, as it indicates the magnetic source direction.

The first magnetometer able to measure the magnetic field intensity (magnitude) was invented by Carl Friedrich Gauss in 1833. Modern (three-axial) magnetometers are able to measure both the intensity and the direction of the magnetic field, namely the magnetic field vector. Magnetometers cannot be considered inertial sensors, since their measurements depend on an external physical quantity. Nevertheless, they do not rely on the inertia principle, are usually embedded in IMU to support orientation estimation and, additionally, to return the rotation with respect to the global reference system. Magnetometers are built with many different designs, such as (i) Spin-Exchange Relaxation Free (SERF) atomic, (ii) Fluxgate, (iii) Proton precession magnetometers, (iv) Hall effect, (v) Magnetoresistive magnetometers, (vi) Optical magnetometers, and (vii) Superconducting Quantum Interference Device (SQUID) magnetometers. Some of these can be classified as scalar (iii, vi), vectorial (i, ii, iv, v), and for both scalar and vectorial sensing (vii).

Once again the MEMS technology have lead to the magnetometer diffusion in commercial devices due to its cost-effectiveness and small size. A MEMS magnetometer exploits many physical effects to sense the magnetic field. Lorentz force-based MEMS design is one of the most common ones: these magnetometers are based on the effect produced by the Lorentz force, namely a combination of electric and magnetic fields-induced force on a point charge. This sensors rely on the motion of the structure due to the Lorentz force acting on the current-carrying conductor in the magnetic field. The mechanical motion of the micro-structure is sensed electronically. The mechanical structure is often driven to its resonance in order to obtain the maximum output signal. Piezoresistive and electrostatic transduction methods can be used in the electronic detection. In Figure 1.14 (image from [63]), a Lorentz force-based MEMS magnetometer is shown.

1.4.4 Errors

Actual MEMS accelerometers and gyroscopes cannot measure accelerations and angular rates without error. One can define this error as the difference between the output provided by the sensor and the real physical input (i.e., the force and/or the rotation impressed on the sensor). It is possible to categorize the error origin in

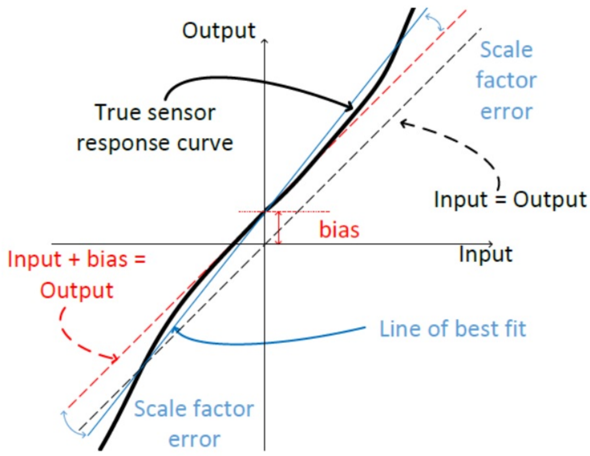


Figure 1.15: Errors introduces by bias and scale factor errors in the IO relationship.

stochastic and *systematic*. In this section, the principal errors in low cost IMUs are investigated. The stochastic (or random) errors are due to non-predictable noise which cannot be avoided ever with accurate calibration procedures. The noise in MEMS sensors, such as in the major part of electronic circuits, is due to white noise. In addition, flickering noise, also called pink noise, and Brownian noise (due to the molecules which randomly collide with the proof mass) are present. Quantization noise (i.e., the error caused by the truncation of the quantization residuals) is also a source of error. These errors are usually taken into account in the stochastic model of the sensor.

Systematic errors are caused by deterministic sources. It is possible to reduce these errors by calibrating the sensor before the measurements. Similar sensors are usually factory-calibrated once. These calibration procedures are made in the same way over every MEMS IMU which belongs to a specific model. In order to obtain a more accurate, but more expensive, IMU, it is possible to calibrate every sensor by taking into account its specific deterministic noise. One of these noise is *bias*. It is a constant quantity added to the output without a real physical cause and, for this reason, it is called also zero-g offset in accelerometers and zero-rate in gyroscopes. In Figure 1.15 is shown the bias effect on the input/output relationship. In Figure 1.15, the *scale factor* error (or *sensitivity tolerance*) is also shown: this is an error introduced in the conversion between the sensor output (i.e., a voltage difference or a current) and the physical input. This error can be produced by a physical misalignment between the inertial triad, which means that the axes are not exactly orthogonal to each other so that the acceleration/angular rate over one IMU axis is split over two MEMS sensor axes. Another scale factor error error can be introduced by temperature variation. In order to avoid this limitation, a temperature MEMS sensor is often included in the IMUs to allow adaptation depending on the temperature changes. Furthermore, higher order errors, such as non-linearities, are present in MEMS sensors, but they are not usually taken into account in low-cost sensor models. These can be reduced during the manufacturing phase by adopting specific strategies in order to make MEMS components more stable.

Even if it is not a real inertial sensor, the magnetometer is subject to several noises. The most challenging problem to deal with is the magnetic disturbance impressed on the sensors by IMU ferromagnetic components and by external magnetic sources, such as electronic circuits or metal structures. Long standing external magnetic sources can be disruptive for magnetometers because they can magnetize part of the sensor, causing an uncontrollable constant bias over the magnetic field sensing.

1.5 Pedestrian Inertial Navigation System (INS)

In recent times, an increasing interest about pedestrian INS has emerged, because of the multiple applications of these systems. In particular, the modern life style leads people to live in indoor environments most of the daytime [64]. The necessity to track these users in indoor environments, together with the advent of innovative MEMS sensors, have enabled a massive research in this field. A pedestrian INS includes a IMU and a processing unit. It is designed to provide navigation information regardless of the specific environment (i.e., indoor/outdoor scenarios, infrastructure availability, map knowledge, etc.). A pedestrian INS is included in the *strapdown* INS category, which means that the IMU is “strapped” to the user body without the necessity of gimbals or other suspended platforms.

Pedestrian Dead Reckoning (PDR) is a navigation technique which relies on the classification of steps pattern to support navigation. According to [65], inertial navigation systems for PDR include three main phases: (i) the orientation estimation; (ii) step detection; and (iii) step length estimation. Various strategies can be applied in each phase, according to the specific operational field, sensor characteristics, and the IMU placement over the user body. A detailed description of these three phases is provided in Subsection 1.5.1, Subsection 1.5.2, and Subsection 1.5.3.

1.5.1 Orientation Estimation

Orientation estimation through inertial signals represents a challenging task to perform. The idea is to fuse the information which comes from the acceleration, such as the estimation of the gravity vector, with the angular velocity collected by the gyroscope, which can provide an information about the angle with respect to every axis. Given that nowadays an IMU usually integrates a magnetometer, which measures the magnetic field strength along every axis, it is also possible to estimate the magnetic north direction and, thus, to translate the reference system from the sensor reference system to the world reference system. An IMU which embeds a magnetometer is also called Magnetic, Angular Rate, and Gravity (MARG). The algorithms which estimate the IMU orientation are called Attitude and Heading Reference System (AHRS) algorithms.

Several ways to perform orientation estimation have been proposed in the literature. The main problem behind this estimation is the trade-off between accuracy and computational efficiency, due to portability needs and battery's finite energy. Most algorithms rely on Kalman filtering to implement sensor fusion [66, 67, 68]. The high computational needs and the dependency on accurate sensor error models make this approach hard to implement and barely suitable to different situations and different types of sensors.

Another proposed solution is complementary filtering, an implementation of which is presented in [69]. Complementary filters are based on the idea of removing noise by using “complementary" Low Pass (LP) and High Pass (HP) filtering. Since gyroscope-based orientation estimation has a good accuracy at high frequencies and accelerometer-based orientation estimation has a good accuracy at low frequencies, by HP filtering the former and LP filtering the latter, one can remove noise from both estimates and, afterwards, reconstruct the signal by fusing the two filtered estimates.

A good trade-off between computational complexity and accuracy is represented by the gradient descent algorithm proposed in [70], which consists in minimizing an error function representing the distance between (i) the orientation estimated using magnetometer and accelerometer and (ii) the orientation estimated using a gyroscope. Estimates from the two methods are related by a constant β, which is linked to the gyroscope error magnitude. The larger the gyroscope error, the higher the weight assigned to the magnetometer/acceleration-based orientation estimate. At the opposite, if the latter estimation is affected by higher error, β has to be reduced to rely more on the gyroscope-based orientation estimate. This filter has been used during the present research work to estimate the orientation collected through the Shimmer sensor-based INS. More details will be given in Chapter 2.

1.5.2 Step Detection

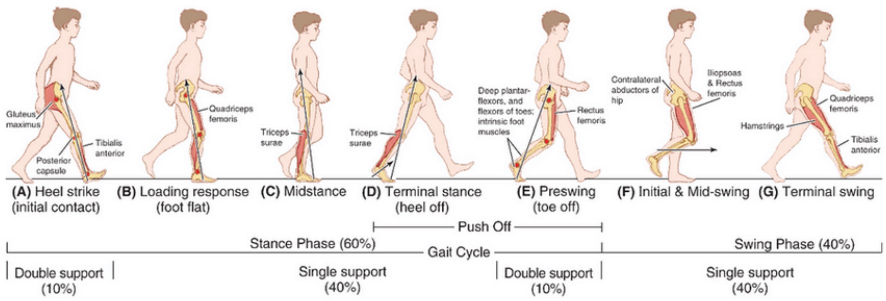


Figure 1.16: Single cycle walking gait, begin and ending at heel strike.

Gait segmentation, with identification of the various phases, is a fundamental part of pedestrian navigation. By localizing the principal events of each gait cycle, one can split a stride in its constituting phases. A normal walk is usually composed of by the phases shown in Figure 1.16 (image from [71]). The principal instants which have to be detected during a walk in order to perform inertial navigation are: Heel Strike (HS) (i.e., the moment in which the foot is touching the ground) and the Toe Off (TO) (i.e, the instant in which the foot is detached from the ground). This segmentation leads to the possibility of identifying, for the right and the left sides, the phase in which the foot is steady on the ground as the interval between an HS and a TO, denoted as “stance." Similarly, “swing" denotes the interval between the TO and the successive HS and represents the time interval during which the foot is in the air.

When the sensor is foot-mounted, step detection algorithms allow the use of fundamental drift-removal Zero velocity UPdaTe (ZUPT) techniques, based on the principle that, when the foot is steady (namely, on the ground), the integration is not necessary. Therefore, every time a stance is detected, integration stops and the velocity value is set to zero, reducing the drift caused by the integration of noisy signals.

While ZUPT techniques are effective when the sensor is on the foot, when it is differently placed (for example: on the chest, on the pelvis, or on the low back) ZUPT techniques cannot be applied. Nevertheless, when the sensor is in these positions, one can apply models which relate the inertial signals to the steps [72, 73]. A well known approach introduced in [74] is based on the relationship between the vertical accelerations and the HS and TO events. In particular, (local) minima and maxima of the vertical acceleration correspond to TOs and HSs, respectively.

1.5.3 Step Length Estimation

A further phase involves the computation of the length of every step. Because of drift, it is impossible to use a continuous integration. When the sensor is foot-mounted, the use of ZUPT techniques [75, 76, 77] allows to carry out the integration separately step by step, removing noise during stances and, thus, obtaining an accurate step length estimate. The method proposed in [78] relies on the relation between step length, walk frequency, and acceleration variance during one step. This method requires frequency domain evaluation and continuous variance estimation, thus increasing the complexity of the final algorithm.

When the MARG sensor is placed on the chest, the impossibility to use a direct relationship between acceleration on horizontal plane and body displacement requires the application of techniques different from ZUPT, such as those which use the vertical acceleration or vertical displacement as indicators of the horizontal movement. Two possible approaches able to reconstruct efficiently the traveled distance, linking the body movement in the vertical direction with the horizontal displacement, are the ones introduced in [79] and [74] respectively. The *first* method is based on the "inverted pendulum model" of gait, which exploits the relationship between trunk vertical displacement and step length according with the following formula:

$$SL = 2\,\kappa\sqrt{2lh - h^2} \tag{1.7}$$

where κ is a subject-dependent calibration constant (adimensional), l is the test subject leg length (dimension: [m]), and h is the vertical displacement of the Center Of Mass (COM) during one step (dimension: [m]).

The *second* method consists in taking advantage of the following relation from the vertical trunk acceleration and the Step Length (SL) [74]:

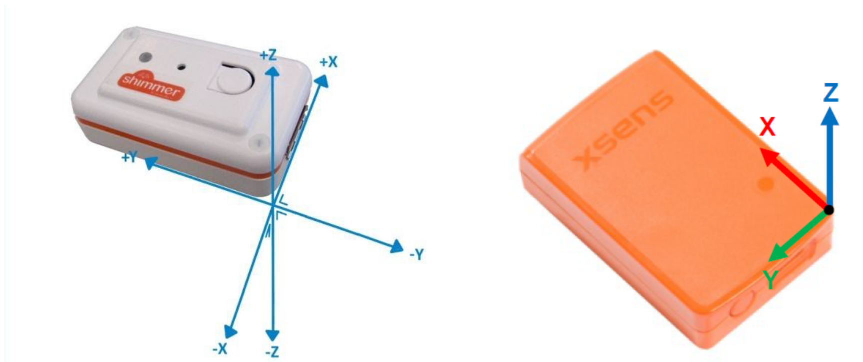
$$SL = K\,\sqrt[4]{a_{z,\max} - a_{z,\min}} \tag{1.8}$$

where: $a_{z,\max}$ and $a_{z,\min}$ are the maximum and minimum vertical accelerations (dimension: $[\mathrm{m/s^2}]$) during a single step; and K is a subject-dependent calibration constant (dimension: $[\mathrm{m^{1/4}s^{1/2}}]$). The second method has a higher reliability than the previous one. In fact, in the first method the calculations of the step length depend on sensor displacement, obtained by integrating the vertical acceleration: this introduces additional noise in long walking sessions. A comparison between these two methods and other solutions, which adopt waist mounted IMU-based step length estimations, is presented in [80].

1.5.4 Shimmer and Xsens IMUs

The algorithms introduced in the following chapters are based on real data collected through IMUs. In this work, the following two sensor platforms have been adopted.

- **Shimmer2r IMU:** The Shimmer device [81, 82, 83, 84] used in this work, with its coordinate reference system, is shown in Figure 1.17(a). It is a small and low-power wireless IMU, equipped with: a TI MSP430 CPU (up to 8 MHz),



(a) (b)

Figure 1.17: IMU and their reference systems: (a) Shimmer2r IMU and (b) Xsens MTw Awinda IMU.

with 10 KB RAM and 48 KB Flash memory; Bluetooth (Roving Networks RN-42) and IEEE 802.15.4-compliant (TI CC2420) radios; an integrated (up to 2 GB) microSD card slot; 3 color status LEDs; a 450 mAh rechargeable Li-ion battery; and a three-axial accelerometer (Freescale MMA7361). External sensing modules can be connected, such as the 9DoF Kinematic Sensor expansion module, which is equipped with a three-axial gyroscope (InvenSense 500 series) and a three-axial magnetometer (Honeywell HMC5843). The device supports various discrete sampling rates (up to 1024 Hz) and can be easily attached to body segments through Velcro straps.

- **Xsens MTw Awinda IMU:** The Awinda IMU [85, 86] is shown in Figure 1.17(b). The MTw is a small (34.5 x 57.8 x 14.5 mm, W x L x H), light (27g including the LiPo battery [battery = 9g]), highly accurate wireless inertial and magnetic sensing unit with pressure measuring capabilities. The MTw communicates with the PC using the unique Awinda radio protocol developed by Xsens. The Awinda protocol guarantees very accurate ($\leq$ 10 μs difference) time synchronization between multiple MTw's comparable to the wired measurement systems from Xsens. The embedded three-axial accelerometer has a full scale equal to ± 160 m/s/s, an internal sampling rate equal to 1800 Hz, and a user-ready sampling rate equal to 150 Hz. The three-axial gyroscope has a full scale equal to ± 1200 deg/s, an internal sampling rate equal to 1800 Hz, and a user-ready sampling rate equal to 150 Hz. The three-axial magnetometer has a full scale equal to ± 1.5 Gauss, an internal sampling rate equal to 120 Hz, and a user-ready sampling rate equal to 60 Hz. The barometer has a full scale in the range 300-1100 mBar. The MTw Development Kit includes multiple MTw's, the Awinda Station, the Awinda USB Dongle, and click-in body straps, making it suitable to measure human body motion and orientation.

1.5.5 Related Works

As mentioned above, progress in MEMS-based inertial sensors have enabled the development of new motion analysis methods to accurately reconstruct a subject's movements in two or three-dimensional spaces, without the use of external infrastructures and with reasonable costs. The personal navigation research field has widely benefited from this evolution.

The first attempts to develop INS based on MEMS sensors exploited foot's movement characteristics during the gait in order to limit the drift. In recent years, several solutions based on foot-mounted sensors have been proposed [65, 87, 88]. By using this particular sensor placement, it is possible to achieve an accurate gait segmentation and to apply the ZUPT technique to reduce the drift. By using this method it is also possible to increase the step length estimation accuracy, since the traveled distance between a stance phase and the next one can be estimated through direct integration of the de-drifted foot velocity. The main drawbacks of this algorithm are the fast orientation changes involved in the foot movement, which may introduce errors in the IMU's measurements. Moreover, sudden fluctuations may degrade the performance of orientation filters, which usually fuse data from different sensors (namely, accelerometer, gyroscope, and magnetometer) in order to compute reliable 3-dimensional orientations: this leads to error accumulation in the estimation of a pedestrian's traveled path. In [89], Jimenez *et al.* analyze different pedestrian navigation systems, highlighting the advantages of sampling the heading angle, measured with a foot-mounted sensor during the stance phase of gait, with respect to double integrating the acceleration to derive both step length and walking direction. Nevertheless, it is important to remark that foot orientation is not always aligned with the subject's movement direction.

The placement of a single IMU on the lower back, close to the COM of the body, is a widely adopted approach to the design of pedestrian navigation systems. In fact, in this position the sensor's movements are highly dependent on the trunk's biomechanical characteristics and, therefore, are limited with respect to those acquired by the foot-mounted sensor. During gait, the heading angle measured through trunk positioned sensors (i.e., on the lower back, on the chest, on the pelvis, etc.) is smoother and more representative of the actual movement direction [90]. The estimation of the traveled path in navigation systems based on trunk-mounted sensors is achieved by sampling the heading angle and updating the position by propagating the steps' lengths along the estimated walking direction [90, 91]. The main limitation of this class of algorithms is the accuracy in step length estimation. Several approaches have been investigated in order to correlate the signals measured from a trunk-mounted IMU with the frontal displacement [74, 79, 92]. The absence of a zero velocity phase and the poor results obtained by a direct integration techniques have lead to the development of empirical formulae, which usually correlate the vertical acceleration/displacement of the trunk with the frontal movement.

Chapter 2

Pedestrian INS: Single Sensor Approach

A journey of a thousand miles begins with a single step.

- Lao Tzu

The main limitation of inertial navigation is represented by the inertial signals' drift. As presented in Subsection 1.4.4, multiple noise sources introduce errors in the estimated values, namely orientation and displacement, and, consequently, lead to inaccurate results (i.e., a wrong user's position estimation). For this reason, innovative Inertial Navigation Systems (INSs) have to rely on algorithms able to obtain an accurate position estimation, even from a single sensor, by exploiting the information about the Inertial Measurement Unit (IMU) placement on the user body.

In this chapter, two novel algorithms for pedestrian navigation are presented. They both use data collected by a single IMU and differ by the position in which the IMU is placed. The first algorithm, denoted as Enhanced-Pedestrian Dead Reckoning (EPDR), relies on a IMU placed on the foot, whereas with the second one, denoted as De-Drifted Propagation (DDP), the IMU is placed on the chest. The aim of the overall approach is to exploit characteristic features of the collected inertial signals to extrapolate navigation information such as heading, step length, gait phases, etc. The research activity introduced in this chapter has been published in [93].

This chapter is organized as follows. Section 2.1 is dedicated to the description of the proposed algorithms, namely DDP and EPDR. In Section 2.2, experimental setup for both DDP and EPDR algorithms is presented, highlighting the characteristics of the chosen scenarios. In Section 2.3, the performance results with the two algorithms are directly compared. Finally, Section 2.4 provides conclusions.

2.1 Single Sensor-based Algorithms

Figure 2.1: Sensor position: (a) chest-mounted IMU (DDP) and (b) foot-mounted IMU (EPDR).

As introduced in Section 1.5, a pedestrian INS relies on three main phases: orientation estimation, step detection, and step length estimation. The algorithms proposed in this chapter adopt the Madgwick *et al.* “Gradient descent algorithm” [70] (see Subsection 1.5.1) for orientation estimation. In this Attitude and Heading Reference System (AHRS) algorithm, there is a key parameter, denoted as β, which relates the orientation estimations calculated with two methods, namely the one based on accelerometer and magnetometer and the one based on the gyroscope. The larger the gyroscope error, the higher the weight assigned to the magnetometer/acceleration-based orientation estimate. At the opposite, if the latter estimation is affected by higher error, β has to be reduced to rely more on the gyroscope-based orientation estimate.

Given that the two proposed navigation algorithms are based on signals collected by the IMU placed in two different positions over the user body, namely on the chest and on the foot, the chosen step detection and step length estimation strategies are different. The EPDR algorithm adopt a Zero velocity UPdaTe (ZUPT) technique to identify the steps and, together with the acceleration double integration, for the step length estimation. The DDP algorithm, instead, relies on the vertical acceleration-based Heel Strike (HS) and Toe Off (TO) events identification to detect the steps and adopts the Weinberg algorithm (see Subsection 1.5.3) to perform the step length estimation.

In this chapter, the following notation is used: $k = 1, 2, \ldots$ denotes the step or stride, depending on the used algorithm (this will be clear in the following); $i = 1, 2, \ldots$ is the sample index of the inertial signal (the sampling frequency is 102.4 Hz, so that each sample correspond to a time interval of approx $(102.4)^{-1} \cong 9.76$ ms); $\underline{\boldsymbol{a}}(i) = (a_x(i), a_y(i), a_z(i))$ denotes the (vector) acceleration signal; $\underline{\boldsymbol{\omega}}(i) = (\omega_x(i), \omega_y(i), \omega_z(i))$ denotes the angular velocity signal; and $\underline{\boldsymbol{m}}(i) = (m_x(i), m_y(i), m_z(i))$ denotes the magnetic field signal.

2.1.1 De-Drifted Propagation (DDP)

This algorithm relies on a single chest-mounted sensor, whose placement is shown in Figure 2.1 (a). In Figure 2.2, the block diagram of the DDP algorithm is shown. When data transmitted through a Bluetooth connection are received on the laptop, the AHRS algorithm computes the sensor's orientation in the Earth (W) reference frame, at every sample i, in terms of the quaternion $\underline{\boldsymbol{q}}^{(\mathrm{W})}(i)$. After the quaternion has been transformed into a rotation matrix $R^{(\mathrm{W})}(i)$, by multiplying $R^{(\mathrm{W})}(i)$ and $\underline{\boldsymbol{a}}^{(\mathrm{S})}(i)$ (i.e., the raw acceleration in the sensor reference frame S), the raw acceleration in the Earth frame, denoted as $\underline{\boldsymbol{a}}^{(\mathrm{W})}(i)$, can be computed. The linear acceleration

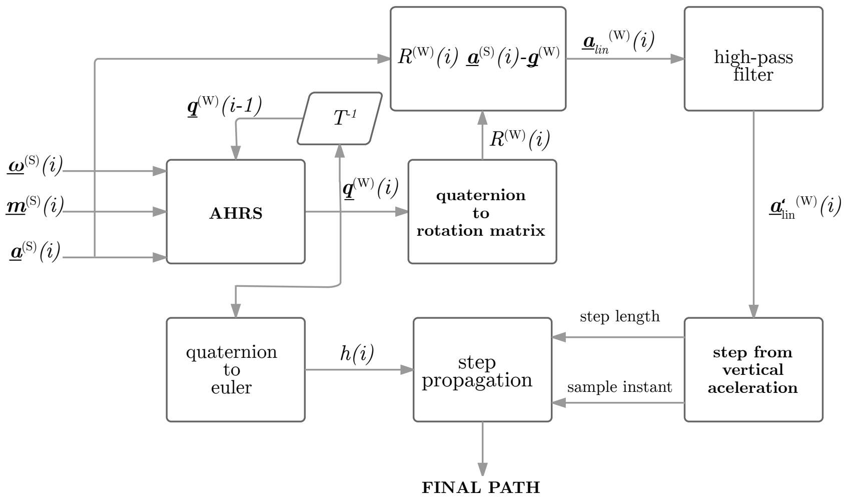


Figure 2.2: Block diagram of the DDP algorithm.

$\underline{a}_{\text{lin}}^{(\text{W})}(i)$ is thus obtained by subtracting the gravity acceleration $\underline{g}^{(\text{W})} = (0, 0, g)$ from $\underline{a}^{(\text{W})}(i)$. A Butterworth High Pass (HP) filter with a cut-off frequency of 0.1 Hz is then applied to $\underline{a}_{\text{lin}}^{(\text{W})}(i)$, obtaining $\underline{a}'^{(\text{W})}_{\text{lin}}(i)$. At this point, step identification is carried out by identifying all peaks and valleys in the $a'^{(\text{W})}_{z,\text{lin}}(i)$. Considering the k-th step, the sample corresponding to the local maximum, denoted as $i_{\text{max}}^{(k)}$, is associated to a HS, whereas the sample corresponding to the consecutive local minimum, denoted as $i_{\text{min}}^{(k)}$, identifies the contralateral TO event. The resulting classification is visible in the upper plot of Figure 2.3. By applying Equation (1.8), the k-th step length SL_k is finally calculated. It is important to remark that this method is more accurate than the one based on the inverse pendulum model of gait, which is less reliable in step identification during navigation, especially during curves.

In the next step, according to Figure 2.2, the heading angle signal $\{h(i)\}$ is obtained by converting the quaternions (generated by AHRS) to Euler angles (i.e., heading, attitude, and bank). A step propagation algorithm, which consists in sampling the heading angle during a stable orientation phase and propagating the position by a segment of length equal to SL_k along the direction corresponding to the heading signal,

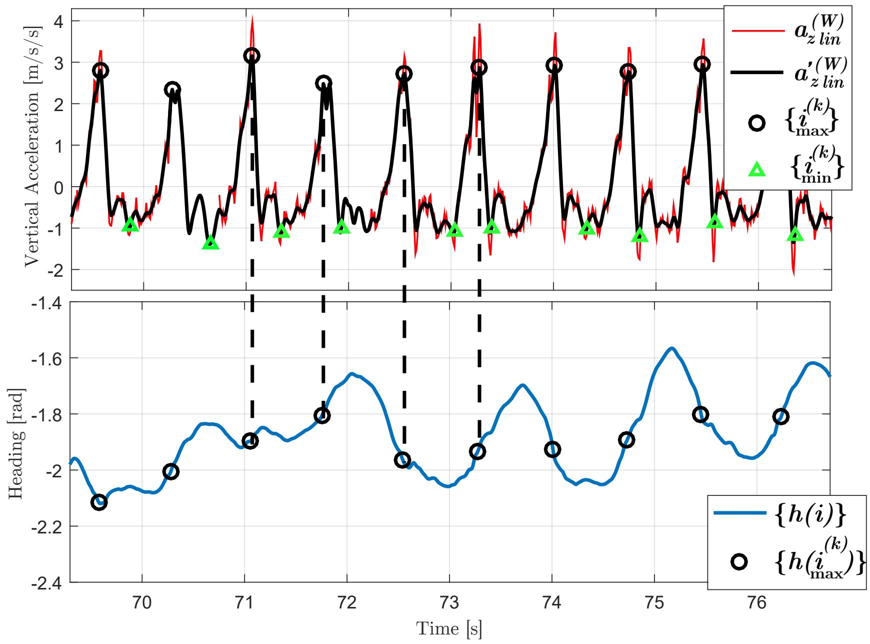


Figure 2.3: In the upper plot, the vertical acceleration signal before ($a^{(\mathrm{W})}_{z,\mathrm{lin}}(i)$) and after ($a'^{(\mathrm{W})}_{z,\mathrm{lin}}(i)$) filtering is shown. In both cases, it is shown the HS instants $i^{(k)}_{\mathrm{max}}$, which correspond to local maxima, and the TO instants $i^{(k)}_{\mathrm{min}}$, which correspond to local minima. In the lower plot, the corresponding heading signal is shown, with circles in correspondence to instants ($i^{(k)}_{\mathrm{max}}$).

is then used to estimate the traveled path. As shown in the lower plot of Figure 2.3, $\{h(i^{(k)}_{\mathrm{max}})\}$ are the selected values of the heading angle signal corresponding to HS instants, as this is the most identifiable gait phase event and is highly reproducible.

2.1.2 Enhanced-Pedestrian Dead Reckoning (E-PDR)

This algorithm relies on data collected from the foot-mounted sensor, as shown in the configuration (b) in Figure 2.1. As shown in Figure 2.4, sensor data are processed with the same AHRS method recalled in the previous subsection. In particular, the

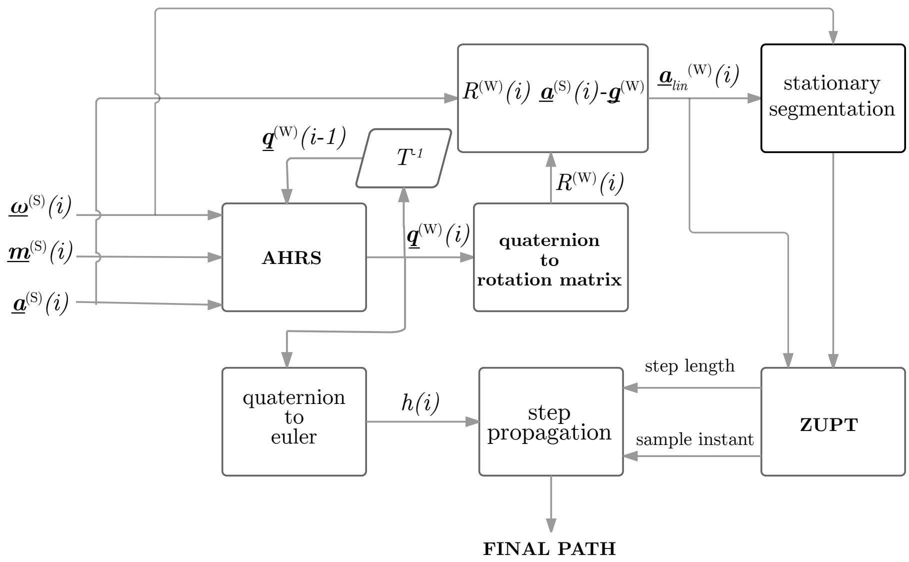


Figure 2.4: Block diagram of the EPDR algorithm.

sequence of orientation quaternion $\{\underline{\boldsymbol{q}}^{(W)}(i)\}$ is first computed and, then, the linear acceleration $\{\underline{\boldsymbol{a}}_{\text{lin}}^{(W)}(i)\}$ is derived. In order to apply the ZUPT technique, the following signals are introduced:

$$
\begin{aligned}
|\omega(i)| &= \sqrt{\omega_x(i)^2 + \omega_y(i)^2 + \omega_z(i)^2} \\
a_{\Sigma}(i) &\triangleq |a_{x,\text{lin}}(i) + a_{y,\text{lin}}(i) + a_{z,\text{lin}}(i)|
\end{aligned}
\tag{2.1}
$$

where: $|\omega(i)|$ is the Euclidean module of the angular velocity and $a_{\Sigma}(i)$ is the absolute value of the sum of the acceleration components. From these signals it is possible to identify the central part of a stance phase:[1] in fact, both of them are approximately equal to zero when the foot is stationary. In particular, the following indicator sequence

[1]The central stance phase notation as been used to highlight that it has been considered only the interval in which the foot is flat on the ground, discarding the initial and terminal part of a complete stance phase.

is defined:

$$\gamma(i) = \begin{cases} 1 & \text{if} \quad a_\Sigma(i) < th_a, |\omega(i)| < th_\omega \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

More specifically, $\gamma(i) = 1$ when a_Σ and $|\omega|$ are below two proper thresholds, denoted as th_a and th_ω, respectively. Additionally, a check on the duration of each central stance phase is performed, identifying the first and the last samples. For each central stance phase, these samples are denoted, respectively, as $i_{\text{start}}^{(j)}$ and $i_{\text{end}}^{(j)}$: the corresponding time duration can be expressed as $T^{(j)} = t(i_{\text{end}}^{(j)}) - t(i_{\text{start}}^{(j)})$. In order to avoid considering spurious peaks, the central stance phases with durations shorter than a proper threshold T_{min} are rejected. If $T^{(j)} < T_{\text{min}}$ is discarded, than the indicator sequence $\{\gamma(i)\}$ is updated setting to zero the corresponding samples (from $i_{\text{start}}^{(j)}$ to $i_{\text{end}}^{(j)}$ of the discarded j-th central stance phase). The chosen values of the considered thresholds are: $th_\omega = 50$ rad/s; $th_a = 9$ m/s^2; and $T_{\text{min}} = 0.30$ s, respectively. The result of this stationary detection procedure is shown in the upper plot of Figure 2.5. Considering only the swing phases, one can integrate the linear acceleration $\{\underline{\boldsymbol{a}}_{\text{lin}}^{(\text{W})}(i)\}$ and obtain the linear velocity $\{\underline{\boldsymbol{v}}^{(\text{W})}(i)\}$ by applying ZUPT.

At this point, the length of the k-th stride is calculated by integrating the velocity $\{\underline{\boldsymbol{v}}^{(\text{W})}(i)\}$ between the end of a still phase $i_{\text{end}}^{(j)}$ and the beginning of the successive one $i_{\text{start}}^{(j+1)}$. The velocity is integrated only along the x and y coordinates, as only 2D navigation is considered in this algorithm.

At this point, the propagation can be applied by sampling the heading angle signal $\{h(i)\}$ calculated from the sequence of quaternions $\{\underline{\boldsymbol{q}}^{(\text{W})}(i)\}$. The orientation is sampled at the instant $i_{\text{cen}}^{(j)}$, defined as the middle point of the time interval corresponding to the j-th central stance phase (in other words, $i_{\text{cen}}^{(j)} = (i_{\text{start}}^{(j)} + i_{\text{end}}^{(j)})/2$, rounded to the nearest integer). The correspondence between central stance phase and $h(i_{\text{cen}}^{(j)})$ is shown in the lower plot of Figure 2.5. It is apparent that one can sample the heading angle during the more stable portion of a stride.

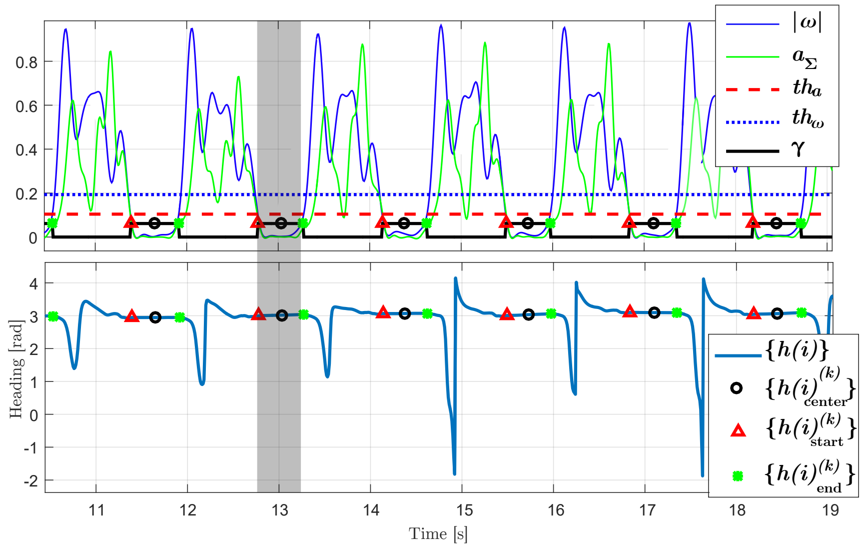


Figure 2.5: In the first plot, the signals a_Σ and $|\omega|$ signals and the corresponding threshold th_a and th_ω are shown. Data are normalized for clarity. Furthermore, the signal γ, corresponding to 1 during central stance phases and to 0 during non-stationary phases, is visualized. In the second plot, the heading signal is shown, highlighting step milestones (namely HS, TO, and central stance phase) localized with stationary routine. In gray, a central stance phase is shown.

2.2 Experiments

Data are collected using, as IMU, a Shimmer 2r unit with 9 Degrees Of Freedom (9DOF), composed by an accelerometer, a gyroscope, and a magnetometer. Hardware technical specs are reported in Subsection 1.5.4. The chosen sampling frequencies are: 102.4 Hz for the accelerometer and the gyroscope; and 50 Hz for the magnetometer. The collected inertial data are transmitted from the Shimmer sensor to a laptop equipped with an Intel i7-6700HQ processor working at 2.6 GHz, 8 GB RAM (DDR3, 1600 MHz). The data are sent using a bluetooth connection and then processed with Matlab® 2016a software.

During still phases (i.e., when the body is not moving) a problem is represented by wrong step identification. To avoid this problem, which especially occurs at the beginning and at the end of the test period, we compare the energy of the accelerometer signal with a properly selected threshold: if the signal energy is lower than the threshold, then a still phase is considered and the step detection algorithm outlined in Subsection 2.1.1 is inhibited.

The test paths have been chosen to be representative for inertial navigation. The first scenario corresponds to an outdoor set up, expedient to investigate the impact of step length estimation over a long walking path. The second scenario encompasses the presence of tight and large curves and electromagnetic disturbances (typical of indoor environments).

In order to evaluate the performance of the proposed DDP and EPDR algorithms, they are tested on closed paths. This allows to use error metrics which are based on the final position. For instance, given that the starting point should coincide with ending point, one can measure the position error as the Euclidean distance between the first (known) position and the second (estimated) one. By measuring the total traveled distance, the precision of the navigation algorithms, in terms of the percentage of the Euclidean distance of the end with respect to the total traveled distance, can be determined. Finally, the orientation performance is evaluated by evaluating the difference between the estimated orientation at the final point with that at the starting point: this difference would be π rad in the absence of errors.

2.2.1 Outdoor Tests

For extensive walking experimental tests, the Olympic running track at the University of Parma campus (Lat. 44°46′03.8″N, Lng. 10°18′41.2″E) is used in order to obtain a precise ground truth. The track length is 400 m. Every test consists of a complete track lap (returning to the starting point) and is performed with two Shimmer sensors mounted simultaneously on the subject body, namely one on the chest and one on the foot, in order to compare the DDP and EPDR algorithms exactly on the same path. Several tasks have been performed: two representative test data (one per navigation algorithm) are shown in Figure 2.6. It is possible to observe that both navigation

Figure 2.6: Comparison between EPDR and DDP performance in the outdoor scenario.

algorithms allow good path reconstruction.

From the results in Figure 2.6, in can be observed that, regardless of drift removal technique, long curves and long travel length lead to error accumulation. The foot-mounted sensor, in particular, due to its faster dynamics, is subject to stronger orientation estimation error with respect to the chest-mounted sensor. The latter has better performance on heading calculation, but step length still depends on the value of K, which varies from subject to subject.

2.2.2 Indoor Tests

Indoor tests are conducted at ground floor of the Department of Information Engineering of the University of Parma. The walking trajectory has been chosen to be representative of inertial navigation specific issues: the path is a closed-loop track, approximately 90 m long, with three sharp bends and four 90° curves. As for the outdoor scenario, many tests have been performed wearing both sensors (one on the foot and one on the chest) simultaneously. Three experimental sessions are shown in Figure 2.7 (a) (DDP) and Figure 2.7 (b) (EPDR). The number of identified steps in Figure 2.7 (a) is twice with respect to those in Figure 2.7 (b), because the DDP

(a) DDP

(b) EPDR

Figure 2.7: Estimated path trajectories, for three test sessions, in indoor environment: (a) DDP algorithm (b) EPDR algorithm.

Table 2.1: Performance with DDP in the indoor scenario.

# Test	ϵ on arrival [°]	ϵ on arrival [m]	Distance [m]	% ϵ on distance
test 1	1.39	2.1	84.50	2.49
test 2	10.8	1.73	86.13	2.01
test 3	26.5	3.49	86.53	4.04
test 4	13.02	0.23	82.37	0.28
test 5	10.53	1.88	83.19	2.26

Table 2.2: Performance with EPDR in the indoor scenario.

# Test	ϵ on arrival [°]	ϵ on arrival [m]	Distance [m]	% ϵ on distance
test 1	8.36	2.54	85.51	2.97
test 2	4.32	1.59	83.07	1.91
test 3	11.78	0.81	83.30	0.98
test 4	6.90	2.76	82.06	3.37
test 5	9.46	2.17	81.06	2.68

algorithm reveals instants at which every foot is on the ground, whereas EPDR, using one foot-mounted sensor, can recognize only one foot dynamics. The higher sampling frequency of the heading angle in the DDP algorithm allows to achieve better performance than EPDR in tight curves, but may introduce noise on a straight path, as discussed in the next subsection.

2.3 Results

In Table 2.1 and Table 2.2, the data collected in the experimental trials in the indoor scenario are summarized. By comparing the two tables, it is possible to highlight the main characteristics of the two solutions. Since several electromagnetic noise sources are present, the DDP algorithm with the chest-mounted sensor is particularly degraded. To limit this negative impact, a small value of β is chosen (in the AHRS gradient descent algorithm) in order to give more weight to the gyroscope signal. The DDP algorithm in based on sampling the heading signal at every step, with a double frequency with respect to EPDR: for this reason, DDP is more subject to spurious noise peaks. On the other hand, the lower heading sampling frequency of EPDR causes a higher error on tight curves, where the orientation signal changes rapidly. Furthermore, a complex path, such as that shown in Figure 2.7, highlights the intrinsic limit of inertial navigation methods: a single measurement error during a step can propagate (during the entire test) and have a detrimental impact over the overall performance.

Table 2.3 summarizes the results in the outdoor tests, focusing on drift and step length estimation. The EPDR's main strategy, based on the identification of stance phases during walk, leads to an effectiveness of 100%, thanks to the double threshold method (presented in Equation (2.2)) and spurious peaks' control, which eliminates false positives. DDP uses vertical acceleration to identify steps and, although this method is more difficult to control, during along steady walk the estimation is correct almost in 100% of the cases. Drift problems for pure inertial tracking still persist, even after the application of drift removal techniques: as can be seen in Figure 2.6, estimated curves are tighter than real ones. EPDR is affected by heading estimation errors during long walking distance because of rapid foot dynamics and, orientation-wise, has a worse performance. At the opposite, EPDR leads to a more accurate estimation of the travelled distance, by direct acceleration measurement, than DDP, in which processing depends on the constant K, which needs to be accurately calibrated over long distances.

Finally, it is important to remark that the proposed DDP and EPDR algorithms can be implemented in an online (i.e., real-time) version, with proper (limited) modification.

2.4 Conclusions

In this chapter, two novel algorithms, denoted as DDP and EPDR, for pedestrian inertial navigation using a single IMU have been compared. EPDR uses data from a sensor placed on the foot, whereas DDP relies on data from a chest-mounted sensor. These algorithms have been developed by exploiting gait segmentation informations provided by well known algorithms together with new approaches which have led to good results according with the literature. Several experiments have been carried out in outdoor and indoor environments, trying to encompass several navigation conditions.

Table 2.3: Navigation performance for both navigation techniques in outdoor environment.

DDP

# Test	ϵ on arrival [°]	ϵ on arrival [m]	Distance [m]	% ϵ on distance
test 1	6.07	4.08	392.33	1.04
test 2	0.71	6.14	396.54	1.54
test 3	17.41	12.6	382.45	3.29

EPDR

# Test	ϵ on arrival [°]	ϵ on arrival [m]	Distance [m]	% ϵ on distance
test 1	22.77	5.18	405.71	1.27
test 2	17.23	5.49	384.90	1.42
test 3	17.20	6.51	343.76	1.82

These surveys have been carried out by collecting data through an IMU connected to a laptop and by analyzing the data with ad-hoc algorithms developed in Matlab code.

In indoor scenarios, it has been observed that electromagnetic sources and, in general, magnetic disturbances may affect significantly the magnetometer readings and, thus, the global reference system estimation. Step recognition and heading sampling represent key phases for correct path reconstruction. The EPDR algorithm has shown good performance in gait segmentation tasks, but the drift cause significant errors in the orientation estimation due to the sensor positioning (i.e., the foot is moving with higher acceleration with respect to the user upper trunk). The DDP algorithm, instead, is based on a stable IMU positioning, which lead to accurate heading estimations. However, it relies on an empirical model to perform the step length estimation, which lead to a drift incrementation over long path.

The obtained results shows that drift can be significantly limited and the achieved performance on long navigation paths confirms the feasibility of a precise inertial navigation system.

Chapter 3

Pedestrian INS: Multiple Sensors Approach

If you're walking down the right path and you're willing to keep walking, eventually you'll make progress.

- Barack Obama

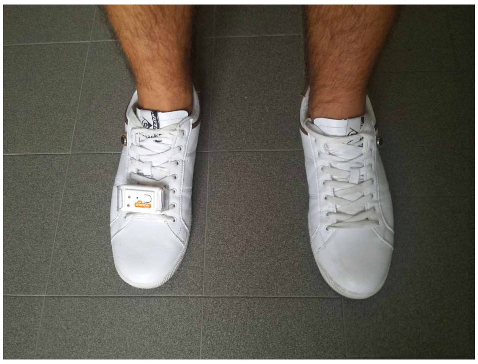
Inertial Navigation Systems (INSs) are often based on a single sensor, in order to provide a easy-to-use system suitable for every user. However, some applications require a higher level of robustness and, therefore, a multiple sensor-based INS is worth considering.

In this chapter, an extensive investigation concerning the sensors' integration and fusion approaches is presented. This analysis leads to the development of a novel navigation algorithm based on the data collected from multiple Inertial Measurement Units (IMUs). These inertial sensors have been placed on the feet in order to exploit the Zero velocity UPdaTe (ZUPT) technique to minimize the negative impact of the drift. A barometric sensor placed in a backpack, has been integrated in order to provide the vertical displacement estimation based on the barometric pressure. The horizontal displacement, evaluated by fusing the information from the IMUs, is later combined with the altitude reference provided by the barometer. As a final prototypical outcome, a hybrid inertial/barometric system able to reconstruct the three-dimensional position of the person wearing the sensors (in both indoor and outdoor environments) has been developed. The system introduced in this chapter has been tested during the *IPIN 2016 Track 2 Competition*, and published in [94].

This chapter is organized as follows. In Section 3.1, an overview on the adopted hardware and the investigated configurations is provided. In Section 3.2, the algorithms based on single/multiple sensors for 2D/3D navigation are presented. Section 3.3 discusses the experimental results obtained using the proposed navigation algorithm. Finally, Section 3.4 provides concluding remarks.

3.1 System Overview

The proposed INS is composed by three basic hardware components: (i) the IMUs attached to the feet through plastic fastenings, (ii) a Freakduino board integrating the barometric sensor in the backpack, and (iii) a laptop, equipped with Mathworks Matlab® and running the developed software for real-time navigation, carried by hand by the test subject in order to monitor his/her position. The IMUs can be placed on the feet according to different configurations. As shown in Figure 3.1, the investigation with multiple IMUs is carried out using three different configurations: configurations 2 and 3, with two sensors, and configuration 4, with four sensors. Configuration 1, shown in Figure 3.1(a), uses only one sensor and is similar to the one introduced in Chapter 1, namely the Enhanced-Pedestrian Dead Reckoning (EPDR) system. It is used as a benchmark to evaluate the improvements obtained by applying a sensor fusion approach to multiple IMUs with respect to the single sensor-based system. The inertial nodes communicate with the laptop via bluetooth connection, whereas the Freakduino is connected to the usb port (i.e., through a cable). The pressure data is transmitted via usb from the Arduino directly to the Matlab script. The board placement has been chosen to maximize the sensor isolation to avoid high temperature and pressure variations caused by environmental modifications during the walk (i.e.,

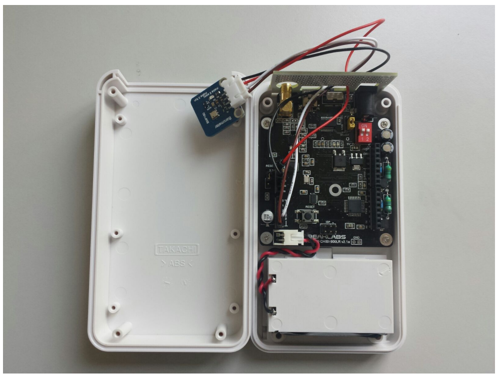


(a)

(b)

(c)

(d)



(e)

Figure 3.1: Sensors configurations: (a) configuration 1: one single IMU attached on one foot; (b) configuration 2: two IMUs, one per foot; (c) configuration 3: two IMUs attached on the same foot; (d) configuration 4: four IMUs, two on each foot; (e) a Freakduino board with the BMP180 barometric sensor connected via I2C connection.

passage from a room to the next, indoor-outdoor transfer, windy environment, etc.).

The proposed system uses, similarly to the INS introduced in Chapter 1, Shimmer 2r IMUs (see Subsection 1.5.4 for details). As mentioned above, the data collected by the IMUs are sent to a laptop, through bluetooth connections, where the navigation algorithm is implemented in real time using Mathworks Matlab®.

The barometer used in the proposed system is the BMP180 from Bosch Sensortech® [95]. The sensor is mounted on a Sunfounder® module compatible with the Arduino GPIO interface, as shown in Figure 3.1(e). A Freakduino board [96] is used to send data, through the usb port, to the laptop. Using the Arduino Matlab toolbox, the sensor data are read and integrated in the navigation algorithm. The chosen sampling frequency for this sensor is 2 Hz: this low sampling rate is sufficient because of the slow altitude variation during a normal walk. The minimum possible sampling period is 25.5 ms using the "ultra-high resolution mode," but it introduces errors during quantization so it has not been adopted. The pressure data are synchronized with the inertial ones using the timestamp provided by the laptop clock, and synchronizing it with the IMUs clocks during the system startup phase. The drift between the sensors clock and the laptop clock during a single data collection is considered negligible.

3.2 Multiple Sensor-based Algorithms

3.2.1 2D INS with a Single Inertial Sensor

First, a description about how each IMU data is processed singularly for navigation purposes is provided—this applies directly to the single sensor configuration in Figure 3.1 (a). In the upper part of Figure 3.2, the block diagram of the single sensor-based 2D navigation algorithm is shown. The single sensor approach is based on the EPDR algorithm introduced in the previous chapter. The basics of this algorithm are now quickly recalled, highlighting the enhancements introduced in order to: (i) increase the performance; (ii) make the proposed approach suitable for the integration with other similar systems; and (iii) make it able to provide real-time estimates.

As for the EPDR approach introduced in Subsection 2.1.2, this algorithm adopts the Madgwick Attitude and Heading Reference System (AHRS) method to estimate

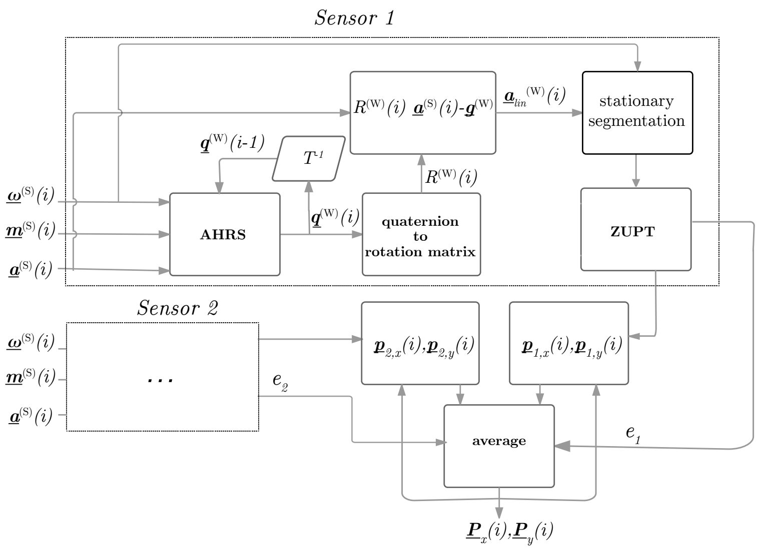


Figure 3.2: Block diagram of the 2D IMU-based navigation algorithm.

the orientation and, then, to compute the linear acceleration $\{\underline{\boldsymbol{a}}_{\text{lin}}^{(W)}(i)\}$. A mentioned above, the ZUPT technique is then applied by computing the signals introduced in Equations (2.1) and generating the indicator sequence $\gamma(i)$, as in Equation (2.2). Then, a check over the duration of each central stance phase is performed by identifying the starting and the ending epoch per every step and by comparing it with a proper threshold $T^{(j)} = t(i_{\text{end}}^{(j)}) - t(i_{\text{start}}^{(j)}) < T_{\text{min}}$ (see Subsection 2.1.2 for details). The chosen values of the considered thresholds (optimized by trial) are: $th_\omega = 50$ rad/s; $th_a = 30$ m/s^2; and $T_{\text{min}} = 0.30$ s, respectively. Considering only the non stationary phase of each foot (i.e., the gait phase in which a foot is moving), one can integrate the linear acceleration $\{\underline{\boldsymbol{a}}_{\text{lin}}^{(W)}(i)\}$ and obtain the linear velocity $\{\underline{\boldsymbol{v}}^{(W)}(i)\}$. By applying the ZUPT technique, which consists in setting to zero all the linear velocity values during a detected stationary phase, the integration drift caused by noisy acceleration signals can be consistently reduced. At this point, the length of every stride is calculated by integrating the velocity $\{\underline{\boldsymbol{v}}^{(W)}(i)\}$ between the end of a stationary phase $i_{\text{end}}^{(j)}$ and the beginning of the successive one $i_{\text{start}}^{(j+1)}$ (i.e., a non stationary phase). The integration is performed only along the x and y coordinates in order to obtain the horizontal displacement. This strategy differs with respect to the one proposed in the EPDR algorithm which is based on the step propagation technique. This allows to obtain the displacement directly from the linear acceleration avoiding the heading sampling. By adopting this technique, it is possible to obtain the foot position as soon as it touches the ground: this leads to an accuracy improvement and allows the real-time position estimation. In addition, it supports the sensor fusion introduced in the following subsections.

3.2.2 2D INS with Multiple Inertial Sensors

In order to combine the measurements from multiple IMUs to obtain a single path estimation, a simple "averaging approach" is adopted. In particular, two initial scenarios with two sensors are considered: (i) one sensor per foot; (ii) both sensors placed on the same foot. The following derivations, carried out considering two sensors, can be generalized to the case of multiple sensors.

In the first scenario, the average is carried out every time the right foot is stationary, i.e., in correspondence to sample i such that $\gamma(i) = 1$ for the inertial sensor placed on the right foot. In particular, by defining the central stance instant for every sensor in the j-th stride as $e_s = i_{\text{cen}}^{(j)}$, where footscript $s \in \{1, 2\}$ refers to the sensor index (namely, 1 is right and 2 is left), the average is computed as follows:

$$\begin{aligned} \underline{\boldsymbol{P}}_x &= \frac{\underline{\boldsymbol{p}}_{1,x}(e_1) + \underline{\boldsymbol{p}}_{2,x}(e_2)}{2} \\ \underline{\boldsymbol{P}}_y &= \frac{\underline{\boldsymbol{p}}_{1,y}(e_1) + \underline{\boldsymbol{p}}_{2,y}(e_2)}{2} \end{aligned} \tag{3.1}$$

where: $\underline{\boldsymbol{p}}_{s,x}$ and $\underline{\boldsymbol{p}}_{s,y}$ are the positions (dimension per axis: [m]) estimated from the single sensor s; and $\underline{\boldsymbol{P}}_x$ and $\underline{\boldsymbol{P}}_y$ are the final position estimates (calculated in correspondence to each right foot stance).

In the second scenario, both sensors are mounted on the same foot: therefore, in this case the stationary events are simultaneously the same for the two sensors. For this reason, it is sufficient to apply the average operation (3.1) when both sensors measure a stationary event (i.e., $\gamma(i) = 1$ for both) to obtain the average position $(\underline{\boldsymbol{P}}_x, \underline{\boldsymbol{P}}_y)$.

In the presence of four sensors, namely two per foot (i.e., configuration 4), the inertial data from the two sensors placed on each foot are first fused with the averaging approach above; then, the estimated paths from the two feet are integrated by using, again, the same averaging approach.

The final average 2D position ($\underline{\boldsymbol{P}}_x$, $\underline{\boldsymbol{P}}_y$) is then used, as current position, i.e, as ($\underline{\boldsymbol{p}}_{1,x}$, $\underline{\boldsymbol{p}}_{1,y}$) and ($\underline{\boldsymbol{p}}_{2,x}$, $\underline{\boldsymbol{p}}_{2,y}$) for the next local computations. Although this procedure introduces little glitches on the paths estimated separately from the sensors, it also allows to reduce the negative impact of noisy data collected by each IMU (i.e., electromagnetic noise that can disturb one of the magnetometers, errors in stationary phase detection due to irregular steps, etc.).

3.2.3 3D INS with Inertial Sensors and Barometer

Finally, the 3D position is estimated by fusing the 2D position with the estimated altitude. The barometric sensor, taking into account the variation caused by the environmental temperature, returns the pressure (dimension: [Pa]). The altitude (dimension: [m]) is then estimated using the following formula [97]:

$$h = 44330 \cdot \left(1 - \left(\frac{\mu}{\mu_0}\right)^{\frac{1}{5.255}}\right) \tag{3.2}$$

where: μ is the measured pressure and μ_0 is the estimated pressure at sea level. During the system initialization, in order to avoid the error caused by the sensor's resolution, N samples are collected in order to identify the absolute altitude of the starting floor (i.e., the floor where the system is turned on). The absolute altitude of the starting floor is then computed as follows:

$$h_0 = \frac{\sum_1^N h(i)}{N}. \tag{3.3}$$

Assuming that the geographical position is unknown, but the starting floor F_0 and the height of each floor F_h (dimension: [m]) are available, in order to correctly identify the current floor F during navigation, the estimated altitude is then compared with the threshold $\eta = F_{\mathrm{h}}/2$, thus leading to the following floor estimate:

$$F = \begin{cases} F + 1 & h(i) - h_0 > F \cdot F_{\mathrm{h}} + \eta \\ F - 1 & h(i) - h_0 < F \cdot F_{\mathrm{h}} + \eta \\ F & \text{else.} \end{cases} \tag{3.4}$$

For example, if the floor height is $F_{\mathrm{h}} = 3$ m, the floor threshold η is set to 1.5 m: if the estimated altitude varies by more than 1.5 m, then a floor change is declared.

The barometer is not precise enough to detect the user height with respect to the sea-level, but it can be used to accurately estimate a floor change. Further details on commercial barometers accuracy are provided in [97].

3.3 Results

An extensive test campaign has been carried out to evaluate the system performance. The obtained results show that during a normal walk both the 2D orientation of the test subject and the floor changes can be correctly estimated. Noisy events, such as electromagnetic disturbances and very rapid dynamics, can still introduce errors, but the integration of multiple inertial sensors highly reduces these nuisances. This is confirmed by the improvement brought by the use of multiple sensors (configurations 2-4 in Figure 3.1) with respect to the case with a single sensor (configuration 1 in Figure 3.1).

In Figure 3.3, the estimated paths, associated with a walk of approximate length equal to 300 m at the ground floor of the Departments of Engineering at the campus of the University of Parma, are shown. The four sensors' configurations in Figure 3.1 (a)-(d) are considered. It can be observed that the fusion of the data coming from multiple sensors allows to improve the performance with respect to the case with a single sensor (configuration 1). In the presence of two sensors, configuration 3 (both sensors on one foot) outperforms configuration 2 (one sensor per foot). Increasing the number of sensors to four (considering configuration 4) further increases the estimation precision.

In Figure 3.4, a multi-floor estimated path is shown, with length approximately equal to 115 m. This test has been carried out in the Building 2 of the Department of

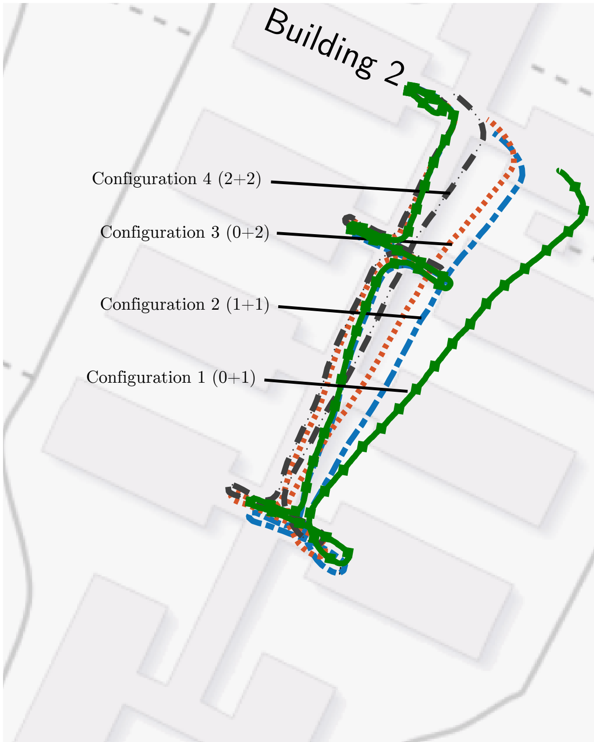
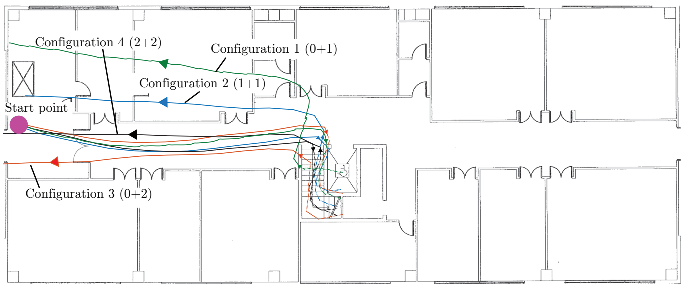
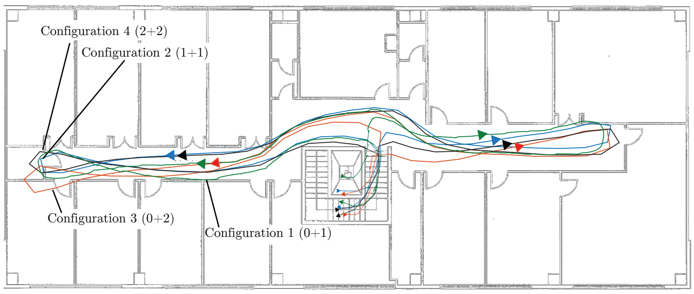


Figure 3.3: Path estimates based on the four different configurations introduced in Section 3.1. The experimental data have been collected during a single walk for a fair comparison.

Information Engineering: it starts from the ground floor, moves to the first floor, and then returns to the starting point in the ground floor. The acquisition set-up is the same as in the Figure 3.3, but in the presence of two floors. In order to evaluate the system performance, a closed loop paths is chosen as shown in both Figure 3.3 and Figure 3.4 (i.e., the test subject returns exactly at the starting point at the end of the track). The estimation error is computed as the absolute and relative difference between the starting and the final points. The obtained results are shown in Table 3.1 for the single floor path (Figure 3.3), and in Table 3.2 for the multi-floor path (Figure 3.4). Although the results presented in Figure 3.3 and Figure 3.4 still show the existence of



(a) Ground floor



(b) First floor

Figure 3.4: Estimated path in the Building 2 (highlighted in Figure 3.3) of the Departments of Engineering and Architecture of Parma. The dotted line represent the path on the stairs. Four different configuration are shown: the green line is the configuration 1; the blue line is the configuration 2; the red line is the configuration 3; and the black line is the configuration 4.

a drift, they highlight the potential of multiple sensors' integration to make personal navigation feasible.

Finally, an outdoor track is considered, in order to use GPS data as a ground-truth to evaluate the system performance. In Figure 3.5, a path with length approximately equal to 500 m is shown. In can be observed that a slight drift affects the path estimated by the inertial system. However, our inertial system guarantees a higher estimated path resolution. More precisely, the inertial system estimates the position every time a stationary phase (i.e., when a foot touches the ground) is detected, whereas the GPS system collects a position estimate approximately every 10 sec. Moreover, the

Table 3.1: Single floor path estimation (path length ∼294.5 m).

Configuration	Final X [m]	Final Y [m]	Relative error [%]
Configuration 1	32.86	-10.35	11.46
Configuration 2	16.39	-5.60	5.8
Configuration 3	14.67	-3.258	5.1
Configuration 4	7.18	0.61	2.4

Table 3.2: Multifloor performance (path length ∼115 m).

Configuration	Final X [m]	Final Y [m]	Relative error [%]
Configuration 1	-1.00	4.282	3.7
Configuration 2	-0.93	-1.45	1.72
Configuration 3	2.18	1.47	2.30
Configuration 4	0.89	0.16	0.78

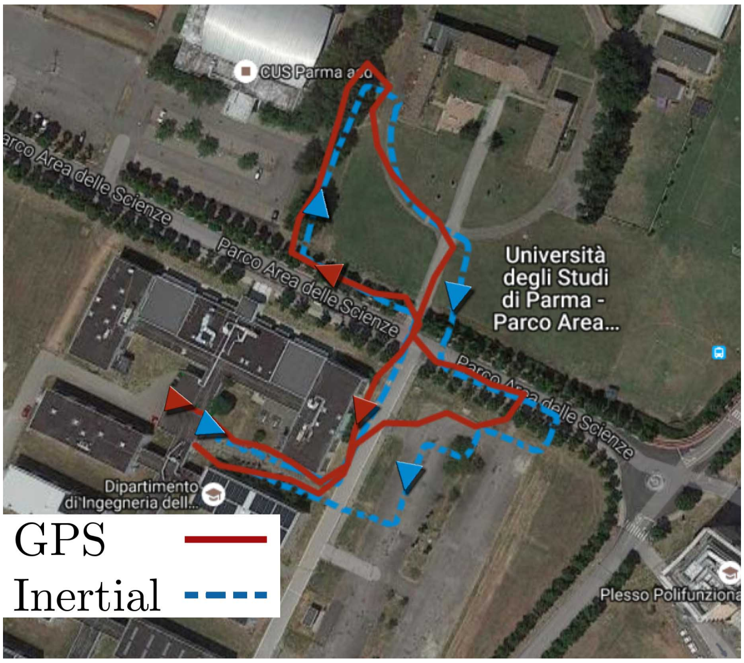


Figure 3.5: Outdoor test: GPS (solid) and inertial (dashed) estimated trajectories.

commercial GPS systems have a precision (in terms of RMS error) of approximately 8 m, which can also justify the difference between the estimated paths in Figure 3.5.

3.4 Conclusions

In this chapter, a novel inertial navigation algorithm able to reconstruct the 3D movement of a subject in both indoor and outdoor environments have been presented. An INS, where 2D navigation is carried out with several IMUs and floor change is estimated through a barometer, has been proposed.

An extensive analysis has been carried out to compare three different configurations with the single sensor configuration presented in the previous chapter, in order to find the optimal strategy to fuse together the paths estimated by up to four inertial sensors. The obtained results show that the positioning error, with respect to the starting position of closed paths, is around 2% of the traveled distance for the configuration with four IMUs, namely *Configuration 4*. It is important to remark that the proposed system has been able to achieve this accuracy performance in real-time, namely providing the position to the user during the walk.

With respect to the system presented in Chapter 2, the configuration which fuses the data from 4 different IMUs leads to an accuracy improvement, but it introduces challenges in the sensor integration. Moreover, the vertical displacement estimation represents a key evolution which leads to a INS able to estimate the user position in a real multi-floor indoor scenario.

Chapter 4

On the Impact of IMU Placement

To improve is to change;
to be perfect is to change often.
- Winston Churchill

The inertial sensing technologies are subject to different errors (e.g., drift, incorrect sensors' positioning, per-subject calibration), which need to be reduced using specific correction techniques in order to make the measurements as reliable as possible. Most of these error reduction strategies depend on the sensors' positioning on the subject's body.

In the previous chapters, the impact of two specific Inertial Measurement Units (IMUs)' placements (namely, on the subject's feet and on the chest) on the performance of purely inertial navigation systems has been investigated. In this chapter, the main focus is put the analysis of the strengths and weaknesses of the previously developed solutions and propose novel hybrid methods, which combine data from IMUs placed on different body segments to improve the accuracy and reliability of the overall pedestrian navigation system.

The novelty of this research consists in combining the strengths of foot-mounted and trunk-mounted IMU-based navigation techniques in order to better estimate the path traveled by the subject. The analysis carried out in order to prove the hybrid approach effectiveness has been also presented in [98].

This chapter is organized as follows. In Section 4.1, an overview of the two navigation systems, based on the use of a single sensor, is provided, together with the definition of the experimental setup. In Section 4.2, the novel method based on sensor fusion is analyzed, emphasizing strengths and weaknesses. In Section 4.3, experimental results are presented, considering both indoor/outdoor scenarios and estimating the computational costs of the proposed solutions. Finally, in Section 4.4 conclusions are drawn.

4.1 System Overview

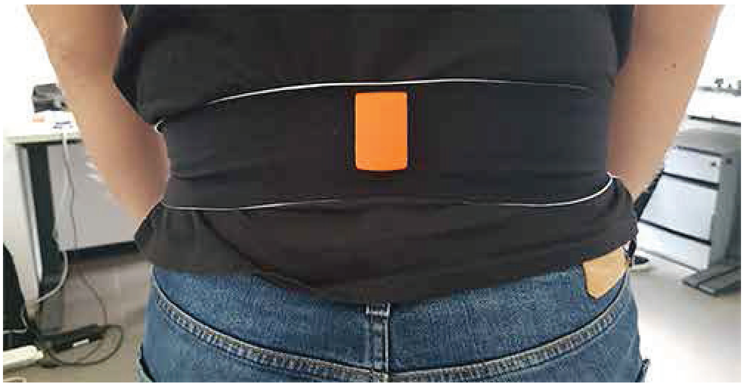
The placements of IMUs on the feet (as shown in Figure 4.1a) or on the lower back, close to the Center Of Mass (COM) (as shown in Figure 4.1b), are commonly adopted strategies in order to exploit the biomechanical characteristics of the human body to allow purely inertial navigation for pedestrians.

In Chapter 2, two different sensor placements, namely on the foot and on the chest, have been analyzed in order to develop two Inertial Navigation Systems (INSs). These systems rely on the signals collected through a single sensor. In Chapter 3, multiple sensor-based estimates have been fused in order to increase the overall accuracy: in this case, all the IMUs are placed on the foot. In this chapter, the last two phases of these algorithms, namely step detection and step length estimation, are recalled. Then, a innovative INS is developed by fusing the information from the IMUs, considering the placement of the sensors on the foot and on the lower back.

4.1.1 Experimental Setup

In the previous chapters, the proposed navigation algorithms rely on the use of Shimmer IMUs. In this chapter, a MTw Awinda IMUs by Xsens is used (see Subsection 1.5.4 for details): in this case, the orientation estimation is directly computed on-board by the Xsens IMU. The high accuracy of the Xsens orientation estimation (based on

(a) Feet-mounted IMUs



(b) Lower back mounted IMU

Figure 4.1: Sensor positions: (a) the IMUs are mounted under the shoe-laces, one per foot; (b) the IMU is attached to a velcro band in the middle of the back, over the spine.

a proprietary undisclosed algorithm) allows to focus on the (subsequent) design of inertial navigation algorithms.

The data stream generated by every IMU is transmitted to the USB Xsens Dongle, which is connected to a hand-held laptop. The online system performance is monitored trough a Matlab® application that collects and analyses the data in real-time.

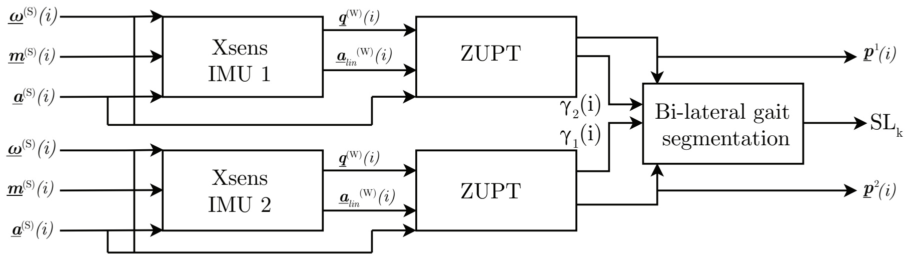


Figure 4.2: Block diagram of the EPDR algorithm for feet-mounted IMUs.

4.1.2 Foot-Mounted Sensor

By using the data collected by the sensor placed on the foot, as shown in Figure 4.1a, it is possible to directly correlate the foot movement to the acceleration measured by the IMU. The standard approach to estimate the foot displacement consists in double integrating the linear acceleration — namely, the acceleration transformed from the body frame to the global frame by rotating the reference system and removing the gravity component. Mathematical details of this approach are presented in Subsection 2.1.2. In particular, the first step of this procedure consists in segmenting the gait on the basis of the detection of the stationary phase of a stride using Equations (2.1).

The block diagram of the used Enhanced-Pedestrian Dead Reckoning (EPDR) algorithm is shown in Figure 2.4 and includes all the three fundamental phases recalled at the beginning of this section. This algorithm refers to one sensor but can be applied separately to the case with sensors on both feet (with one IMU per foot), as shown in the block diagram in Figure 4.2. While in the EPDR algorithm introduced in Subsection 2.1.2 the orientation is estimated through the Madgwick algorithm [70], in the current system the Xsens IMU can directly and accurately estimate the linear acceleration. As shown in Figure 4.2, this allows to directly use the Zero velocity UPdaTe (ZUPT) algorithm and segment the gait in its component phases — in fact, the signals computed through Equations (2.1) are approximately equal to zero when the foot is stationary. Thus, the ZUPT technique allows to identify the starting sample index $i_{\text{start}}^{(j)}$ and the ending sample index $i_{\text{end}}^{(j)}$ for the j-th central stance phase ($j = 1, 2, \ldots$) by computing the control sequence introduced in Equation (2.2). On the basis of the experimental analysis carried out in Chapter 2, proper values of the thresholds adopted in this equation are: $th_\omega = 50$ rad/s; $th_a = 9$ m/s^2; and $T_{\min} = 0.3$ s, respectively.

Considering only the swing phases, one can integrate the linear acceleration $\{\underline{\boldsymbol{a}}_{\text{lin}}^{(W)}(i)\}$ and obtain the linear velocity $\{\underline{\boldsymbol{v}}^{(W)}(i)\}$ by applying ZUPT. Then, the length of the k-th stride is calculated by integrating the velocity $\{\underline{\boldsymbol{v}}^{(W)}(i)\}$ over the entire foot trajectory in order to properly estimate the direction changes during the strides.

4.1.3 Lower Back-Mounted Sensor

When the sensor is placed on the trunk (lower back) of the test subject, it is very challenging to reconstruct the displacement in the frontal plane directly integrating the signal. For this reason, several approaches have been developed to correlate the dynamics of the COM with the displacement of the entire body. The algorithm introduced in this thesis in Section 2.1.1, denoted as De-Drifted Propagation (DDP), estimates the step length through the Weinberg model [74] and computes the traveled path by sampling the heading in the instant corresponding to the trunk vertical acceleration peaks. Its block diagram is shown in Figure 2.2. However, given the empirical nature of this model and the subject-dependent calibration required to tune the associated navigation algorithms, it introduces errors in the final estimated path. To avoid these problems and obtain a more accurate step length estimation, in Section 4.2 a hybrid approach which combines the accurate step length computed from the foot-mounted sensors with the robust heading angle measured through the trunk-mounted sensor is proposed.

4.2 Sensor Fusion

In the previous section, it has been shown that the two chosen sensors' positioning (namely, foot and COM) allows to design different algorithms which exploit the peculiar motion dynamics of foot and trunk, respectively. In this section, the details of

Figure 4.3: Walking human dynamic: heading angle.

a novel hybrid method, which exploits the advantages of both sensors' positions, are presented. In particular, we focus on the analysis of different approaches to sample the heading angle and on the reconstruction of the traveled path using the step lengths computed with the algorithms described in Section 4.1. Moreover, the body displacement computed by the sensor on the back, due to the used empirical formulae, is not so accurate because of the dependency on the test subject characteristics, as shown in Chapter 2.

4.2.1 Heading Sampling

The first phase of sensor fusion involves the synchronization of data streams and the selection of sampling instants. In particular, the latter operation is necessary because the body, during a normal walk, moves by following the patterns illustrated in Figure 4.3. This typical oscillatory behavior is captured by the heading angle measured by the sensor on the lower back, as shown in Figure 4.4, which is directly compared to the one computed from the foot-mounted sensor. It is possible to notice that the foot orientation varies consistently (with fluctuations approximately 4 times wider) with respect to the "specular estimate" computed from the sensor on the lower back. By using, as a reference, the gait segmentation provided by the algorithm relying on the foot-mounted sensors, the instants at which the body is aligned with the walking direction can be identified by analyzing the heading angle of the trunk mounted sensor, as shown in Figure 4.5. In particular, the middle swing instant (namely, the instant between a stance phase and the corresponding contra-lateral one)

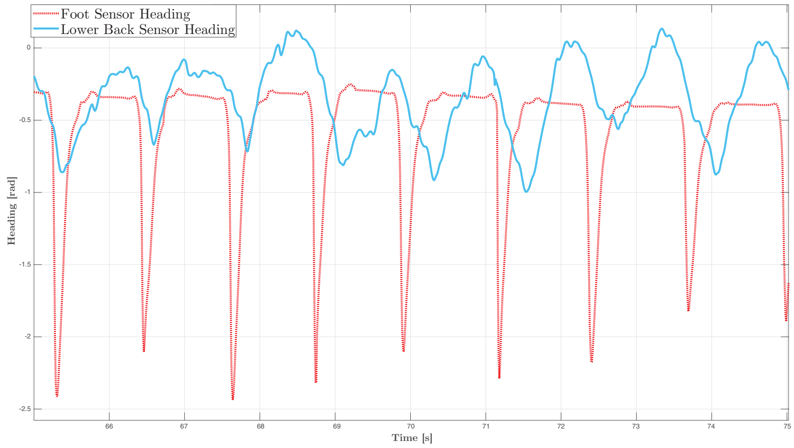


Figure 4.4: Comparison between the heading angles measured through the foot sensor (red line) and the lower back sensor (blue line).

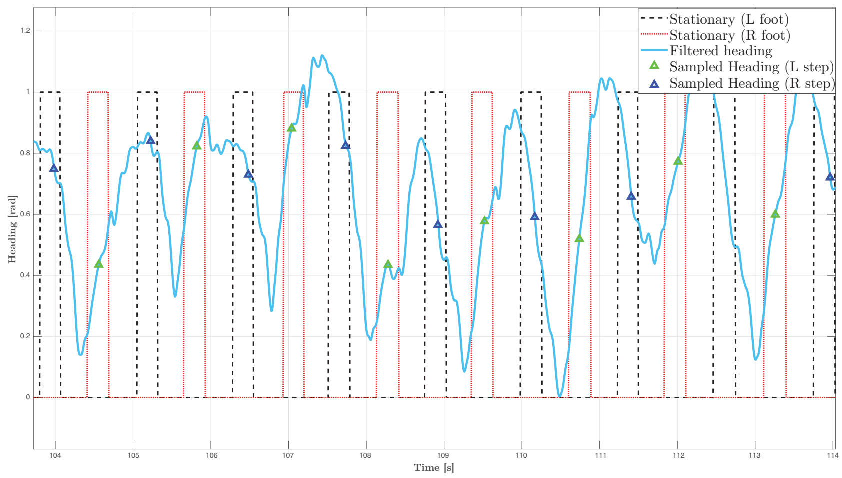


Figure 4.5: Heading estimated from the trunk-mounted (lower back) sensor compared with the stance phases identified through the feet-mounted sensors.

has been selected to sample the heading angle because, as evident from Figure 4.3 and Figure 4.5, it corresponds to the instant at which the body very likely heads towards the walking direction. Using the gait segmentation information obtained through the EPDR algorithm, the middle swing instant and the corresponding heading sample (in quaternions notation) are computed as follows:

$$\begin{aligned} t(i_{\text{mid}}^{(k)}) &= \frac{t(i_{\text{end}}^{(k)}) + t(i_{\text{start}}^{(z)})}{2} + t(i_{\text{end}}^{(k)}) \\ \underline{\hat{\boldsymbol{q}}}(k) &= \text{median}\left(\underline{\boldsymbol{q}}^{(\text{W})}\left[t(i_{\text{mid}}^{(k)}) - \frac{Win}{2F_s}, t(i_{\text{mid}}^{(k)}) + \frac{Win}{2F_s}\right]\right) \end{aligned} \tag{4.1}$$

where: $t(i_{\text{mid}}^{(k)})$ is the middle swing instant; $t(i_{\text{end}}^{(k)})$ is the end stance time of the k-th right step; $t(i_{\text{start}}^{(z)})$ is the start stance time of the z-th left step; median is the median function, defined as the middle value in a sorted dataset (if the samples' number is even, the mean of the two central values in the dataset is used); $\underline{\boldsymbol{q}}^{(\text{W})}$ are the quaternions collected from the back sensor; $\underline{\hat{\boldsymbol{q}}}(k)$ are the quaternions sampled with the proposed method for the k-th step; Win is the number of samples over which the median heading is calculated, and F_s is the sampling frequency (dimension: [Hz]). In our experiment setup, Win is set to 10 samples.

However, if the gait segmentation provided by the foot sensor is not available, an alternative method for sampling the heading angle is proposed. It consists in intersecting the heavily low-pass filtered version of the heading signal with its original (unfiltered) version. By using this technique one can find the same middle swing point as the central instant of the heading excursion, as shown in Figure 4.6.

The filter used to obtain the heavily filtered heading has to be a Finite Impulse Response (IIR) filter, in order to perform a near real-time analysis. This type of filtering introduces a fixed delay equal to $M/2$ samples, where M is the number of taps used by the filter. In our case, a IIR filter designed by using a least squares approach is used, namely by minimizing the discrepancy between a specified arbitrary piecewise-linear function and the filter's magnitude response. The other parameters of the filter are (i) the pass band frequency, set to 0.5 Hz, and (ii) the stop band frequency, set to 0.6 Hz.

The two proposed segmentation methods are developed in order to identify very precisely the middle swing instant and, consequently, their accuracies have to be

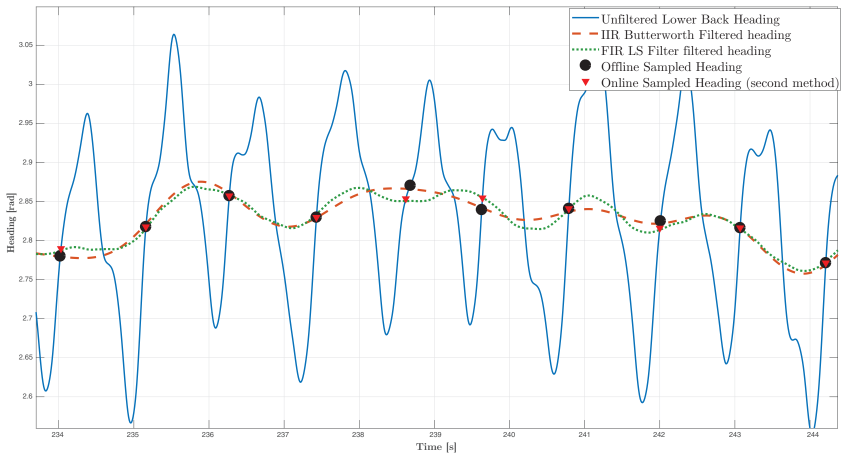


Figure 4.6: The heading provided by the lower back sensor is sampled according to the second method. The intersections between the heading signal filtered through the FIR filter and the original signal are shown as red triangles. For comparison, the same procedure is carried out with an offline IIR filter and the intersections are shown as black dots.

comparable. In spite of that, these methods are not completely equivalent. In particular, there are the following two main weaknesses that affect the second method (i.e, the method which exploits the filtered heading to identify the correct sampling instants) with respect to the gait segmentation based on the data collected from the feet-mounted sensors.

- Changes in the walking pattern (i.e., turnings, velocity changes, obstacle avoidance, etc.) cause irregular heading variations: this leads to a misalignment between the intersection of the two signals (filtered and unfiltered heading) and the middle swing instant.

- The filter introduces a delay with respect to the sampling instant detection. In fact, in order to obtain the signal shown in Figure 4.6, M has been set equal to 150 samples which, for a sampling rate $F_s = 100$ Hz, corresponds to a delay equal to $M/2 \cdot 1/F_s = 750$ ms. In order to perform a near real-time orientation sampling, the system needs to initially store 750 ms of data and, then, by taking into account the delay, to use the filter to obtain the correct sampling instant.

  For comparison purposes, this technique has been tested with an Infinite Impulse Response (IIR) filter, namely, a 6-th order low-pass Butterworth filter with cut-off frequency equal to 0.5 Hz, applied offline to the entire unfiltered heading signal. The resulting signal (red dashed line in Figure 4.6) is then compared with the one obtained by the IIR filter, shifted backwards by $M/2$ samples (corresponding to the filter delay). As can be seen in Figure 4.6, the IIR filter and the IIR filter lead to a comparable performance with respect to the middle swing instant identification: the average error, between the two heading angles estimated by using the two filters, is equal to only 0.03 rad.

In conclusion, when the gait segmentation provided by the feet-mounted sensors is available, its adoption has to be preferred. If the feet sensors' segmentation information is not available but the step length estimation is provided (e.g., an external step length estimation system with no gait segmentation information is available), the second method represents a good option. In general, the two methods show a substantial correspondence between the two sampling instant estimates.

4.2.2 Step Propagation

Once the orientation has been sampled, it is possible to reconstruct the traveled path by propagating the measured frontal displacement (corresponding, in this case, to the step length estimates provided by the foot-mounted sensors) in the direction indicated by the sampled heading angle. In particular, the estimated path can be computed using the following equations:

$$
\begin{aligned}
\psi(k) &= \arctan 2(2(\hat{q}_0(k)\hat{q}_3(k) + \hat{q}_1(k)\hat{q}_2(k)), \dots \\
&\quad \dots 1 - 2(\hat{q}_2^2(k) + \hat{q}_3^2(k))) \\
p_x(k) &= \cos(\psi(k)) * SL_k + p_x(k-1) \\
p_y(k) &= \sin(\psi(k)) * SL_k + p_y(k-1)
\end{aligned}
\tag{4.2}
$$

where: $\underline{\hat{\boldsymbol{q}}}(k) = \hat{q}_0(k) + \hat{q}_1(k)\mathbf{i} + \hat{q}_2(k)\mathbf{j} + \hat{q}_3(k)\mathbf{k}$ is the lower back orientation sampled by applying one of the methods described in Subsection 4.2.1; $\{\psi(k)\}$ is the corresponding heading angle; $(p_X(k), p_Y(k))$ is the estimated position at the k-th step; and SL_k is the length of the k-th step estimated through the feet sensors. The arctan2(x, y) function allows to convert the orientation from quaternion notation to Euler notation, in order to obtain the heading in the $[-\pi, \pi]$ range. Once the heading angle is computed, though the trigonometric formula in Equation (4.2) is it possible to estimate the displacement along the x and the y axes.

4.3 Results

In order to estimate the system performance, outdoor and indoor scenarios are considered. In each case, a reference path is selected. By introducing checkpoints along the chosen paths, one can evaluate the position estimation error of the proposed navigation algorithm as a function of the traveled distance. More precisely, the position error has been evaluated, in terms of both absolute error ϵ^a (dimension: [m]) and relative error ϵ^r (adimensional: [%]) with respect to the actual traveled path. In the following, the average (absolute and relative) errors between the estimates obtained by applying the EPDR algorithm to the right and the left foot are shown. The obtained results are presented by distinguishing between outdoor and indoor scenarios.

4.3.1 Outdoor Scenario

The outdoor experimental evaluation has been carried out on the athletic track of the campus of the University of Parma, Italy. As shown in Figure 4.7, the path is composed by 2 large and smooth curves and 2 straight lines. The reconstructed paths obtained by applying the EPDR algorithm on both feet, the DDP method, and the Hybrid approach (using the two proposed methods for sampling the heading angle) are shown. The absolute and relative estimation errors for each method are presented in Table 4.1. The orientation computed from the lower back sensor is more stable

Table 4.1: Performance, in terms of absolute and relative errors, of the considered algorithms in the outdoor scenarios.

	Foot IMU PDR (EPDR)		Hybrid system (1st & 2nd methods)		Lower back sensor (DDP)	
Distance [m]	ϵ^a [m]	ϵ^r [%]	ϵ^a [m]	ϵ^r [%]	ϵ^a [m]	ϵ^r [%]
84.4	1.5	1.8	0.4	0.47	1.2	1.42
200	5.2	2.6	3.3	1.65	6.9	3.45
284.4	7.3	2.6	4.7	1.65	8.4	2.95
400	12.9	3.23	7.6	1.9	2.7	0.67

and allows to obtain a better performance than that obtained with the feet-mounted sensors. This is due to the fact that feet are subject to rapid orientation changes, even in a smooth track, and this leads to errors in the heading estimation. This performance is intuitively expected from the heading behavior shown in Figure 4.4.

The following considerations can be carried out.

- By looking at the performance in Table 4.1, it is possible to notice that the results obtained with the EPDR are less accurate (i.e., with higher error) than those estimated by the hybrid system (both methods, 1 and 2 have the same performance), although the same step length is used. As a consequence, during the first part of the track the two systems present comparable values of the

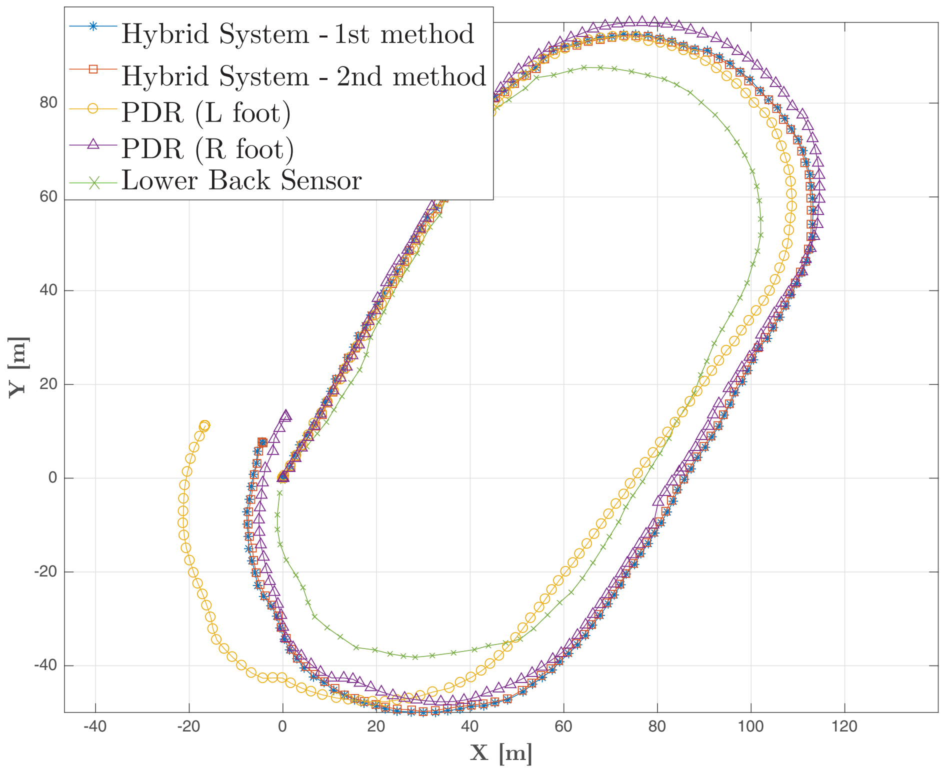


Figure 4.7: Comparison between the paths estimated by: (i) the EPDR algorithm applied on the right (yellow line with circles) and the left (purple line with triangles) feet; (ii) the DDP method based on a single lower back sensor (green line with crosses); (iii) the 1st hybrid algorithm, which samples the lower back heading through the mid swing method (blue line with stars) and the 2nd hybrid algorithm, which samples the lower back heading by using the intersection between filtered and unfiltered heading signal (red line with squares).

Table 4.2: Performance, in terms of absolute and relative errors, of the considered algorithms in the indoor scenarios.

	Foot IMU PDR (EPDR)		Hybrid system (1st & 2nd methods)	
Distance [m]	ϵ^a [m]	ϵ^r [%]	ϵ^a [m]	ϵ^r [%]
36	0.6	1.7	1.1	3.1
54	1.2	2.2	1.7	3.1
81	1.7	2.1	2.5	3.2
93	1.9	2.0	3.3	3.5
105	2.7	2.5	3.9	3.7

error ϵ. However, over a longer path, the estimate obtained using the lower back sensor orientation outperforms the one based on the EPDR system.

- The DDP system achieves very good performance in terms of orientation estimation (the final point is very close to the actual one and the general "estimated shape" is correct) but, due to the empirical formulae used for the step length calculation, it underestimates the actually traveled path and is outperformed by the hybrid system, which relies on the more accurate step length estimates given by the EPDR algorithm.

- Finally, from Figure 4.7 one can observe the correspondence between the two proposed heading sampling methods for the hybrid algorithm. Without loss of generality, in the following only performance results obtained with the EPDR method and the hybrid algorithm with the first heading sample technique are presented.

4.3.2 Indoor Scenario

In indoor environments, the step propagation technique generally used for path estimation with trunk mounted IMUs is less accurate than for outdoor path estimation,

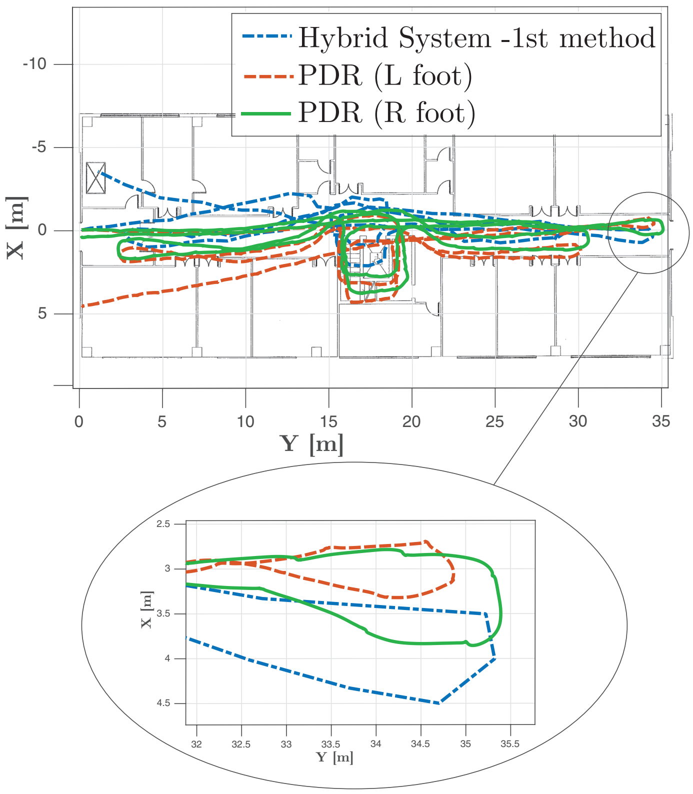


Figure 4.8: Comparison, in the indoor environment, between the path estimated by the EPDR (red dashed line for the left foot and green solid line for the right foot) and the hybrid (blue dashed-dotted line) systems. In the bottom of the figure, a detail of the reconstructed path which shows the impact of the step propagation on the measured error has been highlighted.

as already shown in Chapter 3. In particular, as can be seen from Figure 4.8, the sharp curves typical of indoor paths lead to the introduction of trajectory errors. These problems are caused by straight line propagation (i.e., step propagation) over multiple curves (long paths) and may lead to a relevant error accumulation. These problems also emerge from Table 4.2, in which the performance of the EPDR systems are slightly better than the one achieved by the hybrid system 1st method. This error

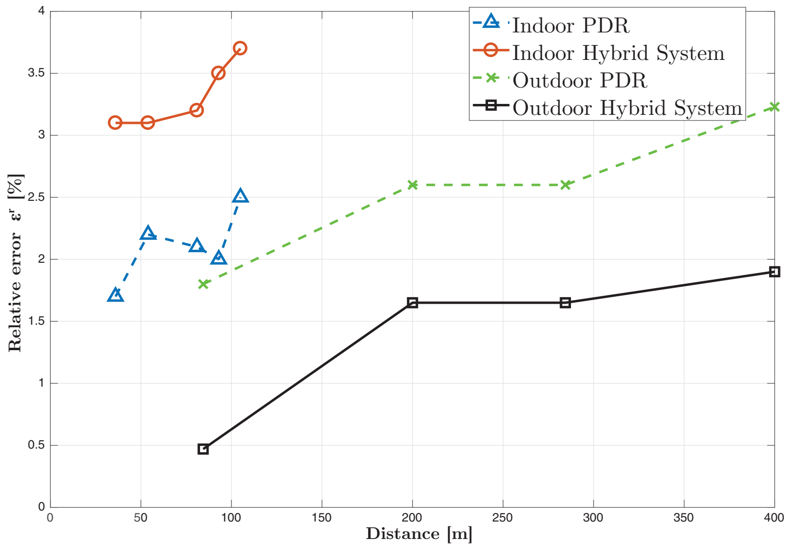


Figure 4.9: Relative error ϵ^r, as a function of the traveled distance, for the considered algorithms in both indoor and outdoor scenarios.

affects not only our hybrid approach but, more generally, all the techniques in which the orientation is first computed and, then, the step displacement is propagated.

4.3.3 Relative Error Comparison

In Figure 4.9, the relative error ϵ^r between the estimated path length and the true path length is shown as a function of the distance, considering the EPDR and the hybrid systems in both indoor and outdoor scenarios. From the results in this figure, it appears clear that in the indoor scenario the relative error increases more rapidly for both navigation systems, due to the error introduced by the sharp curves. Moreover, it is possible to notice that the EPDR system's estimated trajectory is more accurate than that of the hybrid system, in which the step propagation introduces smaller errors at every orientation change. The relative error in the outdoor scenario, instead, increases less rapidly and, due to the smooth changes in the heading angle, the hybrid system is more effective. In particular, the hybrid system reaches a very accurate estimation

Table 4.3: Computational cost organized per task (dimension: [ms]). These times refer to the processing load associated with a new step, which, on average, is detected every 600 ms.

(A) 3 IMUs 4 sub-cycles	(B) 2 IMUs 4 sub-cycles	(C)	(D)	(E1)	(E2)	(F)
4.4	7.4	16.2	0.4	0.46	1.3	0.05

with a final error ϵ^r equal to 1.9% of the length of the total traveled path.

4.3.4 Computational Costs

The proposed algorithms have been tested with a laptop equipped with an Intel i7-6700HQ processor working at 2.6 GHz, 8 GB RAM (DDR3, 1600 MHz) running Matlab® 2017a. In order to clarify the algorithm steps, a block diagram of the Matlab code is shown in Figure 4.10. In particular, the algorithm is composed by the following main steps.

- **Controlled delay:** this block calculates the time spent in the previous cycle, in order to keep the delay between a data acquisition and the next one almost constant and equal to approximately 300 ms. By default, the time delay introduced by this block is equal to 270 ms (as, on average, the previous cycle lasts around 30 ms). If the previous cycle has lasted longer than usual (i.e., a step is detected), the time delay is consequently reduced.

- **Data acquisition (A):** given that the Xsens system adopts an event-based approach to transmit every single data and taking into account that the navigation algorithm cannot perform a cycle every data packet, a custom library able to collect the data in a buffer has been implemented. Every sensor transmits the inertial data to a specified buffer. This part of the algorithm reads the data from these buffers and resets them.

- **Single sensor gait segmentation (B):** in this step, the foot phase (i.e., swing

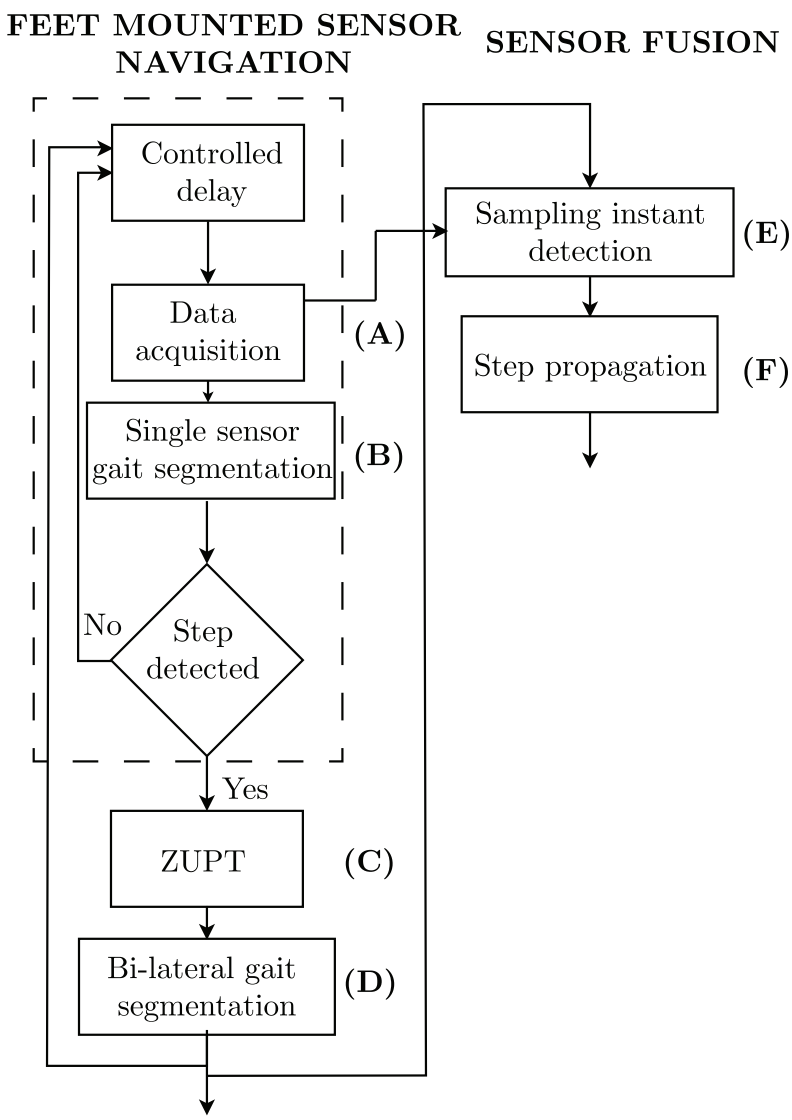


Figure 4.10: Flow diagram of the hybrid system operational steps.

or stride) is analysed. If a stance phase is detected, the ZUPT algorithm is performed; otherwise, the foot is still in a swing phase and the algorithm returns at the acquisition step.

- **ZUPT (C):** in this block, the ZUPT algorithm is applied. The drift is removed from the measurements and the stride length is estimated. At this level, the data from every sensor are processed separately.

- **Bi-lateral gait segmentation (D):** in this block, the complete (bi-lateral) gait is analysed and segmented. The step length is computed by exploiting the two gait segmentation (i.e., the gait segmentations performed separately from the two feet sensors) and stride length estimations.

- **Sampling instant detection (E):** in this phase, one of the two methods presented in Section 4.2.1 is implemented. E1 refers to the sampling method based on the information provided by the foot sensors (method 1) and notation E2 refers to the method based on the intersection between the filtered and unfiltered versions of the heading angle (method 2).

- **Step propagation (F):** in this final phase, the step lengths estimated through the feet-mounted sensors and the heading signal from the lower back sensor sampled in the previous phase are fused in order to perform navigation.

The time necessary to perform every phase of the algorithm on the used laptop is presented in Table 4.3. It is important to highlight that because of the *Controlled delay* function, the number of acquired samples per sub-cycle is about 30: therefore, the procedures (A) and (B) are computed with this number of samples per sub-cycle. The subsequent tasks, instead, are computed every time a stride is detected. By using several acquisitions with a normal walking pace, a stride event has been measured, on average, every 1.2 sec: this corresponds, for a sampling rate equal to 100 Hz, to a number of samples per stride equal (in terms of average value ± standard deviation) to 121 ± 13. In Table 4.3, it is shown the computational time necessary to update the position, namely the time needed to process a double stride (i.e., a stride per foot), in order to perform the bi-lateral gait segmentation. More precisely: task (A) is executed

4 times per sensor every position update; task (B) is executed 4 times only for the feet-mounted sensors; whereas tasks (C), (D), (E), and (F) are performed once per position update. Given that the average stride duration is 1.2 sec, a position update is performed every step, i.e., on average, every 600 ms.

In conclusion, the total time necessary to perform a position update every 600 ms by the hybrid system (with the 1st heading segmentation method) is 28.91 ms: this definitely allows a real time implementation of our approach.

4.4 Conclusions

In this chapter, two different INSs have been compared with the aim of investigating the strengths and weaknesses of various IMU placements in terms of "quality" of inertial navigation. The obtained results show that the orientation collected from a sensor placed near the COM, on the lower back, guarantees a more stable behavior with respect to the orientation obtained from a sensor placed on the foot. On the other hand, step length estimation is more effective when the sensor is placed on the foot, due to direct double integration and ZUPT: at the opposite, navigation systems relying exclusively on a single sensor mounted on the trunk use empirical formulae to estimate the frontal body displacement.

A novel hybrid method which exploits the advantages of the two sensor placements and achieves better or comparable performance in both outdoor and indoor environments has been developed. The main limitation of the trunk-mounted IMU-based method consists of the error introduced by the step propagation technique in the presence of sharp curves. The obtained results should be taken into account during the development of inertial navigation algorithms, in particular in the development of navigation algorithms with trunk-mounted and/or hand-held sensors.

Chapter 5

Smartphone-based INS

Anything can change,
because the smartphone revolution is still in the early stages.
- Tim Cook

Inertial Navigation Systems (INSs) are suitable for many application in multiple scenarios due to their independence from external infrastructure. However, an "ad-hoc" INS needs at least an on-body placed Inertial Measurement Unit (IMU). This highly limits the INS diffusion. Nowadays, the massive spread of Micro Electro Mechanical System (MEMS) IMU in commercial devices enables the possibility to develop user-friendly applications over these ones. In particular modern smartphones integrate multiple technologies and, more and more often, IMUs with acceptable performance. This leads to a high interest over smartphone-based INSs. The main issue with these systems is represented by the holding mode: in fact, users usually hold the smartphone in ways often unrelated to the body displacement.

In this chapter, a novel approach for pedestrian navigation based on inertial data collected from hand-held smartphones is proposed. In Section 5.1, an overview on the chosen software and hardware tools is given; then, an in-depth description of the features of the Android operating system which the proposed algorithm is based on is provided. In Section 5.2, innovative strategies to perform smartphone-based step detection and step length estimation are introduced. In Section 5.3, the system performance, considering multiple hardware platforms and different experimental set-ups, is evaluated. The details of the developed Matlab® scripts and custom-built Android application are also presented. Finally, in Section 5.4 conclusions are drawn. The algorithms and the prototype system described in this chapter have been introduced in [99].

5.1 System Overview

A smartphone-based INS will enable the possibility to bring inertial-based localization in commercial scenarios. In order to obtain a system which is effective in real applications, it is necessary to provide a multi-platform application based on a commercial Operating System (OS). In this chapter, the experimental setup adopted in this project, which is based on commercial Android smartphones, is presented. As every OS, Android acts as interface between the hardware and the developers through APIs. Therefore, the overview in the current chapter also focuses on the physical/emulated sensors, the Software Development Kit (SDK) and the available APIs.

5.1.1 Hardware and Software

Data are collected using three different smartphones, in order to verify the effectiveness of our algorithm over different hardware platforms. The chosen smartphones are (i) a Samsung S7, (ii) a Nokia 6, and (iii) a OnePlus 5T. The smartphones technical specifications are shown in Table 5.1. The offline algorithm and, then, the performance evaluation have been carried out with Matlab® by using the inertial data stored in different CSV files in the internal memory of the smartphones, using a third-party application (*HyperIMU*, [100]) for data collection. The real-time algorithm evaluation, instead, has been performed through an ad-hoc application developed in Android Studio, which has been installed over the three smartphones. The application details are illustrated in Subsection 5.3.2

Table 5.1: Considered Android smartphones.

Producer	Model	Android version	Accelerometer	Sensor manufacturer
Samsung	S7	Nogaut 7.0	K6DS3TR	STM
Nokia	6	Oreo 8.0	BMA2X2	BOSCH
OnePlus	5T	Oreo 8.0	BMI160	BOSCH

5.1.2 Smartphone Holding Mode

Straightforward approaches to perform smartphone-based inertial navigation usually involve a mode recognition phase, which has to be performed before starting the step detection procedure, in order to tune the algorithm according to the specific smartphone positioning. Our approach is *mode-independent*, which means that the only assumption is that the smartphone is placed in the upper part of the body (i.e., in the user's hand in texting/navigation mode or next to the head on phoning mode). The proposed algorithm, in fact, is based on the upper body acceleration dynamics: in particular, it compares the vertical acceleration and the acceleration magnitude to perform step detection. The vertical acceleration is computed online and refers to the world frame coordinates, enabling step detection independently from the current smartphone orientation.

In order to check the system accuracy through experimental tests, two main smartphone holding modes have been chosen. These are usually involved in the normal walk: the texting/navigation mode and the phoning mode, as shown in Figure 5.1. It is important to remark that there is no difference, in terms of system performance, between the cases where the smartphone is held with the right or the left hand.

5.1.3 Android Sensors APIs

The algorithms described in the following rely on data collected from Android smartphones. The Android operating system provides a set of APIs for managing, independently from the hardware, the raw data streams from the smartphone sensors. The

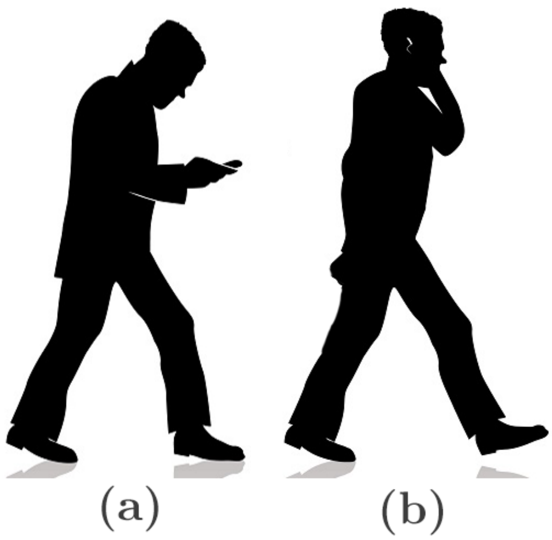


Figure 5.1: The smartphone holding mode tested in this thesis. Mode (a): the smartphone is held in front of the user with one hand, allowing the direct control of the current estimated path. Mode (b): the smartphone is held as in a normal phone call, with the smartphone close to the user's ear.

complete APIs documentation can be found at the Android developer website [101]. The mechanism used to obtain data from the sensors consists in creating and activating a listener (i.e., `SensorManager.registerListener`) over the required sensors, which throws a callback every time the data are available. It is important to remark that a sensor, in this context, could be both an actual physical sensor or an emulated one, which means that the Android operating system retrieves the data from the hardware components whenever available or derives some measurements combining data from the available physical sensors (e.g., by combining the acceleration, the angular velocity, and the magnetic field it is possible to obtain the orientation as if it had been measured from a hardware orientation sensor). However, the list of available sensors depends on the specific smartphone, i.e., different smartphones may likely embed different sensors. Therefore, in order to enable the listener over a specific sensor, it is necessary to call the following function:

```
SensorManager.registerListener(this,
SensorManager.getDefaultSensor(SENS_TYPE),
SAMPLE_RATE);
```

where: `SensorManager` is the object which manages the sensor in Android; `SENS_TYPE` is the sensor type identification code; and `SAMPLE_RATE` is the sampling rate characterized as *fastest*, *normal*, *game* (i.e., rate suitable for games), and *ui* (i.e., suitable for the user interface).

In order to perform an effective purely inertial navigation, one needs to know the orientation with respect to the geomagnetic north: this is provided by the magnetic sensor (i.e., `Sensor.TYPE_MAGNETIC_FIELD`) in terms of magnetic field intensity (dimension: $[\mu T]$). The smartphone reference system is shown in Figure 5.2. The orientation in the world frame can be obtained through the `Sensor.TYPE_ORIENTATION` in terms if rotation angle (in degrees) with respect to the geomagnetic north. Given that this sensor has been deprecated since API level 8 (i.e., Android 2.2 Froyo), in the latest Android version[1] it is possible (i) to exploit the `getRotationMatrix()` function which use the gravity sensor (`Sensor.TYPE_GRAVITY`) and the magnetic

[1]In this study, the Android APIs level 23 (i.e., Android Marshmallow 6.0) have been used.

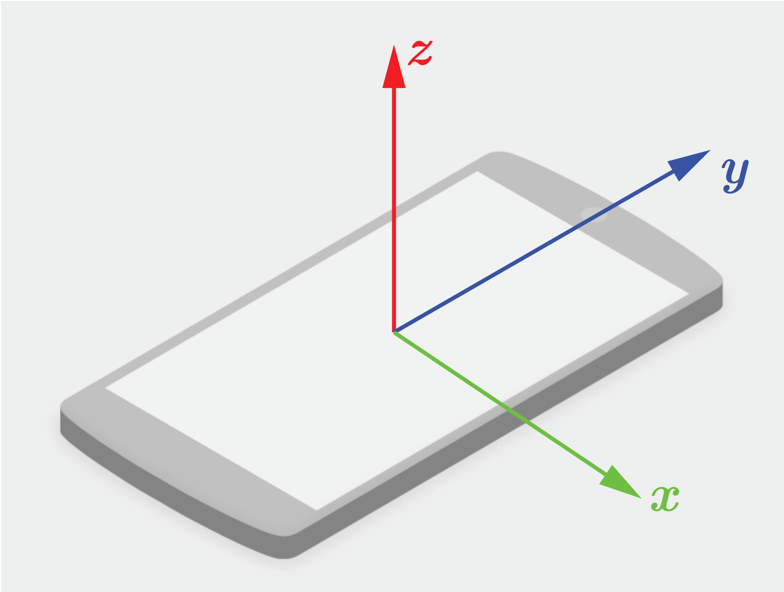


Figure 5.2: Android smartphone reference system.

sensor or (ii) to use the `Sensor.TYPE_ROTATION_VECTOR`, which is a "software sensor" which directly computes the smartphone orientation $\underline{\boldsymbol{h}}^{(\mathrm{W})}$ (where (W) represents the world reference system) as a unit vector, namely $<x, y, z>$, and an angle of revolution around that vector, namely θ. In other words, the orientation is represented by three values expressed as follows:

$$h_x^{(\mathrm{W})} = x \cdot \sin\left(\frac{\theta}{2}\right) \tag{5.1}$$

$$h_y^{(\mathrm{W})} = y \cdot \sin\left(\frac{\theta}{2}\right) \tag{5.2}$$

$$h_z^{(\mathrm{W})} = z \cdot \sin\left(\frac{\theta}{2}\right) \tag{5.3}$$

where: x is the axis tangential to the ground at the device's current location and pointing towards East; y is the axis tangential to the ground at the device's current location and roughly pointing towards the magnetic north; and z is the axis which points towards the sky and is perpendicular to the ground according to the axis-angle notation.

5.2 Smartphone-based Algorithms

5.2.1 Linear Acceleration Computation

In order to obtain the smartphone linear acceleration in the world reference system (W) (introduced in [101]), the raw acceleration is collected from the hardware sensor $\underline{\boldsymbol{a}}^{(\mathrm{S})}(k)$, where (S) refers to the smartphone reference system in Figure 5.2 and k is the sampling epoch. Therefore, the orientation expressed in angle-axis notation $\underline{\boldsymbol{h}}^{(\mathrm{W})}(k)$ obtained through the rotation vector software sensor is used to estimate the rotation matrix $R^{(\mathrm{W})}(k)$. First, the corresponding quaternion is obtained as follows:

$$q_0' = 1 - \left(h_x^{(\mathrm{W})}\right)^2 - \left(h_y^{(\mathrm{W})}\right)^2 - \left(h_z^{(\mathrm{W})}\right)^2 \tag{5.4}$$

$$q_0 = \begin{cases} \sqrt{q_0'} & \text{if } q_0' > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

$$q_1 = h_x^{(\mathrm{W})} \tag{5.6}$$

$$q_2 = h_y^{(\mathrm{W})} \tag{5.7}$$

$$q_3 = h_z^{(\mathrm{W})} \tag{5.8}$$

where $\underline{\boldsymbol{q}} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ is the orientation expressed in quaternion form (q_0, q_1, q_2, q_3 are real numbers and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the fundamental quaternion imaginary units). The rotation matrix at epoch k can then be obtained as follows:

$$R^{(\mathrm{W})}(k) = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}. \tag{5.9}$$

This matrix is used to rotate the raw acceleration according to the following equation:

$$\underline{\boldsymbol{a}}^{(\mathrm{W})}(k) = \underline{\boldsymbol{a}}^{(\mathrm{S})}(k) \cdot R^{(\mathrm{W})}(k) \tag{5.10}$$

moving to the world reference system and obtaining $\underline{\boldsymbol{a}}^{(\mathrm{W})}(k)$. The linear acceleration $\underline{\boldsymbol{a}}_{\mathrm{lin}}^{(\mathrm{W})}(k)$ is then obtained by subtracting the gravity $\underline{\boldsymbol{g}}^{(\mathrm{W})} = (0, 0, g)$ with $g = 9.81\ \mathrm{m/s^2}$ i.e., $\underline{\boldsymbol{a}}_{\mathrm{lin}}^{(\mathrm{W})}(k) = \underline{\boldsymbol{a}}^{(\mathrm{W})}(k) - \underline{\boldsymbol{g}}^{(\mathrm{W})}$.

5.2.2 Step Detection

The following approach allows to check the gait status every sample and, thus, to detect a step in near-real time. In fact, the idea behind the proposed step detection algorithm relies on the assumption that, when the user foot is close to a Heel Strike (HS) or a Toe Off (TO), the principal component in the acceleration magnitude is the vertical one. Therefore, if the vertical acceleration is "close enough" to the acceleration magnitude, the step is detected. The acceleration magnitude at the k-th sample can be computed as follows:

$$a_{\mathrm{m}}(k) = \sqrt{a_{x,\mathrm{lin}}^{(\mathrm{W})}(k)^2 + a_{y,\mathrm{lin}}^{(\mathrm{W})}(k)^2 + a_{z,\mathrm{lin}}^{(\mathrm{W})}(k)^2} \tag{5.11}$$

where $a_{x,\mathrm{lin}}^{(\mathrm{W})}(k)$, $a_{y,\mathrm{lin}}^{(\mathrm{W})}(k)$, and $a_{z,\mathrm{lin}}^{(\mathrm{W})}(k)$ are the linear acceleration components.

A state transition diagram-based approach is applied in order to describe the detection phases. A state transition happens in correspondence to every sample. In particular, the following three states are introduced.

- **No step:** this is the idle status, in which no step is detected. This happens if the test subject is static or if the previous step is ended and the next one has still to be detected.

- **Step initial phase:** in this status, the algorithm has detected a step. The smartphone is swinging because of the walking dynamics and the test subject foot is performing a HS. In this status, the vertical acceleration is maximum.

- **Step terminal phase:** this state lasts until the end of a step is detected, namely the vertical acceleration becomes higher than the magnitude threshold th_m. The horizontal displacement is then computed and the end of a cycle is declared. The system will return in the "No step" status.

As visible in the state transition diagram in Figure 5.3, three indicator functions are defined to allow the discrimination between state transitions (i.e., to distinguish the

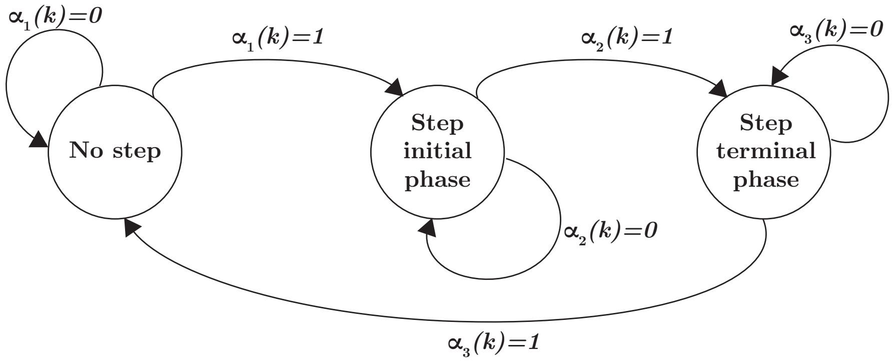


Figure 5.3: Step detection algorithm state diagram.

gait phases). The first indicator function, namely α_1, is defined as follows:

$$\alpha_1(k) = \begin{cases} 1 & \text{if} \begin{cases} a_{\mathrm{m}}(k) & > th_m \\ |a_{\mathrm{m}}(k) - a_{z,\mathrm{lin}}^{(\mathrm{W})}(k)| & < th_d \\ t(k) - t(LSI) & < min_d \end{cases} \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

where: th_m is the magnitude threshold (dimension: [m/s^2]); th_d is the "vertical-to-magnitude-similarity" threshold, namely the distance between the acceleration magnitude and the vertical acceleration which enables the step detection (dimension: [m/s^2]); min_d is the minimum distance, in terms of number of samples, between a step and next one; and the Last Step Index (LSI) is the sample index corresponding to the maximum vertical acceleration during the last step. If a step has not been detected yet, LSI is initialized to 0. The chosen values for the considered thresholds are: $th_m = 1.5$ m/s^2; $th_d = 0.5$ m/s^2; and $min_d = 50$ samples, respectively.

If $\alpha_1(k) = 1$, the status is updated to "Step initial phase" and the algorithm checks a second variable α_2 in order to detect the end of the first part of a step (i.e., the HS of one foot which approximately corresponds to the TO of the other foot). The second condition is described through the following indicator function:

$$\alpha_2(k) = \begin{cases} 1 & \text{if} \quad a^{(W)}_{z,\text{lin}}(k) < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5.13}$$

The use of $\alpha_2(k)$ allows to detect the moment in which the decreasing vertical acceleration crosses zero. This means that the HS event has occurred and the current step's acceleration maximum can be identified as follows:

$$a_{z,\max}(i) = \max_{k \in \text{step}} a^{(W)}_{z,\text{lin}}(k) \tag{5.14}$$

where i represents the step index. The index LSI is then updated as the epoch of the last acceleration maximum: therefore, in every step, LSI is the epoch associated with the maximum $a_{z,\max}(i)$ of the current step.

The algorithm status is then updated in order to detect the epoch i corresponding to the minimum acceleration $a_{z,\min}$ in the current step. This condition is defined through the following indicator function:

$$\alpha_3(k) = \begin{cases} 1 & \text{if} \quad acc^{(W)}_{\text{lin},z}(k) > th_m \\ 0 & \text{otherwise.} \end{cases} \tag{5.15}$$

Once the third condition is satisfied (i.e., $\alpha_3 = 1$), it is possible to identify the minimum in the vertical acceleration in order to estimate the horizontal displacement through one of the various algorithms which rely on the detection of acceleration peaks and valleys introduced in Subsection 1.5.3. The proposed algorithm adopts the Weinberg step length estimation algorithm [74].

The gait segmentation obtained through the proposed method is shown in Figure 5.4, in which one can observe the signal collected from the smartphone held in navigation/texting mode. The vertical acceleration represents the main component in the acceleration magnitude during a normal walk pattern and this allows to perform step detection. Moreover, this method allows to distinguish between the acceleration peaks generated by the step with respect to the other spurious peaks in the vertical acceleration.

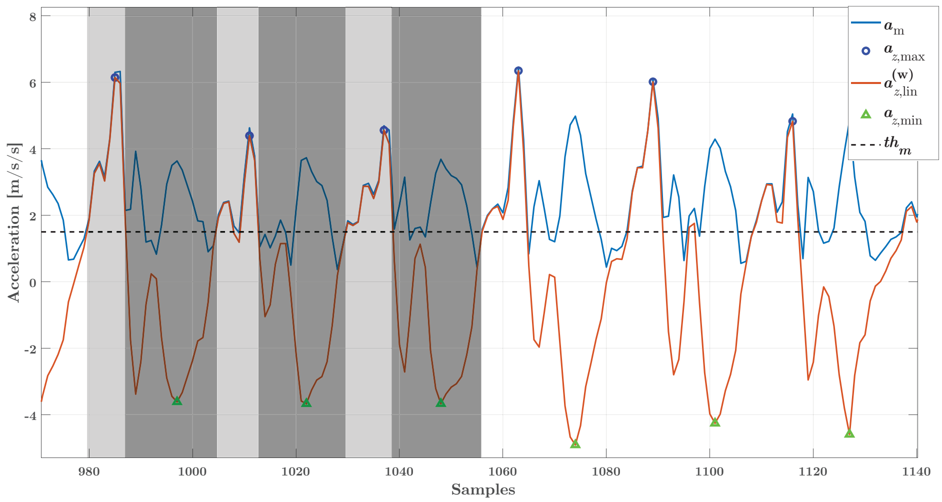


Figure 5.4: Gait segmentation obtained through the proposed algorithm with smartphone held in texting mode. The blue line represents the acceleration magnitude a_{m}, the red line is the vertical acceleration $a^{(W)}_{z,\mathrm{lin}}$. The light gray highlighted graph portion represents the "Step initial phase", whereas the dark gray highlighted graph portion represents the "Step terminal phase."

5.2.3 Step Length Estimation and Path Calculation

Once the gait phases are segmented as described in Subsection 5.2.2, the Weinberg algorithm is applied and the length of the i-th step can be estimated according to Equation (1.8). It is important to remark that, in this formula, K is a constant which relies on the smartphone positioning and the subject height, i[2] is the step index, and $a_{z,\mathrm{max}}(i)$ and $a_{z,\mathrm{min}}(i)$ are the maximum and the minimum acceleration during the i-th step, respectively. The constant K is a key value in the path estimation, especially for the proposed algorithm which aims to be independent of the smartphone position. In Subsection 5.2.4, an innovative K estimation approach will be proposed.

Once the step has been detected and the step length has been computed, the

[2]It is important to highlight that, in the previous chapters, the index i defines the sampling epoch, whereas, in this chapter, i is used to indicate the i-th step.

position estimate $\underline{\boldsymbol{p}}(i) = (p_x(i), p_y(i))$ is obtained by projecting the step length along the heading direction $\{\psi(i)\}$ derived from the quaternion $\underline{\boldsymbol{q}}(i)$ (see Subsection 5.2.1) as follows:

$$\psi(i) = \arctan 2(\,2\,(q_0 q_3 + q_1 q_2),\, 1 - 2\,(q_2^2 + q_3^2)\,). \tag{5.16}$$

The new position is then estimated as follows

$$\begin{aligned} p_x(i) &= SL(i) \cdot \cos(\psi(i)) + p_x(i-1) \\ p_y(i) &= SL(i) \cdot \sin(\psi(i)) + p_y(i-1). \end{aligned} \tag{5.17}$$

5.2.4 Mode Independent K Estimation

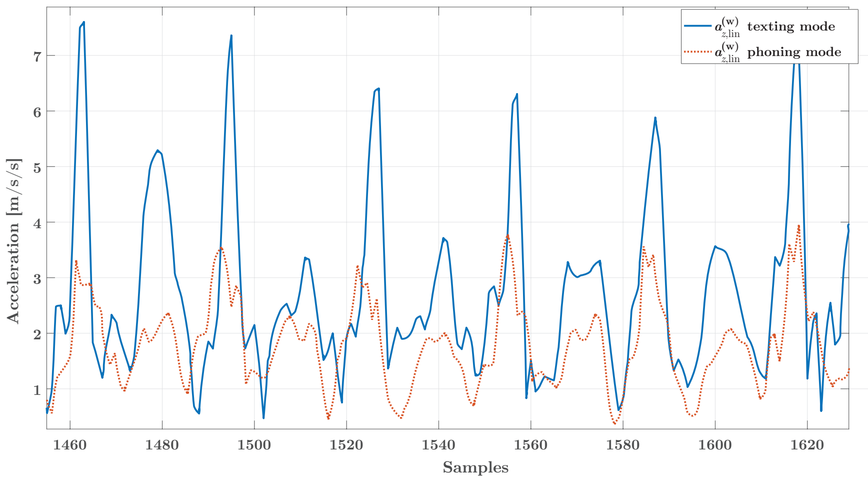


Figure 5.5: Comparison between the acceleration magnitude calculated according with Equation (5.11) for the smartphone held in texting/navigation mode (blue line) and in phoning mode (red dotted line). The two signals refer to two different trials carried out with the same smartphone.

Different smartphone placements are related to different acceleration magnitudes during walking. In fact, the way the user is holding the smartphone determines its orientation and its potential movements. When the smartphone is held in phoning mode, for example, it is more constrained to the body with respect to the texting/navigation holding mode, where the hand is free to move along the vertical direction. This different acceleration magnitude behaviors are illustrated in Figure 5.5. The Weinberg step length estimation formula shown in Equation (1.8) relates the trunk vertical acceleration to the horizontal displacement by assuming the sensor attached to the upper body. This assumption no longer holds when the sensor (i.e., the embedded IMU in the smartphone) is hand-held. In this case, the formula has to be calibrated not only according to the test subject walking characteristic, but also according to the sensor degrees of freedom.

The idea behind the proposed approach is to adopt a constant K for the step length estimation which relies on the acceleration magnitude as "motion quantity indicator." Experimental tests have lead to the development of the following novel formula which allows one to automatically calibrate the step length estimator according to the smartphone position:

$$K = \beta \cdot \frac{1}{\sqrt[3]{a_{\mathrm{m,max}}(i)}} \tag{5.18}$$

where: β is a per-subject estimated constant (dimension: $[(\mathrm{m/s}^2)^{1/3}]$); $a_{\mathrm{m,max}}(i)$ is the maximum value in the acceleration magnitude during the considered i-th step (dimension: $[\mathrm{m/s}^2]$). Experimentally, it has been found that, for the test subject whom the following results refer to, $\beta = 0.7\ (\mathrm{m/s}^2)^{1/3}$ is an efficient choice.

In the following section, the results have been obtained by considering multiple experimental set-ups with a single test subject.

5.3 Performance Analysis

The experiments have been carried out in the Department of Engineering and Architecture of the University of Parma (latitude: 44.765277, longitude: 10.308277), along the main corridor, where markers (i.e., paper tags attached to the corridor's walls) were placed every 5 m. In order to obtain accurate measurements, the final distance (i.e., the one between the final marker and the ending position) has been measured with a measuring tape. Every test has been video recorded to obtain (as a reference) the precise step count. As shown in Figure 5.6, the corridor length is

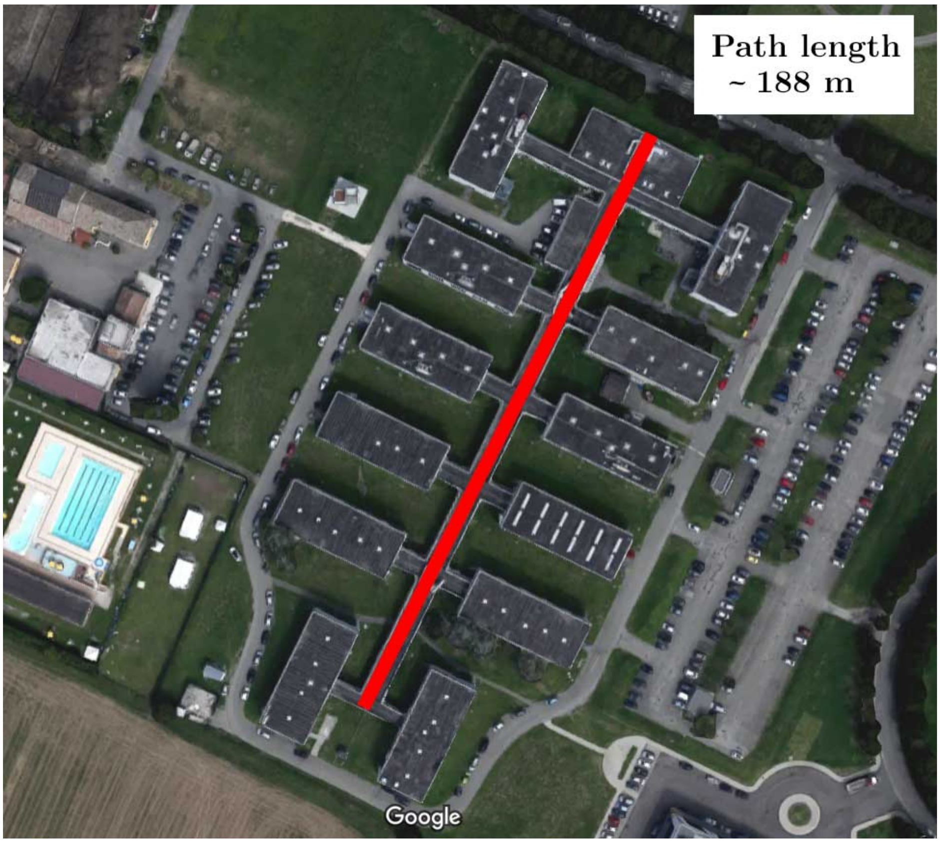


Figure 5.6: Department of Engineering and Architecture of the University of Parma. The trials have been carried out in indoor environment along the main corridor (highlighted in red) which has approximate length of 188 m.

Table 5.2: System performance: smartphone in texting/navigation mode.

# Step	Detected step			True dist. [m]			Estimated dist. [m]			% ϵ on dist.		
	S7	N6	5T	S7	N6	5T	S7	N6	5T	S7	N6	5T
20	20	21	20	13.9	13.5	12.3	13.3	12.4	12.8	4.3	8.1	4.6
50	49	52	52	34.5	33.1	33.5	32.9	31.7	35.9	4.6	4.1	7.1
100	101	104	96	69.7	65.7	61.4	68.2	67.9	63.2	2.1	3.3	2.9
500	503	505	491	348.5	352.2	361.2	340.7	370.5	350.3	2.3	5.2	3.2

Table 5.3: System performance: smartphone in phoning mode.

# Step	Detected step			True dist. [m]			Estimated dist. [m]			% ϵ on dist.		
	S7	N6	5T	S7	N6	5T	S7	N6	5T	S7	N6	5T
20	20	21	21	14.2	13.5	13.6	14.9	14.1	13.8	4.9	4.3	1.7
50	50	49	48	35.4	33.7	33.8	34.9	35.3	33.8	1.4	4.7	0
100	101	97	101	69.1	66.7	66.5	70.4	69.4	71.2	1.8	4	1.6
500	502	498	504	351.3	345.3	352.6	365.3	332.5	344.8	3.9	3.7	2.8

approximately 188 m: for the longest tests (i.e., the walks with 500 steps), the subject has performed a single curve with no step, namely by turning in a single movement. In order to evaluate the step detection accuracy together with the step length estimation capability, several tests have been performed. The data have been analyzed through a Matlab-based script. An Android application has been developed in order to test the navigation algorithm in real-time.

5.3.1 Experimental Results

In Table 5.2 and Table 5.3, the results obtained in the experimental trials are summarized. The experimental data have been organized according to the smartphone use modes.

In Table 5.2, the presented results show the algorithm effectiveness using the smartphones in texting/navigation mode. As one can see, the proposed approach can correctly detect almost all the steps, with an error, with respect to the total number of steps, which is always below 4%. These results are valid for all the three tested smartphones, i.e., the accuracy is independent of the hardware platform.

In Table 5.3, the performance results with the smartphone in phoning mode are shown: one can observe the algorithm capability to accurately identify the steps independently of the smartphone position. This is due to the linear acceleration estimation procedure presented in Subsection 5.2.1, which allows to estimate the vertical orientation with respect to the Earth reference system. In this experimental test, as already highlighted for the previous results in Table 5.2, the proposed approach has been able to identify the steps with high accuracy, namely, with a relative error below 3%. In Table 5.2 and Table 5.3, it is possible to observe that the estimated distance error varies when the step number varies. This is due to error averaging over a path: more precisely, a single step length estimation error has a higher impact in shorter paths than in longer ones.

Considering the performance in terms of path length estimation, the algorithm presents a significant variance in the estimated distances, due to a high vertical displacement variability of the smartphone. Nevertheless, as can be observed in Table 5.2 and Table 5.3, it can automatically adapt the step length estimation to the specific smartphone mode under use. In fact, the errors are comparable in both tables, which means that the bias is introduced by the common walking pattern dynamics, but not from the step length estimation formula, which is thus independent of the smartphone placement. The average (relative) error in Table 5.2 is approximately 4.3% of the total traveled distance, whereas in Table 5.3 the average (relative) error is below 3%. In Table 5.3, it can also be observed that the results are comparable between the three smartphones: this proves that the algorithm performance is independent of the hardware, thus making the proposed approach applicable to multiple devices.

It is important to remark that no modification in the step length estimation algorithm is needed to adapt the Weinberg formula to the smartphone use mode. In fact, the novel strategy introduced in Equation (5.18), based on the acceleration magnitude, enables the automatic correction of the constant K depending on the "oscillation quantity."

5.3.2 Android Application

The Android application has been developed in order to verify the algorithm capability to reconstruct the traveled path in real-time and to provide a continuous feedback to the

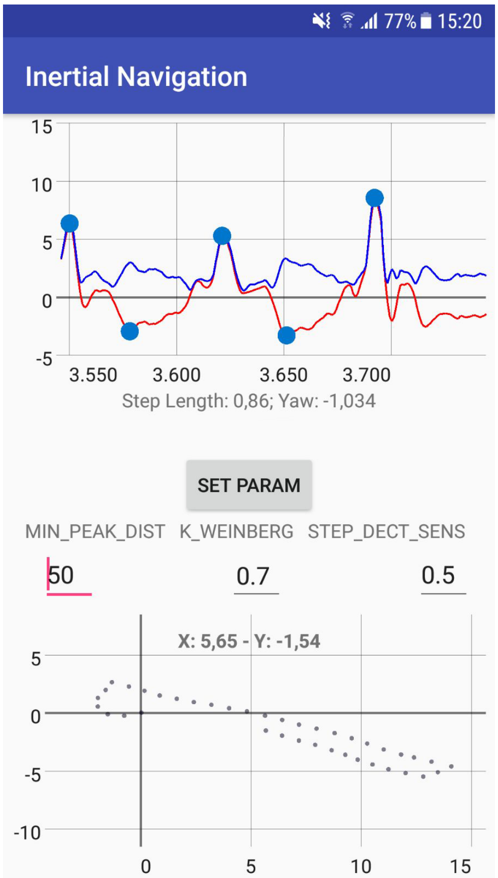


Figure 5.7: A screen capture of the Android Application. The top graph represents the acceleration magnitude a_{m} in blue, and the vertical acceleration $a_{z,\mathrm{lin}}^{(W)}$. The light blue dots represents the maximum $a_{z,\max}$ and the minimum $a_{z,\min}$ in the vertical acceleration. The bottom graph represents the estimated path. The x and y axis are stated in meters.

user. In the application layout, shown in Figure 5.7, one can see (top) the acceleration (linear vertical and magnitude) with step detection/gait segmentation and, (bottom) the estimated path.

The top graph is updated every time a new sample is collected from the sensor. As soon as the algorithm detects the maximum $a_{z,\max}(i)$ or the minimum $a_{z,\min}(i)$ in the vertical acceleration during the i-th step, a blue dot is displayed at the corresponding coordinates in the graph. Since the Android operating system does not ensure a constant sampling rate, the proposed algorithm sets the sampling rate to the maximum possible rate (i.e., `SensorManager.SENSOR_DELAY_FASTEST`) which corresponds, for the considered smartphones, to approximately 150 ÷ 200 Hz.

In Figure 5.7, right under the acceleration graph (on top), the current step length and the current yaw (heading) are displayed. These are updated every time a "*Step terminal phase*" is completed. Moreover, the application allows to set the algorithm key parameters such as: the minimum distance between two subsequent peaks, denoted as min_d (`MIN_PEAK_DIST`); the constant β (`K_WEINBERG`) needed to evaluate K according to Equation (5.18); and the difference threshold between the acceleration magnitude and the vertical acceleration, denoted as th_d (`STEP_DECT_SENS`).

In the bottom graph in Figure 5.7, the current coordinates, with respect to the starting point, expressed in meters—the default starting point is the axis origin $p(0) = (0, 0)$—are displayed. In the path shown in the bottom graph, instead, every point represents a step.

5.4 Conclusions

In this chapter, an innovative algorithm able to estimate the traveled path of a person during walking, exploiting the data collected from the IMUs embedded in commercial smartphones, is presented. Two different softwares have been developed: (i) a Matlab® script for offline data analysis, in order to verify the algorithm accuracy; and (ii) an Android application able to provide an online processing of the collected data, allowing real-time path estimation directly on the device screen. The algorithm performance have been tested on the basis of several trials carried out using three different commercial smartphones (namely, a Samsung S7, a Nokia 6, and a One-plus 5T). The obtained results prove the system feasibility, in terms of accuracy and latency, with respect to the position estimation.

The work presented in this chapter highlights the feasibility of a pure smartphone-based inertial navigation. However, this approach suffers from the noise introduced by the spurious movements which are usually generated during a normal walk. In order to tackle this issue, in particular over long paths, a possible solution could be the integration of other radio-based localization technologies, such as Bluetooth, WiFi, and GPS.

Conclusions

This dissertation has focused on the study of inertial approaches to pedestrian navigation. Even though multiple technologies have been considered, the inertial approach has been chosen for its versatility.

The first analysis has investigated a single sensor-based approach. A single Inertial Measurement Unit (IMU) has been placed in two different positions on the user's body, namely on the foot and on the chest. Two algorithms have been developed and the performance, in terms of positioning accuracy, has been evaluated through an extensive experimental campaign. The results obtained in this first research have demonstrated the feasibility of a single sensor-based Inertial Navigation System (INS), but they have also highlighted the impact of the drift problem, in particular in systems which rely on the signals collected by a single IMU. In order to reduce the estimation errors introduced by the drift, a sensor fusion approach has been investigated.

The first strategy adopted to perform sensor fusion has been the one that involves more than one IMU placed on the feet. Four different configurations, based on up to four IMU, have been investigated. The results demonstrate an accuracy increase in configurations with multiple IMUs. In particular, the configuration with four IMU, (namely, two IMUs per foot) has shown the highest accuracy. The analysis over different body placement has also allowed to highlight the inertial signals behavior related to specific IMU placements.

The analysis described in the previous paragraph has lead to a new sensor fusion approach which involves IMUs placed on various body positions (i.e., two IMUs on the feet, one per foot, and one IMU on the chest, on the lower back). The IMU placed on the upper trunk has shown better orientation estimation performance with respect to the IMUs placed on the feet. On the other hand, the IMUs placed on the feet have shown a higher reliability in terms of step length estimation. The fusion between the information collected from these three sensors has lead to an overall system improvement, but it has also highlighted problems during curves in the step propagation task, namely the combination of heading and displacement to compute the next position.

Finally, a research focused on hand-held-based INS has been carried out. Given the massive diffusion of IMUs in commercial smartphones, this analysis represents an attractive evolution with respect to the "classical" INSs which rely on IMUs firmly attached to the user's body. A new algorithm able to perform real-time inertial navigation exploiting the signal collected by the IMU embedded in the smartphone has been developed. It has been tested over three different smartphone models and the obtained results, in terms of step detection and step length estimation accuracy, have demonstrated the feasibility of an hand-held sensor-based commercial INS.

Possible future research activities could focus on the improvement of the hand-held IMU-based INS by developing a mode recognition algorithm, which would allow to apply proper modification in the step detection and step length estimation algorithms depending on the smartphone placement. Given that the INSs introduced in this thesis rely on per-subject calibrated constants, a further analysis could be carried out to test a large number of users, in order to generalize the presented results and their dependency from these constants. Further surveys could also analyze different trajectories, taking into account critical pattern such as multiple curves and floor changes. Orientation estimation during curves, as discussed in Chapter 2 and Chapter 3, represents a key task in inertial navigation, and correction strategies, such as orientation filter adaptive parameters and parallel orientation filters, could be investigated to address this issue.

Moreover, an overall improvement in the proposed system could rely on the integration with other localization technologies. For example, the INS integration with the Received Signal Strength Indicator (RSSI) information generated by Bluetooth Low Energy (BLE) beacons or WiFi Access Points (APs) placed in known positions in the environment is expected to significantly improve the system accuracy by reducing the drift.

Modern commercial devices, such as smartwatch and virtual reality goggles, usually contains one or more IMUs. This paves the way to innovative INSs which have to be based on completely different inertial signals with respect to the ones introduced in this thesis. For this reason, a further extension for the algorithm presented here could be represented by their application and adaptation in these new scenarios.

Bibliography

[1] B. Hofmann-Wellenhof, K. Legat, and M. Wieser. *Navigation: Principles of Positioning and Guidance*. Springer-Verlag, 2003.

[2] J. E. Miller. Radio guidance. *Proceedings of the Institute of Radio Engineers*, 20(11):1752–1762, Nov 1932. `doi:10.1109/JRPROC.1932.227711`.

[3] Raymond C. Jr. Watson. *Radar Origins Worldwide*. Trafford Publishing, nov 2009.

[4] A. G. Stove. Linear fmcw radar techniques. *IEE Proceedings F - Radar and Signal Processing*, 139(5):343–350, Oct 1992. `doi:10.1049/ip-f-2.1992.0048`.

[5] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt. Millimeter-wave technology for automotive radar sensors in the 77 ghz frequency band. *IEEE Transactions on Microwave Theory and Techniques*, 60(3):845–860, March 2012. `doi:10.1109/TMTT.2011.2178427`.

[6] J. Bock, H. Schafer, K. Aufinger, R. Stengl, S. Boguth, R. Schreiter, M. Rest, H. Knapp, M. Wurzer, W. Perndl, T. Bottner, and T. F. Meister. Sige bipolar technology for automotive radar applications. In *Bipolar/BiCMOS Circuits and Technology, 2004. Proceedings of the 2004 Meeting*, pages 84–87, Sept 2004. `doi:10.1109/BIPOL.2004.1365751`.

[7] J. W. Marck, A. Mohamoud, E. vd Houwen, and R. van Heijster. Indoor radar slam a radar application for vision and gps denied environments. In *2013 European Radar Conference*, pages 471–474, Oct 2013.

[8] F. Ali, G. Bauer, and M. Vossiek. A rotating synthetic aperture radar imaging concept for robot navigation. *IEEE Transactions on Microwave Theory and Techniques*, 62(7):1545–1553, July 2014. `doi:10.1109/TMTT.2014.2323013`.

[9] H. T. Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, May 1946. `doi:10.1109/JRPROC.1946.234568`.

[10] A. Bellini, C. Cirilo, V. R. T. Ferraz, A. G. Josué, J. L. Duque, l. P. Annibal, R. Durelli, and C. Marcondes. A low cost positioning and visualization system using smartphones for emergency ambulance service. In *SEHC10. ICSE Workshop on Software Engineering in Health Care*, pages 12–18, May 2010.

[11] B. Krishnamachari, S. B. Wicker, and R. Bejar. Phase transition phenomena in wireless ad hoc networks. In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, volume 5, pages 2921–2925 vol.5, Nov 2001. `doi:10.1109/GLOCOM.2001.965963`.

[12] D. Niculescu and Badri Nath. Ad hoc positioning system (aps) using aoa. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 3, pages 1734–1743 vol.3, March 2003. `doi:10.1109/INFCOM.2003.1209196`.

[13] S. Jiang, J. Hua, F. Chen, Z. Xu, and J. Li. Comparison and application on the uplink aoa in the geometrical single bounce model. In *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 664–668, June 2017. `doi:10.1109/ICIEA.2017.8282925`.

[14] I. Guvenc and C. Chong. A survey on toa based wireless localization and nlos mitigation techniques. *IEEE Communications Surveys Tutorials*, 11(3):107–124, rd 2009. doi:10.1109/SURV.2009.090308.

[15] Xinrong Li and K. Pahlavan. Super-resolution toa estimation with diversity for indoor geolocation. *IEEE Transactions on Wireless Communications*, 3(1):224–234, Jan 2004. doi:10.1109/TWC.2003.819035.

[16] J. J. Caffery. A new approach to the geometry of toa location. In *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No.00CH37152)*, volume 4, pages 1943–1949 vol.4, Sept 2000. doi:10.1109/VETECF.2000.886153.

[17] Xin Wang, Zongxin Wang, and B. O'Dea. A toa-based location algorithm reducing the errors due to non-line-of-sight (nlos) propagation. *IEEE Transactions on Vehicular Technology*, 52(1):112–116, Jan 2003. doi:10.1109/TVT.2002.807158.

[18] B. Alavi and K. Pahlavan. Modeling of the toa-based distance measurement error using uwb indoor radio measurements. *IEEE Communications Letters*, 10(4):275–277, April 2006. doi:10.1109/LCOMM.2006.1613745.

[19] S. Venkatraman, J. Caffery, and Heung-Ryeol You. A novel toa location algorithm using los range estimation for nlos environments. *IEEE Transactions on Vehicular Technology*, 53(5):1515–1524, Sept 2004. doi:10.1109/TVT.2004.832384.

[20] Jin Ik Kim, Jang Gyu Lee, and Chan Gook Park. A mitigation of line-of-sight by tdoa error modeling in wireless communication system. In *2008 International Conference on Control, Automation and Systems*, pages 1601–1605, Oct 2008. doi:10.1109/ICCAS.2008.4694487.

[21] T. Qiao, Y. Zhang, and H. Liu. Nonlinear expectation maximization estimator for tdoa localization. *IEEE Wireless Communications Letters*, 3(6):637–640, Dec 2014. doi:10.1109/LWC.2014.2364023.

[22] G. Fokin, A. Kireev, and A. H. A. Al-odhari. Tdoa positioning accuracy performance evaluation for arc sensor configuration. In *2018 Systems of Signals Generating and Processing in the Field of on Board Communications*, pages 1–5, March 2018. doi:10.1109/SOSG.2018.8350644.

[23] S. Zhang, H. Jiang, and K. Yang. Detection and localization for an unknown emitter using tdoa measurements and sparsity of received signals in a synchronized wireless sensor network. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5146–5149, May 2013. doi:10.1109/ICASSP.2013.6638643.

[24] Wen juan Ren, Dong hui Hu, and Chi biao Ding. An improved method to sort and pair tdoa based on the correlation between tdoas. In *Proceedings of 2011 IEEE CIE International Conference on Radar*, volume 2, pages 1041–1044, Oct 2011. doi:10.1109/CIE-Radar.2011.6159730.

[25] K. C. Ho and Y. T. Chan. Solution and performance analysis of geolocation by tdoa. *IEEE Transactions on Aerospace and Electronic Systems*, 29(4):1311–1322, Oct 1993. doi:10.1109/7.259534.

[26] Wing-Kin Ma, Ba-Ngu Vo, S. S. Singh, and A. Baddeley. Tracking an unknown time-varying number of speakers using tdoa measurements: a random finite set approach. *IEEE Transactions on Signal Processing*, 54(9):3291–3304, Sept 2006. doi:10.1109/TSP.2006.877658.

[27] K. C. Ho, X. Lu, and L. Kovavisaruch. Source localization using tdoa and fdoa measurements in the presence of receiver location errors: Analysis and solution. *IEEE Transactions on Signal Processing*, 55(2):684–696, Feb 2007. doi:10.1109/TSP.2006.885744.

[28] M. L. Fowler and X. Hu. Signal models for tdoa/fdoa estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 44(4):1543–1550, Oct 2008. doi:10.1109/TAES.2008.4667729.

[29] Z. Li, N. Ge, Z. Huang, and X. Chen. A method to eliminate tdoa ambiguity based on fdoa and fdoa-rate. In *2015 International Conference on Wireless Communications Signal Processing (WCSP)*, pages 1–5, Oct 2015. doi:10.1109/WCSP.2015.7341154.

[30] Pratap Misra and Per Enge. *Global Positioning System: signals, measurements and performance second edition*. Massachusetts: Ganga-Jamuna Press, 2006.

[31] A. S. Zaidi and M. R. Suddle. Global navigation satellite systems: A survey. In *2006 International Conference on Advances in Space Technologies*, pages 84–87, Sept 2006. doi:10.1109/ICAST.2006.313803.

[32] Bradford W Parkinson, Per Enge, Penina Axelrad, and James J Spilker Jr. *Global positioning system: Theory and applications, Volume II*. American Institute of Aeronautics and Astronautics, 1996.

[33] J. Benedicto, S. E. Dinwiddy, G. Gatti, R. Lucas, and M. Lugert. Galileo: satellite system design and technology developments, european space agency, 2000.

[34] André Hauschild, Peter Steigenberger, and Carlos Rodriguez-Solano. Signal, orbit and attitude analysis of japan's first qzss satellite michibiki. *Gps Solutions*, 16(1):127–133, 2012.

[35] V. Chamoli, R. Prakash, A. Vidyarthi, and A. Ray. Sensitivity of navic signal for soil moisture variation. In *2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT)*, pages 1–4, Nov 2017. doi:10.1109/ICETCCT.2017.8280318.

[36] European Global Navigation Satellite System Agency (GSA). Gnss user technology report. Technical Report Issue 2, European Commission, and GSA, August 2018.

[37] T. Jaffre, P. M. Grigis, S. Papanastasiou, and E. Peytchev. On the efficacy of WiFi indoor positioning in a practical setting. In *Computers and Communications (ISCC 2013)*, pages 699–704, Split, HR, July 2013.

[38] Y. Chouchang and H. r. Shao. WiFi-based indoor positioning. *IEEE Communications Magazine*, 53:150 – 157, March 2015.

[39] L. Guo, L. Wang, J. Liu, and W. Zhou. A survey on motion detection using wifi signals. In *2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, pages 202–206, Dec 2016. `doi:10.1109/MSN.2016.040`.

[40] C. Beder, A. McGibney, and M. Klepal. Predicting the expected accuracy for fingerprinting based wifi localisation systems. In *2011 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–6, Sept 2011. `doi:10.1109/IPIN.2011.6071939`.

[41] Min Zhang, Ling Pei, and Xiaotie Deng. Graphslam-based crowdsourcing framework for indoor wi-fi fingerprinting. In *2016 Fourth International Conference on Ubiquitous Positioning, Indoor Navigation and Location Based Services (UPINLBS)*, pages 61–67, Nov 2016. `doi:10.1109/UPINLBS.2016.7809951`.

[42] F. Wu, T. Luo, and J. C. J. Liang. A crowdsourced wifi sensing system with an endorsement network in smart cities. In *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, pages 1–2, April 2015. `doi:10.1109/ISSNIP.2015.7106968`.

[43] Y. Zhuang, B. Wright, Z. Syed, Z. Shen, and N. El-Sheimy. Fast wifi access point localization and autonomous crowdsourcing. In *2014 Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, pages 272–280, Nov 2014. `doi:10.1109/UPINLBS.2014.7033737`.

[44] J. Decuir. Bluetooth smart support for 6lobtle: Applications and connection questions. *IEEE Consumer Electronics Magazine*, 4(2):67–70, April 2015. `doi:10.1109/MCE.2015.2392955`.

[45] A. Benlghazi, E. Chadli, and D. Moussaid. Bluetooth technologie for industrial application. In *2014 5th International Conference on Informa-*

tion and Communication Systems (ICICS), pages 1–5, April 2014. `doi:10.1109/IACS.2014.6841984`.

[46] M. Ji, J. Kim, J. Jeon, and Y. Cho. Analysis of positioning accuracy corresponding to the number of ble beacons in indoor positioning system. In *2015 17th International Conference on Advanced Communication Technology (ICACT)*, pages 92–95, July 2015. `doi:10.1109/ICACT.2015.7224764`.

[47] H. Choi, H. Jin, and S. C. Kim. Rss bias compensation in ble beacon based positioning system. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 494–497, July 2017. `doi:10.1109/ICUFN.2017.7993833`.

[48] S. Memon, M. M. Memon, F. K. Shaikh, and S. Laghari. Smart indoor positioning using ble technology. In *2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pages 1–5, Nov 2017. `doi:10.1109/ICETAS.2017.8277872`.

[49] T. Mori, S. Kajioka, T. Uchiya, I. Takumi, and H. Matsuo. Experiments of position estimation by ble beacons on actual situations. In *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pages 683–684, Oct 2015. `doi:10.1109/GCCE.2015.7398727`.

[50] A. A. Kalbandhe and S. C. Patil. Indoor positioning system using bluetooth low energy. In *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, pages 451–455, Dec 2016. `doi:10.1109/CAST.2016.7915011`.

[51] J. Powar, C. Gao, and R. Harle. Assessing the impact of multi-channel ble beacons on fingerprint-based positioning. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Sept 2017. `doi:10.1109/IPIN.2017.8115871`.

[52] X. Hou and T. Arslan. Monte carlo localization algorithm for indoor positioning using bluetooth low energy devices. In *2017 International Conference on*

Localization and GNSS (ICL-GNSS), pages 1–6, June 2017. doi:10.1109/ICL-GNSS.2017.8376248.

[53] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Al-nafessah, Suheer Alhadhrami, Mai A. Al-Ammar, and Hend Al-Khalifa. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16:1–36, 05 2016.

[54] R. Ye and H. Liu. Uwb tdoa localization system: Receiver configuration analysis. In *2010 International Symposium on Signals, Systems and Electronics*, volume 1, pages 1–4, Sept 2010. doi:10.1109/ISSSE.2010.5607083.

[55] H. Xu and Wang ying. A linear algorithm based on tdoa technique for uwb localization. In *2011 International Conference on Electric Information and Control Engineering*, pages 1013–1015, April 2011. doi:10.1109/ICEICE.2011.5778025.

[56] B. Yan and L. Xiaochun. Research on uwb indoor positioning based on tdoa technique. In *2009 9th International Conference on Electronic Measurement Instruments*, pages 1–167–1–170, Aug 2009. doi:10.1109/ICEMI.2009.5274900.

[57] F. Dong, C. Shen, J. Zhang, and S. Zhou. A tof and kalman filtering joint algorithm for ieee802.15.4a uwb locating. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 948–951, May 2016. doi:10.1109/ITNEC.2016.7560502.

[58] M. M. Saad, C. J. Bleakley, M. Walsh, and T. Ye. High accuracy location estimation for a mobile tag using one-way uwb signaling. In *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*, pages 1–8, Oct 2012. doi:10.1109/UPINLBS.2012.6409769.

[59] B. Barua, N. Kandil, and N. Hakem. On performance study of twr uwb ranging in underground mine. In *2018 Sixth International Conference on*

Digital Information, Networking, and Wireless Communications (DINWC), pages 28–31, April 2018. doi:10.1109/DINWC.2018.8356990.

[60] Y. Nam, H. Lee, J. Kim, and K. Park. Two-way ranging algorithms using estimated frequency offsets in wpan and wban. In *2008 Third International Conference on Convergence and Hybrid Information Technology*, volume 1, pages 842–847, Nov 2008. doi:10.1109/ICCIT.2008.296.

[61] H. Liu and C. Hsuan. Aoa estimation for coexisting uwb signals with multipath channels. In *2014 International Conference on Telecommunications and Multimedia (TEMU)*, pages 173–178, July 2014. doi:10.1109/TEMU.2014.6917756.

[62] S. Galler, W. Gerok, J. Schroeder, Kyandoghere Kyamakya, and T. Kaiser. Combined aoa/toa uwb localization. In *2007 International Symposium on Communications and Information Technologies*, pages 1049–1053, Oct 2007. doi:10.1109/ISCIT.2007.4392171.

[63] M. Li, V. T. Rouf, M. J. Thompson, and D. A. Horsley. Three-axis lorentz-force magnetic sensor for electronic compass applications. *Journal of Microelectromechanical Systems*, 21(4):1002–1010, Aug 2012. doi:10.1109/JMEMS.2012.2196493.

[64] T. Roenneberg, A. Wirz-Justice, and M. Merrow. Life between clocks: daily temporal patterns of human chronotypes. *Journal of Biological Rhythms*, 18:80–90, 02 2003.

[65] Stéphane Beauregard and Harald Haas. Pedestrian Dead Reckoning:A Basis for Personal Positioning. In *Workshop on Positioning, Navigation and Communication (WPNC 2006)*, Hannover, Germany, March 2006.

[66] R. G. Valenti, I. Dryanovski, and J. Xiao. A linear kalman filter for marg orientation estimation using the algebraic quaternion algorithm. *IEEE Transactions on Instrumentation and Measurement*, 65(2):467–481, Feb 2016. doi:10.1109/TIM.2015.2498998.

[67] Z. Zhang, X. Meng, and J. Wu. Quaternion-based kalman filter with vector selection for accurate orientation tracking. *IEEE Transactions on Instrumentation and Measurement*, 61(10):2817–2824, Oct 2012. doi:10.1109/TIM.2012.2196397.

[68] B. Huyghe, J. Doutreloigne, and J. Vanfleteren. 3d orientation tracking based on unscented kalman filtering of accelerometer and magnetometer data. In *2009 IEEE Sensors Applications Symposium*, pages 148–152, Feb 2009. doi:10.1109/SAS.2009.4801796.

[69] Roberto G Valenti, Ivan Dryanovski, and Jizhong Xiao. Keeping a good attitude: A quaternion-based orientation filter for imus and margs. *MDPI Sensors*, 15:19302–19330, 2015.

[70] 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 Rehab Week (ICORR 2011)*, pages 1–7, ETH Zurich Science City, Switzerland, June 2011.

[71] L. Sabatelli. "lower limbs robot motion based on the probabilistic estimation of the joint angles starting from emg data of an injured subject". Master's thesis, Università degli Studi di Padova, November 2015.

[72] Jaehyun Park, Yunki Kim, and Jangmyung Lee. Waist mounted pedestrian dead-reckoning system. In *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 335–336, Nov 2012. doi:10.1109/URAI.2012.6463008.

[73] J. Ruppelt, N. Kronenwett, and G. F. Trommer. A novel finite state machine based step detection technique for pedestrian navigation systems. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7, Oct 2015. doi:10.1109/IPIN.2015.7346771.

[74] Harvey Weinberg. Using the adxl202 in pedometer and personal navigation applications. Technical Report application note AN-602, Analog Devices, 2002.

[75] Z. Wang, H. Zhao, S. Qiu, and Q. Gao. Stance-phase detection for zupt-aided foot-mounted pedestrian navigation system. *IEEE/ASME Transactions on Mechatronics*, 20(6):3170–3181, Dec 2015. `doi:10.1109/TMECH.2015.2430357`.

[76] Y. Li, X. Luo, X. T. Ren, and J. J. Wang. A robust humanoid robot navigation algorithm with zupt. In *2012 IEEE International Conference on Mechatronics and Automation*, pages 505–510, Aug 2012. `doi:10.1109/ICMA.2012.6283159`.

[77] Y. Li, Y. Wang, H. Li, Q. Shu, Y. Chen, W. Yang, Y. Liu, and M. Zhao. Research on zupt technology for pedestrian navigation. In *2017 8th International Conference on Mechanical and Aerospace Engineering (ICMAE)*, pages 725–729, July 2017. `doi:10.1109/ICMAE.2017.8038739`.

[78] S.H. Shin, C.G. Park, J.W. Kim, H.S. Hong, and J.M. Lee. Adaptive Step Length Estimation Algorithm Using Low-Cost MEMS Inertial Sensors. In *Sensors Applications Symposium (SAS '07), IEEE*, pages 1–5, San Diego, CA, February 2007.

[79] Wiebren Zijlstra and At L. Hof. Assessment of spatio-temporal gait parameters from trunk accelerations during human walking. *Gait Posture*, 18:1–10, 2003.

[80] D. Alvarez, R.C. Gonzalez, A. Lopez, and J.C. Alvarez. Comparison of Step Length Estimators from Weareable Accelerometer Devices. In *Engineering in Medicine and Biology Society (EMBS '06), IEEE*, pages 5964–5967, New York, NY, August 2006.

[81] Shimmer Realtime Technologies Ltd. *Shimmer MATLAB ™Instrument Driver Library User Manual Rev 1.4a*. Clonshaugh Technology Park, Dublin 17, Ireland, 2013.

[82] Shimmer Realtime Technologies Ltd. *ShimmerConnect Rev 0.3a User Manual*. Clonshaugh Technology Park, Dublin 17, Ireland, 2013.

[83] Shimmer Realtime Technologies Ltd. *9DOF Calibration Application User Manual Rev 1.2b*. Clonshaugh Technology Park, Dublin 17, Ireland, 2013.

[84] Shimmer Realtime Technologies Ltd. *Shimmer User Manual Rev 2R.f*. Clonshaugh Technology Park, Dublin 17, Ireland, 2012.

[85] M. Paulich, M. Schepers, N. Rudigkeit, and G. Bellusci. Xsens mtw awinda: Miniature wireless inertial-magnetic motion tracker for highly accurate 3d kinematic applications. Technical report, XSENS TECHNOLOGIES B.V., 05 2018.

[86] Xsens Technologies B.V. *MTw Awinda User Manual*. Pantheon 6a, 7500 AN Enschede, NL, document mw0502p, revision j edition, 05 2016.

[87] H. Fourati. Heterogeneous data fusion algorithm for pedestrian navigation via foot-mounted inertial measurement unit and complementary filter. *IEEE Transactions on Instrumentation and Measurement*, 64(1):221–229, Jan 2015. doi:10.1109/TIM.2014.2335912.

[88] Y. L. Hsu, J. S. Wang, and C. W. Chang. A wearable inertial pedestrian navigation system with quaternion-based extended kalman filter for pedestrian localization. *IEEE Sensors Journal*, 17(10):3193–3206, May 2017. doi:10.1109/JSEN.2017.2679138.

[89] A. R. Jimenez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *2009 IEEE International Symposium on Intelligent Signal Processing*, pages 37–42, Aug 2009. doi:10.1109/WISP.2009.5286542.

[90] Juan C. Alvarez, Diego Alvarez, Antonio M. Lopez, and Rafael C. Gonzlez. Pedestrian navigation based on a waist-worn inertial sensor. *MDPI Sensors Journal*, 12(8):10536–49, December 2012. doi:10.3390/s120810536.

[91] M. Basso, M. Galanti, G. Innocenti, and D. Miceli. Pedestrian dead reckoning based on frequency self-synchronization and body kinematics. *IEEE Sensors Journal*, 17(2):534–545, Jan 2017. doi:10.1109/JSEN.2016.2631629.

[92] R. C. Gonzalez, D. Alvarez, A. M. Lopez, and J. C. Alvarez. Modified pendulum model for mean step length estimation. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1371–1374, Aug 2007. doi:10.1109/IEMBS.2007.4352553.

[93] N. Strozzi, F. Parisi, and G. Ferrari. On single sensor-based inertial navigation. In *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 300–305, June 2016. doi:10.1109/BSN.2016.7516278.

[94] N. Strozzi, F. Parisi, and G. Ferrari. A multifloor hybrid inertial/barometric navigation system. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–5, Oct 2016. doi:10.1109/IPIN.2016.7743703.

[95] BoschSensortec. *Data sheet BMP180 Digital pressure sensor*, Apr. 2013. Rev. 2.5.

[96] Freaklabs. *FREAKDUINO 900 MHz Long Range Wireless Arduino-Compatible Prototyping Platform v2.1a Datasheet*, sept 2013. Rev. 2.1A.

[97] Binghao Li, B. Harvey, and T. Gallagher. Using barometers to determine the height for indoor positioning. In *International Conference on Indoor Positioning and Indoor Navigation*, pages 1–7, Oct 2013. doi:10.1109/IPIN.2013.6817923.

[98] N. Strozzi, F. Parisi, and G. Ferrari. Impact of on-body imu placement on inertial navigation. *IET Wireless Sensor Systems*, 8(1):3–9, 2018. doi:10.1049/iet-wss.2017.0087.

[99] N. Strozzi, F. Parisi, and G. Ferrari. A novel step detection and step length estimation algorithm for hand-held smartphones. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7, Sep. 2018. doi:10.1109/IPIN.2018.8533807.

[100] Sebastiano Campisi. Hyper imu. URL: https://ianovir.com/works/mobile/hyperimu/.

[101] Sensors overview - android developers. URL: https://developer.android.com/guide/topics/sensors/sensors_overview.

Acknowledgments

The time spent as PhD student in these last three years, certainly gave me more knowledge, more expertise, and, most of all, an infinite amount of errors. I really believe that the most powerful lesson in these years has been the capacity to learn from my mistakes. As Thomas Edison used to say “I have not failed. I’ve just found 10,000 ways that won’t work.” Errors happens, and that’s a fact. Sometimes, during these years, I felt lost, unable to accomplish all the task which I should have done. My great good fortune, has been the presence of (I don’t even know how to describe them) “awesome” persons around me.

First and foremost, I wish to sincerely thank my supervisor, Prof. Gianluigi Ferrari. With his wisdom, he continuously guided me during the years of Ph.D. course, inspired the research activity and helped me when I thought I shouldn’t succeed. I totally want to say thank you to my mentor, my friend, Federico. Without his help, I probably wasn’t even able to turn on the sensors. Thanks to all the IoTLab guys, present and past (yes Carlo, I’m talking about you, thank you for all the time lost in interesting/useless hobbies). Many thanks to Luca, for working 23h per day and bearing me the remaining time. To Antonio, for the amazing healthy time spent together in (short) sport activities. To Davide, for his accurate (I would say “meticulous”!) suggestions. To Laura, for her infinite kindness and support. To Yani, for her sushi addiction (which make me a little bit fatter).

My family is, and will be forever, a part of me, so thank you mom and dad: without you nothing of this would be possible. Thank you Sere, to accompany me on the road to university almost every day during these years. Thank you Monica, Silvio, Franci e Fil, for all the creative mess all around me. Thank you Pia, I know you’d be proud of me. Finally, thank you Giulia. In theory, I’ve to say something like “I want to thank you for bearing me all the times,” but you know this is not the truth. The truth is that you are always on my side, and that’s what I love about you.