FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Motion Compatibility and Adaptive Filtering for Indoor Localization

Pedro Henrique Oliveira Santos

Mestrado Integrado em Bioengenharia

Supervisor: Dirk Elias / Vânia Guimarães

July 5, 2017

Resumo

Atualmente, a localização *outdoor* (feita via GPS, redes móveis, etc.) está facilmente disponível, em veículos, smartphones ou aparelhos especializados. Para além de ajuda à navegação, a informação de localização é também utilizada para personalizar a partilha de informação, o que é de óbvio interesse do ponto de vista da otimização quer da experiência do utilizador, quer de estratégias de marketing. Por outro lado, os sinais utilizados para a localização *outdoor* não têm precisão suficiente para o problema da localização *indoor* ou podem nem estar disponíveis, pelo que é necessária uma solução alternativa. A solução deste problema, além das vantagens associadas à localização *outdoor*, traz benefícios na gestão de trabalhadores (importante em profissões de risco) e na automação de edifícios, nomeadamente para potenciar Ambient Assisted Living (permitindo a idosos ou pessoas com deficiência serem mais autónomos).

Existem três grandes fontes independentes de informação disponíveis para a localização *indoor*: sensores inerciais, sinais de oportunidade (wi-fi, bluetooth, campo magnético, etc.) e o mapa/planta do edifício. Nesta abordagem, escolheu-se não utilizar os sinais de oportunidade como fonte de informação, dado que estes implicam um trabalho de análise prévia do edifício, para obter a distribuição desses sinais. Para que o sistema possa ser implementado em diferentes locais facilmente, usar-se-á apenas um par de IMUs de baixo custo (Inertial Measurement Units) colocadas em ambos os pés e um smartphone para processamento de dados, dotado da planta do edifício. A escolha da utilização de sensores inerciais externos ao smartphone é justificada pelo facto da colocação nos pés permitir o uso de Zero Velocity Updates (quando se deteta que o pé está plantado no chão), que diminuem o erro de integração. Adicionalmente, a utilização de mais do que um sensor permite cruzar a informação dos dois pés, permitindo aumentar a robustez do sistema e diminuir o impacto dos erros de ambos os sensores.

A solução deste problema passa por vários sub-problemas. É ncessário notar que o problema pressupõe não só o cálculo incremental da posição do utilizador conhecendo a sua posição inicial (*Pedestrian Dead Reckoning*) como também a deteção desta (*Indoor Localization*). No entanto, no contexto desta tese, o foco será dado ao problema de Dead Reckoning numa perspetiva de manter a sua viabilidade a longo prazo, evitando acumulação de erro. Numa primeira fase, é necessário detetar a ocorrência de um passo, estimar o seu comprimento e a sua direção. Estas estimativas estão limitadas pelas propriedades dos sensores inerciais e pelo facto de as interferências magnéticas no interior de edifícios inviabilizarem a utilização do magnetómetro deixando a estimativa da direção apenas a cargo do giroscópio, que sofre desvios com o tempo. Paralelamente, é necessário cruzar a informação incremental da posição do utilizador, a sua trajetória, com a informação do mapa, para que se corrijam erros e se possa obter a localização atual do utilizador a qualquer momento.

No âmbito da dissertação, desenvolveu-se uma aplicação *Android* capaz de localizar o seu utilizador apenas com informação de dois sensores inerciais de 6 graus de liberdade a 100Hz colocados nos pés e da planta do edifício, sabendo a sua localização inicial.

Abstract

Outdoor localization (using GPS or mobile networks) is nowadays widely available in vehicles, phones or dedicated equipment. Besides helping with navigation, location information can also be used to optimize information sharing, improving user experience and empowering companies to carry out custom marketing strategies according to user profile. However, the signals that are used for outdoor localization are either not sufficiently precise or even unavailable in certain indoor scenarios and as such an alternative system is needed. Solving the problem of indoor localization enables not only indoor navigation and optimized information sharing but also workforce management (especially important for jobs with risks) and building automation. It is a great aid for Ambient Assisted Living, and as such can indirectly allow seniors or handicapped people to be more autonomous.

There are three main independent sources of information to achieve indoor localization: inertial sensors, opportunistic signals (wi-fi, Bluetooth, magnetic field fluctuations, etc.) and the map or plan of the building. In this approach, no opportunistic signals will be used, as doing so would require a previous scan of the building to obtain the signal distributions. With scalability and ease of deployment in mind, only low-cost IMUs (Inertial Measurement Units) will be used. The sensors will be placed on both feet and a smartphone, loaded with the floor plan, will be used for data processing. The decision of using inertial sensors external to the smartphone is explained by the fact that foot-placed IMUs enable the usage of Zero Velocity Updates (zeroing the velocity when the foot is on the ground), bounding the integration error. Additionally, using more than one sensor enables sensor errors and drifts to be estimated and compensated to a certain degree and allows the system to be more robust.

The indoor localization problem can be divided in sub-problems. It is important to note that localization implies not only the incremental estimation of the user position from a known starting position (Pedestrian Dead Reckoning - PDR) but also the estimation of the starting position if it is unknown. However, in this dissertation, the focus will be on PDR, with the goal of making it viable even in longer walks, avoiding large error accumulation. The first sub-problems are step detection and estimation of length and direction. These estimates are limited by the properties of the inertial sensors and by the fact that indoor magnetic interferences make magnetometer readings nearly unusable, which causes the step direction to be estimated solely by the gyroscope. The gyroscope is affected by linear acceleration, temperature and drifts over time and as such these errors must be modeled. Consequently, parallel to step quantification, it is necessary to match the incremental position of the user (i.e. his/her trajectory) with map information. This is crucial for the correction of errors (bounding the error propagation typical of inertial sensors), and for the estimation of the current position of the user at all times.

In this thesis work, an Android application capable of locating the user from the signal of two foot-mounted 6-DOF IMUs at 100Hz, the floor plan and the knowledge of the initial position was developed.

Acknowledgements

I would like to acknowledge Professor Dirk Elias for the suggestion of the thesis work and the opportunity to develop this work at Fraunhofer Portugal. I would also like to thank Vânia Guimarães, for helping me fully understand the scope and requirements of the work, for answering my questions, giving me a head start on the work and for our discussions regarding the design of the system. As an extension, I would also like to thank Lourenço Castro and the remaining researchers at the Precise Indoor Location project for letting me give input at the group meetings, helping with having a wider view on the complete Indoor Localization problem and providing me with a solid framework on which to develop my algorithms.

A word of appreciation goes out to all my colleagues and friends at the Faculty of Engineering, with whom I learned so many things and specifically to all of the Master Students who worked on their Master Thesis alongside me at Fraunhofer.

On the personal side of life, I would like to thank my sister for helping me grow as a person every day, to my parents for doing everything in their power to make me reach my potential, to my grandparents for kickstarting my love of Maths and Sciences and to my girlfriend for all the support and patience. To all of them, a big thank you for your love and for being at my side not only during this work, but throughout life.

Pedro Henrique Santos

"To achieve great things, two things are needed
A plan, and not quite enough time"

Leonard Bernstein

Contents

List of Figures

List of Tables

Symbols and Abbreviations

- BF - Bayesian Filtering
- CRF - Conditional Random Fields
- FSM - Finite State Machine
- GF - Gaussian Filtering
- HDE - Heuristic Drift Elimination
- HMM - Hidden Markov Model
- IL - Indoor Localization
- IMU - Inertial Measurement Unit
- KF - Kalman Filter
- MM - Map Matching
- MEMS - Micro Electro-Mechanical Systems
- PDR - Pedestrian Dead Reckoning
- PF - Particle Filter
- SHM - Simple Harmonic Motion
- ZUPT - Zero Velocity Update

Chapter 1

Introduction

1.1 Motivation

The advent of smartphones and their widespread distribution mean that most people carry around a powerful set of sensors. As such, there is an opportunity for techniques and methods to be easily scaled, with virtually no hardware cost, if they are developed specifically for this platform.

Similarly to outdoor localization, developing an indoor localization (IL) solution enables navigation (e.g. guiding costumers around shopping malls, airports, etc.), workforce management (optimizing allocation of human resources in hospitals, policeman and security in a building, and managing operations with associated risk such as rescue or reconnaissance missions, etc.) and context-dependent information sharing (e.g. targeted advertising, as in suggesting items while moving through a store and possibly retrieving profile information from a user according to its past purchases and movement inside the store). Knowledge of user location can also be applied to social networking, medical care in hospitals, to mobile robots or augmented reality. Furthermore, it also enables applications in the field of home automation, optimizing the experience of the user and energy use, for example. This enables Ambient Assisted Living services, creating an environment that is compatible with the needs of the elderly or disabled people and enabling them to be as autonomous as possible [1].

Current localization systems are not able to provide a suitable solution. On the one hand, some systems are not accurate enough, as observed when outdoor localization techniques such as GPS are used to try to determine an indoor location (low signal availability and insufficient precision for indoor applications). On the other hand, accurate systems often require the installation of specific equipment such as beacons, which increases cost of installation and maintenance [2]. Methods that require fingerprinting of the building or measurement of opportunistic signals such as Wi-Fi readings or magnetic field fluctuations require collection and periodic update of detailed building data which hinders their scalability and increases cost [3].

An alternative that is being increasingly explored leverages MEMS (Micro Electro-Mechanical Systems) inertial sensors. Even though the smartphone contains a set of sensors (an IMU, inertial

measurement unit), relying on these for a purely inertial sensor proves to be an extremely difficult task, due to the different carrying contexts (position of the phone relative to the user). Current approaches either leverage opportunistic signals and fingerprints or impose several carrying context limitations (such as holding the phone in front of the user or in the pocket of the pants) or use activity matching (detecting activities such as stair climbing, riding an elevator or other interesting motion signatures [4]) to correct large errors while being unable to deal with large portions of movement without any specific piece of information that can provide absolute localization. These methods also have less information to obtain step length when compared to distributed wearable sensors, since using fixed sensors means the signal pattern can be predicted with added confidence.

However, even if using an IMU in a fixed position, depending exclusively on raw inertial sensor readings proves to be an impossible task, as error propagates quite rapidly due to sensor errors and errors translating the signals to the Earth Frame (as described in Chapter 2). Double-integrating acceleration data without any knowledge of velocity is an impossible task (and even if the initial velocity is zero, if it does not return to zero after the user stops, for example, error will build), which leads to inaccurate user tracking results. It is also necessary to rotate the acceleration signals from the sensor coordinate system to Earth's coordinate system (estimation of current orientation and rotation, attitude), as the displacement that is being estimated is in the Earth frame. The main limitation of using inertial data for attitude estimation is that, in an indoor setting, the magnetometer signal is very distorted mainly because of soft and hard iron effects caused by the structure of the building, and as such requires sophisticated modeling to be usable [5, 6]. Without applying magnetometer data, estimation of orientation around the z-axis in the world frame depends solely on the gyroscope. This means that precision errors as well as bias drift will result in large deviations from the actual true orientation. As such, without additional information, absolute localization is impossible. Therefore, there must be an additional and independent source of information so that errors can be corrected (or at least bounded) and the algorithm can adapt to the user and its surroundings, enabling the conversion of a series of steps into a series of user positions.

An interesting source of additional independent information is the building floor plan, as it is more easily collected (available through computer vision techniques applied to floor plans present in buildings, for example) and of a much more static nature in comparison with the mapping of wireless signals or magnetic field disturbances. Acquisition of floor plans is outside the scope of this project, and the assumption is made that the floor plan exists in the system.

Furthermore, there is another piece of information that can be very important. If the IMUs are fixed to the foot of the user it is possible to exploit the fact that when a foot moves the other foot will be planted on the ground bearing all weight, as it means the foot will not be moving (zero velocity and zero angular rate of turn). Adjusting the velocity estimates at that point will bound integration errors and is key for the displacement reconstruction accuracy and stability over time. These methods are named ZUPT (Zero velocity UPdaTe) and ZARU (Zero Angular Rate Update), respectively.

Alternatively or in parallel, using more than one IMU in a fixed position will allow for the

sensor measurement errors, which are the main cause of the lack of stability of inertial PDR systems, to be compensated. A compromise between signal quality/amount of information and cost/simplicity of the system is necessary. As such, the system developed during this thesis work will make use of only two foot-placed 6-DOF IMUs (3-axes accelerometer, 3-axis gyroscope, one IMU on each foot) as a proof of concept.

1.2 Problem Overview and System Concept

Approaching IL through inertial sensors leads to simplifying the problem to Pedestrian Dead Reckoning (PDR), as one cannot infer absolute position from inertial sensors alone. PDR means looking at the problem as a tracking problem and not as a localization problem. The system needs to track the trajectory (position of the user through time) the user is describing, obtaining future positions as a combination of the trajectory and the initial position. In the context of this thesis work, and since neither a source of absolute location (such as distance to a Wi-Fi access point) nor an absolute source of direction (filtered magnetometer signal) will be used, which makes the absolute localization of the user within the building a very complicated problem, a simplifying assumption of known initial position will be made. This assumption does not severely hamper the applicability of the solution, as the initial location can either be matched to the entrance of the building through integration with GPS in an outdoor context [7], manually provided by the user or even retrieved through other IL methods (such as planting a single beacon on the entrance of the building). Throughout the system, design choices were always done with keeping assumptions to a minimum in mind. In fact, besides the initial location and floor plan availability assumptions, the only other assumption that is made is that the sensor is fixed to the foot and as such its orientation and position will always be the same as the foot, which will be enforced by tightly fitting the IMU to the shoe. There will be no gait limitations or behavioral assumptions (such as walking in straight lines or with no lateral or backwards steps or even assuming that forward motion is done by preserving neutral distance perpendicularly to the step (i.e. no exaggerated sideways opening of the lower limbs). Furthermore, even if there is not a period where the foot is still between every step (the stance phase, as described and explored in Chapter 3) so that ZUPT and ZARU can be applied, as is the case of running, the system will still provide displacement estimates, even if these are much more prone to error (due to the lack of updates and possible sensor saturation from acceleration outside the input range).

Each IMU, integrated in a sensor array named Pandlet which was developed at Fraunhofer Portugal, (which also contains a magnetometer, barometer, humidity sensor, among others), will be mounted on the top of the foot, as shown in Figure 1.1. When designing a foot-mounted IMU PDR system and shoes with integrated sensors, one of the ideas is to include the sensors inside the heel of the boot [8–10]. However, placing it on the top of the foot has some advantages, namely smaller impacts (acceleration due to strong impacts on the heel can exceed the input range and the gyroscope is sensitive to linear acceleration - see Chapter 2) and less heating (which causes sensor bias drift). Furthermore, mounting it beneath the shoelaces has the added bonus of not requiring

additional straps. In all cases, a final solution contained in a smaller case could be easier to fit beneath the shoelaces, be accompanied by adjustable straps or even integrated in the top of the shoe/boot, so as to ensure perfect fixation of the IMU to the foot and in turn better quality of the obtained signal.

Figure 1.1: Placement of the Sensors

As mentioned in the previous section, lack of velocity information, less than ideal precision and sensor drift will be the main limitations of the inertial sensor based system. PDR, as a tracking problem, means that each point depends on the estimation of the previous points and propagating errors can quickly make the system unusable. Additional sources of information will be needed along with an effective way to merge all available information. In order to overcome these limitations the following design choices were made:

- **Integration of Multiple Sensors -** Since the errors and especially the sensor drift will be independent in different sensors, using more than one IMU will allow for errors to be compensated for and better estimates to be computed. (Chapter 4)
- **Placement of Sensors on the Feet -** Placing the IMUs on the feet enables the use of ZUPT, by exploring the characteristics of human gait. The fact that while one foot swings the other is planted on the ground and as such will have zero velocity can be explored by detecting these instances and correcting the calculation of velocity at that point, mitigating the problem of unknown initial velocity and bounding the error of the velocity estimation (and, as a consequence, of position estimation).(Chapter 3). In addition, the stationary foot will also have zero angular rate of turn. Detecting these periods allows gyroscope measurement errors to be detected and compensated, increasing the accuracy of the attitude (orientation and rotation) estimation (Chapter 2).
- **Map Matching -** This comparison between map and position or trajectory, named map matching (MM), is a key part of the system. MM will be responsible for further correcting errors in step length and direction and estimating drift. The main goal of MM is to bound errors during movement in constrained areas (narrow corridors for example) so that the

error can be estimated and inertial tracking can remain usable for longer periods of time. (Chapter 5)

Taking a closer look at the PDR problem and the user tracking, it is necessary to detect and measure changes in position (in normal human gait, the result of steps). As such, information from the Inertial Measurement Unit (IMU) is processed in order to obtain three important pieces of information: Step Occurrence, Step Length and Heading of the Step. These three pieces of information are based on uncertain measurements, and errors propagate. An erroneous detection of a step will always be followed by an erroneous quantification of the non-existent step, for example. Dealing with uncertainty will be crucial throughout the system. In addition, and in order to be able to quantify a step in terms of length and heading, an attitude estimation algorithm is necessary. This is because the acceleration signal needs to be translated from the sensor own coordinate system to the earth system, so that displacement is calculated on the Earth coordinates. By estimating the current sensor attitude, it is possible to obtain the acceleration and angular rate in earth coordinates by rotating the values obtained in the sensor coordinates.

In sum, the system should be able to continuously estimate the sensor attitude from the gyroscope and accelerometer values and then use these to detect and quantify steps. The trajectories generated by analysis of the signals from both IMU are then fused, constrained by the map and corrected so that the series of positions that they describe are valid. This correction should improve the accuracy of the system, by preventing error propagation and improving the detection and quantification of steps through adaptation to the user. Throughout the system, computational complexity of the chosen algorithms must be kept in check, so that it can run in a smartphone in real-time.

1.3 Deconstructing the Problem

This problem can be subdivided in smaller problems not only to help with clarity but also because they are mostly independent and can be tackled individually to some extent. As such, the challenges can be divided into several tasks or modules:

- **Attitude Estimation -** In order to quantify a step in terms of its length and heading (as well as height, if necessary), the attitude of the sensor - and consequently of the foot, as the IMU is fixed - needs to be estimated. As mentioned before, this is crucial in order to be able to obtain acceleration values that correspond to movement in the Earth coordinates (x, y and z - North, East and Down) so that the translation of the foot can effectively be applied on the map enabling the calculation of heading and direction simultaneously (by estimating how much the foot has moved in each direction).

  Attitude estimation is done through the accelerometer reference to the Earth z-axis through the measurement of the Earth's gravitational acceleration (which provides the gyroscope current pitch and roll angles). From an initial alignment of the y- and x-axis (rotation around

the z-axis, yaw angle) that must be provided, the gyroscope will be used to keep track of rotation around all axis (and being the sole source of yaw).

- **Zero Velocity Update Region Segmentation -** Due to the great importance of the ZUPT in mitigating integration errors and its intrinsic relationship with step detection - in this thesis work a step will be defined as the interval between two zero velocity regions - accurately identifying the regions of the signal where the foot is planted is crucial, doubling as a step detector.

- **Floor Change Detection -** Most IL systems work either in 2D or 2.5D (2D plus floor information) since buildings are usually structured in different floors and even in places where stairs or ramps are abundant, a 2D tracking coupled with changing of the floor map according to detected floor transitions will be sufficient. As such, the main goal is to detect the action of climbing or descending stairs through detection of patterns of vertical displacement within steps and elevator through characteristic acceleration patterns namely on the z-axis. The detection of a floor change also helps correct position errors since if stairs are detected, the position can be adjusted to the nearest exit.

- **Step Quantification -** Once a step is detected, the acceleration vector, translated to Earth coordinates, is used for the estimation of the displacement associated with that step, by smoothing of the signal and double-integration. Additionally, it is necessary to analyze whether the estimated motion is compatible with the expectations (namely in terms of maximum velocity and step length) or we need to reprocess the signal to find different zero-velocity regions.

- **Fusion of Information from Different IMUs -** In order to compensate errors due to sensor drift or incorrect estimations of the algorithm, the series of positions of each sensor as well as the estimated heading and step length must be compared, matched and adjusted. Ideally, this module does not impose heuristic assumptions (such as straight line walking or prediction of next step) and instead leverages only anatomical constraints, namely the maximum distance between feet.

- **Trajectory Generation and Map Matching -** This sub-problem is where the bigger improvements can be made and can substantially increase the accuracy of the system even when using less than ideal estimates for the movement of the user. When generating trajectories, one needs to be able to deal with the uncertainty associated with every estimation. As for MM, it is important that it is capable of locating the user even as PDR error builds and of providing valuable learning feedback for the previous module. Ideally, absolute position could be retrieved from Matching a sufficiently long and informative trajectory with a sufficiently constrained map, where the trajectory would only fit in one starting location.

- **Knowledge Representation -** Representing the map (resolution, structure), positions, trajectories and probabilities, orientation and profiles (for the user and possibly for the building as well) are topics that run parallel to the previous problems and that need to be explored.

1.4 Aim of the Thesis

The aim of this dissertation is to develop a PDR system capable of maintaining adequate accuracy (within 2 meters) over significant distance (over 100 meters). The system will be based on two IMUs developed at Fraunhofer Portugal, mounted on both feet, under the shoelaces. The IMUs communicate with a smartphone through Bluetooth connections. The smartphone is responsible for the processing of the data and estimation of the position of the user in real-time. Besides the inertial info retrieved from the two IMUs, the system will start from a known position and heading and will have the information of the floormap.

The system should be able to work on its own but also work as complement to other IL systems that make use of opportunistic signals, as an enhancer of short-term accuracy. In the context of the dissertation, the developed system will work in 2D, without detection of floor changing activities.

1.5 Contributions

The main contribution of this thesis is the fusion of information of the two IMUs using a step as a basic unit of displacement, instead of approaches which apply a Kalman Filter to constrain the continuous estimates of displacement from the measured signals. Alternatively, the proposed Gaussian Filter algorithm with anatomical constraints and drift modeling is much less computationally expensive.

In terms of Map Matching, a new truncated Gaussian distribution algorithm through sampling is proposed. When said algorithm is incapable of generating a valid position, a geometric MM algorithm specifically tailored for indoor pedestrian tracking is applied, which is capable of recovering from absolute errors of over 5 meters in building areas with adequate constraints.

Furthermore, an attitude filter based on Madgwick's Gradient Descent Filter [11] with additional frequency filters and specifically geared towards foot attitude estimation during gait was developed and implemented. An analysis of alternative methods for Map Matching focused on the specific needs of this system is done, including a novel analysis of how Active Contour Models can be applied to this problem.

1.6 Thesis Structure

This work is divided in 4 chapters describing the various sub-problems, which are, to a certain extent, independent. Within each Chapter, besides the implementation details of the developed IL system, there is an overview of the state of the art methods and how they can be applied to the specificities of each respective model within this project. Moreover, the performance of the respective module is discussed. Algorithm choices are described in detail and its advantages and disadvantages are evaluated.

In short, Chapter 2 reviews attitude representations, the attitude estimation problem for 6-DOF IMU and the alternative filters that can be applied. Chapter 3 tackles step detection and quantification, namely in terms of data processing (specifically ZUPT) and integration of the filtered acceleration signal, in order to obtain the displacement relative to a step. The decision to treat displacement as a discrete event (step) instead of continuously integrating data and filtering will also be described in detail in this Chapter. In turn, Chapter 4 is focused on the processing of information from both feet with the goal of compensating errors through enforcement of anatomical constraints, resulting in an improvement over the estimates from both sensors independently. Chapter 5 contains a review of MM algorithms with special emphasis on those geared towards indoor pedestrian localization applications and presents the developed MM algorithm, as well as other alternatives that were tested and conceptualized but are not implemented in the current system.

Following this review and presentation of the implementation details, a system overview is presented on Chapter 6, where the interaction between all modules is explained in further detail. Chapter 7 describes the tests that were performed, presents the results and contains its analysis and discussion. Finally, Chapter 8 reasserts the critical analysis of the performance of the system that is done throughout each chapter, its limitations, possible improvements and research directions. It also contains final remarks regarding this dissertation.

Chapter 2

Attitude Estimation

In order to use inertial data to estimate displacement in the Earth coordinates, it is necessary to obtain the orientation and rotation (attitude) of the IMUs, so that the accelerometer and gyroscope signals can be rotated into the Earth coordinate system. This is necessary because the sensor measures acceleration and rotation in its own instantaneous coordinate system which will change according to the rotation and translation it experiences during movement. Only after applying the coordinate transformation, and with adequate processing, can the signals be integrated in order to obtain the displacement in the Earth frame and track the movement of the user. In this system, the sensors will be carefully strapped with the shoelaces so that there is no relative movement between the foot and the sensor. As such, the goal of this module is to relate the instantaneous coordinate system of the foot and the sensor to the Earth coordinate frame.

The key with attitude estimation is that the accelerometer measures proper acceleration (acceleration in the instantaneous rest frame of the object) instead of the acceleration in a particular coordinate frame, and as such, an accelerometer resting on top of a table will measure Earth's gravity acceleration ($\simeq 9.81 ms^{-2}$ upwards). If we describe Earth's North-East-Down as x-y-z (as in Figure 2.1), acceleration data will provide a long term reference to the z-axis, even as measurements get clouded with acceleration from the movement itself. Even though the short-term precision of the estimation is not ideal, the accelerometer does not experience drift and as such the reference to the z-axis will remain stable over time. On the other hand, the gyroscope will enable short-term tracking of rotation with adequate precision, however, its accuracy will deteriorate with time as its bias - estimated rate of turn when stationary - changes (which is a result of sensitivity to temperature, linear acceleration or errors in calibration) [12]. The signal of the gyroscope can be modeled as in Equation 2.1, where $b_s(t)$ is the sensor bias (non-zero mean noise) and n_s is zero-mean sensor noise.

$$\Omega_{sensor}(t) = \Omega_{true}(t) + b_s(t) + n_s(t) \tag{2.1}$$

The challenge is to use gyro data to track short-term movements while accounting for bias and using accelerometer data to make sure the z-axis reference is kept. As no magnetometer is used on this system and as such there is no long-term reference to the Earth's North direction,

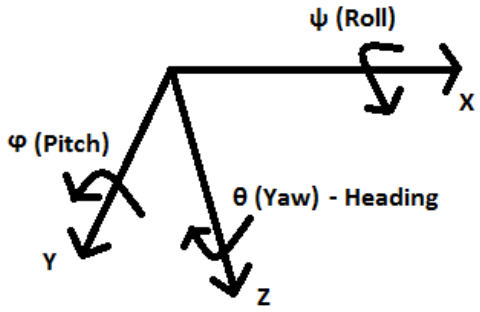


Figure 2.1: Rotation relative to the East-North-Up Frame. Throughout the document, θ (heading), will be used to refer to the angle between north and the current direction relative to the xy plane

.

integration of angular rate of turn will lead to drift over time. Nonetheless, the attitude estimation module should guarantee a stable reference to the z-axis and leave the correction of drift around the z-axis to the Dead Reckoning and Map Matching Modules, as described in Chapters 4 and 5. Adequate alignment with the z-axis will enable the step tracking to return precise displacement in the x-y plane, even if it is rotated around the z-axis. As floors are usually plane, the modulus of the displacement will be the step distance, even if rotated in direction. Obtaining accurate distance estimation in the floor plane is very important, as it means position filtering can correct errors with the main focus being heading and with good confidence on the traveled distance.

For simplicity, in this work the Pitch angles refers to rotations around the Earth x axis, the Roll angle to rotations around the Earth y axis and finally, the Yaw angle to rotations around the Earth z axis. Rotations around the z axis are important since that is what quantifies the direction (heading) of movement in the xy plane.

In this Chapter, the state of the art regarding approaches to attitude estimation is presented, followed by the implementation details of the designed attitude estimation filter and a review on the design of the entire module.

2.1 Attitude Representations

The orientation and rotation of a body in regards to a fixed coordinate system can be described by multiple conventions, which will be briefly reviewed below.

2.1.1 Rotation Matrices, Euler Angles and Axis-Angle

Euler Angles and Tait-Bryan Angles (sometimes also called Euler Angles and as such the name Proper Euler Angles would apply to the former) are perhaps the most intuitive convention for rotation representation. Euler Angles describe a rotation by the sequence of rotations over three axis (that must be defined and ordered *a priori*). These representations can be intrinsic (the

second and third rotations apply on the rotated axis) or extrinsic (when the rotations are relative to the original coordinate frame axis). There are 12 possible representations of rotation with Euler Angles: X-Y-X (rotation around the x axis, then around the y axis and finally around x again), Y-Z-Y, Z-Y-Z, X-Z-X, Y-X-Y, Z-X-Z, both intrinsic and extrinsic. Tait-Bryan Angles, mostly used in aerospace navigation include another 12 combinations: X-Y-Z, X-Z-Y, Y-X-Z, Y-Z-X, Z-X-Y, Z-Y-X, again intrinsic or extrinsic. This means that the axis being used and their order must be carefully defined.

In order to compute and apply these rotations to vectors, one can use a rotation matrices, as exemplified in Equation 2.2 for an extrinsic X-Y-Z Tait-Bryan convention with angles α, β and γ respectively:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\ \alpha & sin\ \alpha \\ 0 & -sin\ \alpha & -cos\ \alpha \end{bmatrix} * \begin{bmatrix} cos\ \beta & 0 & sin\ \beta \\ 0 & 1 & 0 \\ -sin\ \theta & 0 & -cos\ \theta \end{bmatrix} * \begin{bmatrix} cos\ \gamma & sin\ \gamma & 0 \\ sin\ \gamma & -cos\ \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ = \begin{bmatrix} cos\beta & -cos\gamma sin\beta & sin\beta sin\gamma \\ cos\alpha sin\beta & cos\alpha cos\beta cos\gamma - sin\alpha sin\gamma & -cos\gamma sin\alpha - cos\alpha cos\beta sin\gamma \\ sin\alpha sin\beta & cos\alpha sin\gamma + cos\beta cos\gamma sin\alpha & cos\alpha cos\gamma - cos\beta sin\alpha sin\gamma \end{bmatrix} \tag{2.2}$$

The main issue with Euler Angles is gimbal lock, which occurs when a degree of freedom is lost through a rotation that aligns two of the axis. For example, if the sensor rotates around the x axis 90 deg we arrive at a gimbal lock configuration. At this point, if the rotation continues, the rotation around the y axis would change from 0 to 180 deg. It is this kind of numerical instability and difference between two very close values for attitude that makes this convention unsuitable for this application. In addition, executing the rotation through rotation matrices is more computationally expensive than the alternatives we will present below.

Alternatively, a rotation can be represented by a unit vector describing the axis around which the rotation will take place and its angle (as in Figure 2.2), since any rotation in a 3-Dimensional space is equivalent to a rotation around a fixed axis (Euler's rotation theorem). A rotation in this representation ($\hat{\omega}$), called Axis-Angle, can then be applied to a vector ($\hat{e}$) efficiently with the Rodrigues' Formula, (2.3).

$$\hat{e}_{rot} = (cos\ \theta)\hat{e} + (sin\ \theta)(\hat{\omega} \times \hat{e}) + (1 - cos\ \theta)(\hat{\omega}.\hat{e})\hat{\omega} \tag{2.3}$$

While the axis-angle representation has good properties and avoids the gimbal lock problem by using a fourth redundant parameter to describe a 3D rotation, the similar but more sophisticated approach with quaternions has added value because of interesting mathematical properties.

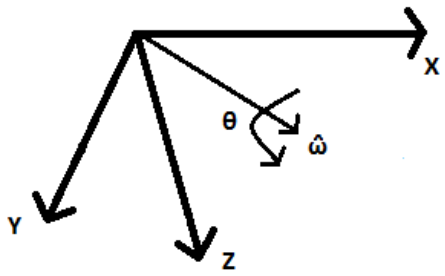


Figure 2.2: Axis Angle Representation of Rotation θ around axis ω

.

2.1.2 Quaternions

Quaternions (4D complex number) are widely used in attitude estimation and rotation representation problems due to the efficiency of rotating vectors, the ability to interpolate between rotations and the inherent simplicity of the numerical system once its relation with the orientation of the object is understood [13]. A quaternion is made up of a real component (q_0) and three imaginary components (iq_1, jq_2 *and* kq_2). A unit quaternion $\hat{q}$ can be used to describe a rotation θ along an axis $\hat{r}$ (2.4):

$$\hat{q} = [q_0\ q_1\ q_2\ q_3] = [cos\ \frac{\theta}{2}\ \ (sin\ \frac{\theta}{2})\hat{r}], \qquad with \quad \sqrt{q_0^2+q_1^2+q_2^2+q_3^2} = 1 \tag{2.4}$$

This rotation can be easily applied to the previous attitude to obtain the new attitude by quaternion multiplication as in Equation 2.5:

$$\hat{a} \otimes \hat{b} = [a_0\ a_1\ a_2\ a_3] \otimes [b_0\ b_1\ b_2\ b_3] = \begin{Bmatrix} a_0b_0 - a_1b_1 - a_2b_2 - a_3b_3 \\ a_0b_1 + a_1b_0 - a_2b_3 - a_3b_2 \\ a_0b_2 - a_1b_3 - a_2b_0 - a_3b_1 \\ a_0b_3 - a_1b_2 - a_2b_1 - a_3b_0 \end{Bmatrix}^T \tag{2.5}$$

Additionally, quaternions can be differentiated so that the attitude can be updated using the angular rate and the time interval, (2.6):

$$\begin{pmatrix} \frac{d}{dt}q_w(t) \\ \frac{d}{dt}q_x(t) \\ \frac{d}{dt}q_y(t) \\ \frac{d}{dt}q_z(t) \end{pmatrix} = \frac{1}{2} \begin{bmatrix} -q_x(t) & -q_y(t) & -q_z(t) \\ q_w(t) & q_z(t) & -q_y(t) \\ -q_z(t) & q_w(t) & q_x(t) \\ q_y(t) & -q_x(t) & q_w(t) \end{bmatrix} \begin{pmatrix} \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{pmatrix} \tag{2.6}$$

Quaternions allow for simple representation of current attitude, rotations and addition of a rotation to a past attitude, enabling efficient calculations. Furthermore, they are unambiguous, do not suffer from Gimbal Lock, are numerically stable (it is easier to normalize a quaternion than correcting numerical errors in rotation matrices) and can be initialized even when there is no rotation [1 0 0 0], unlike the Axis-Angle representation. In conclusion, the quaternion representation will be the one applied in this work.

2.2 Sensor Fusion - Review of the State of the Art

Sensor Fusion is the process of obtaining more precise, accurate or overall reliable values for a measurement by using multiple sources of information (sensors or external knowledge about the system). In this case, sensor fusion will play a crucial role in attitude estimation since if this task was done with simple integration of the raw angular rate of turn retrieved from the gyroscope, the propagation of integration errors as well as the bias instability of the gyroscope itself would lead to wildly inaccurate estimates in a few steps. As such, a fusion algorithm which combines the best features of both the gyroscope and the accelerometer is necessary. This algorithm should enable the update of the attitude estimate every time a new accelerometer or gyroscope sample is available and remain stable over time. The algorithm may simply filter and fuse the data (first order) or also model its drift (second order).

2.2.1 Kalman Filter

The Kalman Filter (KF) and its extensions (Extended Kalman, Unscented Kalman) are popular sensor fusion algorithms, as they are designed to produce estimates of variables from noisy sensor data. The KF works in two steps: Prediction and Update. During the Prediction Step, the algorithm uses the current state of the system and the process model to predict the next state of the model. Immediately after, the sensor measurements are inserted in the measurement model and a new estimate is made. The final estimate of the filter for each iteration is a weighted average of the process model and the measurement model that controls how much the algorithm reacts to changes in measurements. The key is that the estimates for the state variables always have an associated covariance value, that describes the certainty the algorithm has regarding the current estimate.

The first stage in designing a Kalman Filter is deciding on which state variables to model ($\hat{x}$). In the case of a Sensor Fusion algorithm for attitude estimation, these may be the current attitude (in quaternion or Euler Angles form) and the current angular rate of turn or the gyro biases. The next stage is to design the process model (F_k), that is, how to predict the next state for all variables from the previous state. In position tracking systems where the velocity is a state variable, a simple process model would assume no acceleration and predict that the new position would be a result of the current velocity through the time-step between iterations. The process model also includes a process noise matrix, which describes how certain we are that the system will act that way (B_k). On the update step, the goal is to estimate the residual ($\widetilde{y}$), which is the difference between the actual measurements (z_k) and the "measurement prediction" from the measurement model (H_k)

and the current estimate ($\hat{x}$), also with associated noise (R_k). The final state estimate is then the average weighted by the Kalman gain (K_k), which can also be computed from the covariance of the system at that time. The equations (Prediction, Measurement Update and Final Estimate) which describe the KF are presented below (2.7, 2.8 and 2.9):

Prediction:

$$\begin{aligned} StateEstimate -> \quad \hat{x}_{k|k-1} &= F_k \hat{x}_{k-1|k-1} + B_k u_k \\ Covariance -> \quad P_{k|k-1} &= F_k P_{k-1|k-1} {F_k}^T + Q_k \end{aligned} \tag{2.7}$$

Update:

$$\begin{aligned} Innovation -> \quad \widetilde{y}_k &= z_k - H_k \hat{x}_{k|k-1} \\ Covariance -> \quad S_k &= H_k P_{k|k-1} {H_k}^T + R_k \end{aligned} \tag{2.8}$$

Kalman Gain and new estimates:

$$\begin{aligned} Optimal\ Kalman\ gain -> \quad K_k &= P_{k|k-1} {H_k}^T {S_k}^-1 \\ State\ Estimate -> \quad \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \widetilde{y}_k \\ Estimate\ Covariance -> \quad P_{k|k} &= (I - K_k H_k) P_{k|k-1} \end{aligned} \tag{2.9}$$

The power of the Kalman Filter comes from the combination of a process model and smart prediction and the measurements, in an environment where all noise is modeled and the estimates have an associated covariance which is propagated throughout all iterations and represents the confidence the system has in the current estimates. However, it can only be applied to linear systems and will only work ideally if the uncertainty can be modeled as a Gaussian distribution. There are extensions of the Kalman Filter that can be applied to non-linear problems, such as the Extended and Unscented Kalman Filter. Alternatively, the Divided Difference Filter ([14]) is better suited for highly non-linear models and uses multidimensional functions to approximate the non-linear transforms, based on a multivariate extension of Stirling's Interpolation Formula [15]. Its computational cost is similar to the Kalman Filter. For different distribution models of the measurement or process noise (i.e. non-Gaussian), it is possible to apply particle filters (which will be described in further detail in Chapter 5), at the cost of greatly increased computational cost, since they model all distributions through sampling. Another advantage of the Kalman Filter is the simplicity of explicit drift modeling for the design of second order filters. Since sensor drift can be used as one of the state variables, the errors can be estimated and updated explicitly, along with the propagation of the respective covariance matrices. Despite the accuracy advantages, modeling additional state variables comes at the cost of added computational cost, which is already superior to that of the alternatives listed below.

2.2.2 Complementary Filter

A simpler approach to the sensor fusion problem comes in the shape of the complementary filter, which attempts to fuse the long-term stability of the accelerometer with the short-term precision of the gyroscope. It is designed to combine as a high-pass filtered gyroscope data and low-pass filtered accelerometer data as described in Equation 2.10, where α is a smoothing factor (between 0 and 1, usually equal or greater than 0.95) that describes how much we want to trust the gyroscope. Larger values of α lead to a smaller impact of gyro integration errors, however, the effect of noise in the acceleration signal (namely from movement) is magnified. The reverse is also true, with smaller values magnifying gyro drift.

$$\theta_t = \alpha * (\theta_{t-1} + \omega_{gyr} * dt) + (1 - \alpha) * (\theta_{acc}) \tag{2.10}$$

Alternatively, and by implementing a PID controller together with the filter, it is possible to design a second order complementary filter. The update function would then follow Equations 2.11, 2.12 and 2.13, with tunable parameters k_b *and* k_i describing the proportional and integral gain of the controller:

Obtaining estimated direction of gravity from current attitude

$$\hat{v} = \begin{bmatrix} 2(q_1 q_3 + q_0 q_2) \\ 2(q_2 q_3 + q_0 q_1) \\ {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2 \end{bmatrix} \tag{2.11}$$

Error Calculation (between current attitude $\hat{v}$ and normalized accelerometer reading $\bar{v}$)

$$e = \hat{v} \times \bar{v} \tag{2.12}$$

Corrected Rate of Turn

$$\hat{\Omega} = \Omega_{sensor} + k_p e + k_i \int e \, dt \tag{2.13}$$

The corrected rate of turn ($\hat{\Omega}$)is then used as the quaternion derivative and integrated, to obtain the current attitude. It is also important to note that after this calculation, the quaternion needs to be normalized, so that numerical errors do not affect the precision of the calculation and that the quaternion can in fact represent a real rotation.

Complementary filters are much simpler than Kalman Filters and have a smaller computational cost while having competitive accuracy and precision, making them very attractive. The main issue with the Complementary Filter algorithm is that its simple first order form does not produce estimates with sufficient degree of precision and robustness, especially for the estimation of foot attitude during human gait, where the acceleration signal can be quite corrupted. On the other hand, the PID controller can have trouble converging especially due to the nature of the accelerometer signal. Additionally, it does not explicitly model drift.

2.2.3 Gradient Descent Filter

Aligning the direction of gravity as estimated by the accelerometer with the direction that is being estimated by the current attitude quaternion results in infinitely many solutions, as any quaternion that is parallel to the z-axis will have the same direction, which means it can have any possible rotation around it. Madgwick et al. [11, 16] proposed solving this as a minimization algorithm, so that the orientation of the sensor (quaternion $\hat{q}$) is that which minimizes the difference between the direction of its estimate for the z-axis and the estimate of gravity from the accelerometer $\bar{a}$, as in Equation 2.14.

$$f(\hat{q},\bar{a}) = \hat{q}^* \otimes [0\ 0\ 0\ 1] \otimes \hat{q} - \bar{a} = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - \bar{a}_x \\ 2(q_1q_2 + q_3q_4) - \bar{a}_y \\ 2(\frac{1}{2} - {q_2}^2 - {q_3}^2) - \bar{a}_z \end{bmatrix} \tag{2.14}$$

The authors proposed using the Gradient Descent method to minimize function f and derived the Jacobian matrix analytically (quite simple in the case of alignment with a simple direction such as [0 0 1]), enabling the fast computation of the gradient and efficient solving of the minimization problem (2.15).

$$J(\hat{q}) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \tag{2.15}$$

The calculated error vector will then be: (eq. 2.16)

$$\hat{e} = f(\hat{q},\bar{a})^T J(\hat{q}) \tag{2.16}$$

Without a reference for the alignment of the y-axis, the solution may have some numerical noise depending on the result of the gradient descent algorithm. However, this fluctuation as well as the introduction of acceleration noise (erroneous gravity direction) is controlled by the β factor, which determines how much the calculated error will influence the final estimate by controlling how much the gyroscope-based estimation tilts in the direction of the error derived from the direction of gravity as estimated from the acceleration data. The minimum β value is limited by the gyroscope measurement error (with zero mean) and is expressed as the magnitude of the quaternion derivative, as described in Equation 2.17, where $\hat{q}$ is any unit quaternion and $\widetilde{\omega}$ is the estimated gyro error.

$$\beta = \left\| \frac{1}{2}\hat{q} \otimes [0\ \widetilde{\omega}\ \widetilde{\omega}\ \widetilde{\omega}] \right\| = \sqrt{\frac{3}{4}}\widetilde{\omega} \tag{2.17}$$

Furthermore, and unlike typical applications of the Gradient Descent algorithm, because the rate of convergence of a single iteration is greater than the physical rate of the change of orientation, only a single iteration is necessary, resulting in a very efficient algorithm. The attitude estimate is then obtained by integration (eq. 2.18) of the derivative of each quaternion component.

This derivative is obtained from the rate of turn measured by the gyroscope (eq. 2.6), $\dot{q}$ and the direction of the error given by the normalized error vector, $\hat{e}$.

$$\begin{bmatrix} q_w(t) \\ q_x(t) \\ q_y(t) \\ q_z(t) \end{bmatrix} = \begin{bmatrix} q_w(t-1) \\ q_x(t-1) \\ q_y(t-1) \\ q_z(t-1) \end{bmatrix} + \begin{bmatrix} \dot{q}_w(t-1) - \beta\hat{e}_w \\ \dot{q}_x(t-1) - \beta\hat{e}_x \\ \dot{q}_y(t-1) - \beta\hat{e}_y \\ \dot{q}_z(t-1) - \beta\hat{e}_z \end{bmatrix} \times dt \tag{2.18}$$

It is true that without a reference source for the y-axis (magnetometer), this algorithm will introduce fluctuations in the yaw angle (around the z-axis). Nevertheless, these fluctuations, when controlled by a small β value and adequate filtering and processing, can be manageable. With a 6-DOF IMU, drift cannot be modeled, as the errors in each specific axis cannot be numerically calculated (as the system cannot arrive at a completely defined solution without a second direction reference, namely the magnetometer reference to the North). The main advantage is that this filter provides greater accuracy on the alignment of the z-axis estimate with gravity, which is very desirable from a point of view of accurate distance quantification relative to the xy plane, as explored in Chapter 3. As such, this filter will be the base of the attitude estimation filter to be applied on this system.

2.3 Zero Angular Rate Update (ZARU)

Compensating for gyroscope bias is the main challenge in attitude estimation. After integration of the gyro values, uncompensated bias will cause the attitude estimation to drift. The goal of the IMU fusion and map matching modules will be to compensate for the drift in the attitude estimation. Detecting and compensating gyro drift, however, requires information regarding average error in the measurements (as will be estimated by the MM module) or the expected rate of turn at a given moment - this is where ZARU comes into play.

Human gait is made up of several stages in cyclic succession: heel strike -> stance -> toe-off -> swing -> heel strike (...). When the foot is in the stance phase, there is no acceleration due to movement, no velocity and no rotation, even if the accelerometer signal does not point exactly in the direction of gravity and the gyroscope does not return zero rate of turn in all axis (precision and bias issues). If we can detect the stance phase of gait, we can retrieve a very valuable piece of information: the angular rate of turn is zero for all the samples that make up that phase (this will be explored further in Chapter 3 and in Section 2.4.3).

Once the samples corresponding to the ZARU are detected, they are set to zero and bias can be estimated by taking the average value of the gyroscope measurements that were detected as Zero Angular Rate.

2.4 Implementation of the GD-ZARU Filter

This section describes the implementation of the attitude filter which is based on the Gradient Descent Algorithm aided with ZARU.

2.4.1 Pre-Filtering

Before feeding the data into the attitude estimation algorithm and in order to minimize the amount of noise in the signal, the data is preprocessed. Since studies in human gait inertial signal frequency content have concluded that 98% of the signal content is within the frequencies up to 10Hz (and 99% at 15Hz) [17] and band-passing gyroscope signals improves the Signal to Noise Ratio of the signal [18], it is necessary to filter out the unnecessary and noisy frequencies. Accelerometer data should be low-pass filtered at a cut-off frequency of at least 10Hz. Angular rate data will be only low-pass filtered with a cut-off frequency between 10 and 30Hz. The absence of a high-pass filter is due to the fact that filtering the signal at 0.1Hz would require a large number of samples to be used in the filtering window (increasing computational cost). Since the algorithm applies ZARU and estimates bias drift and as such there should not be any DC component in the signal, the data will only be low-pass filtered.

There are several types of filters that can be used to achieve the real-time (or near real-time) filtering that is necessary. Examples include the moving average filter, the median filter, Kalman filters, Butterworth filters, Discrete Wavelet Package Shrinkage ([19]) and Window filters (namely the one-parameter Kaiser-Bessel). In this work, all filters are Kaiser-Bessel Windows [20], since they are real-time (only use past samples to arrive at the current filtered estimate), easily tunable, efficient and linear phase. The filter coefficients are given by Equation 2.19 (where I_0 is the zeroth order Modified Bessel Function, N is the window size and the α parameter controls the relationship between main-lobe width and side-lobe height in the frequency domain) and are precomputed. Since N will be odd and the coefficients will be symmetric, this (type I linear-phase FIR filter) will have no phase delay and will have an integer group delay of 4 samples [21], making it suitable for this application.

$$\omega[n] = \frac{I_0\left(\pi\alpha\sqrt{1-(\frac{2n}{N-1}-1)^2}\right)}{I_0(\pi\alpha)}, \quad 0 < n < N-1 \tag{2.19}$$

2.4.2 Initialization

The initial estimation of attitude is done solely with the accelerometer data, which provides the pitch and roll angles with a indeterminate yaw angle. Equations 2.20, 2.21 and 2.22 describe the calculation of the initial attitude from the normalized acceleration signal. Pitch, yaw and roll are described by ϕ, θ and ψ respectively.

Since the yaw measure is arbitrary (set to zero), the estimated angle will be rotated to fit the initial direction that the system assumes is available (input by the user, retrieved by GPS or

through the direction of the building entrance), as there is no absolute source of heading available. All subsequent heading changes will be corrected with the initial offset.

$$\psi(roll) = arctan(\frac{A_y}{A_z}) \tag{2.20}$$

$$\phi(pitch) = arctan(\frac{-A_x}{A_y sin\psi + A_z cos\psi}) \tag{2.21}$$

$$\hat{q}_0 = \begin{bmatrix} cos\theta cos\psi cos\phi + sin\theta sin\psi sin\phi \\ cos\theta sin\psi cos\phi - sin\theta cos\psi sin\phi \\ cos\theta cos\psi sin\phi + sin\theta sin\psi cos\phi \\ sin\theta cos\psi cos\phi - cos\theta sin\psi sin\phi \end{bmatrix} \tag{2.22}$$

2.4.3 ZARU

Detection of Zero Velocity and Zero Angular Rate (which are connected) are crucial in order to bound the error associated with integration of the acceleration and angular rate signals [22]. There are several algorithms to detect Zero Velocity periods [23], which will be analyzed in further detail in Chapter 3. The features which were used to detect these periods were acceleration magnitude, acceleration variance and angular rate of turn magnitude. Detection of Zero Angular Rate follows the same general idea (as Zero Velocity periods also mean Zero Angular rate), however, in this case they will not be used for step detection as is the case with the Zero Velocity Detection in the Step Quantification Module. Instead, the goal is to detect and correct gyro bias and its drift.

Since ZARU will estimate and correct bias continuously, erroneous detections may distort the signal, which introduces error that is not as easily corrected as distance estimation errors, for example, and propagates through all future estimates of the system. As such, the criteria for ZARU detection will be much stricter than ZUPT. Samples which are classified as Zero Angular Rate will be zeroed and the bias (difference to zero) is stored. As samples arrive at the ZARU module, the average bias is removed.

2.4.4 Attitude Update

As described in Section 2.2.3, attitude estimation is done in two steps: error calculation and quaternion integration. Before calculating the error (difference between the estimate of the direction of the z-axis from current quaternion and current acceleration vector), it is useful to apply a second low-pass filter on the accelerometer data. Even though frequencies up to 20Hz can be useful for the reconstruction of the displacement of the foot, filtering out frequencies up to the 3-8Hz range can be beneficial for the filtering of the movement-induced noise and estimation of the gravity direction.

Once there is an estimate of the error, which will be the direction of the gradient of the minimization problem, it is applied to each of the coordinate derivatives (see eq. 2.6 and 2.18) as estimated from the gyroscope measurements, according to the step value β.

2.5 Parameter Tuning

As described during the previous Sections, there are several tunable parameters that will greatly influence the performance of the filter. An overview of the entire module is presented in Figure 2.3, where all tunable parameters are presented in parenthesis.

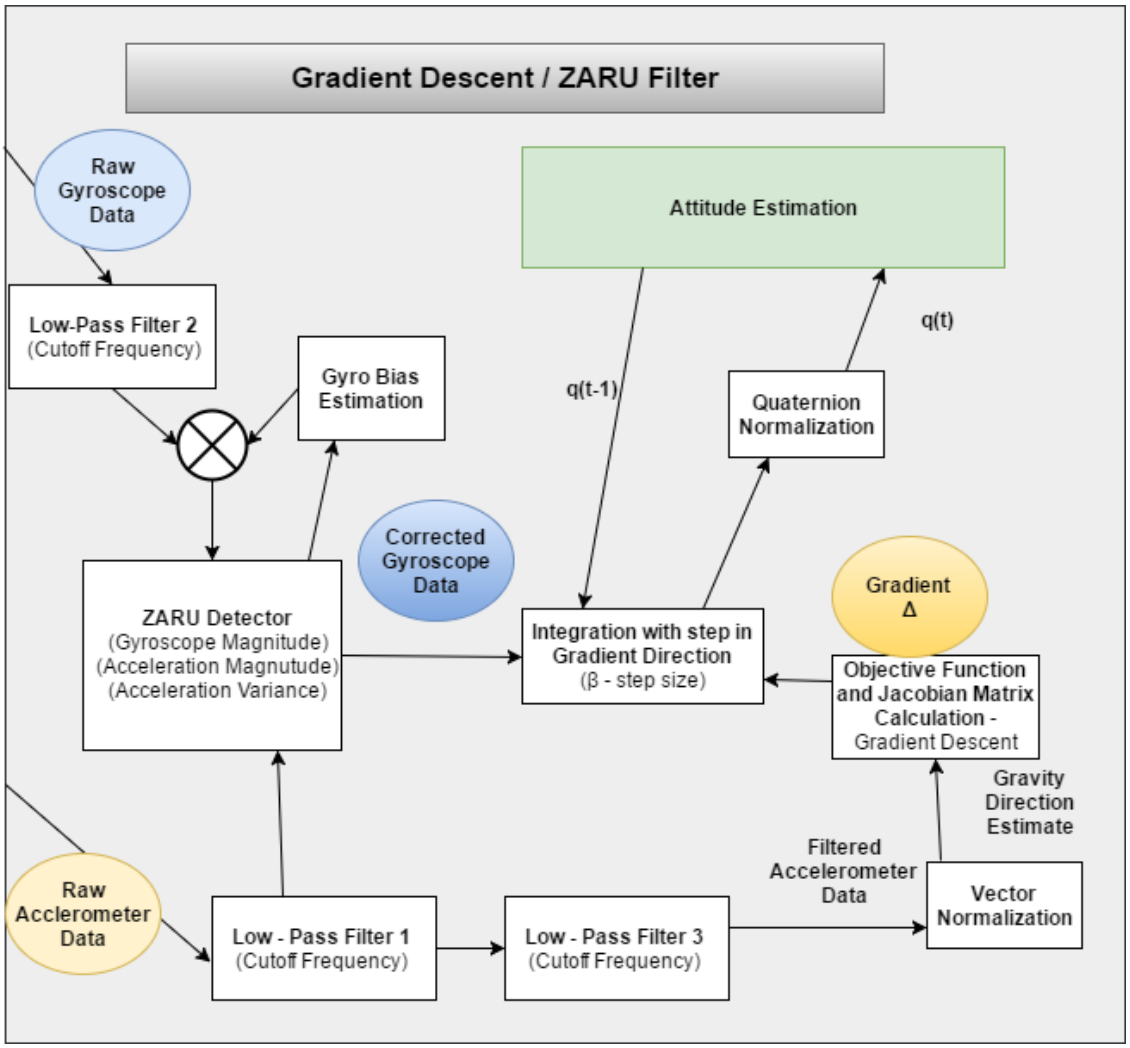


Figure 2.3: Attitude Estimation Module and Tunable Parameters - Raw gyroscope data is corrected and raw accelerometer data is used to obtain the gradient of the correction that minimizes the difference between the attitude estimate of the z-axis and the gravity measurement. These two estimates are used for attitude estimation

.

A list of the all values for the parameters is presented below, with discussion regarding their robustness across different gait samples, impact of varying the parameter value on the overall attitude estimation and general design process. It is important to note that the current system has

a sampling rate of 100Hz per sensor per foot, so each second the system will receive (100 [frequency] $\times$ 2 [accelerometer + gyroscope] $\times$ 2 [IMU on each foot] $\times$ 3 [3-axis] = 1200 samples), which need to be processed at or near real-time alongside the position estimation correction algorithms (step detection, quantification, fusion of information from both feet and map matching), so computational costs need to be reasonable for all processing.

- **Accelerometer Low-Pass Filter 1 -** As mentioned above, 98% of the gait inertial signal content is present in frequencies up to 10Hz. At a sampling frequency of 100Hz, the system will be able to use 10 samples per period. Since the low-pass filter is not ideal, frequencies slightly above 10Hz will still appear (attenuated), allowing the signal to be used for displacement reconstruction with adequate accuracy. In this application, a filter with a wide main lobe and strong attenuation of the side lobes is desirable. The side lobe attenuation of the filter is 100dB, which translates to a main lobe width of 26Hz. The filter window size is 21 samples, a trade-off between precision and computational complexity.

- **Gyroscope Low-Pass Filter -** In [18], the authors test several cut-off frequencies for their band-pass filter, with different optimal results for different subjects and gait speed. This makes filter design very complicated, as either the cut-off frequency is dynamically defined from gait speed or a compromise must be made. Applying no pre-filter to the gyroscope signal is an option, but solving noise problems before the attitude estimator is highly beneficial. Additionally, filtering acceleration without filtering rate of turn can lead to poorer performance of the attitude estimation algorithm. As such, and to avoid losing important rotation information, the cut-off frequency was set to 20Hz, again with 100dB side-lobe attenuation.

- **ZARU Detector -** As explored in Section 2.4.3 all thresholds are stricter than the ones used on the ZUPT/Step Detector described in Chapter 3, while using the same features and classifier: Acceleration magnitude lower threshold is set at 9.5 and its higher threshold at 10.5; Acceleration variance threshold is set at 0.3 with a window size of 31 samples; Angular rate threshold is set at 0.3, which is much lower than the one on the step detector, as there is drift modeling (which will ensure that the returned angular rate returns to zero and does not drift) and setting samples to zero erroneously causes wrong estimates of drift (which create increasing errors).

- **Accelerometer Low-Pass Filter 2 -** In order to reduce the movement-induced noise in the gravity estimation, the acceleration signal is re-filtered with a cut-off frequency of 5Hz and 30dB side-lobe attenuation, so that the direction of the gravity vector is more stable even under high linear acceleration caused by movement.

- **Step size β -** The step size is limited by the gyroscope measurement error. Empirically, the attitude estimation was best for smaller β values. The zero-mean gyroscope measurement error was set to 1 deg/s (the error with non-zero mean is of a larger magnitude).

2.6 Review of the Attitude Estimation Module

The Attitude Estimation Module, which also acts as a pre-filter, is lightweight, real-time and stable. However, it does have a number of tunable parameters that greatly influence its performance and that are, at the moment, static. Firstly, it may be beneficial to relate the cut-off frequency of the filters with features such as step frequency, gait speed or signal magnitude, as interesting frequency bands may change for different subjects and gait speed [18].

Examples of the quaternion estimates through time, separated by quaternion component, are presented on Figures 2.5 and 2.7. Figures 2.4 and 2.6 show the estimate of the route travelled, which were routes **8-route-A** and **zigzag** respectively, as shown in Chapter 7. There is a slight drift on the heading estimates for the right foot (which can be caused by incorrect calibration). Note that there is no information for initialization of yaw and as such it is arbitrary. In these examples, the yaw was forced to be 0°, and as the x and y axis of the IMUs are pointing in the same direction in the first experiment but in different directions on the second (for ease of placing under the shoelaces, the opening of the case for the charger is facing the outside of the foot on both shoes) the initial estimates for the latter have opposite signals.

During the design of the algorithm, there was an attempt to dynamically set the β value according to the magnitude of the acceleration vector, as a function of the confidence of the estimation of the direction of gravity: if the magnitude of the measurement vector is considerably greater than $9.81 ms^{-2}$ then there is movement noise in the signal, the estimate is less precise and less correction in the direction of the estimated error should be applied. As this did not improve the final result, it was discarded. Nonetheless, if movement induced noise on the accelerometer signal can be detected in a more sophisticated manner, it may be beneficial to dynamically change the weight given to the acceleration-based attitude estimate. It is important to note that without dynamic application of accelerometer-based corrections, better results relative to the pitch and roll angles (alignment of the z-axis with gravity) will always result in the introduction of noise in the yaw angle (heading), as can be seen in Figure 2.6, where the left foot estimates are much more accurate in terms of yaw while the right foot estimates are more accurate in terms of pitch and roll.

Finally, classification of samples as zero angular rate could be improved, as explained in the following Chapter, and that enables better estimation of the gyro bias drift even before further information or processing is applied.

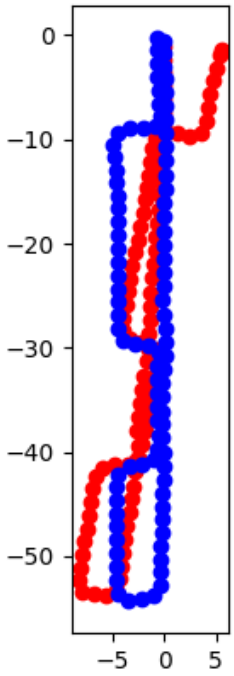
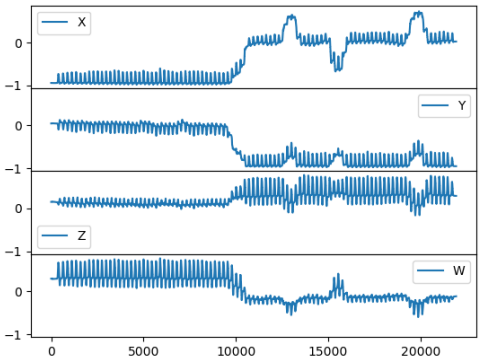
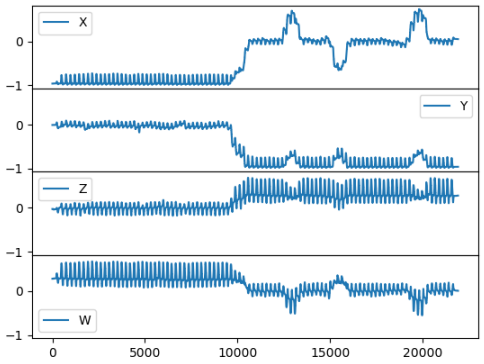


Figure 2.4: Route **8-route-A** - Estimation of positions (each IMU independently) corresponding to the attitude estimation of Figure 2.5. Red and blue dots represent positions for the left and right foot respectively. Axes units in meters

.

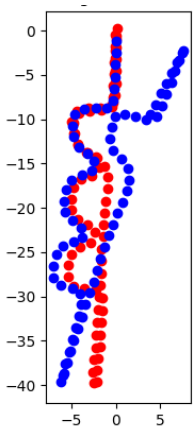


(a) Attitude estimates for left foot

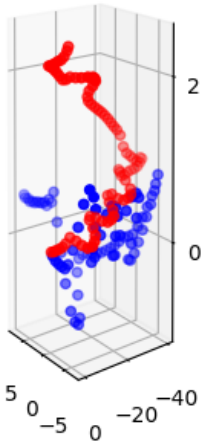


(b) Attitude estimates for right foot

Figure 2.5: Example of quaternion attitiude estimates through route **8-route-A** (see Chapter 7 and Figure 2.4). X-axis describes sample number (at 100Hz sampling frequency) unit is equivalent to the hundredth of a seconf, Y-axis describes value of the quaternion component

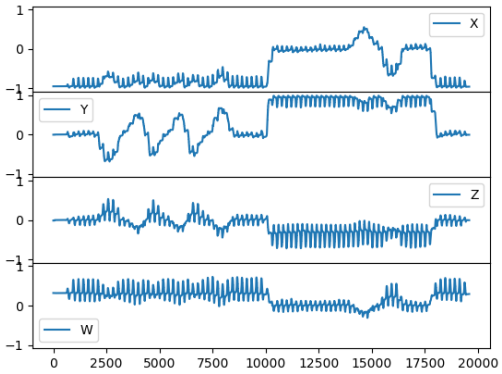


(a) PDR on the x-y plane

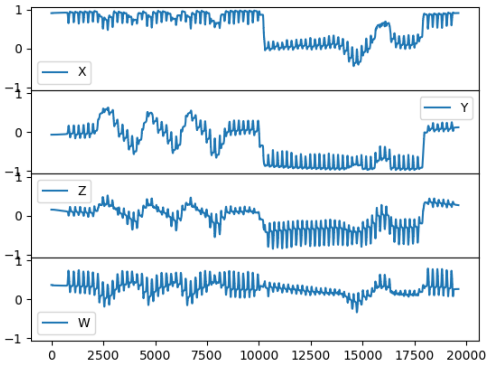


(b) 3D tracking

Figure 2.6: Errors on Pitch and Roll angles vs Error on Yaw angle - Red dots represent positions of the left foot and blue dots represent positions of the right foot. Initial and final height should be the same. Travelled route is **zigzag**, see Chapter 7. All axes in meters.



(a) Attitude estimates for left foot



(b) Attitude estimates for right foot

Figure 2.7: Example of quaternion attitiude estimates through route **zigzag** (see Chapter 7 and Figure 2.6). X-axis describes sample number (at 100Hz sampling frequency). Y-axis describes the value of each quaternion component. Note the symmetric values due to different orientation of the sensor mounted sensor relative to the foot.

Chapter 3

Step Detection and Quantification

For continuous location to be possible, the position of the user must be known, not only with adequate precision but also with adequate time resolution (less than a second, as indoors a couple of meters can be a significant distance). This means that either absolute location can be retrieved at all times (not possible with the desired approach) or the displacement of the user can be tracked in real-time (PDR).

Tracking the movement of users is intricately linked to detecting and quantifying their steps (normal means of human movement) and requires detection of steps and the estimation of the step length and heading. While step detection is not mandatory, as displacement can be continuously tracked through the double integral of the acceleration signal, coupled with attitude estimation and aided by ZUPT and ZARU, step detection and segmentation of the corresponding samples can be helpful for two reasons: forwards and backwards filtering of the samples for smarter ZUPT; ability to fuse information from both feet using a step as a unit of displacement (explored in full detail in Chapter 4), which is much simpler than continuous information fusion.

A step can be defined as a translation of the foot that is greater than a certain threshold, as a movement with a certain cyclic inertial pattern (in terms of acceleration and angular turn) or more simply as the period containing movement between two periods where the sensor is detected to be still. This is an interesting approach since when one foot moves, the other is still and bearing the weight of the user, assuming normal gait. This is a key property, and will be exploited in the ZUPT algorithm. With this approach, lifting one foot off the ground and putting it down on the exact same spot will be counted as a step by the system. Despite this, if attitude estimation and the ZUPT-aided displacement estimation is working properly, the calculated displacement will be zero on all three axes.

In this Chapter, there is an analysis of the importance and the implementation of ZUPT, detection of Zero Velocity periods (which will be equal to step detection) and step quantification (both in length and heading).

3.1 Zero Velocity Update (ZUPT)

ZUPT is ubiquitous in recent foot-IMU based systems, as it bounds the error associated with the integration of signal from imperfect accelerometers and attitude estimates that cannot completely compensate for gyroscope bias drift [1]. Integration of signals with noise or bias will lead to drift that makes estimation of displacement from double integration of the acceleration signal unsuitable for tracking (cubic error growth with time, [24]). Figure 3.1 shows the velocity signal resulting from the integration of the acceleration signal (after the frequency filtering and attitude estimation as described in the previous Chapter) without applying ZUPT.

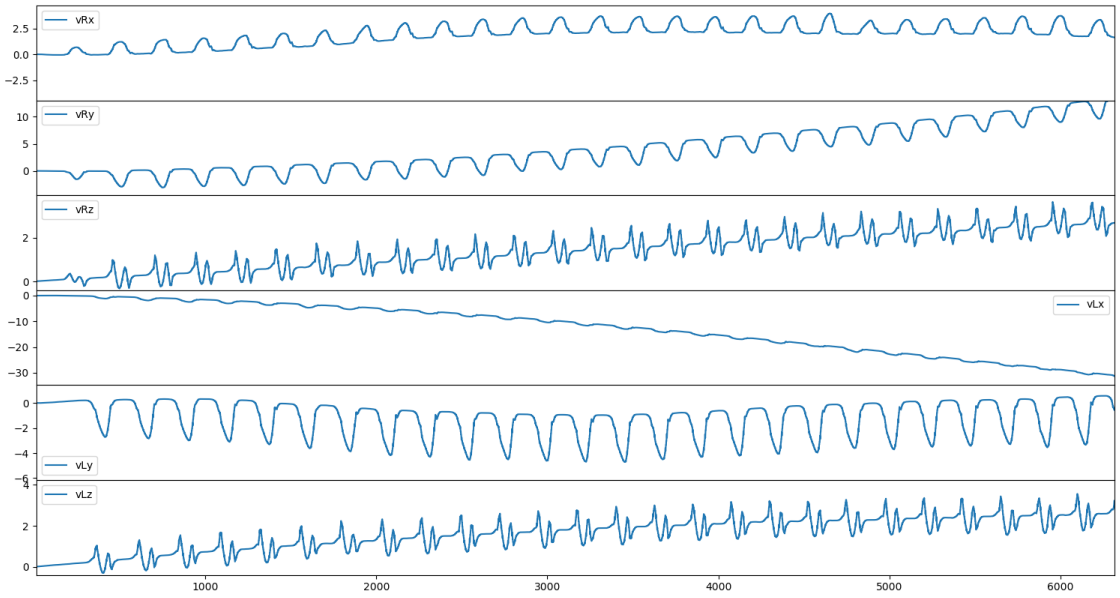


Figure 3.1: Velocity Estimates from Integration without ZUPT. Velocity in ms^{-2}, x axes represents samples (at 100Hz sampling), unit is equivalent to the hundreth of a second. From the top: velocity estimates for the right foot (x, y an z axes); velocity estimates for the left foot (x, y and z axis).

Human gait is a cyclic movement composed of several stages, which can be divided in two big groups: stance and swing. During each foot stance (namely on the midstance, if using the classification suggested by Perry and Burnfield in [25], and displayed in Figure 3.2) phase, it is firmly planted on the ground, bearing the weight of the subject. At that point, it is stopped, with no movement-induced acceleration, no velocity and zero turn rate. As such, ZUPT and ZARU are closely linked and can be detected by the same parameters. As mentioned in Chapter 2, the principle is the same: the detection of the stance phase. By setting the estimated velocity at zero for samples that are detected to belong to the stance period of gait and correcting the signal based on the estimate of error (difference between zero and the actual value), error is bounded to linear growth with number of steps, making tracking methods a feasible approach for the IL problem. As one can see in Figure 3.1, the velocity estimates fail to return to zero without ZUPT, and since every sample depends on the previous state, drift accumulates, leading to wildly inaccurate estimates.

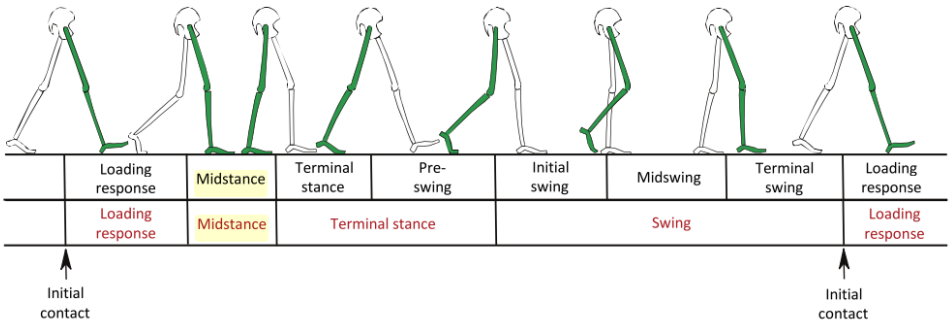


Figure 3.2: Gait Stages as divided by Perry and Burnfield in [25] and presented in [26]

After the samples corresponding to a step are segmented, there are several ways to apply the ZUPT [27]. The simplest alternative would be to average the velocity error (difference between values which are supposed to be zero and the actual value). A second alternative, which was the one that was implemented, calculates the necessary correction through the Residual Velocity. Residual Velocity refers to the final value of velocity after the reconstruction of the signal in the segmented window. Since samples are segmented according to the detection of the Zero Velocity (stance) periods, the final velocity should be zero. The correction is done by subtracting half of the residual velocity to all the non-zero samples, so that both ends of the velocity curve are as close to zero as possible. Figure 3.3 shows the result of Updating and compensating errors when reconstructing the velocity signal, for the same signal as in Figure 3.1. There is no propagation of error on the signal, even if there are small errors in each small estimate.

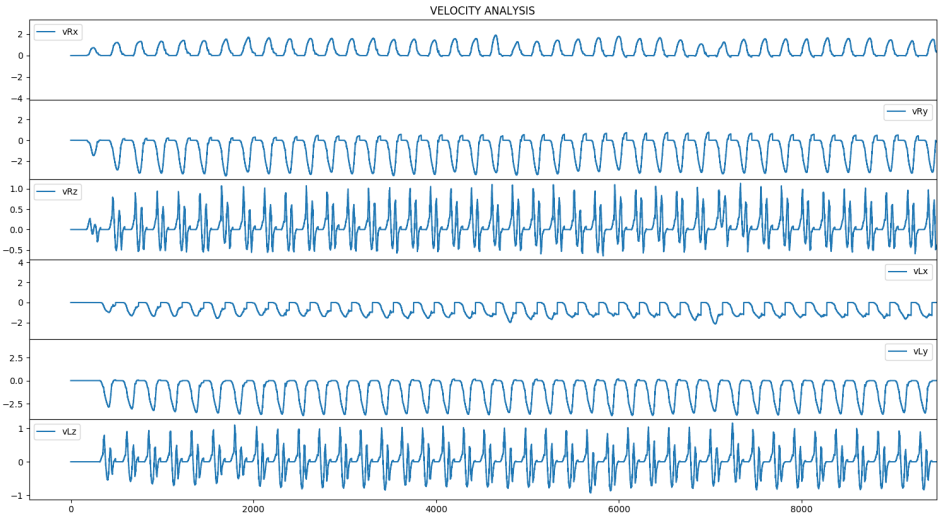


Figure 3.3: Velocity Estimates from Integration with ZUPT. Velocity in ms^{-2}, x axes represents samples (at 100Hz sampling)

More sophisticated reconstruction of the signal could be done, through several fixed-interval smoothing formulas [28] (among those the Rauch-Tung-Striebel formula [29]). In case of Kalman

Filter driven algorithms, the smoothing can work in 3-passes [30]: The first pass occurs when naturally processing the samples the first time, calculating error covariance matrices; Once the end of the step (detection of a posterior ZUPT interval) the algorithm performs a backwards pass to correct all covariance matrices; Lastly, a forwards pass is done to redo all estimates through using new error estimates. These formulas are geared towards state-space estimation methods, namely the Kalman Filter, however, similar approaches could eventually be extended to segmented signals.

3.2 Techniques for Detection of Zero Velocity - Steps

Since the detection of steps is not the ultimate goal of the system, the Step Detection Module is responsible, above all, for timely ZUPT and bounding of errors in displacement reconstruction. If there are persistent erroneous detections because of a different walking pattern and that causes erroneous ZUPT that either greatly increase or decrease the true displacement distance, that is a problem. However, if due to small movements, dragging of the feet at heel strike or during a sharp movement (such as avoiding an obstacle) an additional small step is counted which increases the travelled distance by only a few centimetres, that will not be a problem, as the IMU Fusion and Map Matching Modules will be responsible for the correction of said small errors. The main goal is the correct classification of ZUPT periods.

Detection of Zero Velocity with great accuracy is a challenge, and the key to displacement reconstruction. As briefly described in the previous Chapter, there are several methods available for Zero Velocity detection. An overview of these methods is provided below.

3.2.1 State Machines

The periodicity of human gait can be explored for step detection, namely through Finite State Machines (FSM). By defining the various states that make up a step and the transition between them, step detection can take advantage of the temporal relation between states (for example, the Midstance follows the Loading Response as defined in [25] and shown on Figure 3.2) to increase robustness of the segmentation. Accurate modeling of the states and their transitions (which state can transition to which and how) enables detecting all stages of the gait in succession and a robust step segmentation, as it does not depend on a single test. Instead, FSM look for various patterns (that can be more flexible than a static peak searching) and leverages their sequence to reach a more dynamic segmentation decision. In [26], the authors segment gait in the 5 stages represented in the bottom part of Figure 3.2 in red (including "Initial Contact"). Additional states and transitions are included in order to model the signal behavior for backwards gait and stair climbing. These transitions are represented in Figure 3.4.

One key of the detection of state transitions is the use of the gyroscope measurements for angular turn caused by foot dorsiflexion and plantar flexion (i.e. around an axis perpendicular to the foot length, going out laterally from the ankle), which are the ones where the transition between states can be seen more easily. This requires the IMU to be fixed at a position in which

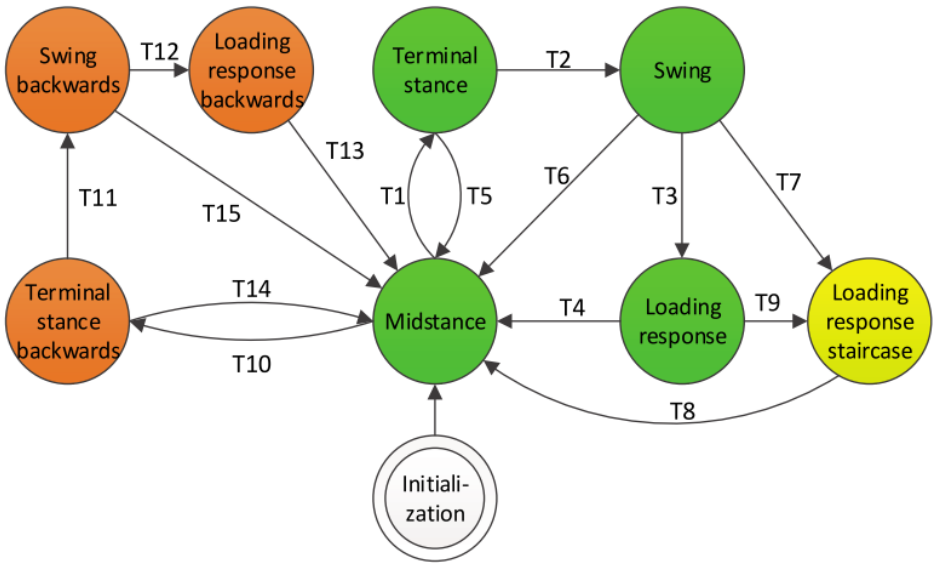


Figure 3.4: Finite State Machine algorithm overview, in [26]. Green states represent regular forward motion, orange states represent backwards gait and the yellow state represents stair climbing.

one of the 3 axes is parallel with the ankle main axis. The authors enumerate several features that are used for transition detection:

- Variance of the accelerometer and gyroscope measurements
- Angular rate of turn in the ankle main axis and its derivative
- Acceleration and angular rate of turn magnitude
- Specific force - Acceleration Magnitude after compensating for gravity

Alternatively, it is possible to model state transitions as a Hidden Markov Model, which can deal with states that are not detected or more complicated patterns of human movement such as very fast walking or running by estimating which states belong to which step and how many steps were taken by using an HMM smoother on the detected states. This approach is used in [31], where the states correspond to segmentation of the gyroscope axis corresponding to the main ankle axis according to the magnitude of its signal.

These approaches provide interesting results and are very robust, but require sophisticated empirical modeling and the isolation of the axis of the gyroscope (since estimates of turn from a rotated gyroscope axis will not be precise enough), in order to reflect the movement of the ankle.

3.2.2 Gait Cycle Detection

The main challenge with Zero Velocity interval detection is on its borders, specifically on its start. This is because the beginning of the stance phase is through the "Initial Contact" where large acceleration spikes occur. In addition, relative motion between the foot and the IMU introduce

additional noise (that also propagates to the attitude estimate) which makes for an added challenge. This problem is tackled in [32] by using Kinesiology knowledge as an added source of information, namely the proportion of the step duration that makes up stationary and moving stages. The idea is to compare the timespan of gait and the ZUPT interval as estimated by the angular rate of turn on the main ankle axis with the standard kinesiology proportions. This match is done through a Bayesian probabilistic model, where accelerometer data is also used.

This approach suggests placing the IMU on the toe in order to increase the signal strength of rotation on the axis parallel to the main ankle axis, which is the axis where rotation is most characteristic of all gait (rotation on other axes is more prevalent in changes of direction, for example). It provides great results in reducing the fuzziness around the boundaries of the Zero Velocity period, however, it also requires the alignment of one axis of the gyroscope with the main rotation axis and suggests an awkward fit on the shoe.

3.3 ZUPT Interval Detection - Implementation Details

A third class of algorithms focus on more general features of the inertial signals in the stance phase to detect steps, without requiring a specific orientation for the mounting of the IMU on the shoe. Since versatility is a desirable trait in the designed system (and tests are being done simply by strapping the IMU under the sholaces), algorithms that do not require the detection of highly differentiated states of the gait cycle (which are mainly detected through rotation on the main ankle axis as explored in the previous two Sections) are very interesting.

While PDR systems based on unconstrained inertial sensors (such as a smartphone) need to focus on the detection of the foot strike impacts (which propagate through the body, regardless of if the phone is in the pocket, hand or a bag) through dynamic tresholds, robust FSMs or even Machine Learning Classifiers, foot-mounted IMUs can explore the idea behind ZUPT: the foot on the ground will have no velocity, no acceleration (apart from gravity) and no rate of turn. The most common and simple algorithm for step detection is the moving variance estimator, which calculates the variance of the acceleration magnitude in windows around a certain point, since very low variance of the accelerometer signal is a characteristic of the stance phase. This simple threshold classifier can be combined with other signal features to obtain a more robust classification [23], especially if used not only as a step detector but also as a ZUPT interval detector.

Working on the algorithm proposed in [22] (and as briefly mentioned in Chapter 2), ZUPT detection will be done using three features:

- **Acceleration Magnitude** - Since the acceleration magnitude on the stance phase should only reflect the gravitational acceleration, the stance phase is characterized by a magnitude around $9.81 ms^{-2}$. This threshold could be made more accurate if the attitude estimate for the pitch and roll angles (z-axis alignment with gravity) was sufficiently accurate and gravity could be removed from the acceleration signal in the Earth frame. However, errors in attitude estimation would lead to errors in velocity reconstruction which would propagate, and a

robust z-axis alignment would come at the cost of a loss of precision in the yaw angle (heading) estimate.

- **Acceleration Variance** - This condition is usually the one that is chosen on systems with a single test, and it is the most accurate and robust [23], especially for step detection and general ZUPT interval detection. Nevertheless, it is benefficial to fuse this condition with other tests that can help locate the boundaries of the interval with added precision.

- **Angular Rate Energy (Magnitude)** - This is a key feature for the detection of the border of the ZUPT interval, as explored in the algorithms presented in the previous Sections. Even though the IMU placement is not geared towards isolating angular turn of a specific axis, the beginning and end of the stance phase are noticeable on the gyroscope magnitude and the fact that it is so sensitive to small rotations mean that during most, if not all, of the swing period, the magnitude of the angular rate of turn will be below a certain threshold.

The selected thresholds were the ones described in [22]: Acceleration Magnitude between 9 and 11ms^{-2}; Acceleration Variance lower than 0.5(m^2s^{-4}) on a 30 sample window; Angular Rate Energy lower than 1 rad/s. The three tests are then combined with a Boolean AND, where True (1) describes a ZUPT interval (foot stationary on the ground) and False (0) describes a movement interval (foot is moving). An example of the obtained signals is presented on Figure 3.5. Refer to Figure 3.7, below, for an illustration of the signals on which the thresholds are applied and the final segmentation result. The first boolean condition testing is set to always true before the first step so as not to detect an additional foot in contexts where the initialization of the IMU introduces error in the measurements and small extra steps would be detected at startup.

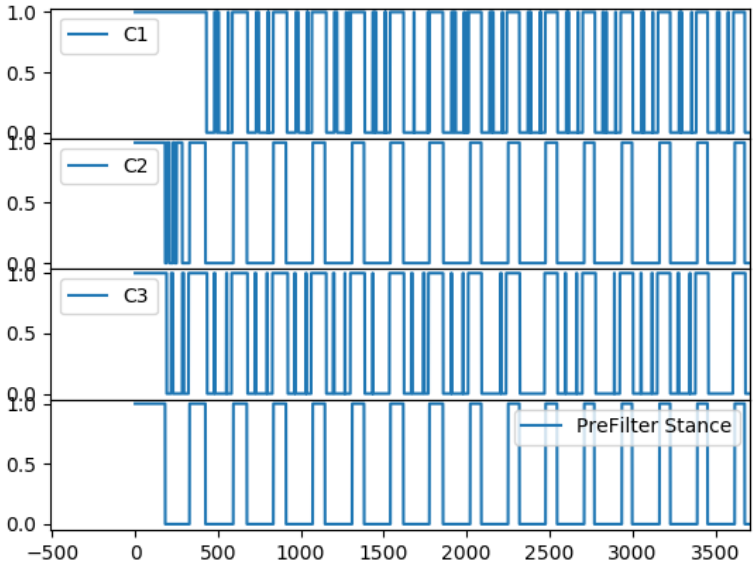


Figure 3.5: ZUPT Detection - 3 Boolean Conditions and Filtered Decision Signal. From the top: C1 - Acceleration Magnitude; C2 - Acceleration Variance; C3 - Angular Rate Magnitude. The Last Signal is the result of the Boolean AND on the 3 conditions. X axis - samples (100Hz sampling).

The resulting signal is then filtered with a skewed median filter of 15 samples, where a True result requires 13 True samples. This was done in order to increase robustness of the algorithm in occasions where the lack of specificity of the first boolean condition (as can be seen in Figure 3.5), so as not to detect partitioned ZUPT intervals or erroneous small updates. The resulting decision signal is shown in Figure 3.6 with no temporal processing. The steps are well intercalated, which reflects the need for synchronization only if the sampling frequency was unstable or the time difference was larger than the duration of the step. Order of step detection is important since the information from both feet is processed and fused as it arrives at the Fusion module (see Chapter 4) and two consecutive steps on the same foot without the processing of an existing step on the opposite foot lead to strong erroneous filtering and overall positioning error.

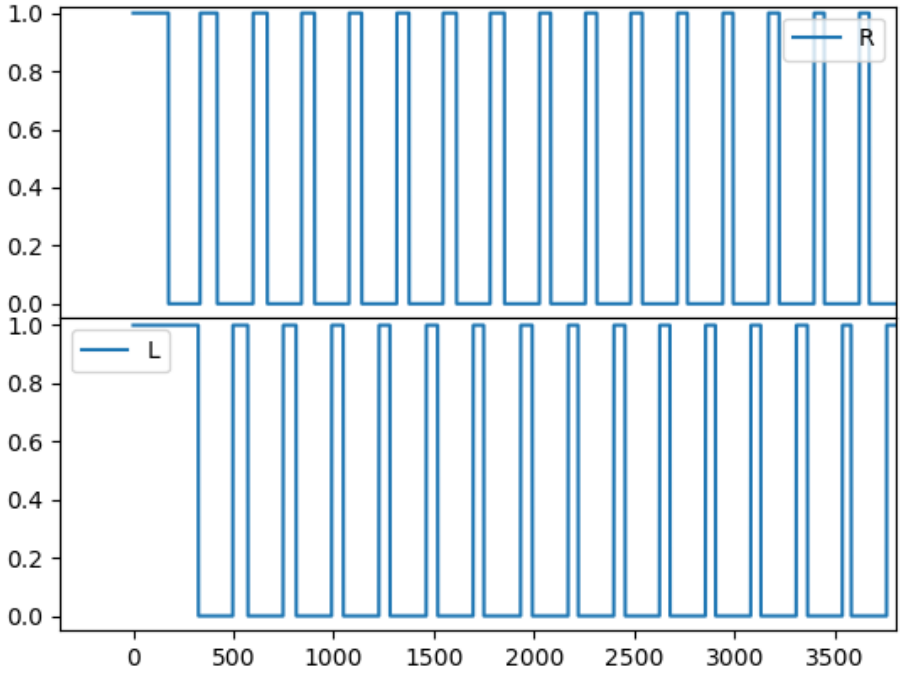


Figure 3.6: Filtered Decision Signal for both feet. Left foot on top and right foot on bottom.

Finally, the system is made more robust through two checks on the integrity of the step detection. If two consecutive steps of length greater than 0.5 meters are detected by the step tracker module to be on the same foot, there is a reprocessing of the data of the opposite step with more relaxed thresholds (namely regarding acceleration variance and gyroscope energy), in order to make sure no step was missed. Additionally, if during step quantification the estimated velocity is above a certain threshold (4 m/s), the corresponding samples are also reprocessed. An illustration of the correct segmentation of the original signal through the method described in this section is presented on Figure 3.7.

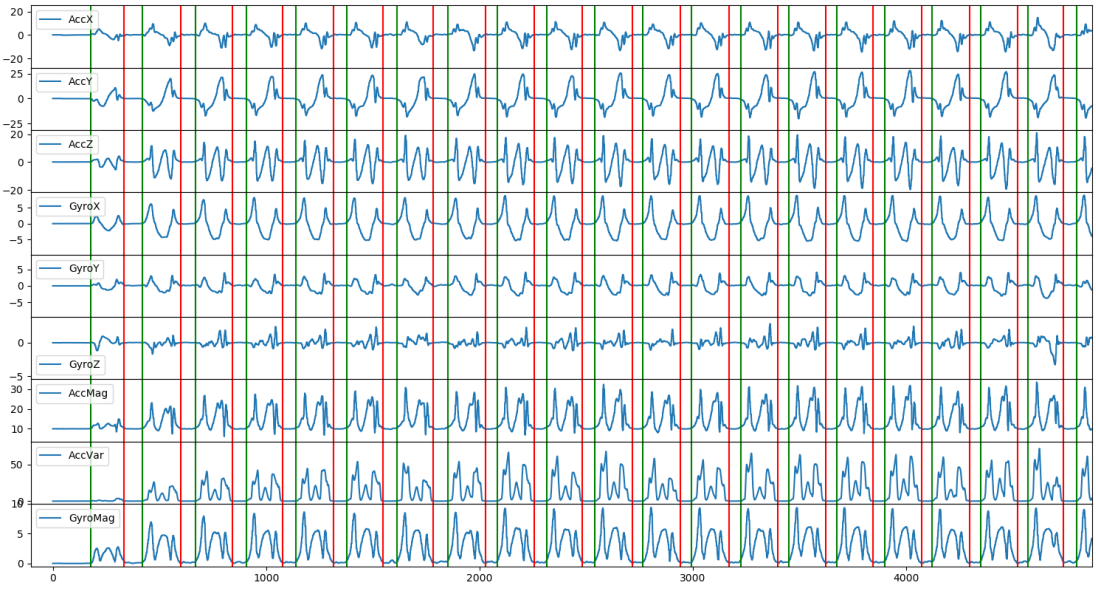


Figure 3.7: Segmentation of the Raw Data according to the ZUPT detection method, for one foot. In order: Acceleration (x, y and z axes, in ms^{-2}), Gyroscope (x,y and z axis, in rad/s) and the features relative to the three boolean conditions (acceleration magnitude [ms^{-2}] , acceleration variance [m^2s^{-4}] and angular rate energy[rad/s]). X axis represents samples (at 100Hz sampling). Red vertical lines represent the beginning of midstance and ZUPT (when the toe hits the floor) and the green lines represent the toe-off - end of the ZUPT interval

3.4 Step Length Estimation

Fixing the IMUs to the feet enables the use of tracking through actual displacement reconstruction. Unlike the usage of an unconstrained smartphone, the length of a step can be retrieved by direct processing of the inertial signals and as such does not rely on modelling the relationship between step frequency or acceleration magnitude and step length. On smartphone based PDR, the MM algorithm would provide feedback in terms of the weights given to the Linear or Non-Linear models (3.1 and 3.2, with step Length SL, tunable weights A,B,C and K, step frequency and variance SF and SV and Acceleration magnitude A_max *and* A_min) obtained empirically, essentially tailoring the algorithm to the user. However, a change in the carrying context, gait speed or irregular movements (such as avoiding obstacles or climbing stairs) can disrupt the model. This makes PDR that relies solely on smartphone inertial signals infeasible.

$$SL = A + B.SF + C.SV \tag{3.1}$$

$$SL = K - \sqrt[4]{A_max - A_min} \tag{3.2}$$

On the other hand, foot-placed IMU systems focus on processing the signals so that double integration can be performed without accumulation of error. Even though the estimation of each

individual step length can be more or less influenced by error - derived from the prefiltering that may not be ideal for the signal features within a certain window, attitude estimation errors, incorrect detection of Zero Velocity and numerical integration errors - ZUPT prevents drifting of the signals. Error builds as a consequence of small estimation errors on each individual steps but remains stable over time. As a result, estimation of Step Length is only a function of double integrating the acceleration signal and applying ZUPT to the first integral - velocity. The numerical integration is done by the rectangle rule. Systems using ZUPT, depending on the precision of the inertial sensors and chosen algorithms, can achieve a distance estimation accuracy of 95-99% of the travelled distance [33, 34].

It is important to note that distance estimation requires acceleration data in the Earth frame, which means that the acceleration signal retrieved from the accelerometers needs to be rotated according to the attitude provided by the attitude estimation module. As such, the precision of these signals will depend on the precision of the detection of the z-axis direction. Even if there is not an adequate heading estimation, if the rotation of the x and y acceleration data indeed aligns with the xy plane (even if the x and y direction are slightly rotated), the estimation of distance will be precise. This provides additional information for the IMU fusion and MM modules, as there is a greater confidence on distance estimates compared to heading estimates and that can be exploited when correcting trajectories.

3.5 Heading Estimation

Step quantification requires not only step length but also step heading. Step heading has, in most cases, a greater effect in the PDR accuracy than the estimation of step length [35].

3.5.1 Heading from IMU Attitude Estimation

Alternatively, accurate estimation of sensor attitude through time will result in stable estimates of displacement not only regarding 2D distance in the xy plane but also regarding heading (yaw angle). However, and as mentioned in Chapter 2, without an absolute source of information, heading estimation will drift due to sensor imprecision and bias drift. As such, it is necessary to design a drift estimation and reduction/elimination strategy.

3.5.2 Future Work - Heading from Magnetometer

As mentioned before, relying on compass readings in indoor environments is not feasible, as magnetic disturbances would result in errors that can reach 40° even in contexts where the IMU is fixed to the body of the user [36]. It is this very fact that enables magnetic fingerprinting approaches which attempt to model the the disturbances in order to provide an additional source of absolute localization. Nonetheless, and since many IMUs come within a set of MEMS that also contains a 3-axis magnetometer, it may be useful to model the magnetic disturbances [35] so that they can be compensated for or filter the readings with a confidence metric so that an

absolute measurement can be used when samples are detected to be undisturbed [37]. However, methods that attempt to model or compensate predictable errors or filter out unreliable magnetic field measurements require initial calibration and that makes them unsuitable for a system that can be used simply by placing the sensors on the feet and requires no knowledge of how to calibrate. All attitude estimation algorithms benefit from a second Earth field measurement (to complement the z-axis estimate from gravity), especially the Gradient Descent algorithm that was implemented on the current system. However, these algorithms are designed with general attitude estimation in mind, assuming availability of reliable magnetic reading and applying them to the PDR problem in indoor contexts requires sophisticated and specific filtering of the data.

Nevertheless, the fact that two MEMS sensors are used coupled with the fact that there is continuous knowledge of their relative position may enable novel filtering of the magnetic readings that could be used to obtain additional information with which to refine the inertial measurements. These include smarter filtering of the magnetic field readings in order to retrieve a confidence metric that would allow for its use when noise-free and refinement of the estimation of relative position of the two feet by the consecutive detection of similar magnetic disturbances, without requiring pre-calibration.

3.5.3 Heuristic Approaches to Drift Elimination

Detecting and compensating drift without resorting to sources of information that are external to the IMU requires some assumptions to be made so that the errors can be estimated as the deviation of the measured values from the assumptions (the heuristics). These heuristic approaches were first applied to the PDR problem by Borenstein and Ojeda in [9], with their work on the Heuristic Drift Elimination (HDE), which has been built on by several authors in order to extend its capabilities to less constrained contexts and improve robustness. Examples of extensions to HDE include generalization to 3D (Extended HDE [38]), additional processing before correcting the heading to a dominant direction (Advanced HDE [39]) or added robustness for buildings where curved corridors or open spaces exist, which deteriorate normal HDE performance (improved HDE [40]).

In general, HDE assumes that since most man-made buildings are made up of straight corridors and rooms where the main directions are orthogonal to each other (i.e. turns at 90 $^\circ$ angles) and people tend to walk along these main directions, small deviations from the expected main direction can be attributed to drift and therefore corrected and be used for drift correction when heading changes are sharper. This simple approach works well for highly constrained buildings with narrow corridors, but will have trouble with open areas and buildings with unconventional plants or unconstrained trajectories where small heading changes over time are actually significant. Moreover, these algorithms also need to be able to deal with movements such as obstacle avoidance, wandering within a corridor and complete changes of direction.

In this system, and following the corollary of unconstraining trajectories and making the fewest amount of assumptions possible, Heuristic Drift Elimination will not be used. Enhanced Drift Elimination following the assumption of plane floors, where the user does not gradually ascend or

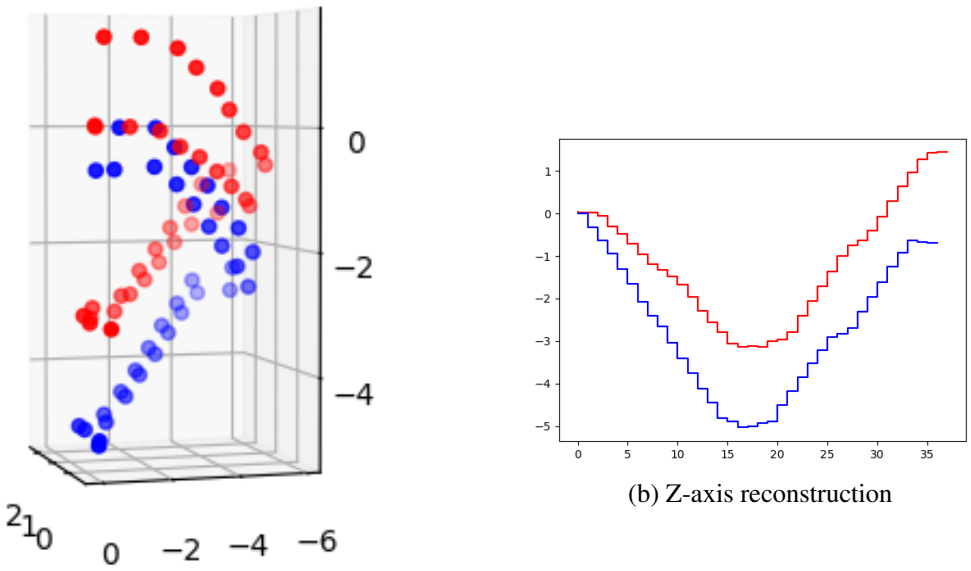
descend is used for the correction of errors in the pitch and yaw angles and is a reasonable assumption. However, these errors are of smallest importance, as there are two sources of information for the calculation of both angles (gyroscope and accelerometer). Furthermore, as there are buildings with small ramps, the correction could possibly introduce error in specific occasions. For these reasons, no Heuristic Elimination of Drift was performed in the current system.

3.6 Vertical Displacement and Detection of Floor Changes

PDR is usually approached at in 2.5D (x/y and floor detection) since there is no advantage of using a complete 3D model, as tracking 3D position is not necessary for the disambiguation of overlapping positions. This is so because possible positions with same x and y coordinates, even if there are ramps or small stairs, are always divided in well defined floors. This makes 2D tracking coupled with floor detection sufficient for indoor contexts. After attitude estimation and the proper filtering (which is also necessary for the 2D tracking), measuring vertical displacement is as simple as reconstructing the z-axis position signal. However, actual tracking will only be necessary when stair climbing and elevator riding activities are detected.

Detecting floor changing activities with suitable accuracy is a very rewarding source of information for PDR systems with map matching, as it can help correct large errors that standard MM techniques can have trouble with by providing a near-absolute source of location (with the caveat there can be multiple possible staircases in the same direction to perform the correction). While unconstrained IMU systems (namely smartphone based) need to have algorithms such as State Machines or Machine Learning Classifiers [4,41] in order to detect floor changing activities, foot-placed IMU systems can easily detect the characteristic vertical displacement on stairs. As for elevators and escalators, the key would be to detect the acceleration signature, by detecting the start and stop of the ascent or descent and the vibration that is inherent to these machines. Coupling this approach with the smartphone barometer (which is used in smartphone based systems [42]) could detect these transitions with promising accuracy. However, this was left out of the scope of the present work. Nonetheless, the vertical displacement is calculated at every step and made available to all modules, so that future activity recognition modules can easily access this information. Figure 3.8 illustrates a simple experiment using the developed system, which clearly shows that each step can be detected even if current acceleration filter is not suitable for the complete reconstruction of the vertical displacement (superior values of acceleration and stronger impacts compared to normal gait).

It is worth noting that while in the presented example the attitude filter probably had an imperfect alignment with the z-axis during the descent stage for the left foot, which was exacerbated by the fact that this vertical reconstruction deals with higher acceleration values than the horizontal reconstruction, a fusion algorithm such as the one developed in this work for the 2-D case could improve the estimates. Nevertheless, once the attitude estimates stabilized, it is possible to see that the estimates of vertical displacement are the same and it is also possible to see the step where no stair step is climbed (roughly in the middle), in both feet.



(a) 3D reconstruction

(b) Z-axis reconstruction

Figure 3.8: Vertical Displacement Reconstruction - The red dots and red curve represent the detected positions of the left foot and the blue dots and line represent right foot positions. The route consists of going down two sets of stairs and then climbing back up to the start position(0,0,0). Unit for all axes is meters (except X-axis on the left, which is in step number). Left - scatter plot of the detected positions in 3D. Right, evolution of z position.

3.7 Review of the Step Quantification Model

Systems with foot-mounted IMUs have several advantages over unconstrained IMUs. The possibility of applying ZUPT and ZARU which enable robust displacement reconstruction and attitude estimation respectively makes for smaller errors (due to more robust attitude estimates) and slower error propagation (from the bounded growth of errors in the velocity signal reconstruction). As a result, the PDR problem can be solved without resorting to step length models and is solvable even without an absolute source of heading (providing the initial orientation is known).

The combination of 3 Boolean conditions and skewed median filter is a simple and effective method for ZUPT (or step) detection. Nevertheless, using dynamic thresholds, adding extra features and a more sophisticated Truth Table, using a FSM without imposing restrictions on the placement of the IMU or restricting measurements to a single axis or even extending the current algorithm with a Bayesian model could provide better results, as the key is correct and strict classification of the ZUPT intervals, especially at its boundaries. This algorithm is robust enough to detect all ZUPT periods, even if it sometimes returns partitioned ZUPT intervals which are reflected in a larger step count and in errors (which are small, since they occur in the middle or close to the ZUPT samples, and can be corrected easily by the Fusion module). Results for step detection can be found in Chapter 7. Improving this detection could also have a positive effect in

attitude estimation, if the detection is extended to ZARU, and improve the estimates both in Step Length and Heading.

After detection of the ZUPT, and since the set of samples corresponding to a specific step is segmented and available for offline processing (which needs to be efficient so as to keep the system near-real time), more sophisticated correction and smoothing algorithms could be developed, namely without requiring state space error modeling. Reconstructing the velocity signal as accurately as possible needs to make use of as much information as possible.

In the current system there is no detection of floor changing activities nor floor detection, even though some experiments with 3D tracking on stairs were done (which are presented in Chapter 7). When deciding the focus of the thesis within the broader scope of IL the focus was given to 2D tracking and its stability in prolonged periods of time. This is due to the fact that a PDR module can always be embedded in a broader IL system (such as one using opportunistic signals or wireless signal beacons) and improve the interpolation of positions between signal updates or when the wireless signals are not available or its measurements are not precise enough.

Chapter 4

Integration of Multiple Sensors

4.1 Motivation

Solving the PDR problem through inertial sensors is a matter of obtaining enough independent information that errors arising from the imperfect precision of sensors can be detected and compensated or at least bounded. This is crucial because a given position depends on the previous estimates, and error builds very quickly. In Chapters 2 and 3, and besides the algorithms required to reconstruct displacement from accelerometer and gyroscope data, additional sources of information were discussed, ZARU and ZUPT. Further sources of information can be artificially created through heuristics, however, these constrain the system and do not support movement that, while completely normal and possible, does not verify the assumption (of straight line walking, for example).

Apart from artificially created sources of information, additional information can come from using multiple sensors. The information from each sensor is then fused together and improved by exploiting the fact that the spatial relationship between the sensors is known, assuming they are fixed to a certain body part. In a preliminary stage, the idea behind sensor integration was to create a framework that could make use of as much information as possible, by allowing a PDR solution to use all available sensors (smartphone, smartwatch, body-fixed IMUs). As the design of the system evolved, the advantages of foot-mounted IMUs (ZUPT/ZARU) over a set of sensors where the updates were not possible was clear: The possibility of actual displacement reconstruction over the use of step quantification models and the better performance in terms of heading estimation even in a system without a magnetometer.

This chapter describes the design process and the IMU fusion algorithm that was developed. The algorithm receives as input the quantified steps from each feet independently and fuses the information so that the estimated position (its output) is calculated from additional information.

The main motivation behind the usage of multiple sensors is the independence of their sensor biases (non-zero mean error), which means that these errors can be estimated and compensated. It is possible to create a multiple IMU array fixed to the body at the foot so that the both attitude estimates and displacement reconstruction can use multiple measurements for angular rate of turn

and acceleration, allowing for much more precise estimation of the sensor biases when coupled with ZARU/ZUPT [43, 44].

Alternatively, and with hardware cost in mind, this system aims to use two low-cost 6 DOF (Degree of Freedom - 3-axis gyroscope and 3-axis accelerometer) IMUs, one on each foot, fusing the information through the knowledge of the restriction that applies to their relative position (namely, leg opening restricts the distance between both feet). Additional behavioural heuristics could be introduced, such as assuming the user will never walk with the feet at larger than shoulder width distance sideways or predicting the user trajectory to be smooth. Though adding heuristics can increase performance in certain contexts (as explored in section 3.5.3), the focus of the system, as mentioned before, is performing DR through displacement reconstruction only, without filtering movement with predefined assumptions.

4.2 Synchronization

When using multiple IMU units that communicate wirelessly with the processing device (in this case, a smartphone), it is crucial to synchronize the measurements, so that the system can retrieve the temporal order of the measurements it is receiving. When initializing the measurement with the two IMU units, their clocks do not initialize at the same time. Despite this, by taking the elapsed time as the difference between the initial time and current time, the system can know the temporal sequence of the data even if there is a difference in initialization time (which is in the order of the hundredth of a second). This would not be enough if perfect synchronization of the events of both feet was needed. In this system, step quantification is done individually (with the only interaction being in the case of two consecutive steps being detected on the same foot, as described in the previous Chapter), and as such the only requisite is that the sampling rates are consistent and that the initial delay is shorter than the time difference between two steps, with which the system easily complies.

It was also necessary to filter and buffer sensor events, as the sensors return measurements that are not necessarily in temporal order, despite correctly labelling their timestamps as shown in the example stream presented in Figure 4.1. The retrieved values correspond to: Sensor type (here only accelerometer, A, and gyroscope, G, data is being retrieved); Timestamp (in nanoseconds); x,y and z axis values (ms^{-2}) for the accelerometer and rad/s for the gyroscope; sensor accuracy (measure of the certainty of the sensor, currently not modeled or used).

It is worth mentioning that the Pandlets sensor array includes an Android Application Programming Interface that makes the MEMS sensor array easily to integrate into Android Applications. The retrieved sensor events are identical to the ones that would be retrieved from the smartphone internal sensors.

```
A,120000000,0.23104046,-3.1794999,-9.424775,3
A,130000000,0.22864626,-3.1663318,-9.425972,3
G,120000000,-7.9894834E-4,-0.02263687,-0.002396845,3
G,130000000,5.326322E-4,-0.022104237,-0.0013315806,3
A,140000000,0.23223756,-3.217807,-9.419987,3
A,150000000,0.22146365,-3.1854854,-9.406818,3
G,140000000,0.0013315806,-0.021038974,-0.002396845,3
G,150000000,0.002130529,-0.021038974,-0.0026631611,3
```

Figure 4.1: Example of the Sensor Events retrieved from an IMU - The timestamps do not follow a temporal order. Refer to the text for an explanation of the presented values.

4.3 Gaussian Step Fusion Algorithm

In Kalman Filtering approaches to displacement reconstruction and tracking there is no explicit definition of step, instead focusing on real-time estimation of all state variables, namely position, velocity and sensor errors. In these systems, introducing an inequality constraint that limits the distance between the two feet at all times, the estimates and related covariance matrices can reflect the upper bound of the distance, and help obtain a better estimate than any of the IMUs by itself. Examples of Inequality Constraint KF are presented in [45–48]. The proposed novel Gaussian Step Fusion Algorithm builds on this maximum spatial distance formulation but is much more lightweight, does its corrections on a step basis and is of easy integration with the MM module, as will be explored in Chapter 5.

4.3.1 Properties of 2D Gaussian Distributions

A Gaussian distribution (usually called normal distribution in 1D) is a very common continuous probability distribution in statistics. It is suitable for the approximation of the probability distribution of several processes, most notably and interestingly for the problem at hand, of physical quantities that are derived from several measurements with associated measurement errors [49]. Gaussian distributions have very interesting properties that make their application in propagation of uncertainties and crossing of multiple values very useful, namely:

- **Simplicity of Representation -** A Gaussian distribution is completely described by two parameters: mean (expected value) and covariance (variance in 1D and variance + correlation in multiple dimensions). These distributions assign a non-zero probability to all combination of variables in the variable space, however, the areas that are far from the expected value and outside the covariance matrix range will have very small probability density values (99.7% of the total probability is within 3 standard deviations of the mean).
- **Simplicity of Sum and Product -** Computing the sum or product of two multivariate Gaussian distributions can be done analytically and with a single matrix inversion, which makes

it very simple in the 2D case. The sum of two independent gaussians is gaussian, as is the multiplication of independent distributions (in this case, an unnormalized gaussian). By calculating the moments (mean and variance) of the resulting distribution (either sum or product), we obtain a new simple representation of the current expected value and variance (which can be seen as the confidence in the value).

2D Gaussians can be used to describe the belief of the system in relation to the current position. Using such distributions, the mean μ will represent the current expected value and the covariance matrix, Σ will describe the confidence the system has in that measurement. Σ *and* μ are described in 4.1, with VAR(X) representing the variance of distribution X and COV(X,Y) (equal to COV(Y,X)) represents the covariance of the X and Y distributions.

$$\mu = \begin{bmatrix} \bar{X} \\ \bar{Y} \end{bmatrix} \quad , \quad \Sigma = \begin{bmatrix} VAR(X) & COV(X,Y) \\ COV(Y,X) & VAR(Y) \end{bmatrix} \tag{4.1}$$

The location after a step can be thought of as the sum of the distribution that describes the past position and the distribution that describes the travelled distance in each direction. This results in a new estimate with larger values on the covariance matrix (less confidence), which is to be expected, as an uncertain value was added to another. This calculation is simple, as the sum of two bivariate gaussians has the following moments (*mean* μ_3 *and covariance* Σ_3), (4.2):

$$\mu_3 = \mu_1 + \mu_2 \quad , \quad \Sigma_3 = \Sigma_1 + \Sigma_2 \tag{4.2}$$

The fusion of two estimates for the same value (an estimate from the algorithm and a new step, for example) is done through the multiplication of both distributions. Multiplication will reduce the covariance, giving the system higher confidence on its estimates. This can be exemplified by thinking of two pieces of information: one that says that a variable is between 3 and 10 with absolute certainty and the other that says the same variable is between 7 and 13 again with absolute certainty. By fusing both pieces of information we see that the variable will be between 7 and 10. This is the key for the retrieval of more accurate estimates from two different sources. The moments of the product of the two distributions can be obtained as in eq.(4.3), where the -1 power describes matrix inversion:

$$\mu_3 = \Sigma_2(\Sigma_1 + \Sigma_2)^{-1}\mu_1 + \Sigma_1(\Sigma_1 + \Sigma_2)^{-1}\mu_2 \quad , \quad \Sigma_3 = \Sigma_1(\Sigma_1 + \Sigma_2)^{-1}\Sigma_2 \tag{4.3}$$

4.3.2 Implementation

The filter will propagate the uncertainty of measurements and of the current position (through the sum) and refine the measurements through crossing the information with the knowledge of the maximum foot distance (product of distributions).

The filter is initialized by providing the initial position and covariance (which will be zero if the position is known with absolute certainty). At this point, both feet will be at the same position.

Once a step is detected, the possible position of the corresponding foot after said step is applied can be anywhere around the foot that is planted on the ground, with a distance smaller than the maximum foot distance. In this system, the maximum foot distance is defined statically at 1.5 meters, which is used as the value for 3 standard deviations of the distribution. This means that said distribution limits 99.7% of foot strikes to occur within 1.5 meters of the opposite foot (and 95.45% within 1 meter), which translates to a variance of 0.25 m^2. This distribution around the planted foot will then be crossed with the step information in order to correct it if they are not in agreement. Even though this distribution says that the expected value is that the two feet are at the same position (or at the same relative position as in the start), when the product of this anatomical constraint distribution and the actual step is done, the result will be much closer to the actual step measurement (which has smaller covariance). Figure 4.2 describes this initial model of where the foot can go. As soon as one foot is detected to be moving, anatomical constraints give the system the first piece of information of where the foot will land (the second one being the step quantification estimate). This first distribution is retrieved by eq. 4.4, applied to the case of a moving right foot.

$$\mu_{Right} = \mu_{Left} \quad , \quad \Sigma_{Right} = \Sigma_{Left} + \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \tag{4.4}$$

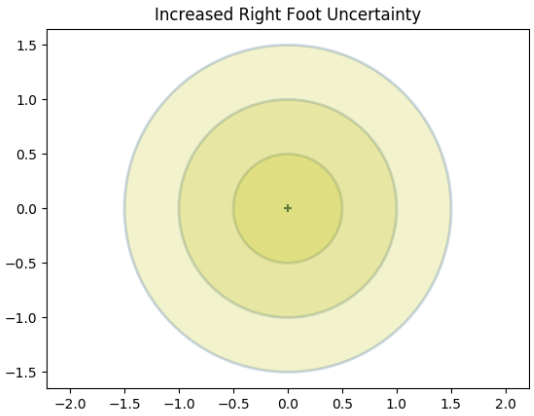


Figure 4.2: Modelling the Uncertainty of a Step - Possible Location of the Moving Foot - The stationary foot is at the center of this distribution. The circles represent the distance within 1, 2 and 3 standard deviations. Axes units in meters.

Secondly, the quantified step is added (according to estimated length, l, and heading, θ) to the current position of the corresponding foot, with a covariance matrix that attempts to mimic the sensor uncertainty (eq. 4.5). The probability distribution cannot be completely modelled by a bivariate gaussian as the error in distance (smaller) and heading (larger) cannot be translated to x and y (Figure 4.3, left), but it can be approximated (Figure 4.3, center). At first, and due to the fact that there is much greater certainty in the distance estimation, an attempt to model the correlation between the two variables was made (Figure 4.3). However, as the algorithm runs and propagates

the distributions for a longer period and especially on sharp turns, a highly correlated distribution did not always return better results, especially if there are errors in step detection or quantification. Furthermore, the product of distributions with correlation can sometimes result in distributions that do not model the system uncertainty as well, especially when the multiplied gaussians are very different. As such, a simpler model with no correlation (still using different variance values for the two axes) was adopted. In Figure 4.3, the distribution of the position of the foot after the step is represented in pink, the yellow distribution marks the entire area where the foot can be after a step, as explained above, and the blue line represents the step (from the past position of the foot to the new expected position).

$$\mu_t = \mu_{t-1} + \begin{bmatrix} l \times cos(\theta) \\ l \times sin(\theta) \end{bmatrix} \quad , \quad \Sigma_t = \Sigma_{t-1} + \begin{bmatrix} 0.0001 + |\frac{l}{12} \times sin(\theta)| & 0 \\ 0 & 0.0001 + |\frac{l}{12} \times cos(\theta)| \end{bmatrix} \tag{4.5}$$

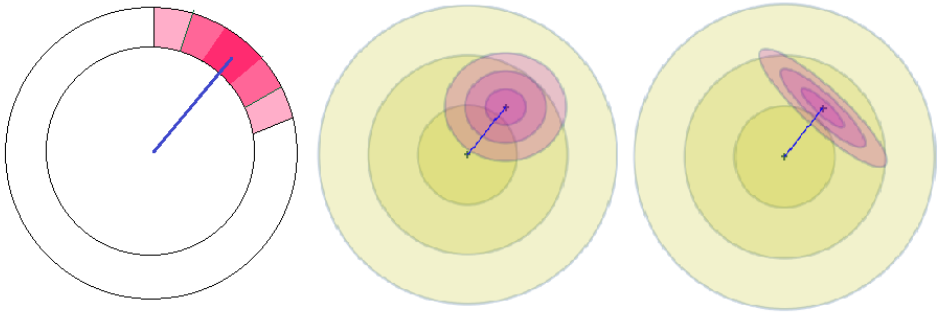


Figure 4.3: Modelling the Uncertainty of a Step - Left, True Distribution - Right, Correlated Distribution - Center, Implemented Distribution

These two estimates of position provide two sources of information which can be fused in order to retrieve a single estimate that is more accurate than each one individually. As such, both distributions are multiplied together, as described in eq. 4.3. The resulting distribution will be the current estimate of the foot position, as in Figure 4.4.

In this first illustrative example, the feet were at the same position and that knowledge was considered by the system to be certain (i.e. zero covariance). The following Figure (4.5) shows how the system performs for a second step. Note how the anatomical constraint distribution encompasses a larger area, reflecting the initial uncertainty regarding the location of the planted foot. The distribution of the position after the step has such an elongated form because it better models the uncertainty associated with the step (larger uncertainty for heading than length) without having to use correlation values.

As is readily apparent from the algorithm, the product of the two distributions will always cause a slight reduction in travelled distance. In order to compensate for this, the past location of the foot that is moving is also crossed with the location of the stationary foot. This acts like

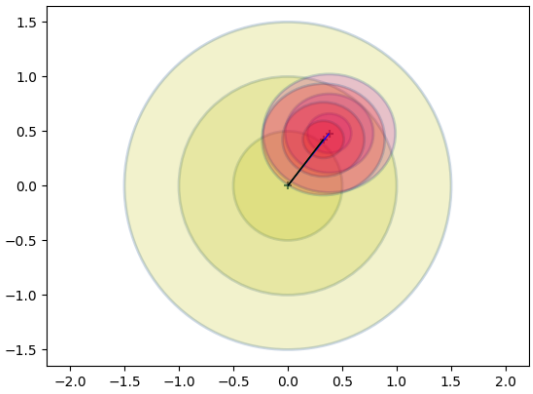


Figure 4.4: Final probability distribution of the foot position after the multiplication of the step estimate and anatomic constraint, in red. The black line represents the corrected step. Axes units in meters

a reverse verification: the place where the foot started the step had to be within the radius of the stationary foot.

4.3.3 Drift Modelling

The algorithm as is would already provide better estimates than the individual step estimates. Nevertheless, and since errors can be easily detected by exaggerated corrections, that is, by final positions that are significantly different from the position estimates from step addition, it is possible to model the system drift by compensating for the corrections that the system is making.

After the final probability distribution, the difference in estimated angle from a simple addition of the step (with the current drift estimate compensated for) to the effective angle - from the original uncorrected position (before the "reverse verification") to the final estimate of the foot expected value - is added to the drift estimate of that foot. Despite the fact that the values are usually very small and end up working as an effective proportional controller, there can be erroneous step detections or quantifications that result in sharp corrections by the algorithm. As such, the angle is only added to the drift estimate if it is likely to have come from actual sensor drift and not from erroneous estimates or sharp turns. In order to decide if the angle difference should be added to the drift estimate, the following test is applied (eq. 4.6), where θ_{diff} represents the difference between the estimated heading (from the Step Quantification Module) of the current step and the last received step of that foot (all values in radians):

$$deviation = \theta_{final} - \theta_{initial}\ , \quad drift += deviation \quad if \quad |deviation \times cos(\theta_{diff})| < 0.05 \tag{4.6}$$

The difference between the current and last step values is used because if the direction of walk did not vary greatly and the algorithm feels the need to perform corrections then it is surely because of sensor drift. On the contrary, if there are inconsistencies in the location of the feet

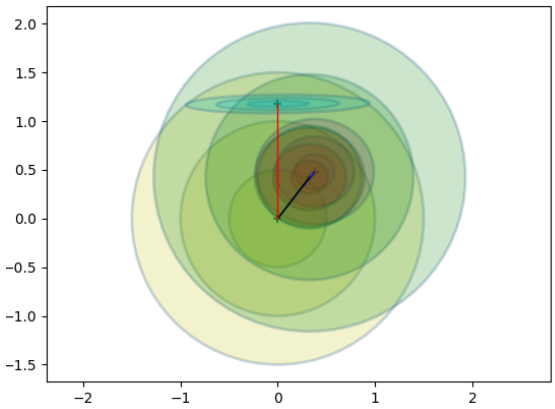


Figure 4.5: Green ellipsis - anatomical constraint / blue ellipsis - step and associated uncertainty / red line - step estimate. Axis in meters.

at a certain point and there is a sharp turn, a large correction is expected that is not a result of drift. This correction is still necessary, as it will align the relative distance between the feet, it just does not reflect drift. Additionally, this test is also useful for heading corrections caused by the MM algorithm, in order to decide if the correction should only be considered on that step (mainly as a correction of position, see Chapter 5) or if in fact there is a heading drift that should be compensated. Figure 4.6 shows how the drift modelling works even while in a curved trajectory, where it can be noted that the feet are not properly aligned right after the first change of direction but that error is corrected by the final shown step (and the final corrected steps have the same heading as the initial estimate after correcting for the estimated drift). At this point, estimated drift for the left foot is 10°while drift for the right foot is 5°. The full route can be seen in Figure 4.8b.

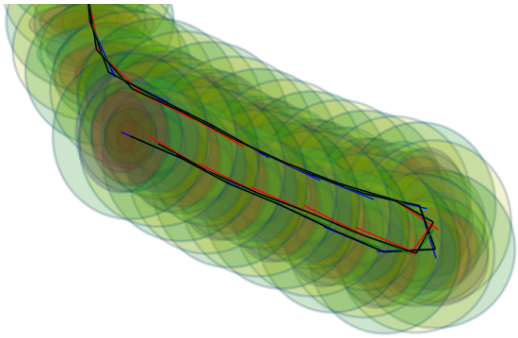


Figure 4.6: .
Drift Modelling and Correction during a Curved Trajectory - The ellipsis represent the probability distribution of the position whose expected value is at the center of the ellipsis. Red and Blue lines represent step estimates for the left and right foot, respectively. Black lines represent the actual step that was added.

4.4 Module Overview

A diagram of the Gaussian Filter for IMU fusion is presented on Figure 4.7, below.

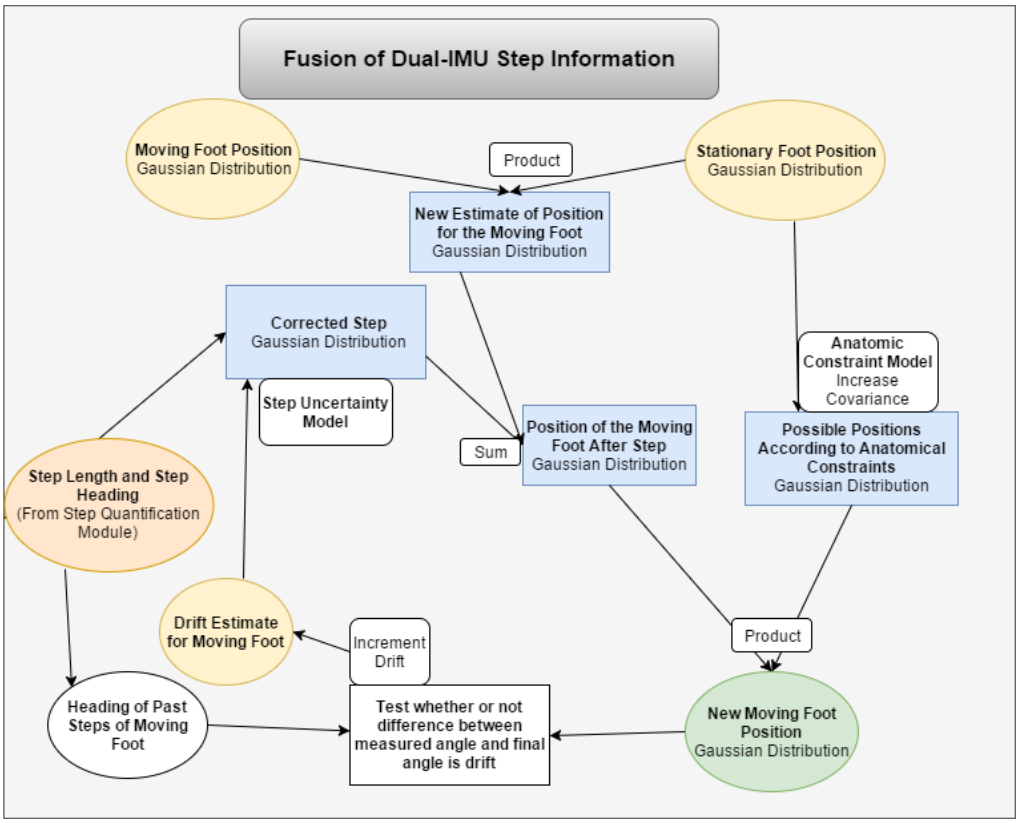
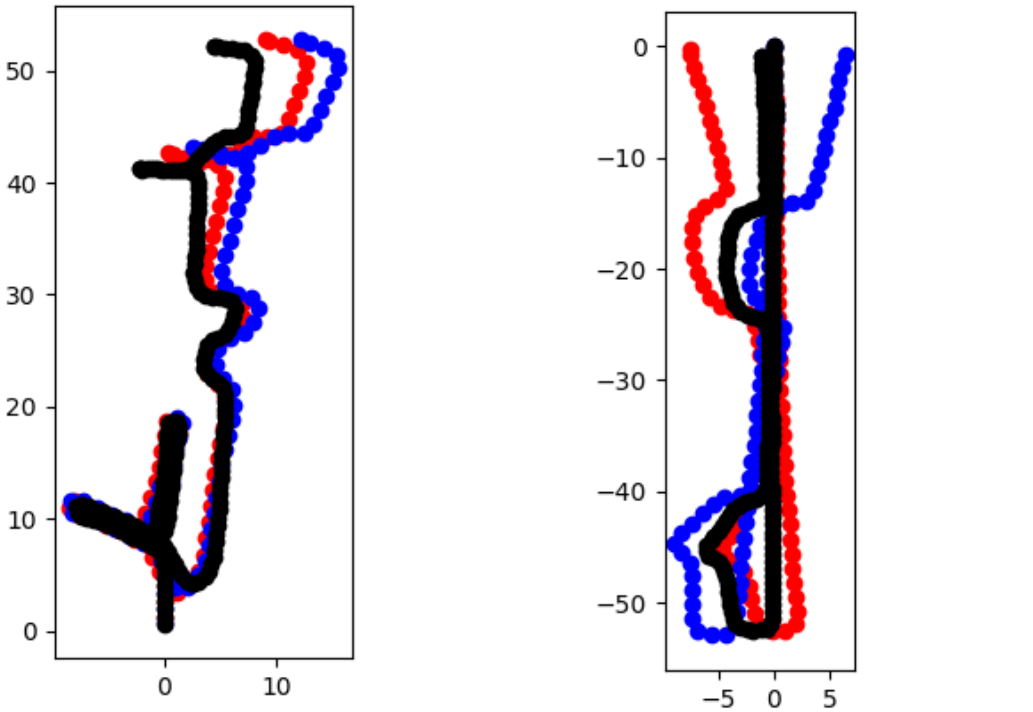


Figure 4.7: Overview of the Gaussian Step Fusion Algorithm - Color code: Orange - input; Yellow - current system estimates; Blue - intermediate distributions; Green - final result, new system estimate

The Gaussian Filter provides better estimates than either of the two IMUs by itself (as can be seen in Figure 4.8). It is capable of modelling drift in the signal to the extent possible with only step information, especially if the bias of the individual IMU are in opposite directions. It does, however, cause a small decrease of the travelled distance that is not fully negated by the initial product between the moving foot position and the position of the stationary foot. Despite the fact that the modelling of the uncertainty of the step is a very rough approximation of the true non-gaussian distribution, this algorithm provides a very robust and lightweight alternative to the Inequality Constraint Kalman Filters that allows for the discretization of displacement in steps (and the associated advantages of forwards and backwards quantification of the step as explored in the previous Chapter).



(a) Example of the Correction for route **open**

(b) Example of the Correction for route **8-route-B**

Figure 4.8: Gaussian Filter Results - Red and blue dots represent independent positions for the left and right foot respectively. Black dots represent the positions for both feet corrected by the Filter. Refer to Chapter 7 for the description of the routes. X and Y coordinates in meters. The routes are rotated 180º and mirrored in the X axis (so that applying the routes on the map image coordinates where (0,0) is the top left corner is immediate)

Chapter 5

Map Matching

5.1 Overview and State of the Art

The IL Problem can be simplified to PDR if the starting location is known. The starting location can be retrieved through detecting the entrance the user took by outdoor GPS tracking or by asking the user to mark it on the map. The end-goal of any IL solution would be for it to be able to function without a starting location. As such, the ideal Map Matching approach must be able to work without any kind of starting information. This means that the trajectory that is retrieved from the PDR modules could be completely relative, requiring a comparison to all possible trajectories (at best, if absolute heading is obtained through attitude, the trajectory can be oriented to a certain extent, decreasing candidate positions in buildings with symmetric structures). If a probabilistic approach is chosen, then the algorithm must be able to work with a starting condition of uniform distribution of probability for the entire plane. In this work, the focus is on accurate and stable long-term tracking. Nevertheless, common algorithms for IL applications can be applied to PDR and as such are reviewed in this chapter.

MM algorithms are usually grouped into 4 groups [50], [51]: Geometric, Topological, Probabilistic and Advanced. Geometric methods, the simplest, include point-to-point, point-to-curve and curve-to-curve matching for the estimation of current link and position within the link. Topological methods, as an extension of the geometric methods, take into account not only shape but connectivity between the links. These are widely applied in road map-matching, as the roadmap is inherently a graph, but they can still be of interest if the floorplan can be simplified to a link and nodes model. Probabilistic methods establish a confidence region on the map based on the variances of the error of the sensors. If more than one link (in the case of a map graph) is within the region of confidence, several criteria such as heading similarity, connectivity or closeness are used for the estimation of position. As an example, in Figure 5.1, Simple Geometric approaches would not be able to select a link with certainty. Topological algorithms would select the bottom link because of the similarity in heading between the link and the P2-P3 segment. Probabilistic approaches would estimate the probability of matching in each node, with the bottom node having a greater probability, due to the topological reasoning. These methods were developed mainly

for absolute estimates of location such as from GPS or other wireless signals but can be adapted for IMU-based PDR and, while incapable of solving the complete IL problem, can be useful in tracking systems.

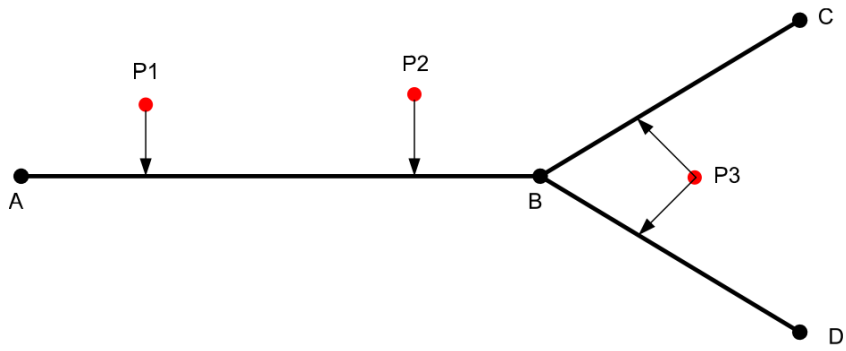


Figure 5.1: An example of a network and 3 measured positions, in [50]

.

There are two main types of approaches for IL MM with an inertial PDR-system with unknown initial position: Shape Matching and Probabilistic Filters (KF, PF and related algorithms, CRF, etc.). The former are predicated on comparing the shape of the trajectory with the topological information of the map (as a link and nodes model). On the other hand, the latter attempt to model the errors of the measurements and estimations to create a probability distribution of the user location (as a Gaussian cloud or through weighted particles) in a continuous or discrete map.

5.1.1 Shape Processing and Adaptive Matching

Taking advantage of the simplification of the map through graph representation, matching shapes to the graph can be simpler than probabilistic approaches for the initial localization problem, both in computational power and speed. Due to its low resolution when locating the user inside a given link and in open spaces, once the user is located in a certain area, one can then turn to the widely used probabilistic approaches for more precise tracking, avoiding initialization on a large area. Shape processing and matching is probably more useful as a lighter alternative to the unknown starting location problem for systems with no absolute source of location (such as wireless signals or signals from beacons), since initializing probabilistic approaches on a very large area can take a long time to converge. These algorithms can also be interesting for situations where the map matching algorithm is estimating location with less certainty or having trouble matching to valid paths, by leveraging the entire shape (especially interesting if there is a large number of points). In all cases, the usability of such algorithms depends on the ability to generate simple graphic representations of the floormap.

In order to use shape matching, it is necessary to define an adaptive shape comparison algorithm, capable of detecting similarities in trajectories made up of possibly erroneous link length

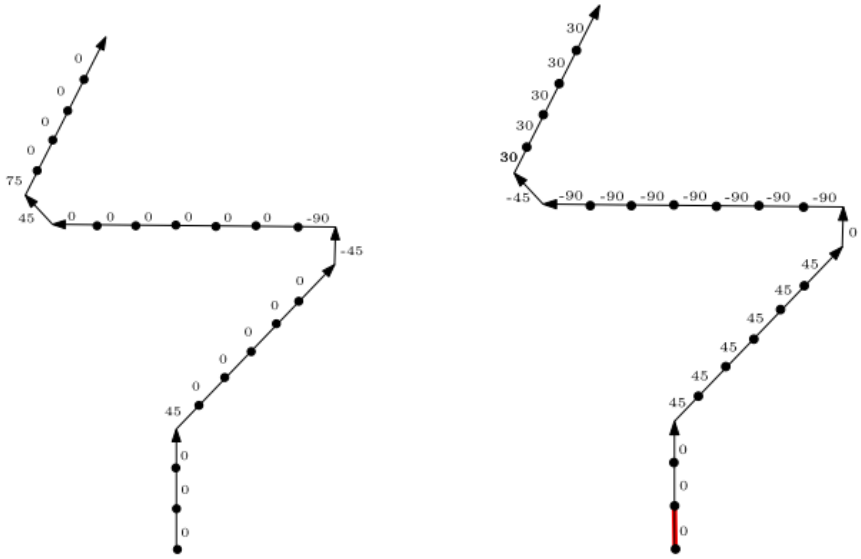


Figure 5.2: Representation of trajectories with Local Angles (left) and Global Angles (right), in [52]

.

and link angles. As such, the algorithm must be able to look past translations and length differences, finding common subsequences within adjustable tolerable ranges. As with all algorithms for mobile applications, the computational cost of querying for shapes and performing the comparison must not be too great.

In [52], the authors propose two comparison metrics, based on either Local Angles (heading variation between steps) or Global Angles (azimuth of step, relative to a specific orientation) as illustrated in Figure 5.2. As with all shape comparisons algorithms, being able to have an idea of the absolute orientation of the path and map (i.e. the North direction) simplifies the searching procedure, as similar shapes with different orientations (larger than a threshold) will not be adequate candidates. Global Angle based comparison needs two parameters to be defined: range and percentage. For every subsection of the path of length equal to range, there must be an equivalence of angles (with tolerance both on angle measurement and time-shifting, as long as order is kept). Local Angle based comparison verifies if for every subsection of length equal to range the sum of all angles is equal (within a tolerance value). The authors also present an effective indexation and querying algorithm based on Suffix Trees (a classic text string search algorithm).

A different approach is described in [53], where a computer vision inspired shape comparison algorithm, the shape filter, is employed. The shape filter enables a primary discard of trajectories that are not similar. By calculating the interception of lines that are generated from the trajectory centroid at specified angles with the trajectory itself as illustrated in Figure 5.3 (once again, having a reference of the absolute orientation is very beneficial), a code descriptor of the shape can be used for the selection of appropriate candidates. In this approach, the shape filter is followed by

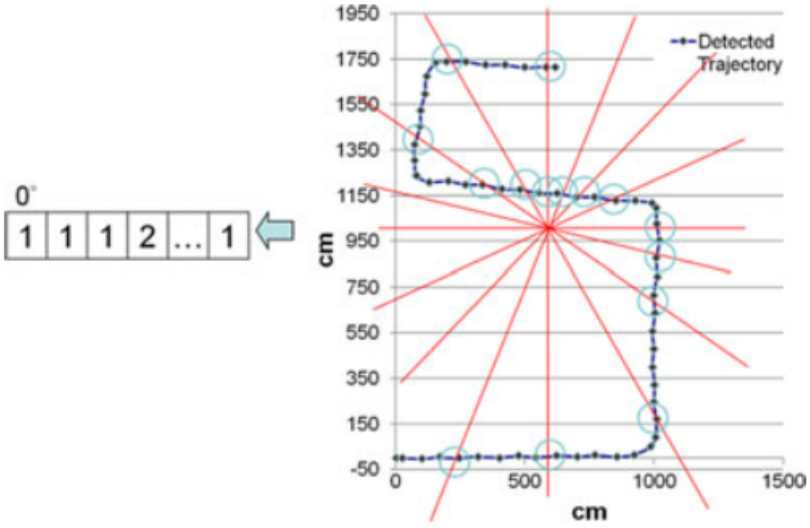


Figure 5.3: Example of the Shape Filtering algorithm, in [53]

.

comparison of angles through a simpler version of a chain code (discrete local angle representation) and edge length in order to filter out shapes that matched due to wrong angle detections, "fake turns".

Yet another different approach is suggested in [54], based on detecting the Longest Common Subsequence for given parameters of space and time tolerance, as shown in Figure 5.4 for a 1-D trajectory. Sampling of trajectories for faster comparison is suggested.

Once the set of possible user positions is restrained to a single link, IL is simplified to PDR (or in the case of a problem or error with tracking that requires more sophisticated/robust map matching, PDR can be resumed).

Classical MM algorithms (geometrical and topological) can be extended to work with PDR instead of GPS signals. Matching of readings to the map will take into account trajectory orientation, proximity of the estimated position and the link and connectivity of links, as illustrated in Figure 5.5, [55], [56]. The weight of each similarity can be weighted differently according to the graph properties, such as density or even building specific information. For example, artificially describing an open space with a mesh of links and nodes can work if connectivity considerations are ignored (as there are no physical constraints blocking any type of movement from link to link). The network would function in a grid-like manner in that area. Furthermore, adopting a probabilistic approach (by matching to multiple position with associated probability) on denser parts of the map network can help mitigate the lack of resolution resulting from the extreme discretization that this model causes.

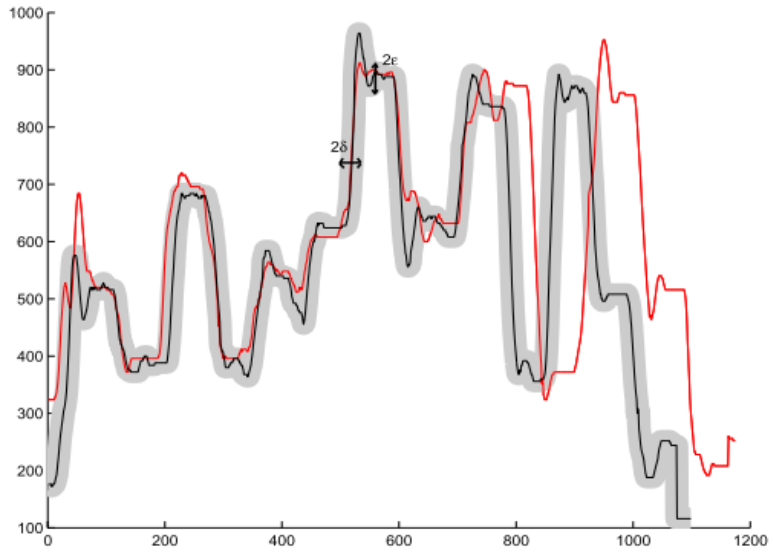


Figure 5.4: Longest Common Subsequence trajectory (1-D) comparison with space (ε) and time (δ) tolerance, in [54]

.

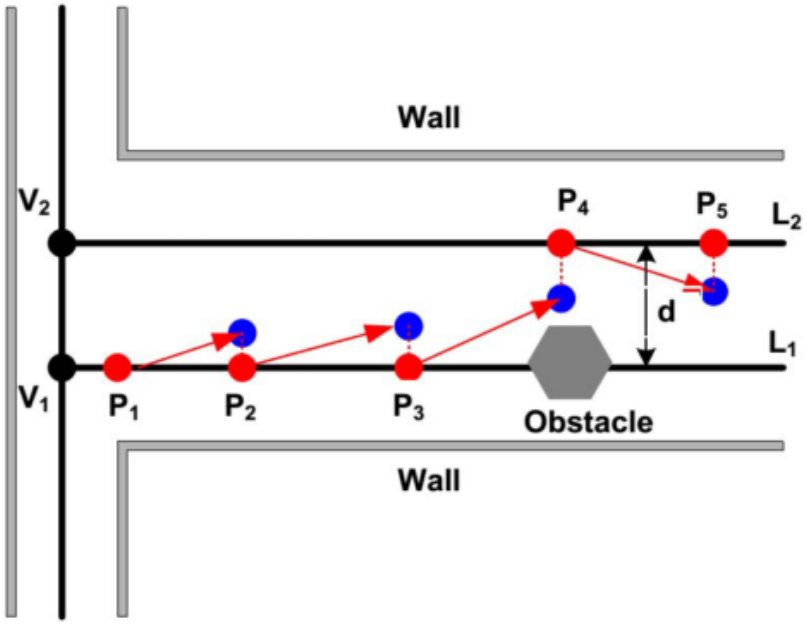


Figure 5.5: Matching of PDR measurements; An example of a transition to a parallel link, in [55]

.

There are a number of other similarity metrics that can be applied. A good example of such

a metric is the Fréchet distance, [57], [56], intuitively defined as the shortest "leash" that can link two trajectories, commonly illustrated by the trajectories described by a dog and its owner, without backtracking but with the ability to stop at different times. The backtracking restrictions are relaxed in some variants of the metric, such as weak Fréchet distance.

Finally, in [58], the authors describe a shape simplification method named Discrete Curve Evolution (Figure 5.6), where a polyline (connected set of oriented segments) is simplified by node elimination, minimizing the loss of information, with the added benefit of noise reduction. The choice of node is done through the sum of distances between a given node and its immediate neighbors and between the neighbors. The shape can be compared to the map until elimination of additional nodes no longer results in an increase in similarity. This technique could be employed in prefiltering of shapes.

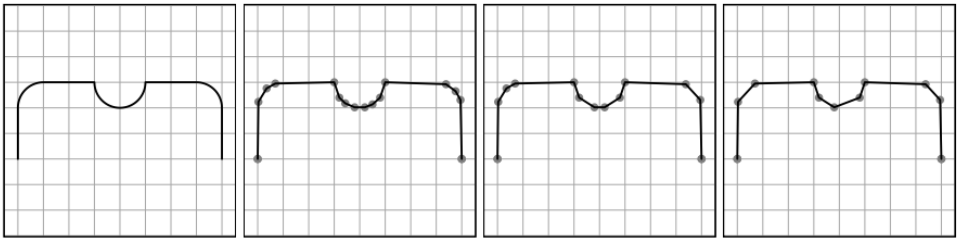


Figure 5.6: Discrete Curve Evolution - reducing noise while maintaining information in discrete shapes. Sequence of curves following elimination of additional nodes, in [58]

.

Through exploitation of the simplification of the variable space that arises from the link-node representation, it is possible to combine different approaches for shape and map matching, estimating the position within a link according to a weighted function of the different methods or following a probabilistic approach and propagating alternative matches with their respective probabilities.

Methods that rely on shape processing and matching are very interesting solutions for deciding in which link of the graph the user is currently in, which can also be useful for this PDR-centric system, as they are lighter than probabilistic methods such as the PF (described below), exploit a map representation to improve robustness and could benefit from using the independent trajectories generated by processing of the signal of the two IMUs. Despite all of this, these methods do not provide enough accuracy - while great at selecting which node the user is in, the methods are not great at defining where in the node the user is, and the simplification of the map further hinders localization accuracy - and have trouble with open spaces - even though these can be represented by a mesh of links and nodes, this shortcut means that the most interesting properties of the representation, its topography, is lost.

5.1.2 Probabilistic Approaches: Dealing with Uncertainty

Even if the map representation provides helpful constraints and if the PDR module provides estimates with a good degree of confidence, matching the position of the user to the map to a single point according to the measured displacement can never be done with total certainty. In fact, often times the uncertainty in the estimated position is very significant. As such, map matching can be approached from a probabilistic point of view, in the frame of Bayesian inference.

The uncertainty can be modelled or thought of as noise, which is usually considered additive but can also be implemented as multiplicative. Bayesian Filtering (BF) is a suitable approach and consists in finding the state **x** at time **n**, of a system that obeys a function **f** with noise **b**. There is a measurement function, **h**, which computes the output **y** of the system, while factoring in noise **v** as in Equations 5.1 and 5.2, ([59]).

$$x_n = f(x_{n-1}, d) \tag{5.1}$$

$$y_n = h(x_n, v) \tag{5.2}$$

The goal of the application of BF to this problem is to calculate the state (Position of the user) at time n through the system and measurement functions. To account for the uncertainty of the system and measurements, variables can be described as probabilistic distributions. As such, x and y can be described as Equation 5.3- the system model (estimating a state based on the previous state); and Equation 5.4 - the measurement model (the probability of a measurement based on current state).

$$x_n = p(x_n|x_{n-1}) \tag{5.3}$$

$$y_n = p(y_n|x_n) \tag{5.4}$$

Assuming that state transitions depend only on the previous state (Markov property), the goal is to calculate the probability of a given state given the measurement through Equation 5.5.

$$p(x_n|y_n) = \frac{p(y_n|x_n) * p(x_n|y_{n-1})}{p(y_n|y_{n-1})} \tag{5.5}$$

A Bayes filter can be directly applied to a map through a space discretization. By tessellating the map, the probability of the user being in a certain cell after a certain movement can be updated through the probability of that movement causing the user to move from neighboring cells to said cell [36].

5.1.3 Recursive Bayesian Filtering

Recursive Bayes models are implemented through two steps. The first one, the prediction step, involves calculating the prior, the probability of the state (position) given the last measure-

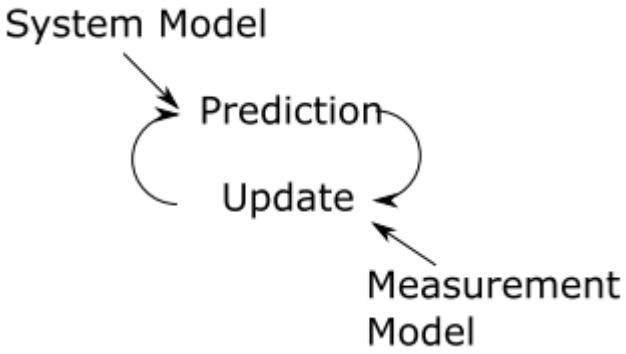


Figure 5.7: Recursive Bayesian Filter, from [59]

.

ment (last position and estimated displacement). The update step involves the likelihood and the evidence and incorporates the current measurement. Figure 5.7 illustrates the process. The actual analytical computation of Equation 5.5 is intractable, so the recursive Bayes filter will be a generic approach for the filtering problem.

There are three possible implementations of recursive BF: Hidden Markov Models, Kalman Filters and Particle Filters (PF). HMMs imply a discretization of the state variables ([60] applies it to the IL problem through discretization of the points of interest, i.e., rooms, and focusing on turn detection). KFs are optimal filters for linear problems with noise with Gaussian distributions (Extended and Unscented KF can be used to approximate non-linear systems). PF are an approximation of $\mathbf{p(x_n|y_n)}$ through a large number of random samples and are a very promising and widely applied algorithm in PDR based IL systems [42], [61], [62], [63], [64].

PF work through the generation of a large number of particles (procesing power will be the limiting factor) within the region where the user may be (in the case of an unknown location problem, throughout the entire map) uniformly. The prediction step will consist in the estimation of the position of the particles through Equation 5.6, where $\mathbf{l}$ is the estimated step length and θ is the estimated heading. δ introduces variation, by introducing zero-centered Gaussian noise [42].

$$\begin{bmatrix} x_t^i \\ y_t^i \end{bmatrix} = \begin{bmatrix} x_{t-1}^i + (l_t + \delta l_t^i)cos(\theta_t + \delta\theta_t^i) \\ y_{t-1}^i + (l_t + \delta l_t^i)sin(\theta_t + \delta\theta_t^i \end{bmatrix} \tag{5.6}$$

Once the new particle positions are calculated, their weight (which is always normalized) needs to be updated. This can take into account map restrictions (set weight to zero if particle prediction crosses an obstacle), behavioral information (assumptions from normal human patterns such as the likelihood of walking very near walls being smaller than walking in the center of a corridor) and likelihood of observations according to the environment, parameters that must be modeled. Searching the entire map for particle-wall collisions can be very time-consuming, especially when particles are distributed throughout the entire map. An approach to indexing obstacles for faster querying is described in [64].

Particle filters can suffer from degeneracy, which means that some particles will start describing increasingly less significant parts of the distribution as weights decrease [65]. As such, and to compensate for particle death (when particles cross obstacles), it is important to have a perform resampling, by generating a new particle distribution within the area of greatest importance, with the same information as the degenerating distribution. The position of the user at a given time can be estimated through the centroid of the particle distribution.

It may be interesting to perform post-processing on the particle distribution, such as smoothing of trajectories or calculation of the most likely trajectory. In [66], the authors illustrate the application of the Viterbi algorithm [67] to this problem. The Viterbi algorithm computes the most likely sequence of states based on the probability of moving from state to state according to observations (emission probability). The algorithm requires discretization of the state space which is achieved by clustering particles into states. Emission probability is computed from the particle weights.

5.1.4 Conditional Random Fields

CRF, [68], are probabilistic graphical models that take as input a vector of observations and attempt to predict a vector of latent variables. In CRF, the goal is to maximize the probability of a state (in this application, the position of the user) conditioned on the observation (again, the initial position and estimated displacement). CRF are useful because they model how observations relate to states and to transitions between states, which in a IL problem is very advantageous, as the PDR information refers to changes in user position.

States are defined by cells resulting from the discretization of the map. We then define feature functions, which define how strongly the observations (Z) support a sequence of states: $\mathbf{f(S_{t-1},S_{t},Z)}$. These features will take into account connectivity (map constraints), distance and heading between the two states (cells) [69] and any other restrictions that may be added, such as including activity information or tinkering probabilities according to expected behavior (based on assumptions such as increased likelihood of people walking in the middle of a corridor or being closer to the inside than outside wall when at a turn) [70, 71].

After defining the feature functions, it is necessary to set the weight of each feature. This can be done by training (matching the output of the PDR module to the ground truth trajectory) with several samples. The final step is to compute the most likely sequence of states and that is done through the Viterbi algorithm.

5.1.5 Correction and Learning

An ideal MM algorithm is not only robust enough to deal with noisy or uncertain data but is also able to improve the estimations of the PDR module through correction of erroneous position, drift estimation and, if relevant, feedback for tunable parameters such as step length models.

Road network based algorithms can provide feedback to the PDR module for heading adjustment and step length: If the matching process implies constant correction of heading direction, a heading correcting parameter can be calculated. Similarly, if the matched positions on the graph

correspond to an increase or decrease of step length and sharp turns are detected at an incorrect time (for example, if the user was expected to be far from the intersection where he turned) step length estimates can be shortened or lengthened.

Grid-based bayesian methods also provide feedback to the PDR module, relatively to step length estimations. This is crucial in order to avoid discretization error. By measuring whether the expected turns are occurring before or after the actual map intersection, the algorithm adjusts the expected step length. This algorithm does not make any heading corrections, however, the grid-like and probabilistic environment is very robust to incorrect readings.

Particles can carry more information than just their own position. Through tracking the step length and heading bias hypothesis that each particle followed, it is possible to retrieve information that can be used to improve the PDR module, as the particles with larger weight carry better parameters for step length and heading bias [64], [61].

Feedback from CRF approaches is possible by comparing the past trajectory once refined with the Viterbi algorithm to the raw trajectory provided by the PDR module, and adjusting persistent errors in drift or step length.

5.2 Problem Specific Algorithms

A PDR system that aims to be stable and robust for longer walks requires not only accurate attitude estimation, step detection and quantification and, in the case of multiple sensors, smart sensor fusion, but also a robust map matching algorithm, as a source of information that is completely independent from all inertial data. The goal of a MM algorithm for this specific application should be to detect and correct small errors as soon as possible and to be able to recover from errors that could not be corrected due to insufficient constraints in the vicinity of the current position (such as error accumulated during a long walk through an open space). Moreover, due to the accuracy requirements (under 2 meters) and the errors associated with discretization [36], opting for a continuous solution is preferable.

A Particle Filter could be a very interesting algorithm that could even bypass the need for an IMU fusion approach as developed for this system. On the other hand, attempting to keep computational costs at a minimum so that the system can work in real-time on a mobile platform calls for the exploration of different types of algorithms. In the following sections, three approaches are explored: Energy Minimization, Generation of Truncated Gaussian Distributions and problem specific Geometric Matching.

5.3 Energy Minimization Formulations

At any moment in the system, there are two pieces of information: the estimated trajectory (after step fusion) and the map. The trajectory may be incompatible with the map (going through walls) and should be corrected with it, while keeping in mind that the original estimate, while noisy, will have very important information such as relative angles between steps and travelled

distance, that should remain as unaltered as possible. From this, two different goals emerge: achieving a good fit between map and conserve the original information of the trajectory.

Energy Minimization methods attempt to solve problems where there are multiple constraints which model the degree of satisfaction of each goal and compute an overall energy score that must be minimized. Examples of these problems include finding spatial conformations of proteins (balancing the attraction and repulsion between atoms in a complex molecule [72]) and dynamic segmentation of images (finding the contour of an object from an initial guess). The latter example, known in computer vision as Active Contour Models or Snakes [73], uses an energy model to iteratively find a contour of an object. This energy model is usually made up of two forces which should be balanced (minimization of total energy function): internal and external, as in eq.5.7.

$$E_{total} = \int E_{int} + E_{ext} ds \tag{5.7}$$

On the one hand, internal energy corresponds to the constraints imposed on the contour F with arclength s - F(s) - , namely imposing a spring-like contraction (modeled with the first derivative of the curve) and aiming for smoothness of the contour (minimization of the second derivative of the curve). The internal energy is a function of the balance of contraction (total contour length) and smoothness (total curvature) as described by parameters α and β respectively (eq.5.8).

$$E_{int} = \alpha |F'(s)|^2 + \beta |F''(s)|^2 \tag{5.8}$$

On the other hand, external energy is a function of the total energy of the curve according to where it is on the image. For contour detection, the image gradient magnitude is computed, as the contour borders will be regions of high intensity change in the image. As a result, the external energy will be symmetric to the gradient, so that the borders have the lowest energy within the image (eq.5.9). The λ parameter weighs the importance of edge minimization. Together, the three tunable weights describe how important it is for the contour to be short, smooth or tight to the image edges.

$$E_{ext} = -\lambda |\Delta Img| \tag{5.9}$$

These equations can be solved for a continuous domain by computation of the total energy gradient direction and definition of the step size, however, it is much simpler and effective to perform discretization and solve them in the discrete domain, by the techniques presented in the following subsection. Furthermore, for the current application, the energy functions will be more complex to model and the shape to match with the map will not be a closed contour. An example of the application of an Energy Minimization algorithm is presented on Figure 5.8.

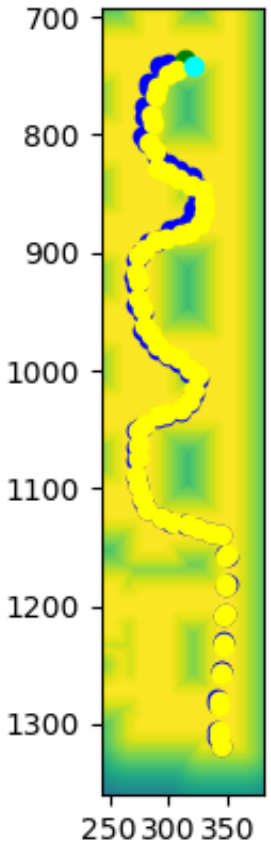


Figure 5.8: Example of an Energy Minimization algorithm application. The image is a portion of the Fraunhofer Porto 1st floor map at 5cm resolution. Axes values are the result of the 5cm grid

.

5.3.1 Discrete Active Contour Model

Discrete Active Contour Models (Snakes) will have discrete equations for energy which will be the sum of the energy at every individual point of the curve (5.10). In this application, the curve will be defined by the trajectory of the user. The energy function reflects the goal which is to avoid walls and obstacles on the map while maintaining the properties of the original trajectory V, with n vertices v_i.

$$E_{total} = \sum_{v_i \in V} E_{int}(v_i) + E_{ext}(v_i) \tag{5.10}$$

The internal energy function can be modeled just as in the continuous case by the following equation (5.11):

$$E_{int}(v_i) = \alpha |v_i - v_{i-1}|^2 + \beta |v_{i+1} - 2v_i + v_{i-1}|^2 \tag{5.11}$$

In order to meet the requirements of the problem, the internal energy function needs to be tinkered to reflect the need for keeping the distance between individual points the same while

also constraining abrupt changes of heading, so that the map is not able to impose a completely different shape on the trajectory even if distance between each point is kept. As a result, the function for the internal energy of point v_i should penalize changes of distance from the original distance between v_i *and* v_{i+1}, $O(v_i, v_{i+1})$ and attempt to smooth out the trajectory by minimizing the minimum distance from v_i to the line connecting v_{i-1} *to* v_{i+1}, $D(v_i, v_{i-1}, v_{i+1})$ (eq.5.13) as in eq.5.12.

$$E_{int}(v_i) = \alpha|(v_i - v_{i-1})^2 - O(v_i, v_{i+1})| + \beta D(v_i, v_{i-1}, v_{i+1}) \tag{5.12}$$

$$D(v_2, v_1, v_3) = \frac{|(x_2 - x_1)(y_1 y_0) - (x_1 - x_0)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \tag{5.13}$$

External energy should heavily penalize going through obstacles. As such, a distance transform is applied to the map, where the value of each pixel will be the minimum euclidean distance to an obstacle pixel. Once this is computed, energy values for free pixels will be negative with absolute value equal to the distance to the nearest wall or obstacle, $D_{obstacle}$, (with a cap of 10 pixels, which at a 5cm resolution corresponds to half a meter). This is done in order to provide an incentive to move away from the walls even though the energy at those pixels is still negative and they are valid positions but not to provide an incentive to move to the center of the corridor or room (which in large rooms would severely change the trajectory). Additionally, the minimum distance from each obstacle pixel to the nearest non-obstacle pixel, D_{free} is also computed and used to give positive energy to those pixels as a function of how impossible to reach they are. The energy of a certain pixel p_i is then given by: (eq.5.14)

$$E_{ext}(p_i) = \begin{cases} \lambda_1 D_{free}(p_i), if\ p_i\ is\ Obstacle \\ \\ -\lambda_2 max(D_{obstacle}(p_i), 10), if\ p_i\ is\ Free \end{cases} \tag{5.14}$$

Now that the energy of each point is defined, it is necessary to find a method to solve the minimization problem. For each original point of the trajectory, the algorithm will compare the total energy on the current point an on the eight immediate neighbors, as in Figure 5.9.

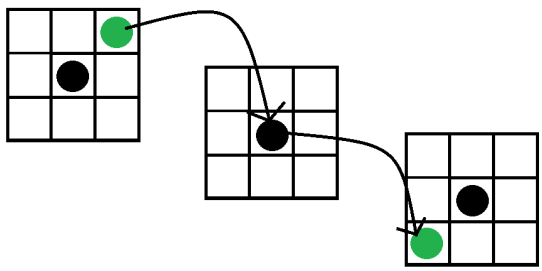


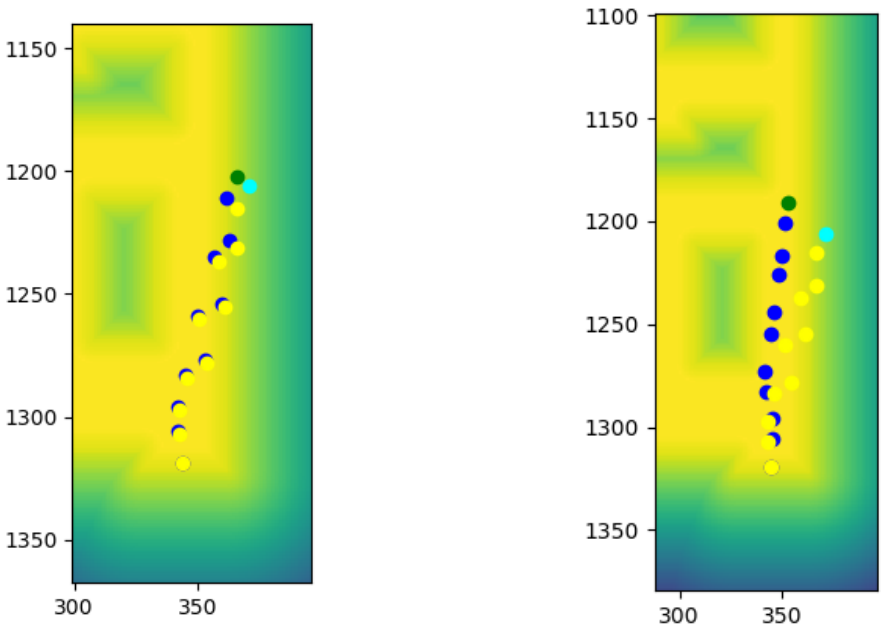
Figure 5.9: Discrete Snake Iteration Process

.

A solution could be found greedily by moving each point to the one with lowest energy until all points were in local minima (which is not guaranteed to be an optimal solution), through Dijkstra's shortest path algorithm [74] (since a graph can be constructed from the connection of trajectory points and path length is the associated energy) or more interestingly through Dynamic Programming with the Viterbi Algorithm. It is important to note, however, that the computational complexity of using a second order model is $O(m^3 * n)$, for n points and m neighbors, while using a first order model (where each point only depends on the previous point) has a complexity of $O(m^3 * n)$ (the greedy approach has a complexity of $O(m^n)$. As such, and because when performing offline testing, even after restricting the maximum number of points in the trajectory to be processed, the required convergence time was too great for a real-time (or near-real time) mobile application, the snake had to be reduced to a first order internal energy function (despite worse results). As a result, the internal energy is described by eq.5.15, which added a compensation for deviating from the original angle between v_{i-1} *and* v_i, $A(v_{i-1}, v_i)$.

$$E_{int} = \alpha_1 |(v_i - v_{i-1})^2 - O(v_i, v_{i+1})| + \beta |\angle v_{i-1} \hat{} v_i - A(v_{i-1}, v_i)| \tag{5.15}$$

The difference between the results of first and second order models, for a straight walk) is easily observed in Figure 5.10, where the first order model is unable to achieve adequate smoothing of the trajectory, whereas the second order model achieves a much closer result to reality.



(a) 1st Order Contour Model

(b) 2nd Order Contour Model

Figure 5.10: Differences between applying first and second order Discrete Active Contour Models

5.3.2 Limitations and Future Work

The main limitation of this approach is the fact that modelling walk using just the relationship between two points is very complicated (as a step deals with three positions - moving foot starting position, opposite foot position and moving foot final position). However, the complexity of the search even when applying the Viterbi algorithm is too great. Nevertheless proper tuning of first order snakes could provide interesting corrections.

Further challenges to be tackled include the estimation of drift, which is very complicated to merge with the energy minimization algorithm and correction of heading for the steps immediately after the correction. Furthermore, for good results to be obtained, an adequate number of points before and after colliding with a wall are needed, which means that this correction should only occur multiple steps after an incompatible trajectory has been found, and as such can be troublesome to implement in a near-real time system. Finally, and besides constraining each individual estimated position not to be in invalid positions, it is necessary to impose a constraint not to allow positions that would require steps to go through obstacles. This is a challenge that while compatible with first order energy functions, would also increase the complexity of the algorithm.

Still, active contours can eventually be an interesting approach to the map matching problem, especially if used in conjunction with other methods and relied upon only in circumstances of large errors that cannot be corrected by the methods proposed in the following Sections, especially for trajectories that do not loop.

5.4 Truncated Gaussian Distribution

The main idea behind the IMU fusion approach that was introduced in the previous Chapter is representing position estimates with associated uncertainty in a simple and easily updatable way. This can be extended by the map matching module through the generation of truncated gaussian distributions. While the numerical calculation of the moments (mean and covariance) of a bivariate truncated gaussian is possible [75], a simpler and more effective way of modelling the influence of the map on the estimated position distributions when they have significant overlap with obstacles is to sample from the resulting position distribution. By discarding samples that are in impossible locations (obstacles), the resulting distribution of samples can be used to estimate a new distribution which will be similar to the truncated gaussian. An example of this process is presented in Figure 5.11, where it is possible to see that new estimates for the probability distribution of the position of the foot is obtained by sampling and discarding according to the obstacles on the map.

The main advantage of approaching map matching as a third source of information for the Gaussian Filter is the easy integration of the methods and the resulting simplicity of estimating drift or correcting the covariance matrices. This method works very well in correcting small heading errors, which can provide invaluable drift information that helps the algorithm calibrate the errors for accurate estimates in posterior open spaces, or correct errors that accumulated during those open areas.

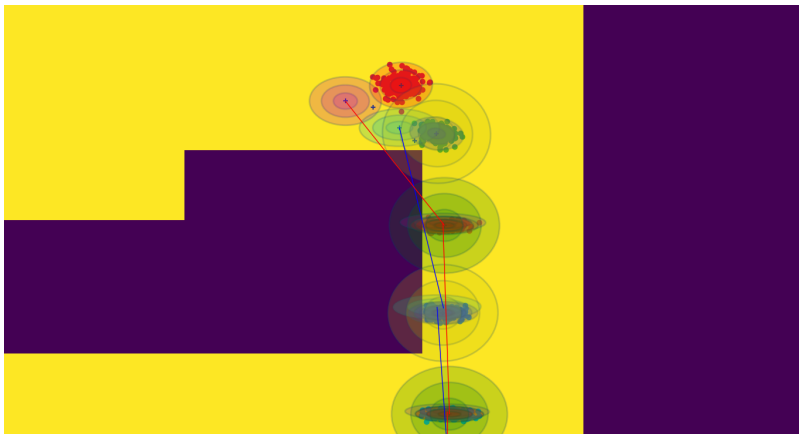


Figure 5.11: Generation of Truncated Gaussian Distributions

.

In cases where the estimated position has severe overlap with an obstacle, the immediate idea would be to increase the covariance matrix (uncertainty) in a smart way and implement a search heuristic that would provide an adequate final distribution. This, while possible, would require more complex modelling of the uncertainty of the system at each point and could eventually mean dealing with distributions that would encompass very large areas. While this would be a consequence of the actual true uncertainty of the system, subsequent steps would require very complex sampling strategies and could approximate a particle filter system. Furthermore, this algorithm would not use the information of the previous (or even immediately posterior) steps, which means that, even if the current position uncertainty is a function of the previous steps, there is a loss of information that could help correct larger errors. Nonetheless, this strategy is very interesting and further improvements on the propagation of the covariance matrices and in the sampling strategy could turn this Gaussian Filter into a very simplified yet very accurate version of a particle filter. In all cases, it would benefit from an additional shape matching or geometric matching method to help in cases of inability of correction or to correct or detect large errors.

5.5 Geometric Matching with Heading conservation

After designing and implementing the truncated gaussian algorithm, it was clear that there was a need for an additional algorithm capable of correcting larger errors which could complement it strengths. The truncated gaussian algorithm, together with the IMU fusion, provides adequate estimates of heading and its drift. It may, however, have trouble matching to future positions if errors accumulate and cause it to go straight into a wall instead of through the room door. Despite this, even in situations where considerable heading error has propagated, the overall signal can be matched to the correct position once the only way to keep moving in the approximate direction of the current heading is through a single door or opening. An example of such a correction is presented in Figure 5.12.

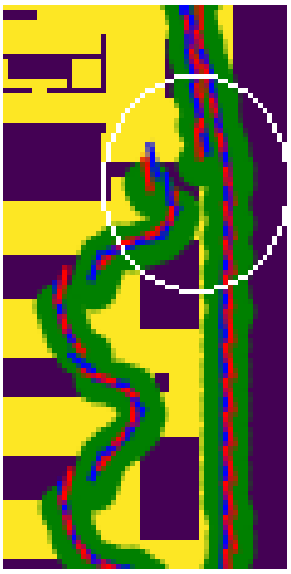


Figure 5.12: Example of Correction through the nearest available path in the direction of current heading, within the white circle

The designed algorithm builds on the heading correcting capacities of the gaussian framework by trusting the general direction of the heading and assuming that movement is indeed occurring in that direction even if error have accumulated. In these situations, and as soon as an impossible position estimate appears, the correct location should be searched along a line perpendicular to the step. Figure 5.13 presents an example of such a correction.

5.5.1 Limitations and Future Work

The main limitation of this method is the lack of a confidence metric which would allow the system to measure whether the current correction makes sense according to its past trajectory. It would be interesting for the system to calculate how much error could possibly have been accumulated (and have evaded the gaussian filter) so that the corrected position distance would not exceed the system estimate. In cases where the current information is not enough to make a confident decision, a more robust approach is needed, such as waiting for subsequent steps in order to apply the correction on more data and estimate the current position with added certainty.

In short, matching the position geometrically through exploitation of the general direction of the walk is interesting, however, the system needs to be made more flexible, so that it can provide good results in the following scenarios:

- Heading has just changed and is temporary (user is in a curved trajectory) and the collision with a wall occurs at this different angle
- Consecutive estimates of position for both feet lead to two different corrections

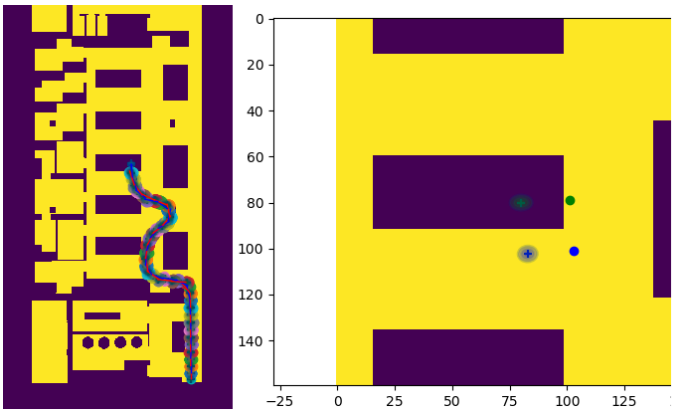


Figure 5.13: Example of Correction through the nearest available path in the direction of current heading, searched alongside a direction perpendicular to the step. Fuzzy points represent original step while solid dots represent the corrected step. The previous position is represented in blue while the next position is in green. Coordinates of the right plot are in grid cells, each cell corresponding to 5cm.

- Position correction leads to loss or gain of distance which cannot be easily recovered by the system
- Consecutive wrong corrections followed by erroneous drift estimates by the gaussian framework lead to wildly inaccurate estimates

Maintaining stability and above all the ability to recover from incorrect matches is crucial, and as such it may be required for the system to keep track of multiple estimates in some situations. In more complicated cases, shape matching strategies described above such as turn detection, active contours or shape filters can be applied.

Chapter 6

System Overview

The system architecture that was used was already present at Fraunhofer Portugal. This solution completely rewrote the attitude estimation, step tracker and position tracker while fusing the heading and step length estimation modules (as they work together in this system). All improvements were added to the existing code, providing an alternative to the existing system, from the algorithms to the sensor sources: smartphone inertial only, smartphone inertial + wireless signals, smartphone wireless signals only or foot-mounted IMU. Integration of wireless signals and foot-mounted IMU PDR is not ready at the moment, as it is dependent on a fusion model that can get the best out of both localization methods. Two designs are possible, where either PDR information interpolates positions between wireless signals readings (the easiest to implement, as a wireless signal based system is already ready at Fraunhofer) or information from the wireless readings corrects DR errors.

As of the writing of this document, the real-time system works solely up until the IMU sensor fusion (pure Dead Reckoning), but is already capable of displaying real-time non-map matched information to the user through the Google Indoor Maps interface. Future work will begin by the implementation of the Map Matching Module within the already existing framework.

Testing was done on the Android Studio tester, with the sensor signal being prerecorded and then fed into the system from a text file. Once the Position Tracker commits a new position, that position and the step that caused it are printed to a text file so that all map matching algorithms can first be tested in Python Code, due to ease of implementation and numerous available libraries.

An overview of the system architecture is presented in Figure 6.1. This diagram is simplified and does not include auxiliary classes such as the ones necessary for vectors and quaternions, among others. Green round boxes represent system inputs, blue, yellow and orange boxes represent processing classes (where the algorithms are implemented), and white boxes represent information classes (used to store and communicate measurements and estimates between the processing classes). Blue boxes apply both in offline Android Studio tests and real-time scenarios while orange apply only to real-time and yellow to offline processing and testing. Purple boxes correspond to additional classes or modules that could interact with the current system, as an extension.

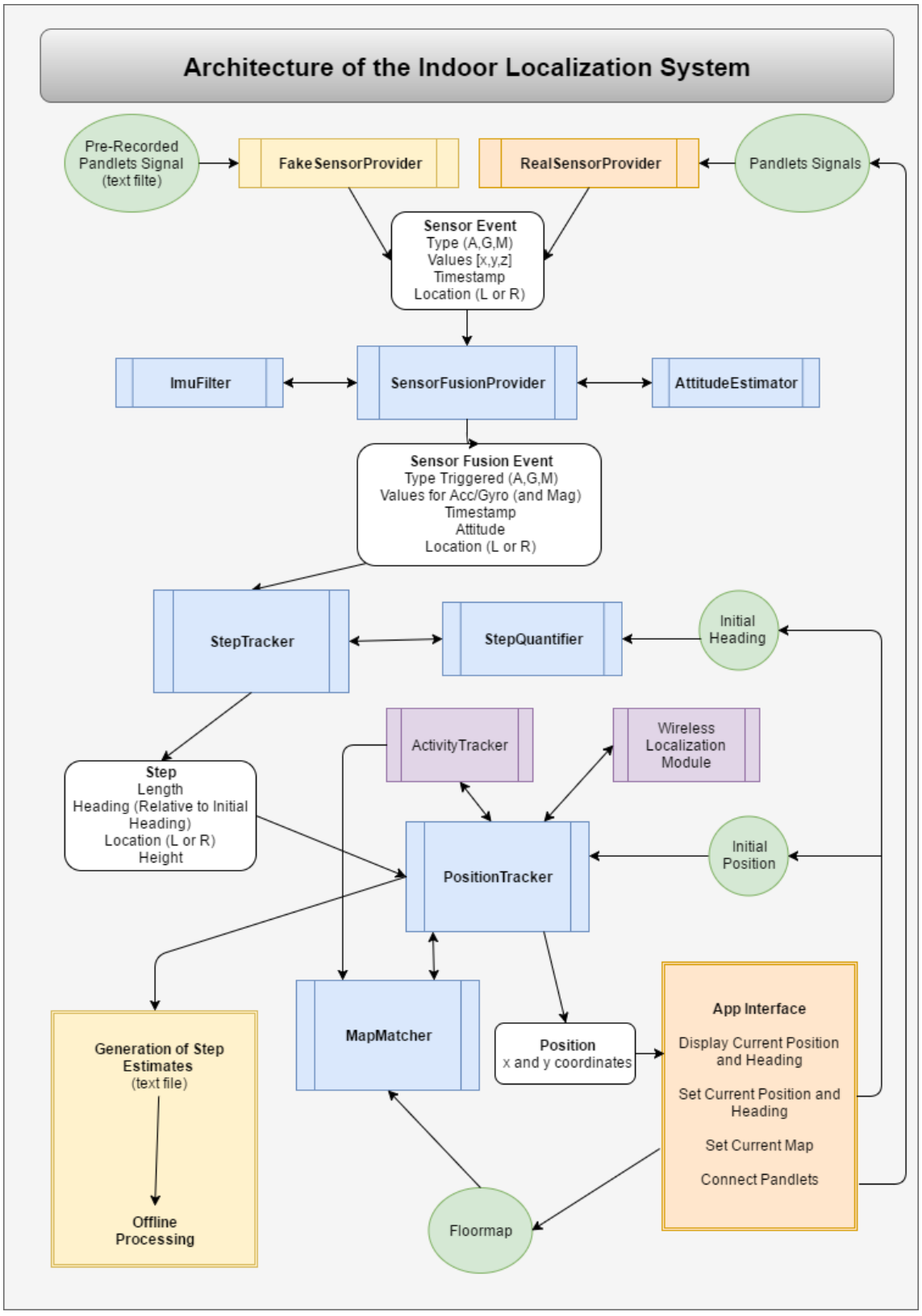


Figure 6.1: System Overview - Simplified System Architecture. Refer to text for color code explanation

Chapter 7

Experiments and Results

7.1 Overview of Tests and Routes

In order to test the performance of the developed PDR system, 5 routes were designed. The subjects followed the presented routes as closely as possible, with no specific markers throughout the route but with well defined start and end points. The length of the routes was estimated through the floorplan (which has a 5cm resolution). Throughout the route, the subject should tap the screen everytime a step is taken, which will be the ground truth for step detection. The system was tested on 5 users of different height and gender.

The routes are presented in Figures 7.1 through 7.3 and the reasoning behind each one is briefly discussed below. On all figures, the route (in blue) is overlayed on the floormap of the 1st Floor of the Fraunhofer Portugal Porto Office. The start point is represented by a green star and the end point by a red star. If the start and end points are the same, the location is represented by a yellow star. The blue arrows on top of the trajectory help in understanding the direction of movement in that part of the map. The red boxes represent obstacles that were placed on the open space in order to guide the subject through the designed trajectory.

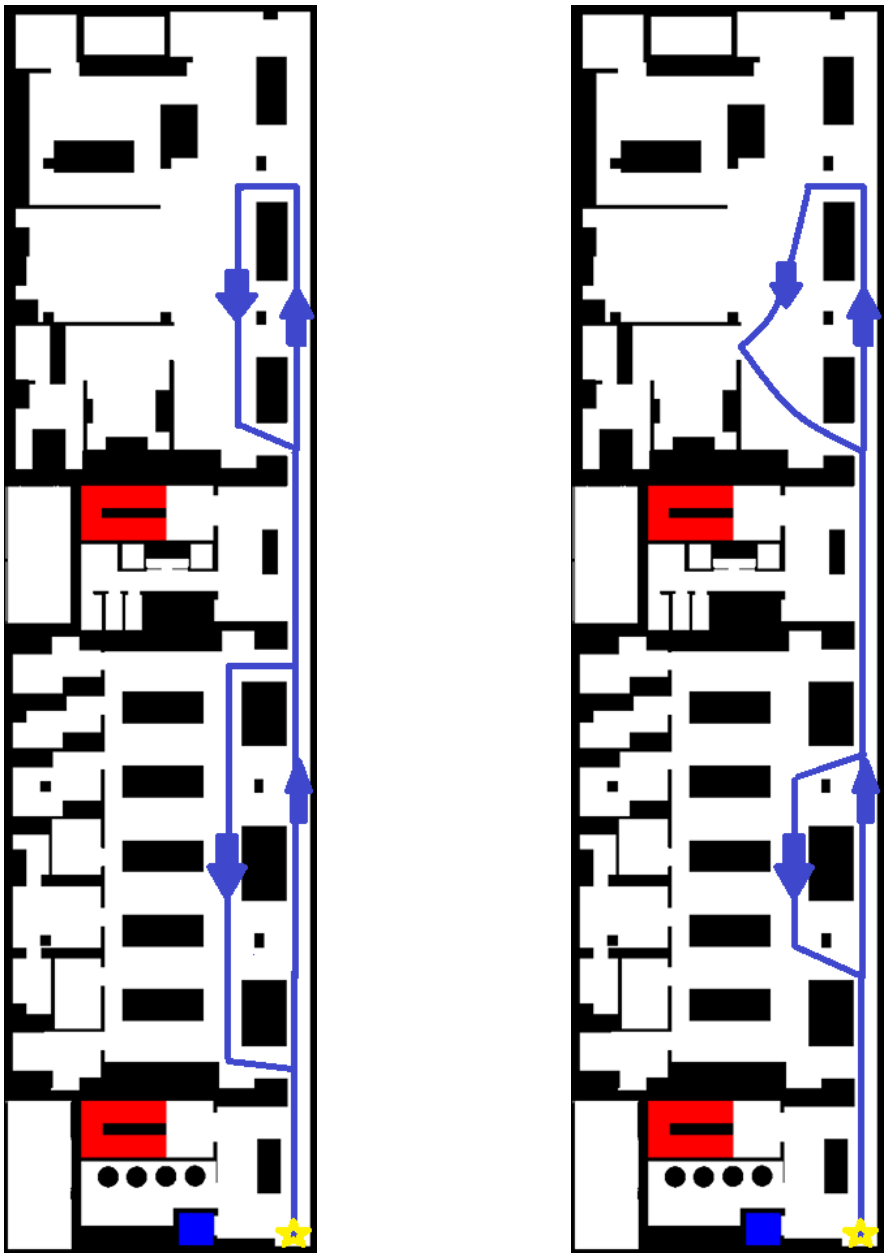
All routes are at least 100m long, so that there is a long enough test to the attitude estimation stability, accurate step detection and quantification and map matching. The routes are a mix of highly constrained trajectories (sharp turns between corridors) and loops in open spaces (where there are no immediate map constraints to correct growing errors).

The first two routes (On Figure 7.1), codenamed **8-route-A** and **8-route-B** respectively for their shapes, test the stability of the heading estimate and drift correction through long straight walks between sharp turns. Route **8-route-A**, on the left, is 116.5 meters long, while route **8-route-B**, on the right, is 117.9 meters long.

The third and fourth routes (On Figure 7.2), which were codenamed **open** and **loop**, test the robustness of the attitude estimation and IMU fusion algorithm by imposing a complex trajectory with round changes of direction and walking segments at non-standard orientations (i.e. not along the main axes of the map). Route **open** (Figure 7.2a) is 120.8 meters long, while route **loop** (Figure 7.2b) is 103.2 meters long. For the sake of clarity since the starting point is revisited, on

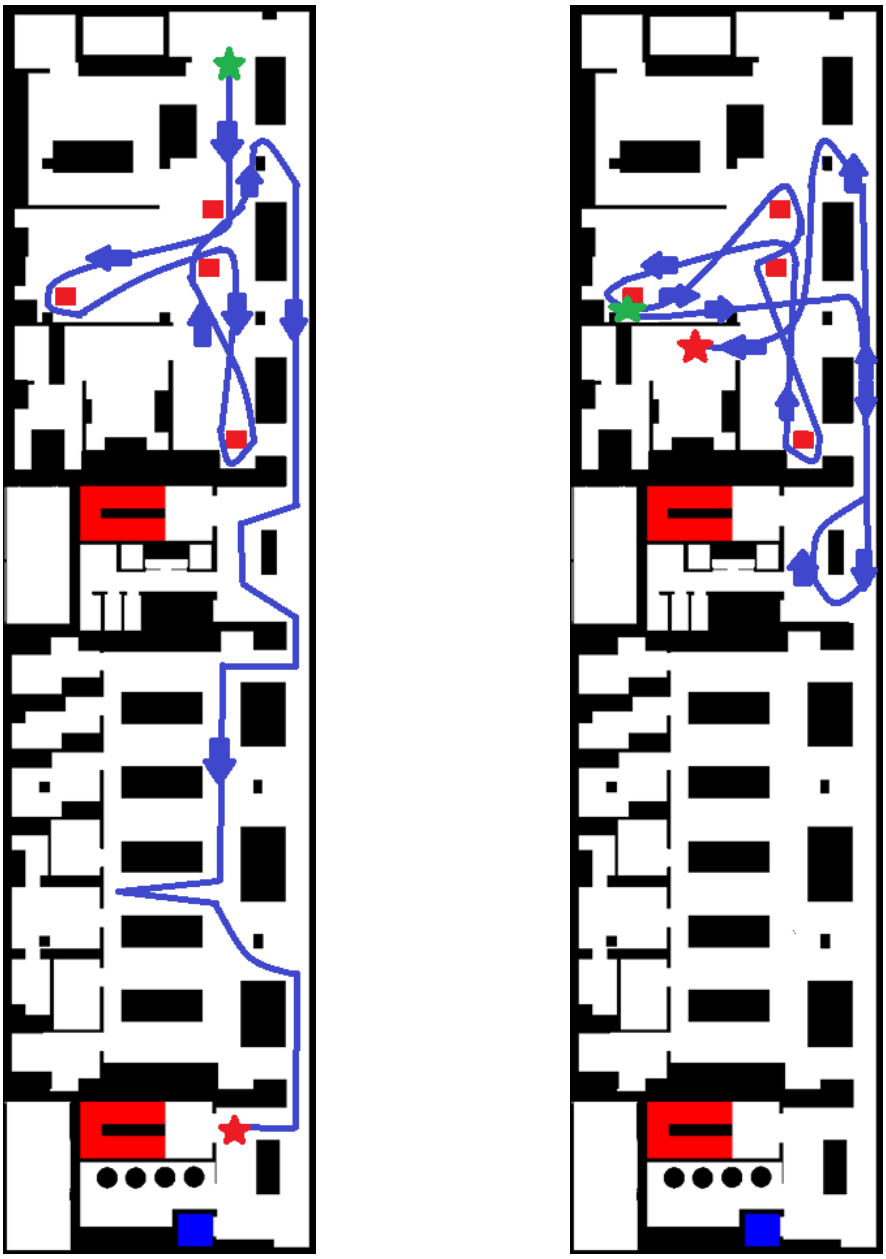
route **open** the initial path is the top one. The lower path, leading to the corridor is travelled on the second time the starting point is passed.

At last, route **zigzag** is a highly constrained route full of turns designed to test the efficiency and robustness of the MM module. It is presented on Figure 7.3 and is 100 meters long.



(a) 8-route-A (b) 8-route-B

Figure 7.1: "Eight-like" Testing Routes



(a) **open** route (b) **loop** route

Figure 7.2: Open-Space Testing Routes

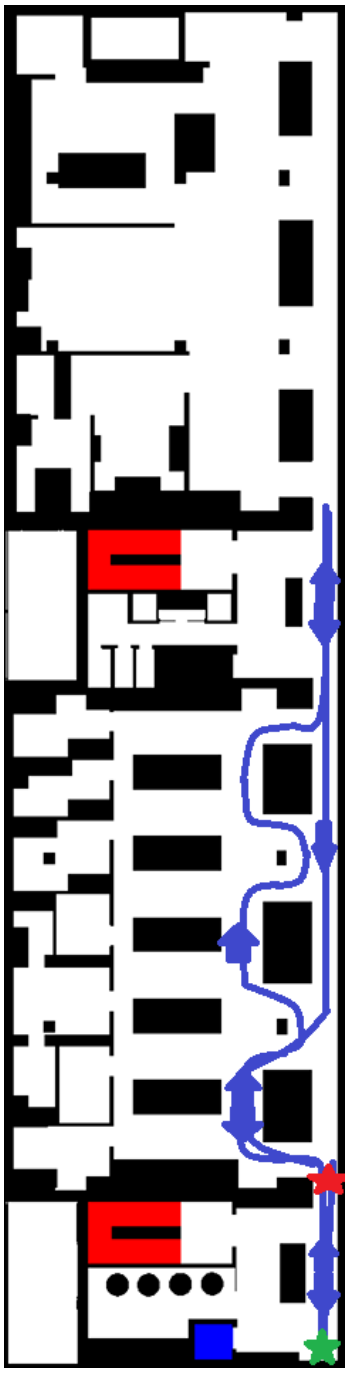


Figure 7.3: **Zigzag** route

.

7.2 Step Detection

Data for step detection testing was retrieved from all 5 users. The results are presented in Table 7.1.

7.2.1 Results

Table 7.1: Step/ZUPT Detection Results

Subject	Nr. of Trials	Average Nr. of Steps	Average Step Error	Max Error
1	9	175	0%	0 steps
2	4	177	1.83%	5 steps/2.81%
3	4	182	0.83%	2 steps/1.19%
4	3	175	0%	0 steps
5	3	164	10.38%(R) / 0%(L)	20 steps/22.6%(R)

7.2.2 Discussion

All users performed trials for at least routes **open**, **loop** and **8-route-B**. A key need for the success of the deployed algorithm is adequate setup of the IMUs beneath the shoelaces. Users 1 and 4 had the most robust mounting of the IMU, which enabled perfect detection of steps. On the other hand, user 5, which had much looser sneakers, probably had an imperfect fit on the Right IMU. Still, and as the PDR and Complete System results will present, it had a manageable impact on the accuracy of the system. Incorrect ZUPT detection happens because a looser fit may result in interrupted ZUPTs as the sensor moves within the shoelaces after stronger impacts even after the foot is stopped. Despite this, the resulting steps (note that a step for the system is all periods between two consecutive ZUPT) will result in very small displacement (under 10cm) and as such do not introduce large errors. There were also two other conducted tests (one for subject 2 and another for subject 3) where imperfect fixation (actual perceptible relative movement) of the IMU caused errors of 8.64 (14 steps) and 14.75% (27 steps) respectively (which were on the **zigzag** and **8-route-B** routes), which were not included in Table 7.1 but were included in the PDR and Complete System results as the algorithm was capable of correcting those errors. We can conclude that route shape has no impact on step detection.

In short, a good fixation of the IMU is crucial. This is not always possible without outside straps or a smaller sensor array, as some shoes cannot accommodate the Pandlet Sensor Array where the IMU is included. Despite the fact that the estimates are very precise for subjects with well-mounted IMUs, and as mentioned in Chapter 3, the detection can be made more robust.

Even though the module provided good results for different users (which naturally have different gait speed) provided the IMU was well fixed, additional tests should be done regarding the impact of gait speed on step detection.

7.3 PDR - Step Quantification + Information Fusion

This Section presents and discusses the results obtained for PDR with the fusion of information from the two IMUs without using the floor map information. The results are presented in two Tables: Table 7.2 sorts the results by route while Table 7.3 sorts the results per subject. Subject 4, as is visible in Table-7.1, participated in three trials. However, despite providing stable results for the step detection, the amount of bluetooth noise along the trial routes coupled with the fact that during those trials magnetometer signals were also being retrieved (despite not being used for processing), resulted in a lack of signal to maintain a stable connection. The resulting slightly corrupted signal while not enough to corrupt the signal step detection (based on magnitude of the signal), seriously corrupted the heading and step length estimates of two of the trials. The results for the remaining trial (on route **loop**) are included in the first table, but were left out of the second.

7.3.1 Results

Table 7.2: PDR-only results by route

Route	**Nr. of Trials**	**Avg. Measured Distance (m)**	**True Distance (m)**	**Avg. PDR Error (m)**
open	5	117.4	120.8	3.73
loop	4	105.5	103.2	1.88
8-route-A	4	114.9	116.5	1.87
8-route-B	5	112	117.9	6.11
zigzag	5	98.44	100	1.79

Table 7.3: PDR-only results by subject

Subject	**Nr. of Trials**	**Avg. Distance Error (%)**	**Avg. PDR Error (%)**	**Max Distance Error**	**Max PDR Error**
1	9	-0.93	2.12	-3.4m / -2.44%	5.28m / -4.56%
2	5	-2.33	1.60	-4.5m / -4.50%	2.37m / -2.30%
3	5	-2.14	3.64	-10.4 / -8.82% *	11.3m / -9.58% *
5	3	-1.22	3.81	-5.3m / -5.14%	5.81m / -4.93%

7.3.2 Discussion

The asterisk on the table is due to the fact that the maximum error was obtained from one of the trials that had wildly inaccurate step detection due to loosening of the IMU. Even though the trial from subject 2 where the same thing happened had a very low PDR error, these errors could only be corrected by the Map Matching Module, as shown in the following Section. Without this trial, the maximum distance error for subject 3 is 3.3m / -3.20% with a PDR error of 3.52m / -2.91%. As the results show, even though users 2, 3 and 5 have trials where step detection was significantly different from the measured number of steps, sensor fusion is capable of achieving good results as a IMU-only solution. Still, there are some important things to comment. All distance estimates were done through the floor map, which means that the ground truth may not be absolutely correct. Nevertheless, there seems to be a slight underestimation of the traveled distance

with the exception of the highly irregular loop route. This may be due to the Gaussian Filter algorithm, which could benefit from distance compensation. Furthermore, the average distance error by subject is decreased by the fact that there are trials with underestimates and overestimates of distance, which does not happen for the average PDR error, as distance is always positive. Despite this, for the trials where step detection was most accurate, the distance estimates are satisfactory. Additionally, the distance error does not correlate with PDR error, both for cyclic and non-cyclic routes.

These PDR results are very interesting and make it seem that if sufficiently robust MM and drift estimation is accomplished, PDR can remain stable for increasingly long routes.

7.4 Complete System

The final positioning results obtained by processing with the complete system are presented in Table 7.4

7.4.1 Results

Table 7.4: Results of the Map Matched PDR system

Route	Nr. of Trials	Avg. Final Error(m)	Avg. Final Error(%)	Max Error
open	5	1.31	1.09	1.45m
loop	4	1.93	1.87	2.77m
8-route-A	4	1.70	1.44	1.62m
8-route-B	5	1.34	1.15	2.14m
zigzag	5	0.75	0.75	1.02m

7.4.2 Discussion

The accuracy of the complete system is very interesting, both for highly constrained and more open routes. It is important to mention that some of the errors in the **open** and **loop** routes were caused by incorrect matches on the map for the final turn, namely the one presented on Figure 7.4 and an incorrect estimate of entering the room at the end of route **loop**. There were trials in which the system error was slightly greater than the PDR error, which reflects the need for improvement of this module. Nevertheless, trials in which the step detection, distance estimation and PDR error were very inaccurate, managed to return good results.

As discussed in Chapter 5, the system is not yet ready for sufficiently robust recovery in case of errors and could have trouble with readjusting some of the routes, depending on the path that would be taken next.

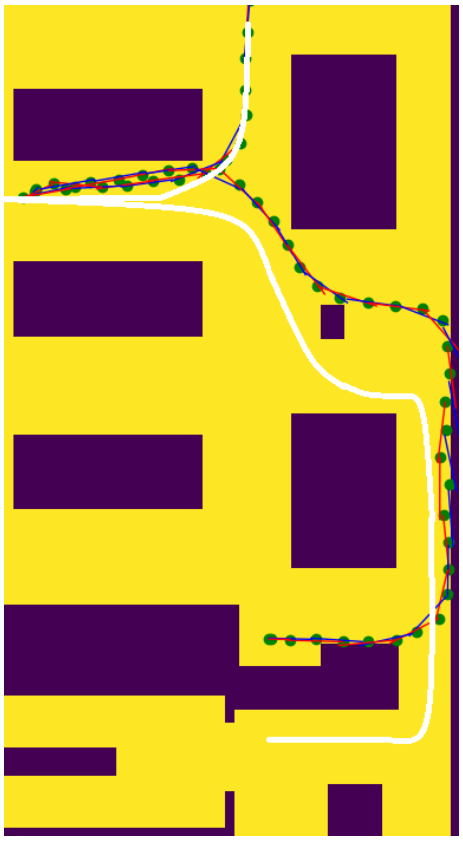


Figure 7.4: Example of incorrect Map Matching - The incorrect matching on the pillar, where the true trajectory is below it and not above, caused matching troubles on the following corridor and an incorrect match on the left turn. On this example PDR error is inferior to the final system error. True trajectory is represented by the white curve

.

7.5 Final Remarks and Further Testing

Despite the fact that the map matching algorithm is still not robust enough to work in every circumstance, three 10 minute walks were done. Trial 1, with a completely random trajectory, managed to keep positioning error under two meters for approximately 2 minutes and 175m until the map matching module introduced an error from which it could not recover. Trial 2, in the same conditions, kept location error under two meters for 5 minutes and nearly 500m, while eventually recovering from a maximum error of 15m to a final error of 6m (despite being in the wrong room of the building, the estimated heading was correct). It is important to note that in circumstances where the geometric matching algorithm returns significantly different estimates for the location of each foot consecutively, the algorithm is not yet capable of settling on a stable position, which means that unless subsequent corrections point to the same position, the system no longer works. That is what happened on both these trials and robustness regarding these uncertain scenarios is undoubtedly the first point of emphasis for future work towards the generalization of the applicability of the system. Finally, a cyclic route (similar to both 8-routes) compatible with

the current strengths of the system was tried out for a 10 minute walk. The system had a final location error of 1.5m while never accumulating an error over 3 meters. This trial is depicted on Figure 7.5.

It is worth mentioning that step detection was 100% accurate for all 3 longer trials, of 741, 860 and 1041 steps (perfectly fixed IMU and carried out by subject 1) and that the cause of failure of the first two trials was simply the inability of the map matching module to converge on a position correction.

Figure 7.5: 10 minute walk example for a route similar to the 8-routes

.

Further testing is necessary, in order to better tune the parameters of the attitude estimation module and to better understand the precision of the velocity reconstruction. As increased robustness is added to the MM module longer routes should be designed, with the goal of achieving stable tracking of routes longer than 1000 meters.

Chapter 8

Conclusion and Remarks

8.1 Conclusions

Designing an Indoor Localization solution based solely on inertial sensors requires great accuracy in trajectory estimation. Since IMUs, especially low-cost units, are not sufficiently precise and suffer from significant bias drift, the fusion of multiple sources of information is necessary. The results of this work show that using multiple IMUs (one on each foot) and the floor plan information can provide sufficiently accurate results (under 2 meters).

Even though estimating foot attitude during gait from a 6-DOF IMU is very challenging and results in an indefinite result per definition (no reference for the yaw angle), leveraging ZARU and smart frequency filtering allows the system to obtain stable representations of attitude even without applying heuristic approaches to drift elimination. Furthermore, the applied attitude estimation filter does not have a prohibitive computational cost for mobile applications.

Step quantification through the use of ZUPT provided satisfactory distance estimates (error around 2% of traveled distance). Despite its simplicity, the implemented ZUPT-detection algorithm enables accurate detection of ZUPT intervals, provided proper fixation of the IMU unit.

The key to drift reduction is the fusion of information from both IMUs. The proposed Gaussian Filter algorithm returns position estimates that are better than the estimates from each individual IMU. It is capable of explicit drift estimation and is robust enough to erroneous inputs that sporadic incorrect segmentation and quantification of steps is easily compensated for. The errors of the PDR system before map-matching are not much greater than the distance estimation errors, even though heading drift is the primary source of error. As such, even if there are large portions of the trajectory in open spaces, the error accumulation should not be too great for a robust map matching algorithm to correct.

Finally, the proposed map matching framework based on the generation of truncated Gaussian distributions through sampling merges well with the Gaussian Filter algorithm for IMU fusion, further improving its drift estimation and compensation capabilities and making Dead Reckoning stable for long walks provided the building has enough constraints for the position to be periodically corrected. At the same time, the geometric matching algorithm proves to be a smart

approach, as looking for openings in the currently estimated direction of walking makes great use of the natural constraints of buildings, such as doors and corridors (that are often the only way to leave an area in a certain direction). Nevertheless, this algorithm still needs refinement, and for a system to depend solely on DR stability (or on trajectory generation), a shape matching algorithm capable of correcting large errors (provided an initial estimate of position exists) could be crucial for a stable solution. In fact, if shape matching is robust enough, and due to the accuracy of the trajectory estimation, complete Indoor Localization (i.e., localization from unknown starting position) could eventually be possible, by matching a sufficiently long trajectory to a sufficiently constrained map.

In sum, the developed system shows that an inertial sensor based Pedestrian Dead Reckoning approach can be accurate and stable enough to function as a standalone Indoor Localization solution. Moreover, merging wireless signal localization techniques such as Wi-Fi or magnetic fingerprinting could provide very valuable absolute localization information while using the inertial info as a source of accurate interpolation of position between signal readings.

Such a system provides an alternative for localization in places deprived of ubiquitous localization signals (such as GPS) and enables fast deployment without the need for fingerprinting of wireless signals (Wi-Fi or magnetic disturbances), instead requiring only the floor plan, which is of a much more static nature than the wireless signals and can be more readily obtained (through image analysis techniques, for instance). This, coupled with the use of low-cost MEMS sensors, means the cost of the system is much lower. Among the benefits a stable IL solution provides, this approach is especially interesting for users in dangerous activities such as rescue missions (where no fingerprinting is possible) and for navigation in large areas such as airports or shopping malls where the floor plans are increasingly available in the Google Indoor Maps platform.

8.2 Limitations, Future Research and Final Remarks

Besides the fact that the MM module still needs to be implemented in Android for a true near-real time online system, there are still some limitations of the current algorithms and some further work that could improve the accuracy and robustness of the system. The limitations and possible improvements on the algorithms and modules that constitute the current system are summarized in Table 8.1.

Parallel to the improvement of the current system, additional design choices or modules could improve the accuracy of the system:

- **Mangetometer Data** - Using magnetometer data to provide an absolute heading reference while exploiting the fact that there are IMUs in two different locations not only to filter out disturbances more easily (avoiding the use of incorrect heading readings that would cause the yaw angle to fluctuate) but also to possibly retrieve information regarding the relative location of the feet according to the detected disturbances. The latter could refine the GF estimate of the relative position of the feet by matching the patterns of the magnetic signals retrieved on the two feet in areas with disturbances.

- **Multiple IMUs** - Using multiple IMUs on each foot would improve the attitude estimate of each foot and the precision of the retrieved signals, as the zero mean and non-zero mean errors would be independent. Additionally, it would allow for much more reliable bias drift estimation. As a consequence the accuracy of the system would naturally improve. This is a trade-off between cost and accuracy.
- **Wireless signal fingerprinting** - As mentioned before, the PDR-only system can be extended by fingerprinting wireless signals (namely through detection of Wi-Fi access points) if it is advantageous for a specific application.

Table 8.1: Overview of limitations and future work

Algorithm / Module	**Limitations**	**Possible Improvements**
Prefiltering	Current cut-off frequency of filters is static and can remove representative frequencies for certain types of gait.	Study the relationship between gait speed (or other important features) and frequency content of the inertial signal to implement dynamic cut-off frequencies.
Attitude Estimation	Increasing precision of z-axis alignment introduces numerical noise in the yaw estimate.	Dynamically set the weight of the accelerometer estimate of attitude (gravity direction) according to its uncertainty (amount of movement induced noise in the signal) and/or improve the filtering of the acceleration signal in order to improve the attitude estimate.
Step Tracker	Thresholds of ZUPT-detection are static and can have trouble detecting the exact sample where the ZUPT-interval should begin; Detection of ZUPT-intervals (steps) sometimes causes the commitment of errouneous small additional steps; System is not ideally robust to errors of the ZUPT-detector.	Dynamically adjust the ZUPT-detection intervals, refine the classification of samples at the boundaries, experiment with other techniques such as FSM or Bayesian inference frameworks; Increase the robustness of the step estimates, especially in cases where multiple steps are detected consecutively on the same foot.
Step Quantification	Reconstruction of displacement is not ideally precise.	Implement a more sophisticated smoothing algorithm for increased precision of the velocity and displacement signal reconsturction.
Gaussian Filter	Step Uncertainty Model is not a perfect representation of the actual step uncertainty; Step Length reduction due to corrections is not compensated at any time.	Improve the Step Uncertainty Model through correlation of displacement in both directions (x and y) and robust propagation and bounding of the resulting distributions. Compensate distance loss that is estimated not to be completely caused by distance measurement errors (instead being caused by heading errors)
Map Matching	Covariance matrix of position estimates can be modeled by the MM algorithm, increasing uncertainty when invalid movements are detected; Sampling algorithm should enable faster acquisition of viable samples in scenarios where obstacles exist; Geometric MM is not sufficiently robust and has trouble in specific scenarios, specifically in collisions at angles that do not correspond to leaving an area (such as turning around near a wall when error has accumulated) and when corrected estimates for consecutive foot-strikes return very different results	Model uncertainty of estimates when irregular steps are detected, exploiting the map constraints: Increase uncertainty when a significant area of the distribution overlaps with an obstacle and use a sampling algorithm with smart heuristics for efficient computation of more informative truncated Gaussian distributions. Make system robust to scenarios when the correction of consecutive steps is different; Create a confidence metric for the position correction; Use a register of the uncorrected trajectory to correct large errors through a shape matching algorithm.

Bibliography

[1] R. Mautz, "Indoor Positioning Technologies," *Institute of Geodesy and Photogrammetry, Department of Civil, Environmental and Geomatic Engineering, ETH Zurich*, p. 127, 2012.

[2] L. Mainetti, L. Patrono, and I. Sergi, "A survey on indoor positioning systems," *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 111–120, 2014.

[3] B. Viel and M. Asplund, "Why is fingerprint-based indoor localization still so hard?," *2014 IEEE International Conference on Pervasive Computing and Communication Workshops, PERCOM WORKSHOPS 2014*, pp. 443–448, 2014.

[4] J.-g. Park, "Indoor Localization using Place and Motion Signatures," *PhD draft 2013*, pp. 1–150, 2013.

[5] G. Ligorio and A. M. Sabatini, "Dealing with magnetic disturbances in human motion capture: A survey of techniques," *Micromachines*, vol. 7, no. 3, 2016.

[6] P. Guo, H. Qiu, Y. Yang, and Z. Ren, "The soft iron and hard iron calibration method using extended kalman filter for attitude and heading reference system," *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 1167–1174, 2008.

[7] R. Feliz, E. Zalama, and J. Gómez, "Pedestrian tracking using inertial sensors," *Journal of Physical Agents*, vol. 3, 2009.

[8] S. Kwanmuang, "FILTERING AND TRACKING FOR A PEDESTRIAN DEAD-RECKONING SYSTEM," *PhD Thesis, University of Michigan*, 2015.

[9] J. Borenstein and L. Ojeda, "Heuristic Drift Elimination for Personnel Tracking Systems," *Journal of Navigation*, vol. 63, no. 04, pp. 591–606, 2010.

[10] P. Aggarwal, D. Thomas, L. Ojeda, and J. Borenstein, "Map matching and heuristic elimination of gyro drift for personal navigation systems in GPS-denied conditions," *Measurement Science and Technology*, vol. 22, no. 2, p. 025205, 2011.

[11] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," *IEEE International Conference on Rehabilitation Robotics*, pp. 179–185, 2011.

[12] A. Lawrence, *Gyro and Accelerometer Errors and Their Consequences*, pp. 25–42. Springer New York, 1998.

[13] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs," *Sensors (Switzerland)*, 2015.

[14] M. Ahmadi, A. Khayatian, and P. Karimaghaee, "Attitude estimation by divided difference filter in quaternion space," *Acta Astronautica*, 2012.

[15] M. Nørgaard, N. K. Poulsen, and O. Ravn, "New developments in state estimation for non-linear systems," *Automatica*, vol. 36, no. 11, pp. 1627–1638, 2000.

[16] S. O. H. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," *Technical Report*, 2010.

[17] E. K. Antonsson and R. W. Mann, "The frequency content of gait," *Journal of Biomechanics*, 1985.

[18] R. K. B. Edgar Charry, Daniel T.H. Lai, "A study on band-pass filtering for calculating foot displacements from accelerometer and gyroscope sensors," *Conf Proc IEEE Eng Med Biol Soc. 2009*, 2006.

[19] W. Z. Wang, Y. W. Guo, B. Y. Huang, G. R. Zhao, B. Q. Liu, and L. Wang, "Analysis of filtering methods for 3D acceleration signals in body sensor network," in *Proceedings of 2011 International Symposium on Bioelectronics and Bioinformatics, ISBB 2011*, 2011.

[20] J. F. Kaiser, "Nonrecursive digital filter design using the i0-sinh window function," *Proceedings of the 1974 IEEE International Symposium on Circuits and Systems.*, pp. 22–23, 1974.

[21] S. Mitra, *Digital Signal Processing*. McGraw-Hill, 1998.

[22] H. Guo, M. Uradzinski, H. Yin, and M. Yu, "Indoor positioning based on foot-mounted IMU," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 2015.

[23] I. Skog, P. Händel, J.-O. Nilsson, and J. Rantakokko, "Zero-velocity detection — an algorithm evaluation.," *IEEE transactions on bio-medical engineering*, vol. 57, no. 11, pp. 2657–2666, 2010.

[24] S. Wan and E. Foxlin, "Improved Pedestrian Navigation Based on Drift- Reduced MEMS IMU Chip," *Time*, vol. 180, no. 3, pp. 220–229, 2007.

[25] J. Perry and B. J., *Gait analysis: Normal and pathological function*. Thorofare: Slack, 2010.

[26] J. Ruppelt, N. Kronenwett, and G. F. Trommer, "A Novel Finite State Machine Based Step Detection Technique for Pedestrian Navigation Systems," *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 13–16, 2015.

[27] K. Abdulrahim, T. Moore, C. Hide, and C. Hill, "Understanding the Performance of Zero Velocity Updates in MEMS-based Pedestrian Navigation," *International Journal of Advancements in Technology*, vol. 5, no. 2, pp. 53–60, 2014.

[28] D. Simon Colomar, J. O. Nilsson, and P. Handel, "Smoothing for ZUPT-aided INSs," *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings*, no. November, pp. 13–15, 2012.

[29] Q. Cai, G. Yang, N. Song, J. Pan, and Y. Liu, "An online smoothing method based on reverse navigation for zupt-aided inss," *Journal of Navigation*, vol. 70, no. 2, p. 342–358, 2017.

[30] D. S. Colomar, "Step-wise smoothing of ZUPT-aided INS," *Masters' Degree Project, KTH Electrical Engineering*, p. 49, 2012.

[31] S. K. Park and Y. S. Suh, "A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems.," *Sensors (Basel, Switzerland)*, 2010.

[32] Z. Xu, J. Wei, B. Zhang, and W. Yang, "A robust method to detect zero velocity for improved 3d personal navigation using inertial sensors," *Sensors (Switzerland)*, vol. 15, no. 4, pp. 7708–7727, 2015.

[33] J. Ruppelt, N. Kronenwett, G. Scholz, and G. F. Trommer, "High-precision and robust indoor localization based on foot-mounted inertial sensors," in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium, PLANS 2016*, 2016.

[34] P. H. Truong, J. Lee, A. R. Kwon, and G. M. Jeong, "Stride counting in human walking and walking distance estimation using insole sensors," *Sensors (Switzerland)*, 2016.

[35] W. Chen, R. Chen, Y. Chen, H. Kuusniemi, and J. Wang, "An effective pedestrian dead reckoning algorithm using a unified heading error model," *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 340–347, 2010.

[36] M. Opiela, "Grid-based indoor localization using smartphones," *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 4–7, 2016.

[37] M. Ma, Q. Song, Y.-h. Li, Y. Gu, and Z.-m. Zhou, "A Heading Error Estimation Approach based on Improved Quasi-static Magnetic Field Detection," *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016.

[38] R. Zhi, "ScholarWorks @ UVM A Drift Eliminated Attitude & Position Estimation Algorithm In 3D," *Graduate College Dissertations and Theses. Paper 450.*

[39] H. J. Ju, M. S. Lee, C. G. Park, S. Lee, and S. Park, "Advanced heuristic drift elimination for indoor pedestrian navigation," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 729–732, Oct 2014.

[40] A. R. Jiménez, F. Seco, F. Zampella, J. C. Prieto, and J. Guevara, "Improved heuristic drift elimination (ihde) for pedestrian navigation in complex buildings," in *2011 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–8, Sept 2011.

[41] D. Gusenbauer, C. Isert, and J. Krösche, "Self-contained indoor positioning on off-the-shelf mobile devices," *2010 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2010 - Conference Proceedings*, pp. 15–17, 2010.

[42] M. Vasconcelos, H. Gamboa, and D. Elias, "A Motion Tracking Solution for Indoor Localization Using Smartphones," *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 4–7, 2016.

[43] J. O. Nilsson, A. K. Gupta, and P. Handel, "Foot-mounted inertial navigation made easy," *IPIN 2014 - 2014 International Conference on Indoor Positioning and Indoor Navigation*, pp. 24–29, 2014.

[44] J. B. Bancroft and G. Lachapelle, "Data fusion algorithms for multiple inertial measurement units," *Sensors*, vol. 11, no. 7, pp. 6771–6798, 2011.

[45] H. Lan, C. Yu, Y. Zhuang, Y. Li, and N. El-Sheimy, "A novel Kalman filter with state constraint approach for the integration of multiple pedestrian navigation systems," *Micromachines*, vol. 6, no. 7, pp. 926–952, 2015.

[46] W. Shi, Y. Wang, and Y. Wu, "Dual MIMU pedestrian navigation by inequality constraint kalman filtering," *Sensors (Switzerland)*, vol. 17, no. 2, 2017.

[47] I. Skog, J. O. Nilsson, D. Zachariah, and P. Handel, "Fusing the information from two navigation systems using an upper bound on their maximum spatial separation," *2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012 - Conference Proceedings*, pp. 13–15, 2012.

[48] G. V. Prateek, R. Girisha, K. V. S. Hari, and P. Handel, "Data fusion of dual foot-mounted INS to reduce the systematic heading drift," *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS*, pp. 208–213, 2013.

[49] A. Lyon, "Why are normal distributions normal?," *British Journal for the Philosophy of Science*, vol. 65, no. 3, pp. 621–649, 2014.

[50] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 5, pp. 312–328, 2007.

[51] E. Sakic, M. Number, and P. Kranz, "Map-Matching Algorithms for Android Applications," *Vmi.Lmt.Ei.Tum.De*, 2012.

[52] S. Funke and S. Storandt, "Path Shapes – An Alternative Method for Map Matching and Fully Autonomous Self-Localization," *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '11*, p. 319, 2011.

[53] K. C. Lan and W. Y. Shih, "Using smart-phones and floor plans for indoor location tracking," *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 2, pp. 211–221, 2014.

[54] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering Similar Multidimensional Trajectories," *ICDE '02 Proceedings of the 18th International Conference on Data Engineering Page 673*.

[55] J. H. Jiang, H. L. Le, and S. C. Shie, "Lightweight topological-based map matching for indoor navigation," *Proceedings - IEEE 30th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2016*, pp. 908–913, 2016.

[56] I. Spassov and I. SPASSO, "Algorithms for map-aided autonomous indoor pedestrian positioning and navigation," *EPFL Thesis, Lausanne*, vol. 3961, pp. 1–139, 2007.

[57] S. Har-peled and B. Raichel, "The fréchet distance revisited and extended," *ACM Transactions on Algorithms (TALG)*, vol. 10, no. 1, pp. 1–22, 2014.

[58] D. Whiter, L. Latecki, Jan, R. Lakämper, and X. Sun, "Shape-Based Robot Mapping," *27th German Conf. on Artificial Intelligence, Ulm, Germany*, vol. 27, p. 439, 2004.

[59] M. Pelka and H. Hellbrück, "Introduction, Discussion and Evaluation of Recursive Bayesian Filters for Linear and Nonlinear Filtering Problems in Indoor Localization," *The Seventh International Conference on Indoor Positioning and Indoor Navigation*, 2016.

[60] Y. Lu, D. Wei, Q. Lai, W. Li, and H. Yuan, "A Context-Recognition-Aided PDR Localization Method Based on the Hidden Markov Model," *Sensors*, 2016.

[61] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," *ACM Conference on Ubiquitous Computing (UbiComp)*, pp. 421–430, 2012.

[62] S. Beauregard, W. Widyawan, and M. Klepal, "Indoor pdr performance enhancement using minimal map information and particle filters," *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 141–147, 2008.

[63] P. Davidson, J. Collin, and J. Takala, "Application of particle filters for indoor positioning using floor plans," *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*, pp. 1–4, 2010.

[64] J. Qian, L. Pei, J. Ma, R. Ying, and P. Liu, "Vector graph assisted pedestrian dead reckoning using an unconstrained smartphone," *Sensors (Switzerland)*, vol. 15, no. 3, pp. 5032–5057, 2015.

[65] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear / Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[66] K. Zoubert-ousseni and C. Villien, "Comparison of Post-processing Algorithms for Indoor Navigation Trajectories," *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016.

[67] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theor.*, vol. 13, no. 2, pp. 260–269, 2006.

[68] J. Lafferty and A. Mccallum, "Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data Conditional Random Fields : Probabilistic Models for Segmenting and," *Proceeding ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning Pages 282-289*, vol. 2001, no. June, pp. 282–289, 2001.

[69] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Lightweight map matching for indoor localisation using conditional random fields," *IPSN 2014 - Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (Part of CPS Week)*, pp. 131–142, 2014.

[70] S. Bataineh, A. Bahillo, L. E. Diez, E. Onieva, and I. Bataineh, "Conditional random field-based offline map matching for indoor environments," *Sensors (Switzerland)*, vol. 16, no. 8, 2016.

[71] S. Bataineh, A. Bahillo, and L. E. Díez, "Enhancing Conditional Random Field-based Map Matching with Behavioral Information," *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 4–7, 2016.

[72] M. Levitt, "Protein folding by restrained energy minimization and molecular dynamics," *Journal of Molecular Biology*, vol. 170, no. 3, pp. 723 – 764, 1983.

[73] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[74] E. W. Dijkstra, "A Note on Two Probles in Connexion with Graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[75] R. Kan and C. Robotti, "On Moments of Folded and Truncated Multivariate Normal Distributions," *Journal of Computational and Graphical Statistics*, 2017.