Position Tracking during Human Motions using an Integrated Wearable Sensing System

2016

Giulio Zizzo

A thesis submitted to the University of Manchester for the
degree of Master of Philosophy
in the Faculty of Engineering and Physical Sciences.

School of Mechanical, Aerospace and Civil Engineering

Contents

Word Count: 34918

List of Figures

List of Tables

Abstract

Position Tracking during Human Motions using an Integrated Wearable Sensing System
The University of Manchester
Giulio Zizzo
Master of Philosophy
July 28, 2016

Fixed motion tracking systems can offer highly accurate data but several drawbacks are present, including a high upfront cost and require the user to stay within a very limited area. Of keen interest are shoe mounted systems which aim to offer a similar suite of information but are unconstrained in their operating environment. The potential of knowing the user's foot placement and orientation is an extremely valuable set of information. This data can be used in a wide range of applications such as healthcare monitoring, emergency responder localisation, and lower limb prosthetic stability and control.

This thesis investigates the potential of using low cost (∼£30) inertial measurement units (IMUs) to track a user's motion and position. When using an IMU, general purpose strap-down navigation is shown to give inadequate results after only seconds of use. Thus, to provide corrections, an Extended Kalman filter (EKF) is used to provide zero velocity and heuristic drift reduction updates. This system is shown to have typical loop closure errors of $\leq$1% with maximum errors of 4-5%.

In parallel with the IMU an ultrasound (US) trilateration system calculates the displacement of each step and the results of the IMU and US systems are combined. This addition gave slight improvements in the results, typically reducing the cumulative error over a walk by 15%.

Lastly, a particle filter can impose movement constraints on the predicted motion by including environmental information. In combination with the previous two sensing systems the addition of a particle filter gave consistent errors of $\leq$1%.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

1. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

2. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

3. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

Publications

Giulio Zizzo and Lei Ren. Foot position tracking during human motions using an integrated wearable sensing system. *International Conference of Bionic Engineering*, June 2016

Pending:

Giulio Zizzo and Lei Ren. Position tracking in Indoor Environments with Wearable Sensors. *IEEE Sensors*

Chapter 1

Introduction

The integration of computers into society affects our daily lives on almost every level. Seamlessly blending into the background, computers aid us in a multitude of tasks. This is part of the paradigm of ubiquitous computing, in which technology is present all around us in every object and location. Already, the widespread diffusion of smart phones can be seen as a manifestation of this trend. It is set to carry on expanding, completely changing jobs, industries and daily life [1].

With computers diminishing in size and cost, one form of ubiquitous computing is wearable technology. With products such as the Google Glass, Apple SmartWatch, and VR headsets wearable technology is becoming a rapidly emerging market.

1.1 Background

An area of wearable technology which is receiving increasing attention is pedestrian tracking. Shoe mounted tracking systems have undergone significant advancements in recent years benefiting from increasingly accurate, compact, and low cost sensors such as accelerometers, gyroscopes, magnetometers, and ultrasound. Coupled with this are more powerful microprocessors that can perform the required data analysis on the fly to provide real time gait information. Unlike fixed motion labs, a shoe mounted system is unconstrained in its operating environment and offers a wide range of applications from gait monitoring in healthcare, prosthetic limb stability, augmented reality and, most predominately, pedestrian tracking.

Effective pedestrian navigation can be employed in a variety of situations from museum tours [2] to emergency responder navigation [3]. It has been undertaken using beacons such as radio frequency identification, wideband ultrasound [4] and Wi-Fi signal strength [5]. However, what these systems offer in accuracy and reliability is offset by their cost and extensive pre-installed infrastructure. This is a major obstacle preventing their more widespread use. On the other end of the spectrum are foot mounted inertial measurement units (IMUs) consisting of a 3-axis accelerometer and 3-axis gyroscope. They offer total freedom of movement and work without reliance on external infrastructure. This has attracted significant attention in terms of tracking pedestrian motion in any potential environment.

IMUs can track the changes in orientation of an object and hence, project the accelerations measured by the accelerometers into a global navigation frame enabling the object's position to be obtained. IMUs have, in theory, the ability to track the position of an object with almost no limitations. In reality however, unbound error growth occurs in the estimated position arising from inaccuracies in the measured acceleration and angular velocity. Such errors are referred to as drift. Despite these issues IMUs are widely used as tracking sensors, either with periodic corrections of the position via an external source, such as global navigation satellite systems (GNSSs), or by having such large and accurate IMUs, such as those found on submarines, that positional accuracy can be maintained over prolonged use.

It is only with the development of micro-electrical-mechanical systems (MEMS) that IMUs have become small enough to be used as wearable sensors. Although their accuracy continues to improve, they do not yet have the levels of accuracy possessed by the large mechanical sensors used in submarines or aircraft. Despite this, their durability due to the lack of moving parts, coupled with their small size, makes them a staple of many motion tracking systems.

Although being significantly cheaper than pre-intalled motion tracking sensors, a barrier to IMU tracking in everyday use is the cost associated with a sensor of a high enough quality. Commonly used IMUs for motion tracking, such as Xsens sensors, have a price of over £1000 [6]. On the other hand, low cost IMUs that are found in mobile phones cost several orders of magnitude less, but suffer from large amounts of error accumulation.

Therefore, a sensor fusion approach is considered. If by using several different types of low cost sensors, such as ultrasound, position sensitive devices, and IMUs then the drawbacks of each individual method can be mitigated by the efficiency of another sensor, and so effective pedestrian motion tracking can be performed.

The objective of this thesis is to develop a shoe mounted system that can measure the user's walking characteristics and by extension track a user's movement indoors. From this we can define the following aims to address in this thesis.

1.2 Aims

1. Devise a shoe mounted system that can track a user's walking using primarily IMUs, supplemented with additional data from other sensors such as ultrasound, position sensing, and magnetometers. The walking will be at a user selected speed and examined on flat indoor terrain.

2. Should map information be available then this information will be combined with the sensor data to improve tracking accuracy.

3. The system should be built with the aim of trying to achieve similar performance the results obtained in the in literature using Xsens IMUs but at a drastically reduced cost.

To reach the aims above the following objectives are considered:

1.3 Objectives

1. **IMU Performance:** The IMU will be used in a standalone fashion to track the user's footsteps. Via an Extended Kalman filter (EKF) the

user's step length, width and heading are computed. Its performance for tracking in a global co-ordinate frame is then examined against a fixed motion capture system. When a fixed motion tracking system is unavailable the wearable system will be analysed by having the user perform a closed walk, i.e. the walk ends when the user has returned to their starting point. Any misclosure errors given by the wearable system can be used to analyse its accuracy.

2. **Ultrasound Sensor Fusion:** To improve on the data gathered with the IMU, ultrasound sensors are mounted on the foot which measure the step length and width. This is done by using multiple receivers which enable a trilateration algorithm to be applied to find the step parameters. By combining this with the IMU system more accurate information is synthesised. Similarly, the effect of the ultrasound sensors will be evaluated against the results from a Vicon motion capture system and in terms of loop misclosure.

3. **Imposing Map Constraints:** Lastly, the effect of having a known map of the environment is considered. The map can be used to impose movement constraints, and as such, correct for the predicted motion. For example, should the system predict the user has stepped though a wall then that motion can be taken as an error and a better estimate of the position calculated. This will be done by implementing a particle filter to represent the user's position, and by modelling the building constraints via a 2D polygon.

1.4 Research Questions

When developing the system the following research questions were considered.

1. **What performance can be achieved by using low cost IMUs?** Numerous motion tracking systems have been developed that use high cost MEMS, such as the Xsens IMU. The monetary cost of these sensors

can be prohibitive for widespread use and so the potential of cheaper IMUs is considered.

2. **Can multiple low cost sensors compensate for each others' weaknesses and match the performance of similar systems using state of the art MEMS?** If by using multiple different sensors the overall cost of the system could be dramatically reduced and hence, widen the use of motion tracking in everyday situations enormously.

3. **To what extent is prior map knowledge required to provide accurate tracking?** The more accurate and detailed a map of the environment is the more constraints can be imposed upon the user's motion, and therefore the larger the error reduction in the user's position. However, having an extremely detailed map is often impractical and so a balance between accuracy and prior knowledge must be found.

1.5 Thesis Overview

The remainder of this thesis is organised as follows

Chapter 2: Details on existing and related work on pedestrian tracking are examined. The mathematical techniques used in foot tracking are discussed, which leads onto the technologies and algorithms used.

Chapter 3: Tracking using an IMU is explored. The IMU sensor is examined in particular with regards to its error characteristics. An EKF is applied to correct for errors caused by the IMU's noisy data. Results obtained using an IMU in conjunction with an EKF are presented.

Chapter 4: The system discussed in Chapter 3 is developed further to include a series of ultrasound sensors mounted onto the shoes. The ultrasound sensors give the user's step length and width and thus can be combined with the IMU data to give a more accurate result.

Chapter 5: Prior knowledge of the environment is utilised to provide movement constraints. A particle filter takes in data from the IMU/ultrasound hybrid system and combines it with known environmental constraints to track the user within a known map.

Chapter 6: We revisit the aims stated at the beginning of the thesis and evaluate our results against them. The main contributions of this thesis are summarised and additional routes for further development are proposed.

Chapter 2

Literature Review

A wide body of literature encompasses motion tracking. The measurement of real time foot placement and stride data is used for both researching human walking as well as pedestrian tracking and localisation.

There are two broad approaches to motion tracking. The “gold standard” approach is to use a motion laboratory to conduct a full analysis of the motion of body segments. Motion laboratories typically use highly accurate computer based force and optical tracking sensors. This provides high quality data offering complete gait analysis. The drawbacks however are significant, as aside from the costs of acquiring and maintaining the equipment, the small capture volume limits the distance travelled and is an inherent handicap.

In contrast wearable lightweight sensors, predominately IMUs, can be used to perform motion tracking in a wide range of unrestricted environments. They can provide information on multiple parameters involved in human walking and have numerous applications. However, a price must be paid for this freedom and IMUs are prone to high levels of uncertainty.

It is on this second category of IMU tracking systems which we will focus on. Due to the noisy measurements obtained from wearable sensors, research is currently focused on developing algorithms and techniques to minimise the errors that occur.

2.1 A Probabilistic Approach

When dealing with motion tracking and navigation, uncertainty dominates. What is required is a way to deal with the errors arising from imperfect measurements. We naturally turn to probability theory to represent, and deal with, noisy measurements. To begin with, we examine Bayesian filters which form the basis for more sophisticated treatment.

2.1.1 Bayesian Filters

When conducting motion tracking a wide range of methods are available. Inertial sensors, ultrasound, and infra-red are all examples of sensors which can be used to give relevant information. The reason why so many technologies operating on completely different physical principles have been developed is simple: not one of them is a silver bullet which satisfies all the needs of a ubiquitous motion tracking sensor [7].

With each individual sensor possessing an inherent flaw, Bayesian filters provide a framework in which to combine measurements to provide a more accurate result. Bayesian filters maintain a probability distribution, which means that at each stage we have an estimate for a particular parameter, but also the uncertainty associated with it. As measurements from sensors are received, Bayesian filters take into account this new measurement along with its uncertainty, to update our belief of the parameter.

What is of widespread use in pedestrian tracking is not so much the ability of Bayesian filters to combine different sensor readings, but to be able to apply constraints to the system. For example, should a user know the maximum value of a parameter, such as speed, height, or acceleration, such a constraint can be applied via pseudo-measurements. Pseudo-measurements are virtual measurements which can enforce such prior knowledge of a system.

A Bayesian filter provides this framework for updating the state of a system based on noisy measurements by representing the state at step k as a probability distribution, $Bel(x_k)$, conditioned on all previous measurements $z_{1:k}$,

$$Bel(x_k) = p(x_k|z_1, ..., z_k). \tag{2.1}$$

This is referred to as a posterior distribution and, as can be expected, with an increasing amount of sensor measurements the complexity of calculating the posterior grows exponentially [8]. To be able to calculate the posterior we thus assume that the system's state at step k depends only on the state of the system at $k-1$. This is known as a Markov assumption.

From this, it follows that the $Bel(x_k)$ is obtained from the previous distribution $Bel(x_{k-1})$. The Bayes filter algorithm computes $Bel(x_k)$ in two separate steps, a prediction and measurement update stage. Firstly the prediction step,

$$Bel^{-}(x_k) = \int p(x_k|x_{k-1})\, Bel(x_{k-1}) dx_{k-1} \tag{2.2}$$

computes the prior distribution $Bel^{-}(x_k)$. This prior distribution is then updated with new measurements to give the posterior distribution,

$$Bel(x_k) = \eta\, p(z_k|x_k)\, Bel^{-}(x_k) \tag{2.3}$$

where η is a normalisation factor. In other words, to compute the posterior the Bayes filter multiplies the belief $Bel^{-}(x_k)$ by the probability that measurement z_k occurs [9].

We can see that Baysian filters provide a framework for recursively estimating the state of a system based on new information. Of key importance is that it does not specify how the belief $Bel(x_k)$ is represented. Due to this, how $Bel(x_k)$ is chosen leads to different assumptions, computational costs and accuracy.

Kalman Filters

Kalman filters are some of the most widely used implementation of Bayesian filters. It is part of a wider family of Gaussian filters which make the assumption that the prior belief, $Bel(x_k)$, follows a Gaussian distribution. From this $Bel(x_k)$ can be represented by the mean and covariance, known as moments of parametrisation.

R.E Kalman published his ground breaking work in 1960 [10]. The Kalman filter is a recursive algorithm to estimate the state of a system and has very powerful features, offering estimation of past, current and future states even when noise is present.

The Kalman filter predicts the state of a system, x_k, which evolves according to the linear equation:

$$x_k = Ax_{k-1} + Bu_k + w_k. \tag{2.4}$$

In equation 2.4, A is the transition matrix and describes how the system at step $k - 1$ relates to k in the absence of input or process noise. The matrix B states how the control input, u_k, influences the future state. Lastly, w_k is the process noise of the system [11]. It should be noted that equation 2.4 represents the general form of a system's time evolution. In the specific implementation of the Kalman filter used later in this work, both the matrix, B, and control input, u_k, are removed as we are not introducing a control action in our system.

When a measurement, z_k, is made it is related to the state of the system through the matrix H via:

$$z_k = Hx_k + v_k \tag{2.5}$$

where v_k is the measurement noise. In essence, the matrix H determines which states of the system have been measured by z_k.

It is then possible to obtain two sets of equations: prediction and correction equations. Prediction equations forecast the future state and error covariance to calculate a priori predictions for the future step. Correction equations then use new measurement data to acquire a more accurate posteriori estimate. This cycle is shown in Figure 2.1

The most costly operations when running the Kalman filter are the matrix multiplications required. These can be implemented in $\mathcal{O}(N^2)$ time where N is the number of states in the system [12]. This can become computationally expensive in certain scenarios. For example, in robotic localisation if the robot has to track the location of several hundred landmarks, all in real time, then the computational delay can be detrimental. In the case of tracking human motion,

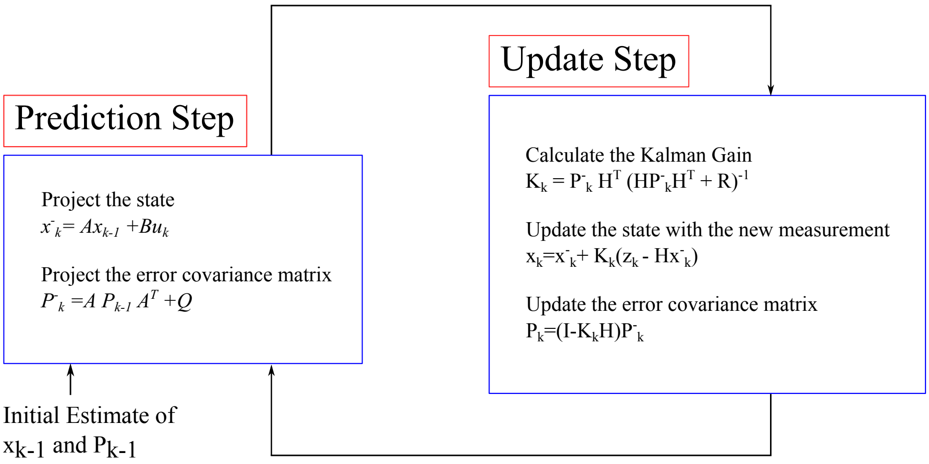

Prediction Step
Project the state
$x^{-}_{k} = Ax_{k-1} + Bu_k$
Project the error covariance matrix
$P^{-}_{k} = A P_{k-1} A^T + Q$
Update Step
Calculate the Kalman Gain
$K_k = P^{-}_{k} H^T (HP^{-}_{k}H^T + R)^{-1}$
Update the state with the new measurement
$x_k = x^{-}_{k} + K_k(z_k - Hx^{-}_{k})$
Update the error covariance matrix
$P_k = (I - K_kH)P^{-}_{k}$
Initial Estimate of
x_{k-1} and P_{k-1}

Figure 2.1: Figure illustrating the Kalman filter equations. The prediction step advances the state and the associated error covariance matrix P_k. The update step takes new information into account and computes a more accurate state estimate. The matrices Q and R represent the process and measurement noise of the system.

the Kalman filters used have relatively few states and so the computation can be done extremely quickly.

One of the main limitations of the Kalman filter is the numerous assumptions made about the system noise. For the Kalman filter to function the noises must be independent, white, Gaussian, and with zero-mean. Obviously these assumptions do not always hold true.

Additionally, the Kalman filter only works on linear systems. In practice we are often dealing with non-linear systems which have the effect of not preserving the Gaussian nature of our variables. To deal with this problem of non-linearity the Extended Kalman filter (EKF) is used. It performs a local linearisation of the system around the current best estimate of the state [9]. To demonstrate these features consider Figure 2.2. We can see how a local linearisation of the function preserves the Gaussian distribution.

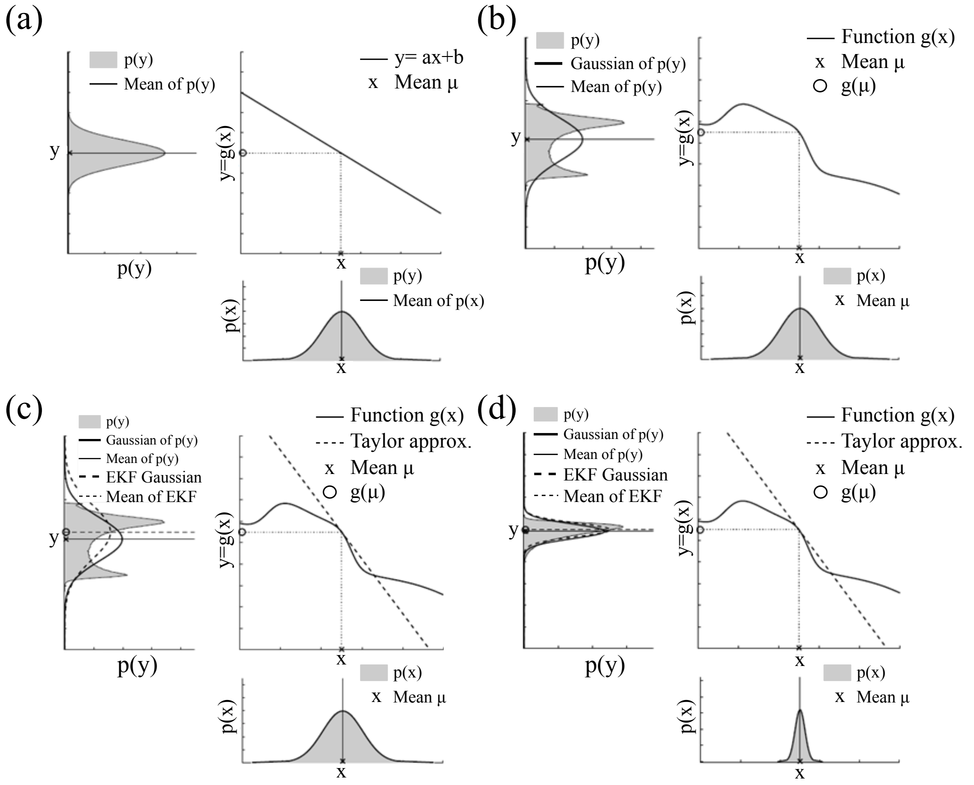


Figure 2.2: (a) shows that by passing our Gaussian variable through a linear function the output remains a Gaussian. (b) illustrates how the passing a Gaussian through a non-linear function will result in a non Gaussian output. (c) demonstrates that by performing a local linearisation we can preserve the Gaussian distribution. (d) shows that the tighter the Gaussian is the more accurate the linearisation will be [9]. Real systems are rarely linear and so this issue of non-linear systems is present in many applications including our own. The relevant equations for IMU based tracking have several non-linear terms, and therefore an EKF is required.

For moderately non-linear systems where the Gaussians are relatively compact the EKF provides an effective approximation. It is the EKF algorithm with its additional flexibility that has been successfully implemented in a wide variety of works concerning shoe mounted gait analysis and pedestrian dead reckoning systems [13],[14],[15],[16].

2.1.2 Kalman Filters in Pedestrian Dead Reckoning

Pedestrian dead reckoning (PDR) refers to tracking the relative movement of a pedestrian. A novel approach of using Kalman filters to perform PDR was conducted by Foxlin [17] who developed a shoe mounted system named NavShoe. Its aim was to track the location of a person in GPS denied environments and, when GPS was available, combine the odometry readings from the NavShoe with the GPS. An innovative advance was performed by incorporating zero-velocity updates (ZUPTs) into the algorithms to reduce the accumulated errors.

The principle behind ZUPTs is that a person has a stationary stance phase and a moving stride phase while walking. By making the approximation that the foot's velocity is zero during the stance phase, then any measured velocity must be due to errors arising from the IMU. Therefore any measured velocity can be incorporated as a pseudo-measurement into a EKF to provide error corrections.

Exploiting the cyclical nature of gait allows ZUPTs to be periodically supplied, which enables the EKF to provide corrections at every step. Whereas traditional approaches involving integration of the acceleration and angular velocities yield a error growth proportional to t^3, incorporating ZUPTs reduces the error growth, from being cubic in time, to one that is related to the number of steps taken.

This has advantages over simply setting the velocity to zero in every stance phase. Primarily ZUPTs allows the EKF to estimate the accumulated errors in several system parameters. These errors, once determined can be suitably corrected. The EKF can be used to correct the drift in position as well as errors in the velocity, gyroscope and acceleration biases, and orientation.

Despite their predominance some authors have questioned the validity of ZUPTs, in particular [18] points to the modelling errors introduced when using ZUPTs. Firstly, there is the question of the foot not being completely still and having speeds in the order of a few mm/s in the stance phase. From this the step length is underestimated as parts of the gait cycle are declared stationary and the related motion is ignored. [18] makes the point that the modelling errors introduced will mean that inertial navigation systems utilizing ZUPTS have an inherent limitation due to the assumptions made about the foot's velocity. Therefore, the accuracy of ZUPT aided tracking systems will not be improved beyond a certain point by relying on ZUPTs.

Indeed, the fact that the foot is not truly stationary in the stance phase has been examined in other works. [19] shows that none of the tested points on the foot and shank had zero velocity at any time during the stance. It was seen that for the foot the average minimum velocity was under $0.011\,\mathrm{ms}^{-1}$. Additionally, stride length underestimation was seen of up to $0.7\,\%$ when using foot mounted IMU navigation system.

However, the advantages that ZUPTs offer outweigh the modelling errors introduced and so this method has been widely used in multiple research groups [13],[14],[20],[21],[22],[23].

Early work is being conducted to apply alternative filter formulations to navigation and tracking systems. The potential use of a Cubature Kalman filter to directly filter the position, velocity and attitude, as opposed to their errors, was investigated in [24]. However, the lack of vigorous experimental validation means the effectiveness of the Cubature Kalman filter is yet to be properly determined.

Although the principle of ZUPTs has been used in an extremely wide range of works, several stand out for introducing new methodologies or approaching pedestrian navigation in an innovative way.

[13] presented a system which integrated several techniques into one pedestrian dead reckoning system. Aside from using ZUPTs, they incorporated zero angular rate updates (ZARUs). ZARUs are the angular equivalent to ZUPTs. When the foot is on the floor the measured angular velocity should be zero. Hence, any measured angular velocity can be taken as an error and fed into a EKF to provide corrections. ZARUs have been criticised by [25]. The authors pointed out that the foot can rotate while being on the floor and the effectiveness of ZARU rests on that assumption: if the user does not rotate their foot it can provide a strong error reduction technique analogous to ZUPT. However, should that assumption not hold true then applying ZARU will introduce significant modelling errors.

In addition to ZARUs [13] used heuristic drift reduction (HDR), discussed in more detail in 2.2.1, in combination with a magnetometer to reduce the drift in yaw. The culmination of all these techniques was a system that was able to track the movement of the user with errors of 0.3-1.5% of the total distance travelled.

In [26] the authors examined how by using two IMUs on each foot could yield more accurate results. The principle the authors were exploiting is simple to conceptualise: at all times the IMUs cannot be separated by more than a maximum distance. The authors formulated a ZUPT aided Kalman filter, which by incorporating a maximum bound on the foot to foot separation, gave improved results.

This work was continued in [27], where a commercially implementable motion tracking system, named OpenShoe, was presented. The modules and code are fully open source and can be found at: http://www.openshoe.org/. The OpenShoe system functioned to a very high level of accuracy, accumulating errors of 0.5% of the total distance travelled. The authors did not make use of heuristics or magnetometers, hence with additional sources of information it is feasible to reduce the error further.

2.1.3 Stance Detection

To be able to apply the EKF corrections a system must be able to determine if the foot is in a stance phase. Foxlin's approach was to measure the gyroscope and accelerometer signals, and should they exceed particular thresholds then the foot is declared to be in motion [17].

A similar methodology was taken by [25]. To determine when a stance phase occurs the angular velocity readings from a shoe mounted IMU were examined. In ideal conditions the angular velocity in a stance phase would be 0 rad/sec. In practice a threshold value was set to account for instrumentation errors of up to 1 rad/sec. The signal was filtered to remove small fluctuations in the angular velocity when it momentarily dropped below 1 rad/sec. Once the stance and stride phases are identified error correction methods can be applied.

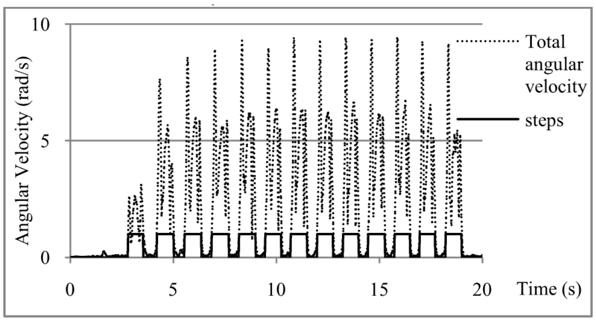


Figure 2.3: Angular velocity with a threshold of 1 rad/sec in combination with a filter being used to determine stance phases [25].

An extremely robust algorithm was implemented by [13] in which 3 separate conditions drawing on the acceleration, the acceleration variance and gyroscope data are considered before declaring that the foot is stationary. Firstly, the magnitude of the acceleration must fall between 9 m/s^2 and 11 m/s^2 for the first condition signalling a stationary foot to be fulfilled. Secondly, the local acceleration variance, which highlights foot activity, must be below 3 m/s^2. Lastly, the magnitude of the gyroscope readings must be below a threshold of 50 deg/sec. Once all three conditions are satisfied a stance phase is declared as is shown in Figure 2.4. This method has also been used in [14] for applying ZUPT.

An alternative method is to use force sensors to detect when the foot is on the ground. [28] examined the angular velocities of the shank and thigh. By using four force sensitive resistors (FSR) placed underneath the foot the authors declared that the foot was on the ground once the measured force exceeded certain thresholds. The authors used this information to reset the inclination angle of the shank and thigh to 0° in the mid-stance phase as at that point the shank and thigh segments are in their vertical positions.

Similarly [29] used two force sensors placed at the bottom of each foot to detect heel-strike and toe-off stances. [29] highlights the importance of the placement of the force sensors on the sole. They found a positive correlation between cadence and positional error. In other words, a higher stepping rate led to a larger displacement error. This was due to the force sensors being

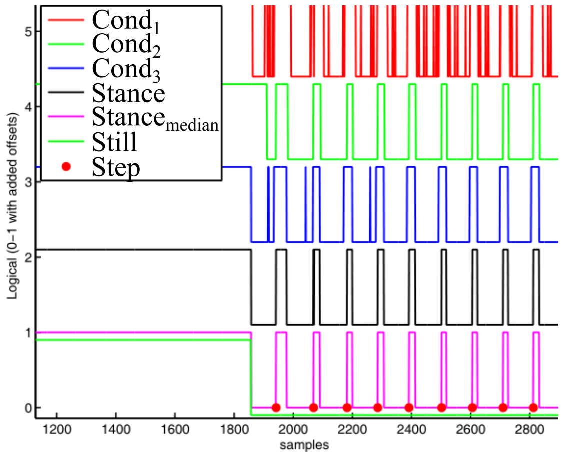


Figure 2.4: Stance detection using three separate conditions. $Cond_1$ represents acceleration magnitude, a, lying between $9\,m/s^2 < a < 11\,m/s^2$. $Cond_2$ is when the acceleration variance is below $3\,\mathrm{m}/s^2$. $Cond_3$ is when the magnitude of the gyroscope is below $50°s^{-1}$. These all represent logical conditions which when true have a value of 1. The *Stance* phase corresponds to when the foot is stationary on the floor. The *Still* phase is for a non-walking stationary foot. It is detected when multiple stance samples occur back to back in a two second or larger window [13].

installed on the toe, hence the system did not detect heel strike until the toe had hit the ground. This is exacerbated at higher cadence.

Likewise in [30] FSRs were used to determine heel-strike and toe-off in human gait using a shoe mounted system. Interestingly, [30] found that the calibration for the FSR was highly non-linear which could lead to erroneous force measurements without accurate calibration. Additionally, uncertainties may arise due to the fact that after prolonged use the adhesive layer in FSR breaks down contributing to an increased non-linearity. This will introduce additional errors in detecting the heel strike and toe off stances.

In particular, a very thorough detection system was mathematically formalised by [31], in contrast to the ad-hoc nature of previous studies. This stance detection problem was expressed mathematically as a binary hypothesis:

$$\begin{aligned} &H_0 : \text{IMU is moving} \\ &H_1 : \text{IMU is stationary} \end{aligned} \tag{2.6}$$

and the detector's performance is characterised by the false alarm probability, $P_{FA} = P(H_1|H_0)$ and the detection probability $P_D = P(H_1|H_1)$. The authors aimed to maximise P_D.

To achieve this the authors developed models for both the sensor and signals in order to develop a detection framework based on maximum likelihood estimation. The new detector used both gyroscope and accelerometer signals and is referred to as the SHOE detection scheme. To declare the foot as being stationary it combines both accelerometer and gyroscope data, and via a threshold determines if the IMU is stationary.

Three different types of detector were derived in the context of maximum likelihood estimation and use either the acceleration variance, the acceleration magnitude or the angular rate energy and, along with the SHOE detector, were analysed against each other.

Comparing the results of the detection schemes against the data from force sensitive resistors mounted on the shoe, the authors saw that the SHOE and angular rate energy detectors performed similarly with the acceleration variance and acceleration magnitude detectors giving worse results. The authors pointed out that this would imply that the gyroscope signals have more reliable information for accurately determining a user's stance phase.

2.2 Heading

One of the most challenging parameters to accurately determine is the heading, or yaw, of a user. ZUPTs, although offering very powerful error reduction techniques, still lead to a drift in the yaw angle. There have been two approaches to limiting the drift in yaw, using magnetometers and a technique named heuristic drift reduction.

2.2.1 Heuristic Drift Reduction

Heuristic drift reduction (HDR) was originally proposed by [21]. It uses the principle that many environments which humans walk in, such as corridors and paths, are straight. The HDR algorithm seeks to determine if a person is walking in a straight line and, if so, apply a correction to the heading. It was implemented in [21] by creating a closed loop control system. Via the use of a binary I-controller the yaw drift is tracked and corrected for. This correction is performed at the end of every step. With the incorporation of a low pass filter the HDR algorithm is efficient at removing swaying- i.e. motion that is intended to be straight but due to the nature of walking contains fluctuations. However curving, or the motion along an extended arc, causes the most problems to the algorithm. This is because a continuous turn in a large radius will interpreted as a drift and therefore be mistakenly corrected.

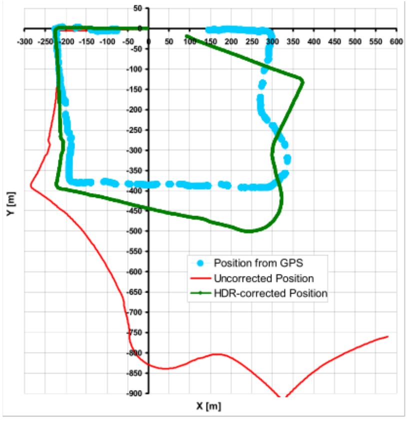


Figure 2.5: Example results as shown in [21]. As we can see the HDR algorithm reduces the drift in yaw.

An alternative implementation was undertaken by [13]. They calculated the orientation change between successive steps and if it was below a particular value then the EKF was fed with a value to correct the heading error.

Additionally, HDR has been performed differently by [14]. In that work a sliding window comprising of λ samples was used to compare the heading at time k to $k - \lambda$. If this difference in heading is less than a given threshold then it is detected as an error and fed into the EKF. The advantage of using a sliding window approach is that HDR can be performed at every time interval and not solely at the stance phase.

2.2.2 Magnetometers

Magnetometers can be used to obtain a heading with respect to the magnetic north. Outdoors they are very useful as they can provide accurate, drift free headings. Indoors the situation is somewhat different. Large magnetic distortions are present indoors which can significantly skew any obtained heading. With regards specifically to pedestrian motion tracking the majority of the literature suggests that they are not suitable for indoor use, with only very few works, such as [13], having reported an improvement using a magnetometer indoors.

Attempts to compensate for varying magnetic disturbances are being conducted. [32] used an adaptive EKF to determine orientation by combining IMU and magnetometer data. The authors implemented a checking technique to the accelerometer and gyroscope reading before including them in the EKF. If the accelerometer was detected as being in motion then the measurement covariance was set to ∞ and hence the filter relied on the magnetometer for performing the state vector update. Likewise, magnetically perturbed environments the magnetometer readings become very unreliable, so should the sensor data deviate significantly from the earth's local magnetic field then the measurement covariance is set to ∞. However, that research was done in the context of determining the orientations of human body segments, and thus was only experimentally validated in a cubic volume measuring 60 cm in height, width and length. Nonetheless, the authors did find that the maximum errors were under 10° in either the roll, pitch or yaw axes.

By contrast, magnetometers for indoor navigation purposes was examined in [33] and [34]. To gain accurate results the authors attempted to model the effects of soft and hard iron disturbances. The soft iron effects are modelled via the matrix A,

$$A_{si} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{32} & a_{33} \end{bmatrix} . \tag{2.7}$$

The hard iron effects, equivalent to sensor bias, are represented by b_{hi},

$$b_{hi} = \begin{bmatrix} b_{hi_x} & b_{hi_t} & b_{hi_z} \end{bmatrix}^T . \tag{2.8}$$

The full error model relating the error free magnetic field $\hat{h}$ to the measured readings, h, follows

$$\hat{h} = Ah - b + n \tag{2.9}$$

where n is Gaussian wideband noise. The matrix A incorporates scale factors, misalignments and soft iron disturbances while b is the combined bias.

By collecting magnetic field data in several orientations an ellipsoid can be fitted to the data. From the fitted elliptical equation we can extract the parameters A and b of equation 2.9. The authors of [33] and [34] individually calibrated 12 magnetometers and arranged them as shown in Figure 2.6

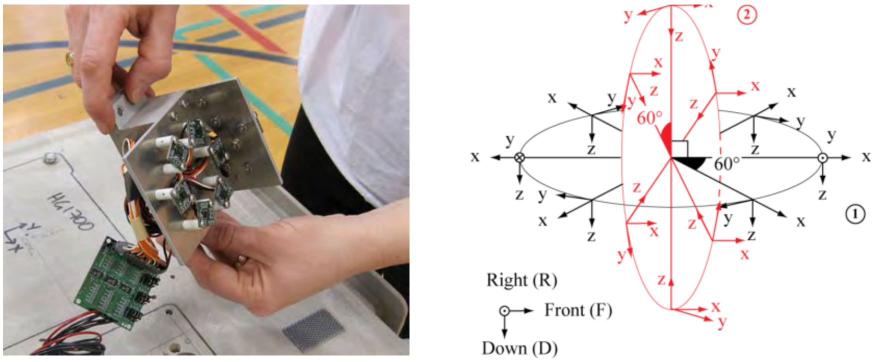


Figure 2.6: Arrangement of magnetometers as used in [33] and [34]. After the magnetometers are calibrated their results are combined via an adaptive least squares method to yield accurate heading information.

By combining the results of the 12 magnetometers the heading was kept stable with a deflection of 4° when an artificial magnetic disturbance was created as is shown in Figure 2.7.

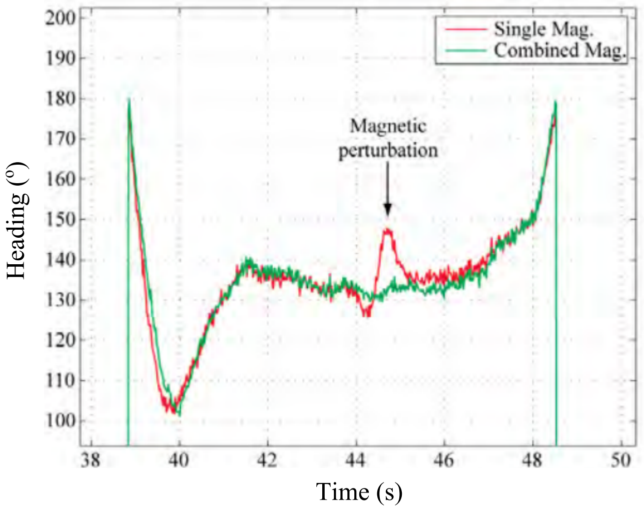


Figure 2.7: Results presented in [33] in which the magnetometer array is passed over an artificial magnetic disturbance. We can see that when functioning together the magnetometers are able to keep an unperturbed heading.

2.3 Height Detection

Although not a focus of intensive investigation, researchers have examined various techniques that determine the vertical distance traversed by a user. IMUs as well as barometer sensors have often been used for this purpose.

One such system which used barometer and IMU data was [25]. The authors performed a test to examine how their shoe mounted system works in a 3D trajectory by climbing up stairs. The total height measured by the barometer information was compared to that given by the IMU. The results given by the IMU are much smoother however, as is always the case with IMU data, it has attached to it an accumulated error. The barometer readings have the advantage of not being subjected to drift, but have a slower response time and are much rougher. This can be seen in Figure 2.8.

A particularly unique approach used was discussed in [30] which made use of a capacitive sensor which detected the height of the foot above the floor via

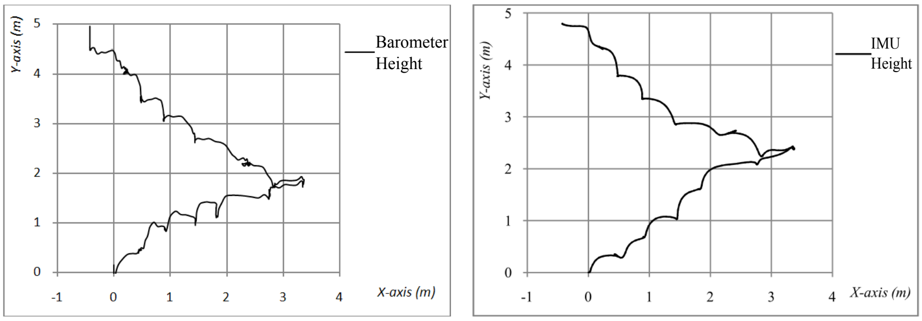


Figure 2.8: Comparison of data gathered while a user climbed a flight of stairs. As we can see the IMU readings (Right) are smoother and distinguish the steps more clearly. However, they are subject to accumulated drift and over several flights of stairs the positional accuracy will gradually deteriorate. On the other hand, the barometer data (Left) is less clear but more reliable over prolonged periods as it does not suffer from accumulated error growth [25].

capacitance loading from the ground. Furthermore, ultrasound sensors were incorporated to measure the range and angle between each foot. Preliminary results collected by the authors in [35] showed potential. However, the system was deemed insufficiently mature to use during validation.

2.4 Particle Filters

A further widely used implementation of the Baysian filter is a family of filters called non-parametric filters. Recall that in the case of a Kalman filter we assumed that the posterior had a Gaussian form, however there can be many cases in which a variable is not Gaussianly distributed.

One approach to account for non-Gaussian distributions is to use the particle filter. The key idea is to represent the posterior by a set of random samples drawn from this distribution. This representation is approximate and becomes increasingly accurate the more samples, or particles, we generate. The main advantage of a particle filter is that it can represent any type of distribution and can model highly non-linear functions without a loss of accuracy.

The set of particles, χ, can be expressed as

$$\chi = x_t^1, x_t^2, ..., x_t^N \tag{2.10}$$

where each particle, x_t^n, represents a hypothesis of the state of the system. For example, if dealing with an object's position each particle represents a potential location of the object.

The principle is that the probability that a particle is included in the set χ is proportional to the posterior. Therefore, the higher the density of particles in a subregion then the more likely that the true state falls within this region.

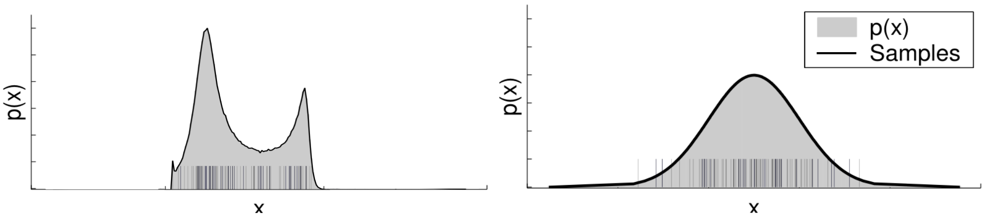


Figure 2.9: (a) Demonstrates how a non-gaussian distribution can be modelled by a series of samples. (b) Shows a Gaussian distribution represented in particle form [9].

Like all other Bayesian filter algorithms, a particle filter recursively estimates the state of a system. It does so in three stages. Firstly, all of the particles are propagated forward according to the system's propagation model. The propagation model defines how the particles behave between successive time steps. In the context of pedestrian tracking the propagation model will receive information from underlying sensors about the size and direction of the step and move the particles by the appropriate distance. Then, for every particle a weighting is calculated and assigned to it. Returning to our example of each particle representing a position, then each particle is weighted representing how likely that a particular sample is the true location of the object.

Lastly, the particles are re-sampled which is when a set of new samples are generated from the weighted particles. Each particle's weight corresponds to the probability of it being re-sampled. This results in samples with a low probability being discarded and samples which we are confident in being multiplied. Informally it can be thought of as a "survival of the fittest", with the good samples replicating and so increasing our confidence that the object is in a particular location while the poor samples are removed.

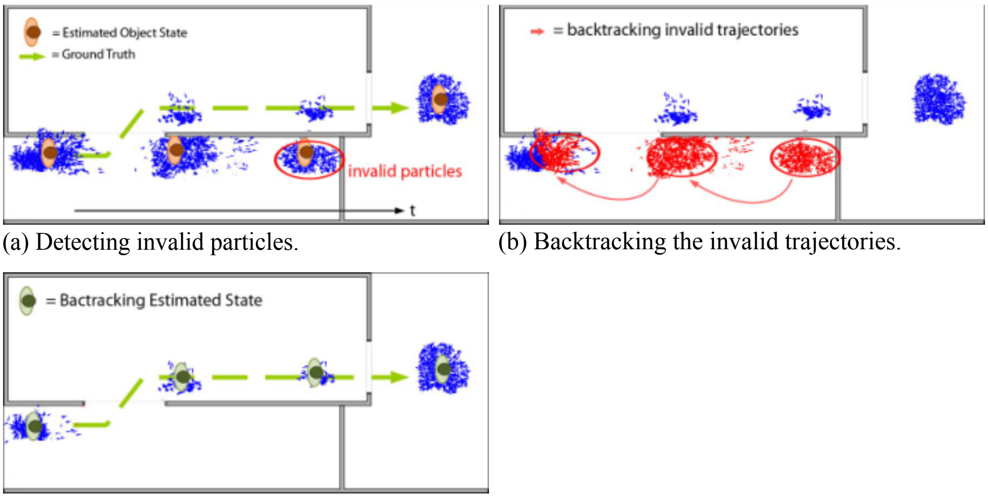
Of course the flexibility of a particle filter comes at a price: it is computationally expensive. For an accurate representation of the posterior a large number of particles need to be generated.

2.4.1 Particle Filters in Tracking

Particle filters are primarily used in tracking to represent the position of a user or robot within a map. Specifically, it can be used to constrain the user to within physically sensible areas. For example [36] implemented a particle filtering technique in combination with an Wireless Local Area Network (WLAN) localisation scheme. The particle filter incorporated knowledge of the environment into the particle movement. It eliminates motions that a human could not do: for example crossing a wall. If a particle violates this constraint then the algorithm attempts to find a new particle position. If following multiple attempts the particles position still does not satisfy the criteria then the weighting is set to zero and the particle is removed.

Provided the building outline is known further limitations can be added. In [37] the authors noted that a good indication as to whether one is indoors is GPS availability. Should GPS be available then the particles that are within the building have their weightings penalized and, vice versa if GPS is unavailable then the particles which are outdoors are handicapped. This has the effect of preventing the particles from splitting into groups, inside and outside of the building and diverging away from each other.

A further extension to particle filters has been the development of Backtracking Particle filters [38]. It refines the state estimates based on the particle trajectory history. It is based on the assumption that an invalid particle, such as one crossing through a wall, is the result of following an invalid trajectory. If a particle is invalid the previous state estimate can be refined by removing the incorrect particle trajectories. Recalculating the projection of particles without the invalid trajectory produces noticeably better estimates [38].


= Estimated Object State
= Ground Truth
invalid particles
t
= backtracking invalid trajectories
= Bactracking Estimated State

(a) Detecting invalid particles.

(b) Backtracking the invalid trajectories.

(c) Backtracking estimated states.

Figure 2.10: A series of diagrams illustrating the Backtracking Particle Filter. The blue points represent particles, i.e. potential locations of the user. The particle's positions are averaged to give the best estimate of the location of the user. (a) shows a common problem when employing particle filters. The posterior density is shown in 4 different time intervals. Via particle filtering the incorrect particles can be discarded. (b) Illustrates how by using Backtracking Particle filter algorithm the incorrect trajectories are removed. (c) demonstrates how with the incorrect trajectories being removed the particles successfully follow the correct path [38].

Exploiting prior knowledge of the map in this way can provide very accurate tracking. The authors in [37] did not create a sophisticated Kalman sensor fusion algorithm, but rather used the native software of the Xsens Mti motion sensor. This software was designed for limb motion capture and so, when subject to the high dynamics of the foot, showed several errors. Nonetheless, over a 10 minute walk the particle filter showed high levels of accuracy. By comparing the location accuracy when using regular pedestrian dead reckoning (PDR) and then combining it with a particle filter (PF) and a backtracking particle filter (BPF), we can see that the additional filtering algorithms provide increased accuracy. It is clear from Table 2.1 that increasing the building map information results in more accurate position estimates, which shows that a balance needs to be struck between highly detailed maps and realistic levels of information which can be easily supplied.

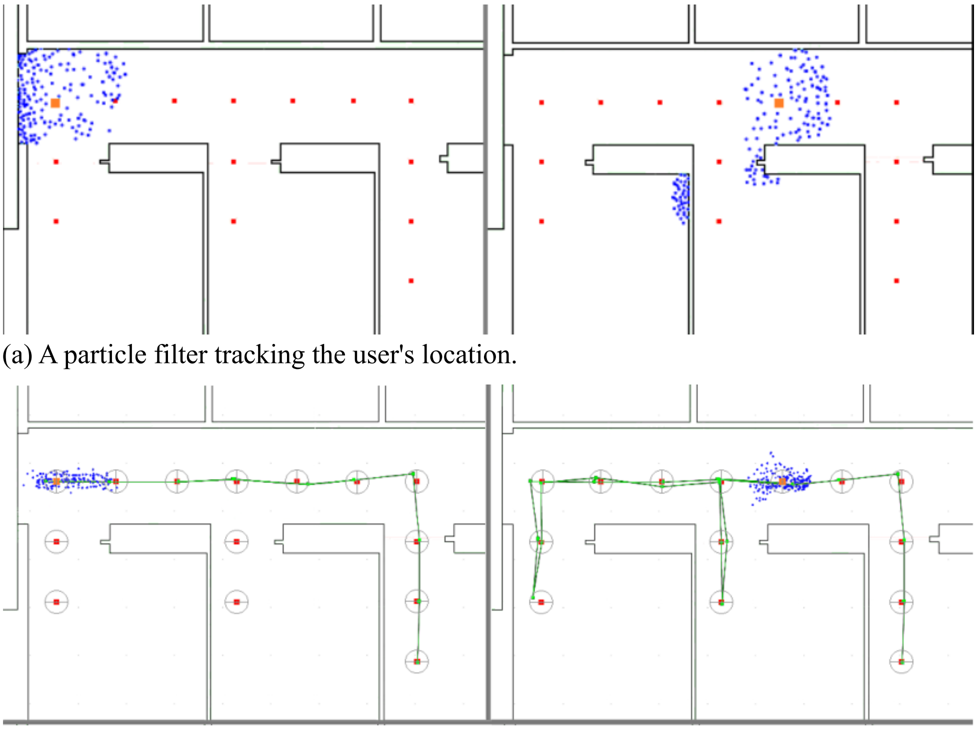
Detail of Map	**PDR (m)**	**PDR+PF (m)**	**PDR+BPF (m)**
External Wall Map	7.738	3.103	2.557
Detailed Wall Map	7.738	1.083	0.7432

Table 2.1: Table showing errors accumulated when using different algorithms [37]. Note however that this was only from a single experiment of a walk measuring ∼330 strides in length. Therefore, more experimental validation would be needed for a generalised result to be drawn as to the exact improvement offered by the BPF.

Ultrasound sensors have also been incorporated into particle filter algorithms to improve their performance. [22] used a very simple motion model in which the user took steps of a constant size and utilised particle filters to localise the user within a building. They incorporated particle filtering, but also used ultrasound range sensors to determine the distance of the walls to the user. With this information the particles could be weighted more appropriately, with particles matching well to the ultrasound range measurements given a high weighting while those with a significant disparity would be given a low one and be discarded. The effect of the ultrasounds can be seen in Figure 2.11.

An advantage of using ultrasound ranging measurements, as opposed to laser based ones, in the context of localising emergency rescue personal is that they are much more robust being able to provide range data in smoke filled rooms and, due to their wider beam angle, are less easily obstructed. Whereas laser based range sensors will offer very high resolution it is, to a certain degree, unnecessary as other errors will dominate.

A novel addition to particle filters was presented in [39]. Here, the system does not begin with the user's location initialised, but rather determines it when the user moves via a particle filter. The innovation is that rather than using a fixed number of particles, their number is scaled based on how clustered they are. Essentially, when the particles are tightly grouped around the



(a) A particle filter tracking the user's location.

(b) Particle filter incorporating ultrasound range measurements.

Figure 2.11: (a) Demonstrates how as the user moves in the building (the left diagram illustrating the earlier time point) the particle filter removes impossible user locations and particles only occupy the corridors and rooms. (b) shows the same walk however, incorporating ultrasound range data makes the particle cloud much tighter and does not diverge as the particle filter algorithm can make use of the new data [22].

true location of the user it is unnecessary to have several thousand particles. Therefore, when the system is unsure of the user's location the number of particles is increased. Vice versa, if the user's location has been determined very accurately their number is drastically reduced. This reduces the computational load significantly while still maintaining equivalent accuracy. The system additionally uses WLAN to contain the user's location to an approximate region of the building, further reducing the amount of particles needed. This system was able to track a user within 0.73 m 95% of the time over 16 minutes of walking.

2.5 Overlap with Robotics

An area of research in which tracking and localisation plays an important role and overlaps in many ways with human motion tracking, is that of robotics.

Robotic localisation within an unknown environment has been described as the “Holy Grail” [40] of building a truly autonomous robot. Several techniques exist to carry out the task of simultaneous localisation and mapping (SLAM) such as FastSLAM [41] and RatSLAM [42]. This is an extremely complicated task and remains an active area of research.

Robotic localisation within a known map has been done with a high degree of success. Robots such as Minerva [43] were used to give tours in museum, an inherently chaotic and ever changing environment. By using a particle filter approach the robot was able to effectively carry out the tours. Using a camera aimed at the ceiling, Minerva was able to accurately localise itself within the main hall and, in conjunction, also used laser range finders to localise itself within the exhibits and avoid collisions with visitors.

This is when differences between pedestrian and robotic localisation begin to arise. With pedestrian tracking it is highly desirable to keep the number and size of sensors as low as possible to make them unobtrusive. Furthermore, humans have few fixed mounting points suitable for the sensors used in robotic localisation. By contrast, robots are generally fitted with a range of sensors such as ultrasound, laser range finders, wheel encoders, and vision systems to give a highly detailed suite of information.

Added onto all this, is that the motion of a human is far more prone to uncertainty, and can be much more complex than that of a robot. For example, consider the range of motion a human can carry out compared to a four wheeled robot and it is immediately clear that accurately tracking a human poses more of a challenge.

A more subtle difference is that a human tracking system cannot exert any control over the motion of the user, whereas a robotic localisation system can issue commands to move the robot in a way that will aid the localisation process. For example, should a robot be unsure between two or more possible locations it can move towards environmental asymmetry or landmarks and

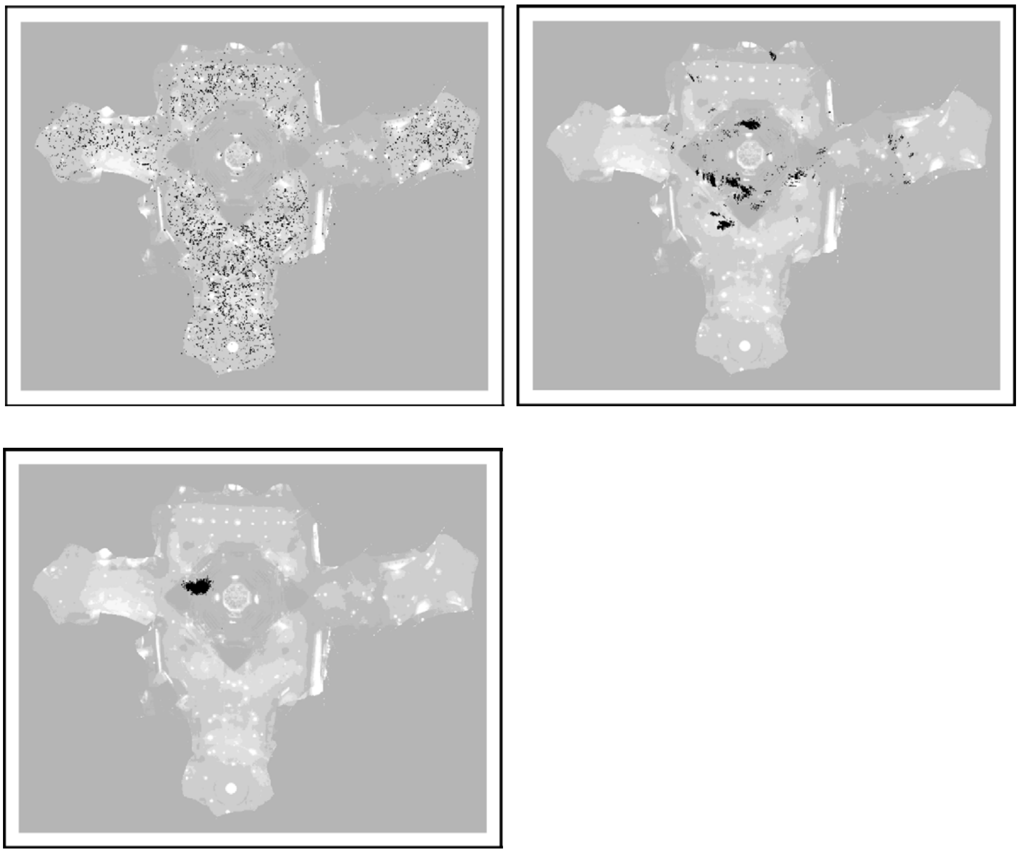


Figure 2.12: A series of illustrations showing how the robot Minerva localises itself by using a particle filter in the museum [43]. Minerva used a camera pointed at the ceiling of the museum to generate the shown maps onto which it can localise itself. The black dots represent particles and, as the robot takes in more information the particles converge to its true location.

thus confirm one hypothesis while eliminating others. This is referred to as active, as opposed to passive, localisation.

However, tracking a user's foot position may soon become more directly applicable to robotics with the increasing prominence of humanoid robots. With [44] real time SLAM was conducted using a visual camera supplemented with inertial data. If the inertial sensor was mounted on the robot's foot and the data processed using zero-velocity updates then more accurate odometry could potentially be achieved.

2.6 Trilateration

Trilateration is a method used to determine the location of an unknown point. Using ranges measured from known points an object's position can be calculated. It is a common tool used in GPS, aeronautics [45], robotics, computer vision, crystallography [46], and kinematics [47]. Conceptually it is a simple problem, the range measurements form circles, or spheres in a three dimensional case, and the object's location is on the intersection point of the circles. The objects location $O(x, y, z)$ is related to the ith beacon position, $B(x_i, y_i, z_i)$, via

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_i^2 \tag{2.11}$$

in which r_i is the range measurement form the i th beacon. Thus, to find $O(x, y, z)$ we must solve $i = 1, 2, 3...n$ simultaneous equations. This solution would be very involved due to the highly non-linear equation produced. The problem can be approached in a variety of fashions; usually the equations are linearised to reduce the problem to that of finding intersection of several planes [48]. Increasingly accurate methods also employ linear as well as non-linear least squares [49].

2.6.1 Wireless Networks

Regardless of the algorithm employed trilateration is used with a variety of sensing technologies. A popular technology is to use Wireless Local Area Network (WLAN). By measuring the received signal strength (RSS) from known access points the distance travelled by the signal can be inferred. The overwhelming advantage of using WLAN to conduct positioning estimates is its prevalence. Virtually every building in developed nations has access to a WLAN. However, the data is very noisy, and numerous factors can influence the RSS. The authors in [5] used a Kalman filter to try and reduce the noise errors. Indeed, they reported a marked increase in accuracy compared to a minimum mean squared error approach reducing the error from 34 m to 6.78 m. Similar work on WLAN trilateration was conducted in [50]. Rather than measuring the RSS they opted for a time-of-arrival (TOA) approach. By measuring the time delay between Request-to-Send (RTS) and Clear-to-Send (CTS) frames of the 802.11 standard, the approximate distance between the user and WLAN access points could be determined. By employing a nonlinear least squares approach and applying Newton iteration the position was found to have an accuracy of between 2 m and 2.3 m in 90% of cases.

2.6.2 Ultrasound Beacons

Although WLAN trilateration is improving in recent years, it still remains too inaccurate for our needs. More accurate range measuring devices are ultrasound sensors. Ultrasound is used very widely as a distance sensor for a multitude of applications, and several innovative approaches for indoor positioning have been undertaken. The potential of smart phones to act as ultrasound transmitters was investigated by [2]. Commercial off the self smartphones are capable of producing ultrasound in the range of 20 - 22 kHz. By placing a series of microphones within a room, and using a least squares technique, the smartphone's position is determined. This study was interesting, as rather than using conventional time of flight which relies on a synchronisation between the transmitter and the receiver, the authors used the difference in time of arrival between the different microphones. This alternative approach was very successful in locating a mobile phone with a typical accuracy of less than 10 cm in ideal operating conditions. However, due to the high directionality of the emitted ultrasound waves and the lack of penetrating power possessed at high frequencies the orientation of the phone played a large role in the accuracy of the system. If the speaker on the phone is facing downwards then the errors can be as large as 70 cm. Likewise, if the phone is held at an angle then the user's body position can obstruct the ultrasound wave, further deteriorating results.

In [51] the use of ultrasound beacons to correct for the drift in position from IMUs was investigated. A ZUPT strategy was used to correct part of the drift and the internal filtering of the Xsens sensor was employed to obtain a rotation matrix which included magnetometer readings. However, as discussed previously, the indoor magnetic field is subject to large fluctuations which the Xsens sensor was not fully able to correct, leading to an erroneous heading estimate. Typically, the authors achieved an error of 7-8% of the distance travelled. In the presence of machinery, and hence large magnetic fluctuations, the tracked position accumulated an even larger error while travelling. Ultrasound beacons were used as a “breadcrumb” trail to periodically provide corrections. The results were solely simulated, so further work was needed for experimental validation. Nonetheless, the simulations did show some promising early results. As the ultrasound position estimates can vary significantly the authors opted to only use them if the IMU position estimate was over a meter different to the ultrasound measurement. This gave the overall trajectory a much smoother pattern and showed a significant improvement to the IMU data.

2.6.3 Wearable Ultrasound Systems

Ultrasound sensors have been used in numerous foot tracking systems. They have been primarily used to detect the step length of the user and thus can be used instead of, or fused with, IMU derived step lengths.

The limitation of ultrasound based methods arise when obstacles lie between the feet, such as climbing stairs, and so block the ultrasonic pulses. The other drawback is when both feet are off the ground simultaneously, as this means that there will not be a fixed point from which the distances can be measured. Studies such as [52] showed a 90 % accuracy when using ultrasound in ideal conditions, however the system was only tested in straight line walking with no obstacles. Ultrasound by itself would struggle in more complex gait patterns or conditions. More robust systems have used ultrasound in combination with IMU data.

A footstep location system which uses wearable ultrasound has been created in [29]. The ultrasound system used force sensors on the bottom of the sandals to detect heel-strike and toe-off. When the foot was detected as being on the ground the ultrasound sensors took range measurements. Via a triangulation calculation the distance between the two feet is determined.

When the ultrasound sensors lost line of sight a complementary IMU based method was used. This system used a dual axis accelerometer, an InterSense InterTrax2 device to calculate orientation, and force sensors. The force sensor is used to detect when the sandal is in a stride phase, and once the user's foot is in motion the accelerometer collects data at a rate of 100 Hz.

The two systems run in parallel, but the ultrasound system is preferentially used due to its higher accuracy. The IMU system is only used if the ultrasound system cannot successfully perform a range measurement. A graphical illustration of the two modes of operation is shown in 2.13.

[29] also used radio frequency identification (RFID) tags to attempt to mitigate drift, and so is not completely free of infrastructure requirements.

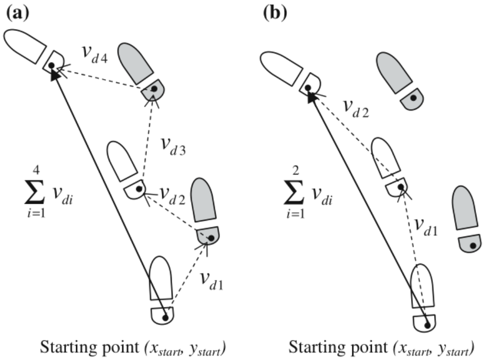


Figure 2.13: Illustration of the two different methods employed to measure the user's foot step. (a) demonstrates how the ultrasound based method measures the foot to foot distance giving a series of vectors. Summing the individual vectors results in total displacement. (b) shows the case for the IMU based method [29].

Despite the RFID tags the system performed poorly accumulating a 20% positional error when performing straight line walking and between 7.45- 13.75% error in rectangular paths. This can be attributed in part to the simplistic nature of the algorithms used i.e. no Kalman filtering was performed to account for IMU errors, neither was sensor fusion attempted and only the minimum number of ultrasound transducers were used.

A different system which uses ultrasound is described by [53]. The authors aimed to build a personal navigation system which was a full body suit aiming to give indoor localisation to fire-fighters and other search and rescue teams. The system used a combination of odometry and laser range data to localise itself within the environment.

Additionally it uses a custom built Stride Length Measurement Unit (SiLMU). The SiLMU uses time of flight ultrasonic pulses to measure the distance between the feet and is refreshed at a rate of 60Hz. The measured pattern forms a sine wave and it is filtered reducing data fluctuations. Once minimums and maximums of the sine wave are obtained they are used to calculate the step length as shown in 2.14. The heading was computed by using a Kalman filter to combine gyroscope and magnetometer readings.

The SiLMU was tested separately to the other sensors. It was evaluated by walking in a 55 m straight line walk and it performed strongly, with average percentage error of 1.63%.

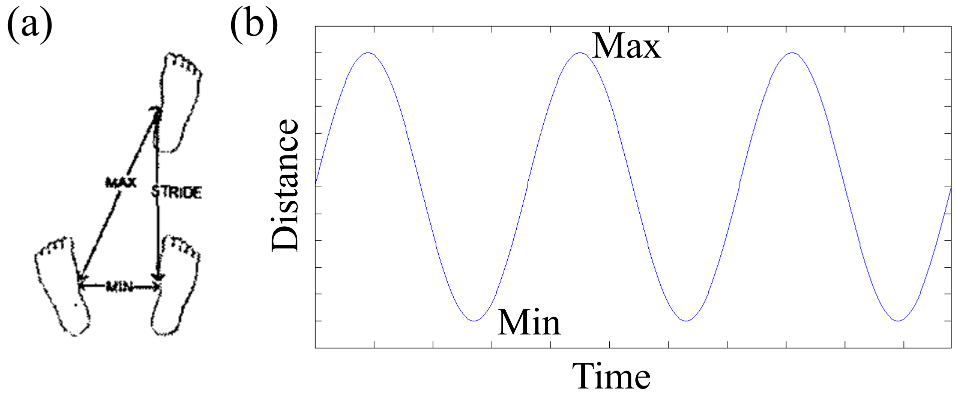


Figure 2.14: Figure showing how the stride length measurement system functions. The ultrasound data provides the foot to foot displacement. As the user walks the distance between the two feet forms a sine wave from which the minimum and maximum foot distances are found. From that the stride length can be readily obtained.

Interesting work was carried out in [54] where ultrasound and IMU data was fused via a Kalman filter in order to obtain the stride length and width. Using the Xsens Forceshoe which contained two IMUs sampling at 50 Hz ZUPT, HDR and zero height updates were used. Zero height updates state that the total vertical displacement of the IMU over a step is zero, and so any measured vertical displacement is fed into the EKF as an error. The authors also augmented the EKF to include position updates based on range information gathered from the wearable ultrasound sensors. The authors repeatedly walked within a small volume covered by Vicon cameras. Vicon cameras are highly accurate infra red sensors which are able to track the position of an object to sub millimetre accuracy. By comparing the stride length and width calculated from the IMU/ultrasound system to the data gathered with the Vicon cameras its performance was evaluated. The system is shown to be very accurate with mean stride length errors of 0.017 m and stride width errors of 0.015 m. However, the system's capability to track a user's motion over prolonged distances and track the user's absolute position was not analysed. It displayed very strong performance for motion in a local co-ordinate system but accumulated an error of 2.4 cm per step without even considering the yaw drift which was a prime source of error for previous systems of this type.

2.7 Conclusions

While GPS allows accurate outdoor positioning it lacks high resolution and, due to signal degradation, rarely gives sub-meter accuracy when indoors. However, there are a range of indoor technologies to perform tracking and positioning. The majority require pre-installed expensive infrastructure but can offer extremely high precision.

New technologies are emerging which can offer motion tracking without the infrastructure requirement. IMUs have traditionally been too noisy to use effectively, however, the development of ZUPTs [17] has enabled them to become a viable option. There have been several refinements and improvements on the original algorithm and recent works [26] have achieved less than 0.5% positional error.

A different paradigm in ubiquitous tracking has been to use pre-existing sensors such as WLAN [5], or sensors that can be inexpensively deployed such as ultrasound beacons [2]. These approaches suffer similar difficulties in which the noisy data makes accurate positioning extremely challenging. However, they do have an advantage over IMU based approaches in that the obtained position is not subject to drift, and therefore can function for indefinite periods of time without accumulating unbound error growth.

An approach that has been taken by authors to limit the drift in IMU based systems is to use particle filters in combination with a pre-known map. This limits the motion of the user to physically possible locations and has been shown to give extremely good performance in long term motion tracking, with some sophisticated approaches overcoming the associated computational cost and implementing it in a real time fashion [39].

From the literature we can identify potential areas for improvement which we can address in this work. Firstly, all of the IMU based approaches used high end commercial IMUs designed for motion tracking, or used specialised custom built IMU set-ups. This monetary barrier is a significant hurdle if accurate tracking is to occur in daily life. Hence, we proceed to investigate the potential of lower end IMUs in Chapter 3.

Secondly, the use of wearable ultrasound sensors as an aid to inertial navigation has only been examined in a few works. Among those works, ultrasound sensors were used as part of a full body system equipped with a large amount of additional sensors in [53]. By contrast in [29] the navigation was performed poorly, and in [54] only individual steps were considered. Therefore, in Chapter 4 we carry out further investigation into the performance of using an ultrasound based aiding system.

Finally, in virtually all the examined works the systems are examined without accurate ground truth data. This means the step by step performance of the systems is unknown. Therefore, when evaluating our systems we use a Vicon motion capture system to provide the ground truth data from which we can make such an analysis.

Chapter 3

Inertial Navigation

IMUs are staple sensors in motion tracking and navigation systems. In this thesis IMUs are initially used to track the user's footstep without any additional sensors or map information. Therefore, we first explore the principles behind inertial navigation and the sensor's stand-alone performance.

3.1 Inertial Measurement Units

IMUs are composed of gyroscopes and accelerometers. Their low size and weight, especially those using micro-electrical-mechanical systems (MEMS) technology are extremely popular but, due to their bias, they suffer from unavoidable drift in the displacement data. A further consideration is that the size and performance of MEMS sensors are correlated, so the more compact the sensor the lower the achieved performance [55]. This becomes a trade off when devising motion tracking systems for pedestrians which require the system to be small and portable, yet maintain high accuracy.

3.1.1 IMU Technologies

The MEMS sensors used in this thesis are rigidly attached to the shoe, giving what is known as a strapdown navigation system. Strapdown systems have the IMU sensors rigidly mounted onto the tracked object. Hence, the readings are given in the local body frame rather than the global navigation frame.

On the other hand, in stable platform navigation, the IMU is mounted onto a platform which is held in alignment with the global frame. The platform is mounted on gimbals which allow it to freely rotate. If the gyroscopes detect rotation, motors then rotate the platform back into its original alignment. By using angle pick offs the angle between the different gimbals is measured and the orientation is tracked.

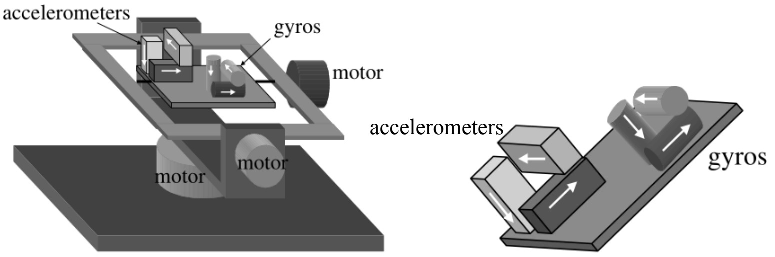


Figure 3.1: Differences between a stable-platform (left) and strap-down (right) systems [56].

Strapdown systems tend to be smaller and do not possess any moving parts. This comes at a increased computational load to process the navigation algorithm. However, as computational power is becoming less of an obstacle strapdown systems are now the prevalent method of inertial navigation. Aside from the mounting configurations, either stable platform or strapdown, gyroscopes and accelerometers come in various types operating on different principles.

Gyroscopes

Gyroscopes encompass a wide range of technologies spanning significantly in terms of cost and accuracy. The most accurate types of gyroscope are the large mechanical sensors which function by tracking the precession of a rotating mass. More compact sensors are emerging; laser ring gyroscopes for example, detect the phase difference between two light beams travelling in opposite directions. They operate on the Sagnac effect, in which the relative phases of the two laser beams are shifted depending on the angular rotation.

More exotic forms of technology exist such as cold atom gyroscopes and nuclear magnetic resonance gyroscopes. However, of most relevance to this thesis is the principle of operation behind MEMS gyroscopes. They work by having a vibrating proof mass which has an oscillating linear velocity. Should the sensor be rotated in an orthogonal axis to this velocity a Coriolis acceleration is created. By detecting the resulting movement of the proof mass via capacitors the rate of turn can be inferred. These sensors can be made extremely small and compact due to silicon etching techniques. MEMS sensors have to be made to an extremely high degree of precision as the silicon wafers are between 50-100 μm. Additionally, the proof masses only move around 10^{-9} m under a 1 rad/s turn. This results in a peak charge of 15,000-65,000 electrons, an exceedingly small amount [57].

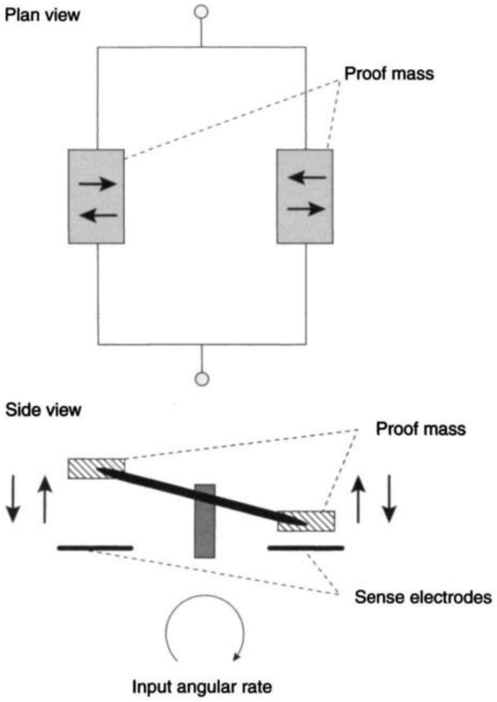


Figure 3.2: Diagram illustrating the operating principle of a MEMS gyroscope [57].

Accelerometers

Accelerometers in their most basic configuration operate by having a proof mass suspended with a spring and, as the sensor is subject to acceleration, the proof mass will be displaced. By measuring this displacement via a pick off then a signal proportional to the acceleration is derived. An alternative approach is to measure the force required to return the proof mass to its original, or null, position. This is typically achieved by having a pair of coils on the proof mass within a magnetic field. By measuring the amount of current required to generate a magnetic field of sufficient strength to counteract the acting acceleration we can then infer its magnitude and direction.

MEMS accelerometers work on a very similar principle. The sensor is manufactured out of silicon wafers, with the proof mass suspended via polysilicon springs. By using capacitors the displacement of the proof mass can be measured and the acceleration obtained.

3.1.2 Inertial Measurement Unit Errors

Due to the reliance on gyroscopes and accelerometers, several sources of error arise in any motion tracking system employing IMUs. Position correction measurements from GPS could be supplied however, with indoor tracking being the area of investigation means that GPS is too unreliable. Alternatively, this problem can be mitigated by using higher performance sensors which will exhibit minimal drift over the period of data collection. However, it does not eliminate drift but rather attempts to reduce it to an acceptable level. In addition, higher precision of gyroscopes and accelerometers will come at an increased monetary cost, offsetting the low cost aim of this project. IMUs can be broadly categorised from commercial grade to strategic grade systems. Each jump in category offers a significant increase in precision but conversely also has a 100 fold increase in price [56].

Even extremely high performance IMUs will suffer unacceptable levels of drift for motion tracking given enough time. Figure 3.3 shows a simulation of results when only considering random error sources [56]. In practice, due to scale factor calibration errors and slight gravity fluctuations, the real performance will be worse.

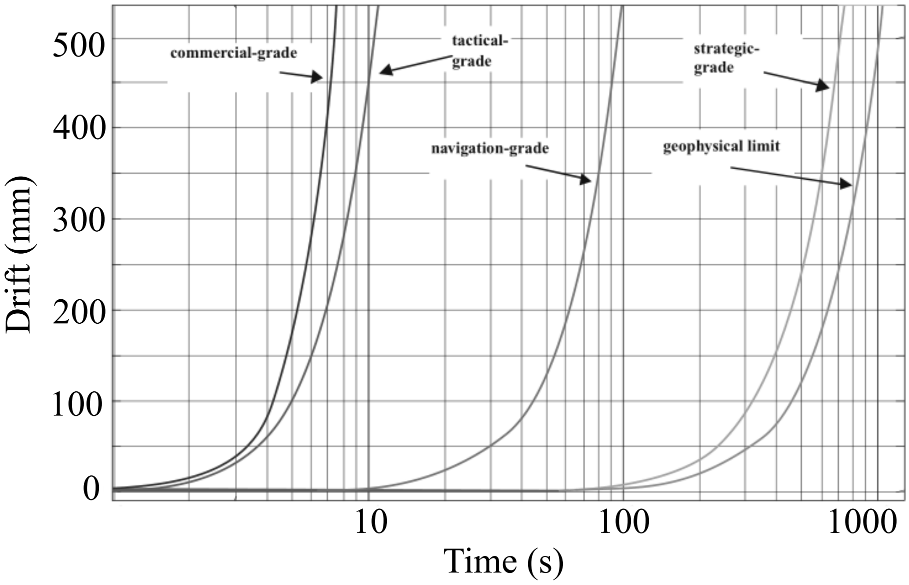


Figure 3.3: Shows a 20 minute long simulation when modelling the random error sources in different grades of IMU [56].

The errors that result in positional drift arise from 5 principal sources: sensor bias, white noise, calibration errors, temperature effects, and bias instabilities.

3.1.3 Bias

An accelerometer and gyroscope will output a constant signal even when not subject to acceleration or rotation. This is known as a bias and is a constant offset in the readings. In the case of a gyroscope bias, ρ_g, it causes the error, τ_e, to grow linearly with time, while with an accelerometer bias, ρ_a, due to double integration, it causes the error, P_e, to grow quadratically in time t,

$$\tau_e = \rho_g t \tag{3.1}$$

$$P_e = \rho_a \frac{t^2}{2} \tag{3.2}$$

The biases for the gyroscope and accelerometer can be obtained with proper calibration. However, due to the presence of other errors it is difficult to truly eliminate the bias completely.

3.1.4 White Noise

Thermo-mechanical white noise will cause the output of a MEMS IMU to be corrupted by noise fluctuating at a much faster frequency than the sampling rate of the sensor. This means that data is perturbed by a series of zero mean, uncorrelated random signals, referred to as white noise.

This results in the sensor output being influenced by a zero mean random walk error, with a standard deviation growth proportional to $\sqrt{t}$ in the case of a gyroscope and $t^{\frac{3}{2}}$ for an accelerometer [58]. Note that in the context of sensor errors random walks refer to the mathematical formalisation, not the user walking in a random direction.

3.1.5 Calibration Errors

Calibration errors are systematic effects that collectively include scale factors, mis-alignments and cross axis sensitivities in the IMU. These effects cause additional drift in the data and are only observable when the IMU is in motion. They are usually correctable with high end IMUs possessing their own internal corrections.

3.1.6 Temperature Effects

Variations in environmental temperature as well as the sensor self heating will cause changes in the bias. This causes the residual bias to give an angular error growing linearly with time and a displacement error growing quadratically.

It is possible to compensate for temperature effects by having a thermometer on the IMU to measure the changing temperature and provide suitable bias corrections. However, the IMUs used in this thesis are not equipped with thermometers and so the temperature effects are not directly compensated for.

3.1.7 Bias Instability

The bias in MEMS accelerometers and gyroscopes changes gradually over time due to flicker noise in the electronics. Flicker noise is usually observable at low frequencies, being overshadowed by white noise at high frequencies.

The bias instabilities can be modelled as a random walk. However, bias instabilities do not behave as a true random walk but are constrained within a particular range around zero. This is of great benefit, else the error in the bias would grow in an unbound fashion.

3.2 Tracking with Inertial Measurement Units

Having discussed the various sources of error present in an IMU we now turn our attention to how IMUs can be used for tracking. Firstly the IMUs utilised in this thesis are examined and their set-up explained. Then the IMUs must be calibrated to get the most accurate data possible. Lastly, the acceleration and angular velocity data can be passed into a navigation algorithm which tracks the IMU's position.

3.2.1 Inertial Measurement Unit Setup

In this work two separate IMU mounted adjacent to each other are used. The two IMUs each have a 3 axis accelerometer and a 3 axis gyroscope. Both the accelerometer and gyroscope outputs are averaged between the two IMUs. This reduces the noise associated with the acceleration and angular velocity measurements. Hence, we effectively have a single, more accurate IMU by combining the results. The reader should thus note that, unless stated otherwise, when the words "accelerometer" or "gyroscope" are used, they refer to the combination of two accelerometers or two gyroscopes effectively functioning as a single sensor.

The accelerometers used are ADXL345 manufactured by Analogue Devices. They have high resolution (13-bit) measurements at up to 16 g. They interface either via Serial Peripheral Interface (SPI) or Inter-Integrated Circuit (I^2C) communication. It is set to the full scale range of $\pm 16\,g$ and to dynamically adapt the resolution to maintain a 256 per g reading. Further we set it to have a 200 Hz sampling frequency.

The gyroscopes are ITG-3200 manufactured by InvenSense with a full scale range of $2000°s^{-1}$. They use a I^{2}C serial interface for communications. They are set to sample at 200 Hz with an internal 20 Hz digital low pass filter.

The ADXL345 and ITG-3200 sensors are mounted on a board readily available for purchase at any large electronics supplier (see Appendix for details). Communication between the IMU and the microcontroller is conducted via the I^{2}C bus.

The I^{2}C bus was developed by Philips in the early 1980s and has since grown to be a systems control bus used in millions of electrical products. The two wire I^{2}C bus functions by having a master and slave components connected by serial data (SDA) and serial clock (SCL) lines. Both are bi-directional, able to function as inputs and outputs.

I^{2}C devices are either in standard (100 kHz) mode or most modern devices can also operate in fast (400 kHz) I^{2}C mode allowing for much faster data transfer. This improvement was achieved by reducing the rise time in the data and clock lines. Additionally, fast I^{2}C devices have larger address fields of 10 bits rather than 8 bits allowing for more components to be on the same I^{2}C bus. Both the ADXL345 and the ITG-3200 are capable of running in fast I^{2}C.

The IMU is connected to an Arduino Uno Board with an ATmega328P microcontroller. After the microcontroller has acquired the data it sends the data to a laptop PC via a WiFi connection or, if WiFi is unavailable, it uses a cable running to the laptop and transfers the data via asynchronous serial communication. The data is then subsequently post-processed on the laptop. It should be noted that while in this thesis the data was post-processed with suitable software changes it could be adapted to run in a real time manner. The inertial navigation algorithm, combined with the EKF algorithm (discussed further in 3.4), takes an average of 3.4605x10^{-4} s to run per sample (tested on a laptop running an Intel i5-2410M at 2.3 GHz). As the microcontroller is able to provide data at a rate of ~160 Hz to the PC, the additional delay caused by the computation is extremely small.

Below in Figures 3.4 and 3.6 we have both the block diagram of the system and the physical layout on the shoe.

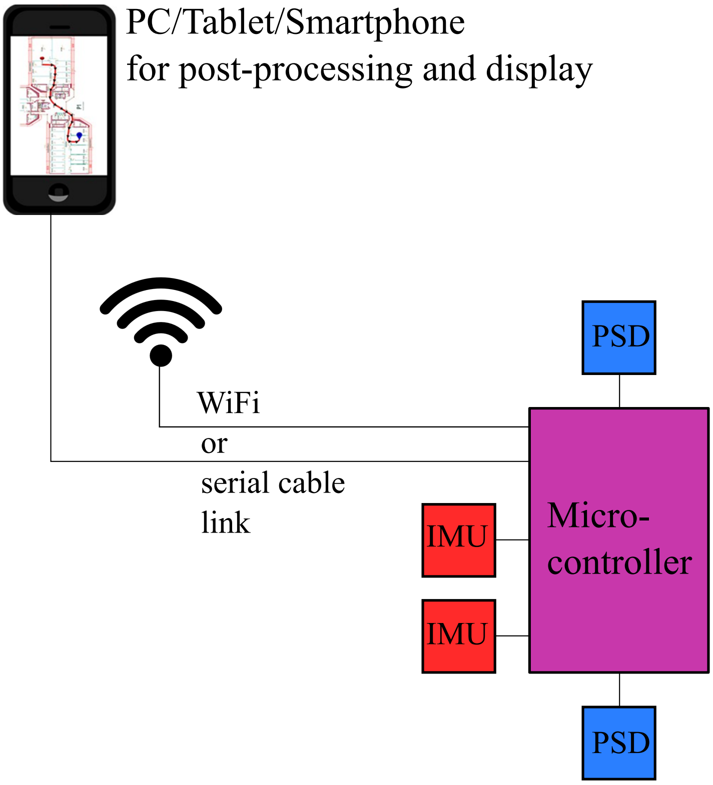


Figure 3.4: Block diagram showing the sensors connected to the microcontroller. The two IMUs outputs are averaged, effectively resulting in a single, more accurate IMU. For the inertial navigation system two position sensitive devices (PSDs) aid in detecting step events for applying ZUPT updates (see section 3.4.5 for details). The microcontroller transmits the data to an end user device which post-processes the data [59].

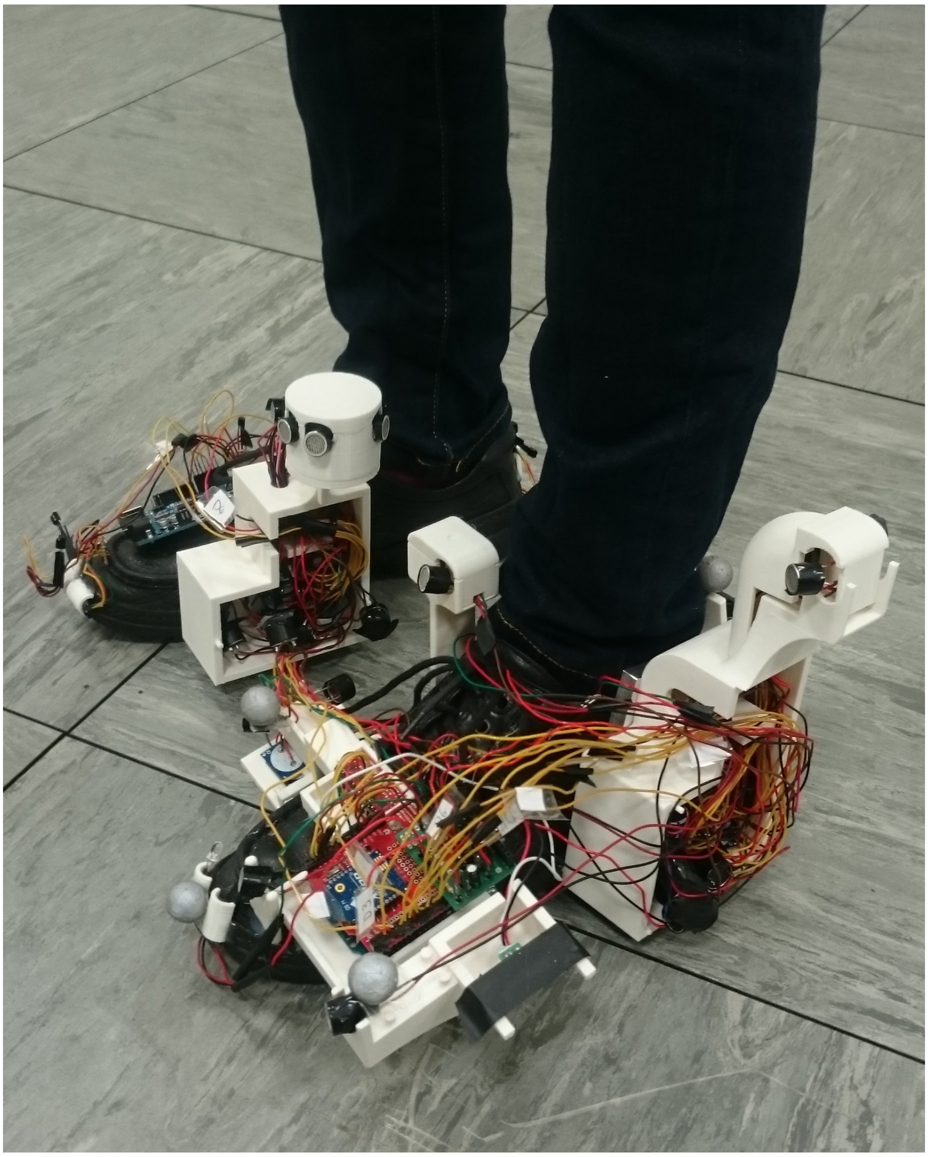


Figure 3.5: Picture showing the system being worn prior to data collection. On the left foot is mounted the inertial measurement system hardware. The majority of the wires on the left foot are for the complementary ultrasound based navigation system, discussed further in Chapter 4. Additionally on the right foot we can see further sensors used in the ultrasound navigation system.

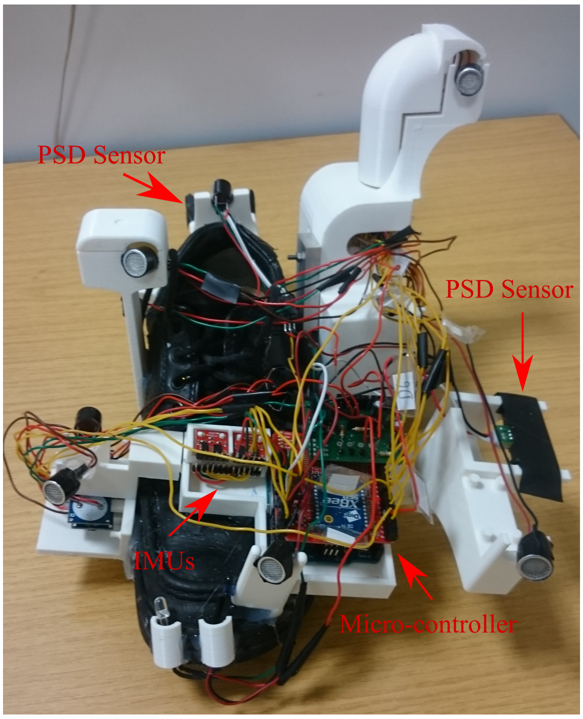


Figure 3.6: Physical setup of the sensing system. Highlighted are the IMU and PSD sensors, as well as the mirco-controller. Additionally we can see a series of ultrasound transducers and IR LEDs which form the basis for an ultrasound navigation system, discussed further in Chapter 4.

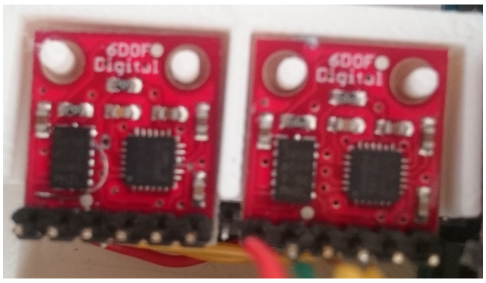


Figure 3.7: The two IMUs mounted adjacent to each other.

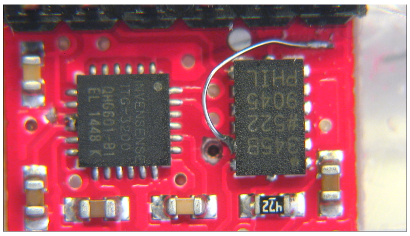


Figure 3.8: Close up on the IMU mounted on the left of Figure 3.7. As we can see it has been slightly modified. This is done so both IMUs can function on the same I^{2}C bus. By default the accelerometers share the one I^{2}C address with the gyroscopes sharing another. However an alternate I^{2}C address can be selected by changing pins 12 on the accelerometer or pin 9 on the gyroscope from ground to 3.3V. The rectangular accelerometer therefore had a small wire running from pin 12 to 3.3V while the square gyroscope had pin 9 attached to an adjacent pin which is held at 3.3V.

3.2.2 IMU Calibration

To effectively use IMUs the various sources of error described in section 3.1.2 must be quantified and corrected for as much as possible. High end IMUs can be calibrated by using six position and static rate tests. The six position method functions by having the IMU mounted on a flat surface, with each sensitive axis pointing first upwards and then downwards. For accelerometers the reference signal is the gravity vector, while the gyroscopes should output the angular rotation of the earth. For lower end IMUs the noise associated with the gyroscope completely drowns any angular velocity due to the earth's rotation [57]. The traditional approach to calibrating lower end IMUs is via the use of a mechanical platform which can rotate the IMU to precisely known positions at very accurate angular velocities. The IMU output is then compared against the known gravitational forces and rotational velocities. However, cheap off the shelf IMUs come uncalibrated and the calibration equipment is both expensive and specialised, something which is highly undesirable if a low cost foot mounted tracking system is to achieve widespread use. Various algorithms which relax the use of specialised equipment have been developed [60]. We use a procedure introduced by [61] and further developed in [62] to calibrate the IMU without the use of external sophisticated equipment.

The equipment free calibration procedure relies on two separate constraints.

1. The magnitude of the static acceleration must always equal that of gravity. This can be used to accurately calibrate the accelerometer.
2. If a static calibrated accelerometer measures a gravity vector G_1 and it is rotated so that the new gravity vector is G_2 then $G_1C = G_2$ where C is a rotation matrix computed from the gyroscope's angular velocities.

From these conditions then we can calibrate the accelerometer and then the gyroscope without any external equipment. Note, that for the calibration procedure is the only time in the thesis where we consider the two accelerometers and two gyroscopes as separate. Here we calibrate the two accelerometers individually and then calibrate the two gyroscopes. At all other points in the thesis the combined output from two accelerometers and two gyroscopes is used.

To calibrate the accelerometer we must build a model of the various sources of error to calibrate. Firstly, we note that misalignments between the axis leads to non-orthogonal measurements from the accelerometer. If the non-orthogonalities in the sensor frame are different only by small angles from the orthogonal co-ordinate axes in the platform frame then a rotation can be performed. Defining the acceleration measured in the sensor frame to be s^a and in the platform frame as s^p then, assuming that the non-orthogonality is due to small angles, s^a and s^p are related via:

$$s^p = Ts^a \tag{3.3}$$

where T is

$$T = \begin{bmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ \alpha_{xz} & 1 & -\alpha_{zx} \\ -\alpha_{xy} & \alpha_{yx} & 1 \end{bmatrix}. \tag{3.4}$$

The angle α_{ij} represents the rotation from the ith accelerometer axis around the jth platform axis. We can see the angles illustrated in Figure 3.9.

Defining the x-axis of the platform to coincide with the x-axis of the sensor, and defining the y^p axis to lie on the plane spanned by x^a and y^a then angles α_{xz}, α_{xy} and α_{yx} reduce to zero. Equation 3.4 is then simplified to

$$T = \begin{bmatrix} 1 & -\alpha_{yz} & \alpha_{zy} \\ 0 & 1 & -\alpha_{zx} \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.5}$$

We can now consider the scale factors and cross axis sensitivities to be represented by the matrix $\mathbf{S}$,

$$\mathbf{S} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix}. \tag{3.6}$$

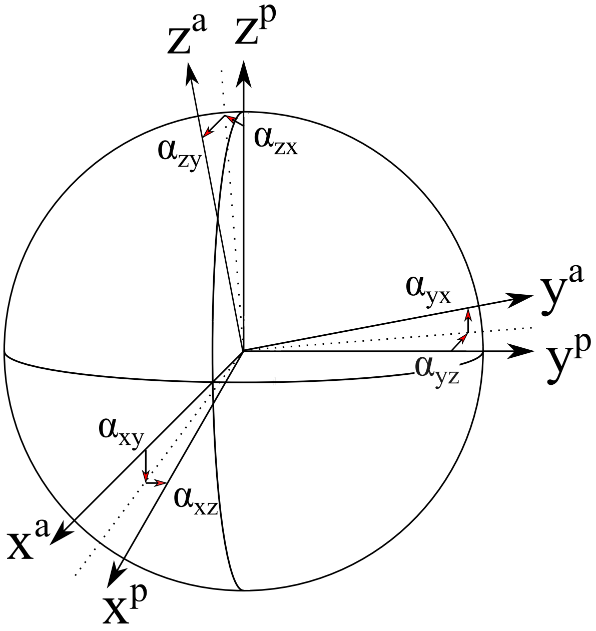


Figure 3.9: The non orthogonal sensor sensitivity axes x^a, y^a and z^a can be rotated to the orthogonal platform axes x^p, y^p and z^p through the illustrated angles [61].

Multiplying matrices T and $\mathbf{S}$ gives the calibration matrix $\mathbf{E}_a$ to be

$$\mathbf{E}_a = \begin{bmatrix} s_{xx} - s_{yx}\alpha_{yz} + s_{zx}\alpha_{zy} & s_{xy} - s_{yy}\alpha_{yz} + s_{zy}\alpha_{zy} & s_{xz} - s_{yz}\alpha_{yz} + s_{zz}\alpha_{zy} \\ s_{yx} - s_{zx}\alpha_{zx} & s_{yy} - s_{zy}\alpha_{zx} & s_{yz} - s_{zz}\alpha_{zx} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix}, \tag{3.7}$$

which can be more compactly written as

$$\mathbf{E}_a = \begin{bmatrix} e_{00} & e_{01} & e_{02} \\ e_{10} & e_{11} & e_{12} \\ e_{20} & e_{21} & e_{22} \end{bmatrix}. \tag{3.8}$$

$\mathbf{E}_a$ is therefore a diagonally dominant correction matrix. To calibrate the acceleration data we must subtract the bias vector, b_a, from the raw acceleration values, a_s. Then after multiplication with the matrix, $\mathbf{E}_a$, the calibrated accelerometer values, a_p, are achieved:

$$a_p = \mathbf{E}_a(a_s - b_a). \tag{3.9}$$

The parameters to be obtained can be expressed as a single vector,

$$\Gamma = [e_{00}, e_{01}, e_{02}, e_{10}, e_{11}, e_{12}, e_{20}, e_{21}, e_{22}, b_x, b_y, b_z] \tag{3.10}$$

and to calculate Γ we define our cost function as,

$$\mathbf{L}_a(\Gamma) = \sum_{k=1}^{k=N}(||g||^2 - ||a_p||^2)^2 \tag{3.11}$$

where N is the number of static orientations that the accelerometer is exposed to. It must at least be equal to the number of parameters being estimated, in our case 12. The cost function can be minimised using a variety of techniques. Here Newton's method [63] was selected,

$$\Gamma_k = \Gamma_{k-1} + [\frac{d^2\mathbf{L}_a(\Gamma)}{d\Gamma d\Gamma^T}]^{-1}\frac{d\mathbf{L}_a(\Gamma)}{d\Gamma} \tag{3.12}$$

After the accelerometer is calibrated it can be used to calibrate the gyroscope. By keeping the gyroscope stationary we can take an average of the output on each axis and obtain the basis. To calculate the other calibration parameters we can, in a similar fashion to the accelerometer, construct a matrix, $\mathbf{E}_\omega$, with the scale factors, misalignments and cross factor sensitivities,

$$\omega_p = \mathbf{E}_\omega \omega_s \tag{3.13}$$

where ω_p and ω_s are the calibrated and uncalibrated gyroscope readings respectively.

To calibrate the gyroscope the following procedure is applied.

1. Firstly, an initial gravity vector g_{int} is measured by the static calibrated accelerometer.

2. The IMU is rotated and, using the second order algorithm integration method as is presented by Jekeil [64], the rotation matrix C is obtained.

3. The initial gravity vector is rotated by C to give g_{fin} and is compared to the output of the accelerometer in the IMU's post rotation position, u_a.

In the absence of errors u_a and g_{fin} will have the same value. Hence, we define the cost function as

$$\mathbf{L}_\omega = \sum_{k=1}^{k=N} ||u_a - g_{fin}||^2 \tag{3.14}$$

in which N is the number of rotations the IMU is exposed to and should be equal to, or greater than, the number of parameters being estimated, in this case 9. We minimise equation 3.14 by using MATLAB's built in non-linear least squares solver [65] which solves non-linear least suqares problems of the form,

$$\min_x (f_1(x)^2 + f_2(x)^2 + ... + f_n(x)^2) \tag{3.15}$$

After minimising 3.14 we obtain our calibration parameters $\mathbf{E}_\omega$. This algorithm will however yield a worse calibration result than the accelerometer procedure. This is an inherent limitation as we are assuming a perfectly calibrated accelerometer as well as no integration errors.

The selected integration method in step two of the procedure above, is that presented in [64]. It is a second order algorithm which relates the current quaternion, q_k, to the previous quaternion, q_{k-1}, through

$$q_k = [cos(0.5|\delta\beta|)\mathbf{I} + \frac{1}{|\delta\beta|} sin(0.5|\delta\beta|)\mathbf{B}]q_{k-1} \tag{3.16}$$

where $\mathbf{I}$ is a 4x4 identity matrix. $\delta\beta$ is a vector composed of the product of the angular velocities ω_p and the time step Δt. The matrix $\mathbf{B}$ is given by,

$$\mathbf{B} = \begin{bmatrix} 0 & \delta\beta_x & \delta\beta_y & \delta\beta_z \\ -\delta\beta_x & 0 & \delta\beta_z & -\delta\beta_y \\ -\delta\beta_y & -\delta\beta_z & 0 & \delta\beta_x \\ -\delta\beta_z & \delta\beta_y & -\delta\beta_x & 0 \end{bmatrix}. \tag{3.17}$$

The procedures above were implemented as follows: firstly the accelerometer was held stationary in a total of 26 orientations. For each orientation 3 minutes of data was collected. The orientations consisted of each of the accelerometer's 6 faces, 12 edges and 8 corners being aimed downwards i.e. towards earth. It is important to emphasize that a precise orientation is not needed, the algorithm simply relies on the constraint the magnitude of the measured gravity must be equal to $9.81\,\mathrm{ms}^{-2}$. A range of orientations is used here so that there is data on each axis experiencing a significant gravity component, and hence enabling the calibration parameters for each axis to be calibrated more accurately. The accelerometer data is pre-multiplied by the scaling factor of $\frac{g}{250}$ which is the approximate scaling as recommended by the data sheet. After the procedure above was carried out and data gathered in all the relevant orientations the following bias and calibration matrices where then obtained:

$$\mathbf{E}_a = \begin{bmatrix} 0.956 & -0.0016 & -0.003 \\ -0.0016 & 0.960 & -0.0084 \\ -0.003 & -0.0084 & 0.9898 \end{bmatrix}, \quad \mathbf{b}_a = \begin{bmatrix} 0.2476 \\ -0.190 \\ -0.660 \end{bmatrix}.$$

$$\mathbf{E}_a = \begin{bmatrix} 0.9668 & 0.00193 & -0.00395 \\ 0.0019 & 0.96797 & -0.00561 \\ -0.00395 & -0.00561 & 1.0046 \end{bmatrix}, \quad \mathbf{b}_a = \begin{bmatrix} 0.2598 \\ 0.2980 \\ -0.6987 \end{bmatrix}.$$

For the gyroscopes the following method to gather the data was conducted. Firstly, the IMU was held stationary for 2 minutes to obtain the initial gravity vector, g_{int}, and the gyroscope bias. Then, the IMU is rotated by approximately 180° and held in the new orientation for 30 seconds to obtain a value for g_{fin}. Note that the exact angle that the IMU is rotated though is unimportant- all that is required is for the IMU to go through a sufficiently large rotation that the gyroscope errors can accumulate and be corrected for through appropriate calibration. Further the exact rotation speed is unimportant as long as it remains within the tolerances of the gyroscope and is appropriate for the sampling frequency of the IMU.

The IMU is exposed to a range of different rotations as shown in Figure

3.10. For each setup the IMU is rotated first clockwise and then counter-clockwise giving a total of 18 unique rotations. After the data was gathered for each rotation the gyroscope calibration algorithm was executed which gave the following calibration parameters:

$$\mathbf{E}_\omega = \begin{bmatrix} 1.040 & 0.017 & -0.004 \\ -0.011 & 1.032 & 0.023 \\ 0.025 & 0.021 & 0.990 \end{bmatrix} \left(\frac{1}{14.375}\right)\frac{\pi}{180}, \quad \mathbf{b}_\omega = \begin{bmatrix} 29.517 \\ 48.263 \\ 0.138 \end{bmatrix}.$$

$$\mathbf{E}_\omega = \begin{bmatrix} 1.011 & -0.025 & -0.009 \\ 0.015 & 1.024 & -0.033 \\ 0.004 & 0.015 & 0.987 \end{bmatrix} \left(\frac{1}{14.375}\right)\frac{\pi}{180}, \quad \mathbf{b}_\omega = \begin{bmatrix} -59.073 \\ 27.018 \\ -8.74 \end{bmatrix}.$$

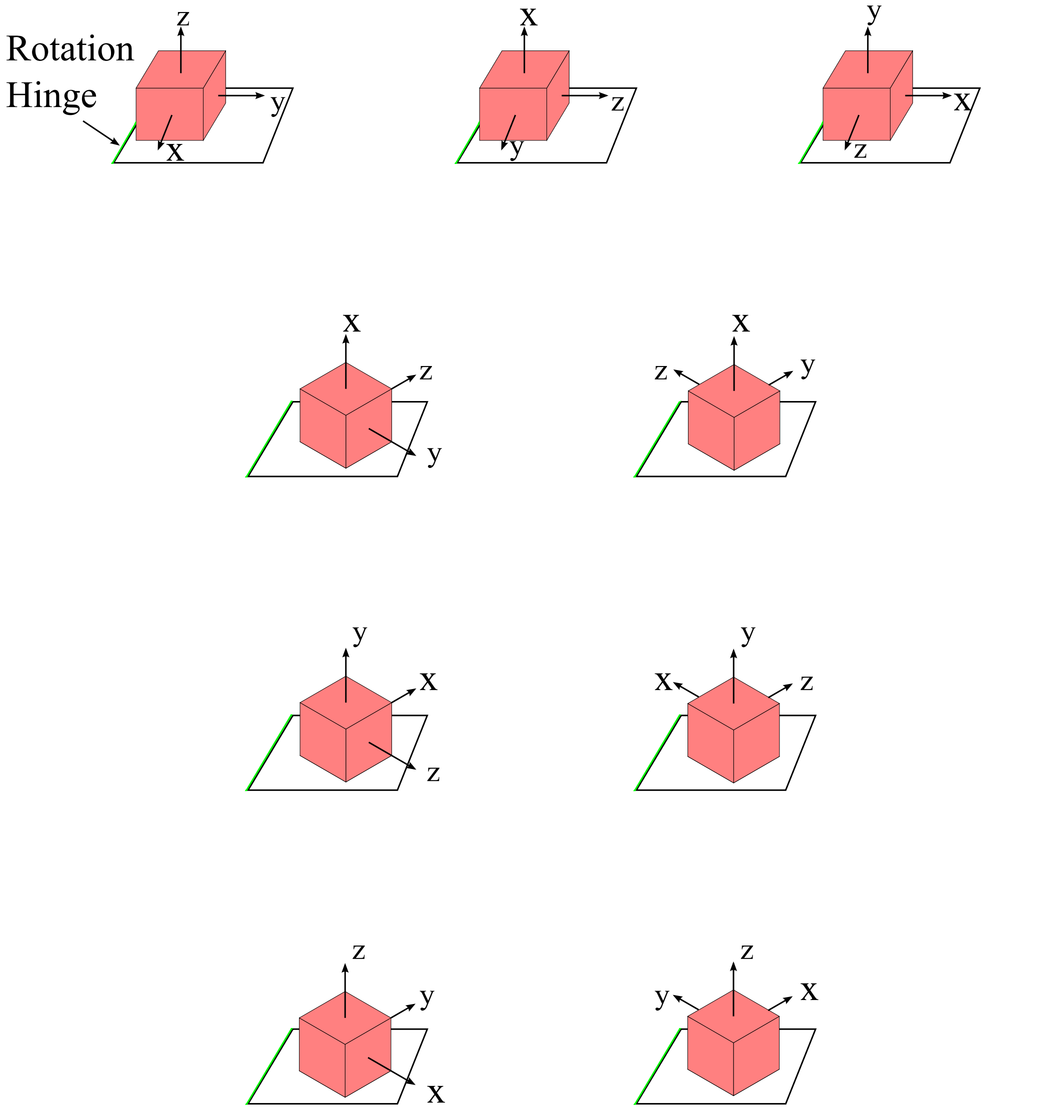


Figure 3.10: Diagram illustrating the different rotations the IMU was exposed to when calibrating the gyroscope.

As accurate turntables were unavailable it was not possible to determine the accuracy of the gyroscope calibration parameters. However, by using gravity as a reference we can gain a measure of how accurate the accelerometers are. The accelerometers were held stationary for five separate two minute long trials. The average of the accelerometer output is presented in tables 3.1 and 3.2. This gives the average error for the two accelerometers to be 0.0324 ms^{-2} and 0.0196 ms^{-2}.

Test Number	**Measured Gravity** (ms^{-2})	**Error** (ms^{-2})
1	9.765	0.045
2	9.816	0.006
3	9.801	0.009
4	9.783	0.072
5	9.780	0.030

Table 3.1: Calibration results for accelerometer number 1.

Test Number	**Measured Gravity** (ms^{-2})	**Error** (ms^{-2})
1	9.776	0.034
2	9.796	0.014
3	9.835	0.025
4	9.800	0.010
5	9.825	0.015

Table 3.2: Calibration results for accelerometer number 2.

3.2.3 Inertial Navigation

Once the IMU is calibrated we can develop the required navigation algorithm. The strapdown navigation algorithm is comprised of several stages. However, before diving into the mathematics a brief overview can illustrate a few key points. Firstly, as a strapdown system is used we must note that the accelerometer measures the acceleration in its own body frame which is not the same as the global navigation frame.

Therefore, we use the gyroscope to track the IMU's orientation with respect to the navigation frame. After rotating the measured accelerations into the navigation frame the effect of gravity can be accounted for, and after double integration of the acceleration the position obtained. Figure 3.11 gives an overview of the steps involved.

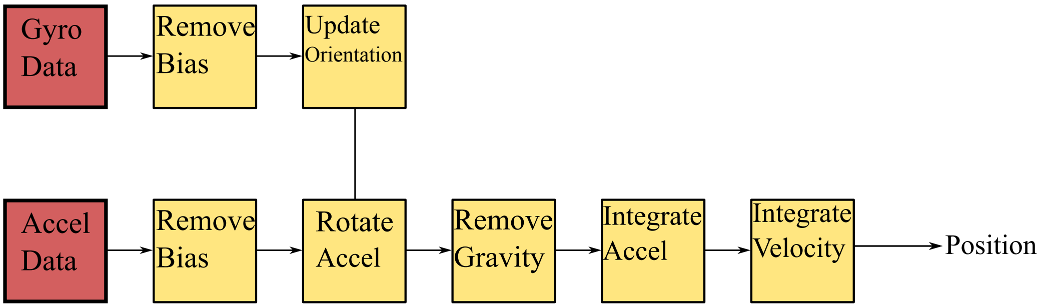


Figure 3.11: Block diagram of inertial strap-down navigation.

The gyroscope measures the angular rotation with respect to an inertial frame, which is a frame that is not subject to acceleration or rotation. Hence, strictly speaking if we take our navigation frame's x and y co-ordinates to align with north and east, it is not an inertial frame of reference due to the rotation of the earth. In our case as the IMU is a relative tracking sensor it is not possible, unless further information is provided, to compensate for the earth's rotation. Compensation would only be possible if the IMU was initialised with an absolute rotation with respect to the earth and if the IMU's latitude was known. However, the errors arising from neglecting the earth's rotation are far smaller then those arising from sensor noise. For comparison the earth's rotation of 7.292x10^{-5} rad/sec is close to two orders of magnitude less than the gyroscope's standard deviation as quoted by the data sheet of 6.632x10^{-3} rad/sec. Therefore, in this thesis we make the approximation that the navigation frame is an inertial frame.

3.2.4 Strap-down Algorithm

The first stage in a strap-down navigation algorithm is to bias compensate the raw acceleration, a_k^b, and gyroscope, ω_k^b, information at step k,

$$\omega_k^{'b} = \omega_k^b - \delta\omega \tag{3.18}$$

$$a_k^{'b} = a_k^b - \delta a \tag{3.19}$$

where $\delta\omega$ and δa represent the gyroscope and accelerometer biases respectively. $\omega_k^{'b}$ and $a_k^{'b}$ are thus the bias compensated gyroscope and accelerometer readings.

Subsequently, the sensor orientation is updated with respect to the navigation frame via a Padé approximation [66],

$$C_{b_{k|k-1}} = C_{b_{k-1|k-1}} \frac{2\mathbf{I} + \delta\Omega_k \Delta t}{2\mathbf{I} - \delta\Omega_k \Delta t} \tag{3.20}$$

where Δt is the time step between updates and $\mathbf{I}$ is a 3x3 identity matrix. $C_{b_{k|k-1}}$ is the rotation matrix which transforms from the sensor frame to the navigation frame, which has not yet been corrected by the EKF. $C_{b_{k-1|k-1}}$ is the last rotation matrix available which has been corrected by the EKF at update $k-1$. Ω_k is the skew symmetric matrix used to define small increments in orientation.

$$\delta\Omega_k = \begin{bmatrix} 0 & -\omega_k^{'b_z} & \omega_k^{'b_y} \\ \omega_k^{'b_z} & 0 & -\omega_k^{'b_x} \\ -\omega_k^{'b_y} & \omega_k^{'b_z} & 0 \end{bmatrix}. \tag{3.21}$$

In the third stage the bias compensated accelerometer readings are transformed to the navigation frame and gravity, g, is subtracted from the vertical component.

$$a_k^c = C_{b_{k|k-1}} a_k^{'b} - [0,0,g] \tag{3.22}$$

Lastly, the acceleration readings, a_k^c, are double integrated via the trapezium rule to yield first velocity, $v_{k|k-1}$ and then position, $r_{k|k-1}$,

$$v_{k|k-1} = v_{k-1|k-1} + (a_{k-1}^c + \frac{a_k^c - a_{k-1}^c}{2})\Delta t \tag{3.23}$$

$$r_{k|k-1} = r_{k-1|k-1} + (v_{k|k-1} + \frac{v_{k-1|k} - v_{k-1|k-1}}{2})\Delta t. \tag{3.24}$$

As we can see going from acceleration and angular velocity requires several stages. Small errors in the initial acceleration and angular velocity measurements can propagate to give large uncertainties in the final position.

3.3 Error Accumulation Analysis

The inertial navigation algorithm presented would work well if the IMU was not subject to the sources of errors discussed in section 3.1.2. Errors in the acceleration which are propagated through the double integration will cause error growth in position values proportional to t^2. However, accumulated errors in the gyroscope will cause the most severe positional errors. This is because the gyroscope readings are used to calculate the orientation matrix C. This in turn leads to both the attitude being computed incorrectly, but more significantly, leads to residual gravity components being doubly integrated.

To evaluate the IMU drift a simple experiment was conducted. The IMU was held stationary for a period of 180 seconds and the acceleration and gyroscope data was collected. For the same set of data the navigation algorithm was run four different times. For each run a different source of error was examined, the results of which are shown in Figure 3.12.

To see the impact of the gyroscope errors the accelerometer data was artificially set to $a^{'b} = [0,0,9.81]$. Thus the positional drift will only be due to the gyroscope errors.

In turn, the effect of the accelerometer errors were also examined. By setting the gyroscope readings to $\omega_k'^b = [0, 0, 0]$ we thus remove any errors arising from the gyroscope and the rotation of the earth.

To examine how much of the gyroscope errors are actually due to the rotation of the earth we can set every data point to:

$$a_k^c = [0, 0, 9.81] \tag{3.25}$$

$$\omega_k'^b = [\xi \cos \Lambda, 0, \xi \sin \Lambda] \tag{3.26}$$

where ξ is the earth's rotation, 7.292x10^{-5} rad/sec and Λ is the latitude where the tests are performed (Manchester, 53.48°). Therefore, the only error source is the unavoidable rotation of the earth.

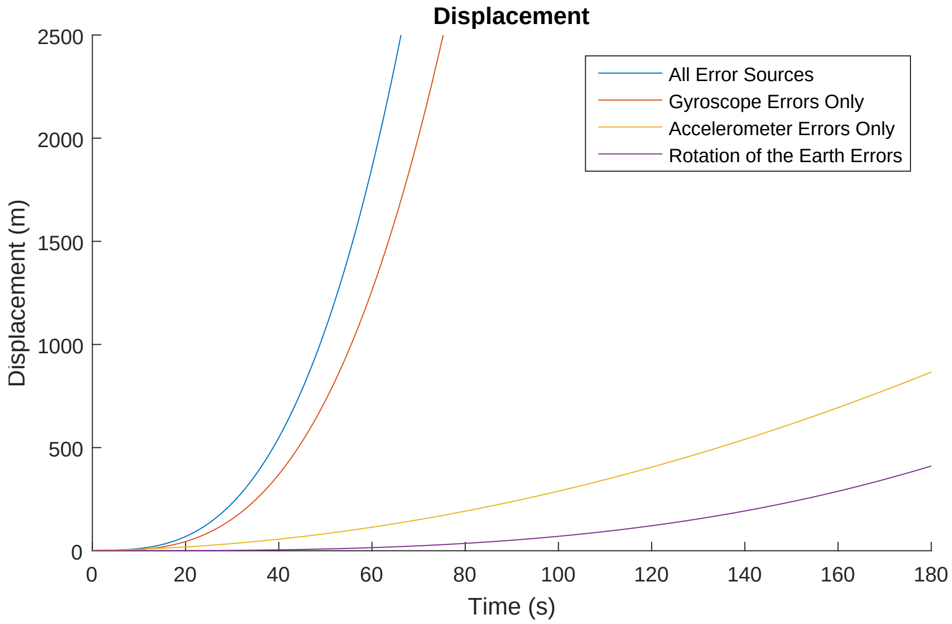


Figure 3.12: Drift results due to the various sources of error. We can see how gyroscopic errors account for the majority of the observed drift.

Figure 3.12 shows the measured position under the different testing conditions. We can see two separate features from the graph. Firstly, that errors arising from the gyroscope dominate over accelerometer errors. This can be attributed in part due to the fact that the gyroscopes are calibrated to a worse standard than the accelerometers (see 3.2.2 for details). Additionally, we can note that the accumulated errors for both the gyroscope and earth's rotation grow proportionally to t^3 with the acceleration errors growing in proportion to t^2 as expected from the inertial navigation algorithm. We see that the earth's rotational errors, although starting extremely small, due to the cubic in time error growth rapidly accumulate and will eventually overtake the accelerometer errors which have a quadratic in time error growth.

Lastly, Figure 3.12 demonstrates the importance of requiring powerful error correction techniques to obtain meaningful data. Indeed, tracking a moving object will cause even more errors to arise due to two reasons. Firstly, the larger accelerations will cause more significant positional errors due to being rotated using an incorrect rotation matrix. Secondly, the gyroscope did not have any significant rotation applied to it and mis-calibrations in the cross-axis sensitivity, non-linearity and misalignments will only be observed when the gyroscope is subject to rotation.

3.4 Kalman Filters in Pedestrian Dead Reckoning

As can be clearly seen from Figure 3.12 any positional data inferred from the IMU quickly becomes meaningless. Thus, additional stages to correct for the accumulated error must be introduced.

Error corrections in position from devices such as GPS to directly correct for the position do offer improved accuracy however, direct position corrections will only effect the variables it directly observes. GPS for example will be unable to offer corrections to the sensor biases, as well as tilt and roll angles.

To be able to correct for the various states in the system we exploit our knowledge of human walking in combination with a Kalman filter to apply corrections with every footstep.

Kalman filters have been used in conjunction with a strapdown navigation system to feed in ZUPT and HDR corrections. In this work, an inertial navigation algorithm which received error correction information from an EKF was used.

The EKF uses a 15 element error state defined as:

$$\delta x = [\delta\phi, \delta\omega, \delta r, \delta v, \delta a] \tag{3.27}$$

where all 5 components are 3x1 matrices. $\delta\phi, \delta r$, and δv are the estimated error in orientation, position and velocity respectively, while $\delta\omega$ and δa are the estimated biases for the gyroscope and accelerometer. Hence, the EKF tracks the error states which can be used to correct the various parameters in the navigation algorithm.

We can linearise the navigation algorithm around a state estimate to give

$$\delta x_{k|k-1} = \Phi_k \delta x_{k-1|k-1} + w_k \tag{3.28}$$

in which $x_{k|k-1}$ is the predicted error state and $x_{k-1|k-1}$ is the last filtered error state. w_k represents the process noise with covariance matrix Q_k and Φ_k is the transition matrix,

$$\Phi_k = \begin{bmatrix} I & \Delta t C^n_{b_{k|k-1}} & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & \Delta t I & 0 \\ -\Delta t S(a_k^{'n}) & 0 & 0 & I & \Delta t C^n_{b_{k|k-1}} \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \tag{3.29}$$

where each element is a 3x3 matrix. $S(a_k^{'n})$ is the cross product matrix:

$$S(a_k^{'n}) = \begin{bmatrix} 0 & -a_{zk} & a_{yk} \\ a_{zk} & 0 & -a_{xk} \\ -a_{yk} & a_{xk} & 0 \end{bmatrix}. \tag{3.30}$$

In the matrix $a_k^{'n}$ is the bias corrected acceleration which is transformed into the navigation reference frame. This causes errors in orientation to become correlated to velocity errors. These correlations are stored in the covariance matrix P_k. Due to the covariance matrix ZUPTs are able to correct all of the correlated errors despite only directly observing a limited number of states.

The measurement model to obtain the errors, z_k, for this EKF is

$$z_k = H\delta x_{k|k} + n_k \tag{3.31}$$

where H is the measurement matrix and n_k is the measurement noise with noise covariance R_k.

At this point we can use our knowledge of human gait to provide error correction techniques. The core of these techniques is the zero velocity update of which we now examine the mathematical implementation.

3.4.1 Zero Velocity Updates

As detailed in Section 2.1.2, zero velocity updates (ZUPTs) declare that the velocity of the foot during the mid stance phase should be $0\,\mathrm{ms}^{-1}$. We can therefore state that the measured error, m_v, is any velocity, Δv, detected by the IMU

$$m_v = \Delta v - [0, 0, 0]. \tag{3.32}$$

The corresponding measurement matrix, H, is then

$$H = \begin{bmatrix} 0_{3\mathrm{x}3} & 0_{3\mathrm{x}3} & 0_{3\mathrm{x}3} & I_{3\mathrm{x}3} & 0_{3\mathrm{x}3} \end{bmatrix} \tag{3.33}$$

where the subscripts indicate 3 by 3 matrices. ZUPTs is one of several error state corrections that we can apply. These similar techniques that we can incorporate alongside ZUPTs deal with other parameters in the error state matrix.

3.4.2 Heuristic Drift Reduction

Rather than implementing a binary-I controller as was done by [21], we incorporate heuristic drift reduction (HDR) measurements into the EKF framework. If the change in yaw, $\delta\psi$, between successive steps is lower than a certain threshold then we interpret this as an incorrect measurement due to drift and feed in an appropriate correction value,

$$m_{\mathrm{HDR}} = \begin{cases} \Delta\psi, & \text{if } \Delta\psi \leq 5^{\circ} \\ 0, & \text{otherwise.} \end{cases}$$

Hence, HDR has a measurement matrix of,

$$H = \begin{bmatrix} [001] & 0_{1\text{x}3} & 0_{1\text{x}3} & 0_{1\text{x}3} & 0_{1\text{x}3} \end{bmatrix}. \tag{3.34}$$

3.4.3 Combination

To combine the techniques described above the individual measurement matrices can be merged to give

$$H = \begin{bmatrix} [001] & 0_{1\text{x}3} & 0_{1\text{x}3} & 0_{1\text{x}3} & 0_{1\text{x}3} \\ 0_{3\text{x}3} & 0_{3\text{x}3} & 0_{3\text{x}3} & I_{3\text{x}3} & 0_{3\text{x}3} \end{bmatrix} \tag{3.35}$$

with m_k, the measured error, corresponding to

$$m_k = [\Delta\psi_k \;\; \Delta v_k]. \tag{3.36}$$

Each component of the measured error arises from one of the error reduction techniques of ZUPTs or HDR.

From the derived matrices we can then execute the regular Kalman filter equations with Kalman gain, K_k, obtained via,

$$K_k = P_{k|k-1} H^T (H P_{k|k-1} H^T + R_k)^{-1} \tag{3.37}$$

where $P_{k|k-1}$ is the estimation error covariance matrix and is computed at every step k from

$$P_{k|k-1} = \Phi_k P_{k-1|k-1} \Phi_k^T + Q_k. \tag{3.38}$$

$P_{k|k-1}$ is then updated once the EKF corrections are applied via the Joseph form equation,

$$P_{k|k} = (I - K_k H) P_{k|k-1} (I - K_k H)^T + K R_k K^T. \tag{3.39}$$

Finally, the filtered error state can be calculated from,

$$\delta x_{k|k} = \delta x_{k|k-1} + K_k(m_k - H\delta x_{k|k-1}) \tag{3.40}$$

With the EKF we can feed the error corrections, δr_k and δv_k, into the navigation algorithm. The obtained position and velocity are corrected via

$$r_{k|k} = r_{k|k-1} - \delta r_k \tag{3.41}$$

$$v_{k|k} = v_{k|k-1} - \delta v_k \tag{3.42}$$

The attitude adjustment is performed by updating the rotation matrix with $\delta\phi$, corresponding to the errors for roll, pitch and yaw. Using a small angle approximation the rotation matrix is updated following,

$$C^n_{b_k|k} = \frac{2\mathbf{I} + \delta\Theta}{2\mathbf{I} - \delta\Theta} C^n_{b_k|k-1} \tag{3.43}$$

where $\delta\Theta$ is

$$\delta\Theta_k = \begin{bmatrix} 0 & -\delta\phi_k(3) & \delta\phi_k(2) \\ \delta\phi_k(3) & 0 & -\delta\phi_k(1) \\ -\delta\phi_k(2) & \delta\phi_k(1) & 0 \end{bmatrix}. \tag{3.44}$$

The non-bias error terms of the state are reset to 0 after they are passed onto the navigation algorithm and used to refine the position, velocity and attitude.

3.4.4 Filter Initialisation

For proper functioning of the EKF the parameters of the process noise, Q_k, measurement noise, R_k, and covariance matrix, P_k, need to be properly selected to ensure stable performance. Fine tuning of these parameters needs to be carried out as their values have a large influence on the effectiveness of the EKF.

In an ideal scenario the process noise represents solely a magnitude of a white noise sequence. This is under the assumption that all the various error sources are accurately modelled into the state propagation matrix.

However, many error sources are not represented in the state transition matrix such as non-linearity, hysteresis, misalignments, g-sensitivity, and scale factor temperature coefficients. The accelerometers and gyroscopes are assumed to be dominated by two primary sources of error [67]:

1. **Sensor noise:** The IMU was held stationary for a period of 180 seconds and the measured acceleration and angular velocity was examined. Form this the accelerometer was found to have a standard deviation of 0.0381 ms^{-2} with the gyroscope having a standard deviation of 8.1500x10^{-3} rad s^{-1}.

2. **Mis-calibration errors**: The two accelerometers had an average error of 0.265% when measuring gravity. Therefore, we take the accelerometer noise as $\sigma_a = 2.65\text{x}10^{-3}a$. For the gyroscope a reference signal was not available, however as the gyroscope relied on a less precise procedure for calibration we know that it will be calibrated to a worse degree than the accelerometer. Therefore, the error is modelled as 1% of the measured value, $\sigma_\omega = 0.01\omega$.

The procedure to determine the gyroscope covariances is therefore to:

1. Set $\omega_{max} = \max(\omega_\phi, \omega_\theta, \omega_\psi)$.

2. Let $\sigma_\omega^2 = (8.1500\text{x}10^{-3})^2 + (1\text{x}10^{-2}\omega_{max})^2$. In other words we sum the individual covariances of sensor noise and mis-calibration to obtain the total error for that particular reading.

3. Set the matrix

$$\hat{Q}_\omega = \begin{bmatrix} \sigma_\omega^2 & 0 & 0 \\ 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & \sigma_\omega^2 \end{bmatrix} \tag{3.45}$$

   .

   Here we make the assumption that there is no covariance between the sensor axes.

4. Set the submatrix of Q which represents the gyroscope covariances to be

$$Q_\omega = C_k \hat{Q}_\omega C_k^T \tag{3.46}$$

where C_k is the rotation matrix at time-step k.

This procedure is then repeated for the accelerometer, but with $a_{max} = \max(a_x, a_y, a_z)$ and a standard deviation of $0.0381\,\mathrm{ms}^{-2}$.

To obtain the angular covariances, the ω_{max} values are multiplied by the duration of the time-step, Δt, and due to the first order integration used, have an integration error added of $\omega_{max}^2 \Delta t$. Therefore, the integration adds a covariance per step of $\omega_{max}^4 \Delta t^4$ [67]. This is carried out equivalently for the velocity covariances [68], however as a second order algorithm was used for integrating the acceleration the covariance per step is $a_{max}^6 \Delta t^6$. This method will provide a slight overestimation of the noise covariance as we are using the maximum value, ω_{max} or a_{max}, to determine the covariance for all the axes.

Accurately determining the values of the covariance matrix Q_k is more of an art than a science in many cases, with rather arbitrary scaling often required [67], [68]. Indeed, through experimental testing the values generated using the above gave less than optimal results. Hence, the covariances corresponding to the roll, pitch and yaw were set to a larger fixed value of $0.5\text{x}10^{-3}\,\mathrm{rad}^2$ and the gyroscope covariance due to the sensor noise was increased to $1\text{x}10^{-2}\,\mathrm{rad}^2\mathrm{s}^{-2}$.

The measurement noise covariance matrix was modelled in an equally approximate manner. It was assumed that there was a small amount of uncertainty in each reading that was independent of other measurements. Hence, the matrix was set to

$$R = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}. \tag{3.47}$$

3.4.5 Step Detection

To be able to apply the EKF corrections the system must be able to determine if the foot is in a stance phase. To that end, a dual approach of using two position sensitive devices (PSD) in conjunction with the IMU data is used. For a foot to be declared flat on the ground then three conditions need to be met.

Proximity Sensing

The PSDs are mounted on the heel and the toe of the shoe. When the system is first turned on, and the foot is flat on the ground, the distance from the PSDs to the floor is obtained. When both the PSDs measure under 1.05 of the initial measured distances, D_{int}, we consider the first condition, $\hat{C}_1$, to be fulfilled,

$$\hat{C}_1 = \begin{cases} 1, & \text{if } D_k \leq 1.05 D_{int} \\ 0, & \text{otherwise.} \end{cases}$$

In which D_k is the range as measured by the PSD.

Acceleration

Secondly, we look at the magnitude of the bias compensated acceleration, a_{mag}, and if the sensor is measuring close to gravity we interpret it as the foot being stationary according to the accelerometer. Hence, if the magnitude is within the limits $9.3\,\mathrm{ms}^{-2} \leq a_{mag} \leq 10.3\,\mathrm{ms}^{-2}$ then the second condition, $\hat{C}_2$, is fulfilled,

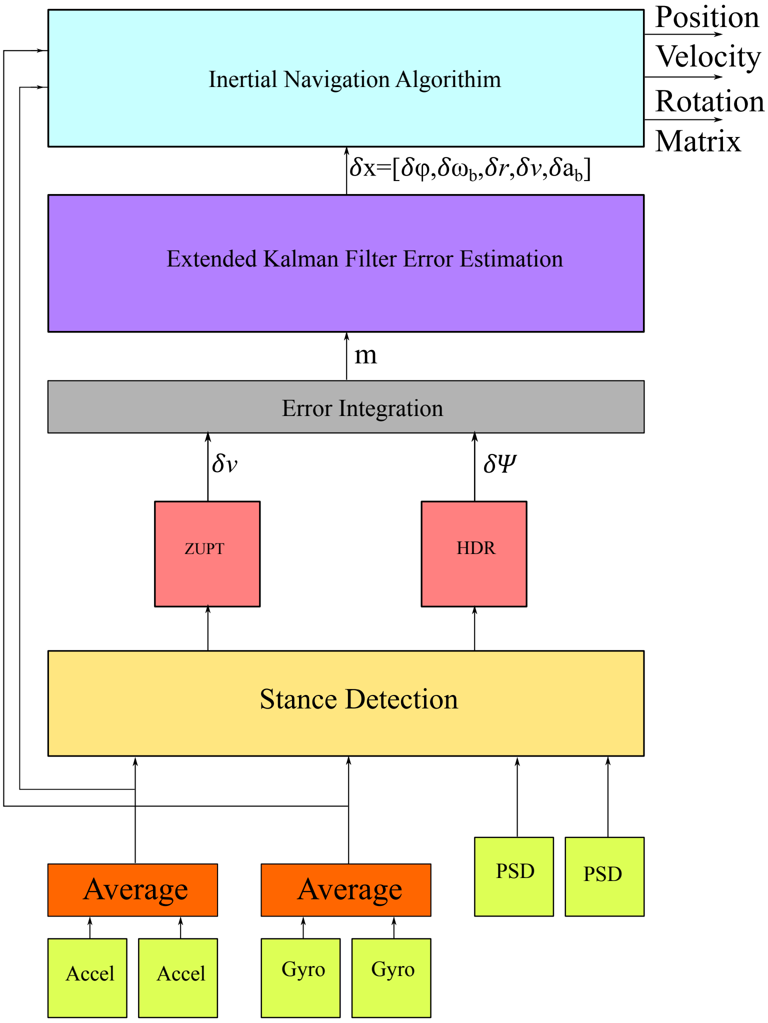
$$\hat{C}_2 = \begin{cases} 1, & \text{if } 9.3\,\mathrm{ms}^{-2} \leq a_{mag} \leq 10.3\,\mathrm{ms}^{-2} \\ 0, & \text{otherwise.} \end{cases}$$

Gyroscope

Lastly, the gyroscope data is examined to evaluate any potential foot motion. Should the magnitude of the calibrated gyroscope readings, ω_{mag}, be under

$20°\mathrm{s}^{-1}$, then the third and final condition is satisfied.

$$\hat{C}_3 = \begin{cases} 1, & \text{if } \omega_{mag} < 20°\mathrm{s}^{-1} \\ 0, & \text{otherwise.} \end{cases}$$



Inertial Navigation Algorithim
Position
Velocity
Rotation
Matrix
δx=[$\delta\varphi$,$\delta\omega_b$,δr,δv,δa_b]
Extended Kalman Filter Error Estimation
m
Error Integration
δv
$\delta\Psi$
ZUPT
HDR
Stance Detection
Average
Average
PSD
PSD
Accel
Accel
Gyro
Gyro

Figure 3.13: Block Diagram showing the software blocks for the EKF correcting the navigation algorithm from ZUPT and HDR measurements.

3.5 Evaluation

To test the performance of using an EKF to correct the calculated foot position three different types of walk were examined. The subject for all the walks was the author and the walks were conducted at a user selected walking speed.

1. The first type of walk was when the user travelled in a straight line forward for approximately 4.5 m, turned around, and then walked back. This type of walk is referred to as Type 1.

2. Secondly, a walk in which the user travelled in a rectangle measuring approximately 4x2 m was considered. The user walked around this rectangle three times before stopping. This type of walk is referred to as Type 2.

3. Lastly, a longer closed loop walk measuring approximately 55 m in length in which the user travelled in a typical indoor environment was examined. The path consisted of entering and exiting multiple rooms connected by a corridor. This type of walk is referred to as Type 3.

The three types of walks were evaluated differently. Types 1 and 2 were conducted in a motion laboratory and a Vicon system was used to track the position of the user. Type 3 walking was evaluated in terms of loop misclosure as discussed further in section 3.5.2.

3.5.1 Vicon Evaluation

The wearable sensing system generates results in the form of footstep positions. Therefore, to be able to use the Vicon system as a ground truth comparison the raw Vicon data needs to be converted to give a series of footstep locations.

To this end four markers were placed on the wearable system as shown in Figure 3.14. The marker labelled “Outer Marker” was used as a tracking reference with the other markers aiding in footstep detection and reconstruction of data should the Outer Marker be subject to flickering at any point during a trial.

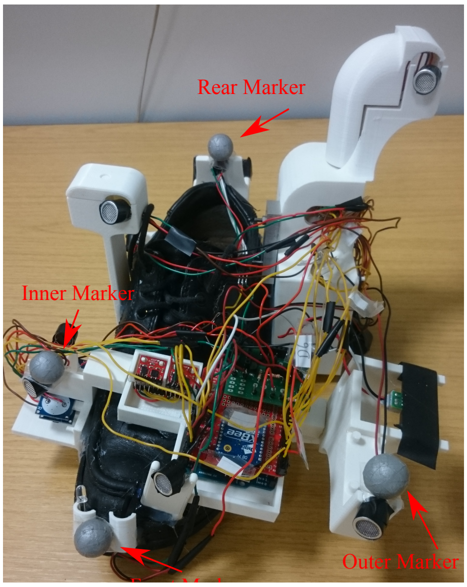


Figure 3.14: Picture showing the marker locations on the system.

To determine when a footstep occurs the 4 markers' vertical height was considered. If all visible markers were moving by less then 0.05 mm between successive samples then that sample was flagged as a quasi-stationary phase. To then identify which samples corresponded to footsteps the average vertical height of all four markers was taken and the varying baseline subtracted. A quasi-stationary sample was then considered a footstep sample if it lay within certain thresholds after the baseline was subtracted. The footstep samples are then grouped together to form footstep events. This is highlighted in Figure 3.15. The average x-y position of the footstep events is then taken as the foot's position for a particular footstep. If we further examine Figure 3.15 then we can gain a more quantitative value of the walking speed in the various trials. Figure 3.15 shows that to conduct 13 steps with the left foot around 6000 data points were gathered by the Vicon system. As the Vicon system gathers data at 200 Hz it corresponds to 2.3 s per left footstep. This is slightly slower than normal walking, and is due to the slight additional difficulty caused by navigating with the sensing system.

The position obtained from the Vicon system was then compared against the results from the IMU system. Two different error metrics were then cal-

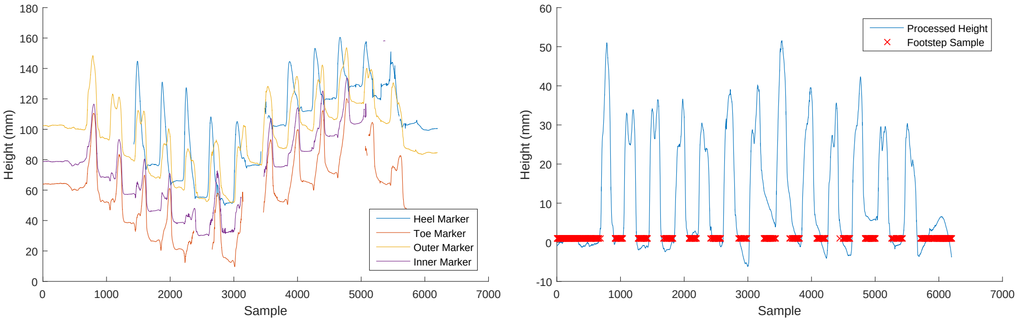


Figure 3.15: Graphs illustrating how the footstep position from the Vicon data is obtained. (Left) shows the vertical height of the four markers and if the change in height of all visible markers is less then 0.05 mm then the sample is considered to be quasi-static. (Right) Shows after further post processing and the application of thresholds the quasi-static samples that correspond to footsteps are found.

culated.

1. **Absolute Error**: This is obtained by calculating the distance between the Vicon and IMU results at every footstep. Hence, every time the footstep is detected the difference in position between the two systems is obtained.

2. **Percentage Error**: This is calculated by taking the absolute error as a percentage of the total distance travelled as given by the Vicon system.

Both of these metrics are illustrated in Figure 3.16.

The data was collected and subsequently post processed. As was mentioned in section 3.2.1 the navigation and EKF algorithms run at close to 3000 Hz on the utilised PC. Therefore, with more sophisticated data acquisition methods it will be feasible in future work to perform the motion tracking described in a real time manner.

It is important to remember that the shoe mounted system does not determine the initial heading or position. Rather, it tracks the motion relative to the starting point. In these experiments the initial heading is set to 0° and the initial x, y co-ordinates are also both set to 0 m. This introduces an inherent

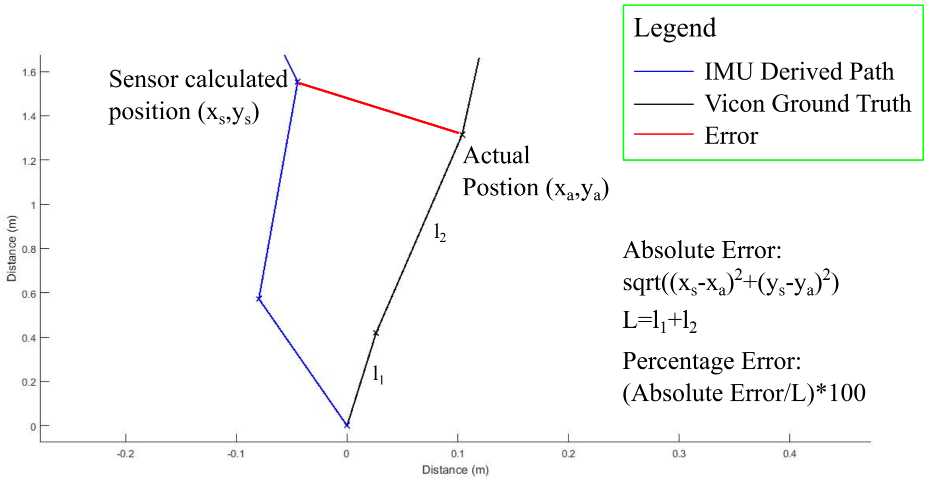


Figure 3.16: Diagram illustrating how the two different error metrics, absolute error and percentage error, are calculated.

error in all of the results as we assume that the IMU's x-axis is initially aligned with the forward direction of the user. Previous works that did not use magnetometers have applied post-processing rotations to the final results [27], or used ground truth information for alignment [68]. However, these techniques either rely on known movements or require external equipment to be present which is unrealistic in everyday walking environments. Hence, the results presented here represent more "rugged" operating conditions which could be more representative of real-world use.

3.5.2 Type 3 Walking Evaluation

Type 3 walking was evaluated slightly differently. As the walk went far outside the Vicon capture volume the absolute and percentage errors could not be calculated. Hence, it was evaluated in terms of its loop misclosure. Theoretically, as the walk was a closed loop, the final footstep should return the user to the starting position. The distance between the start and end point is thus taken as the error.

Although this type of metric is very common in evaluating IMU performance, when applied to pedestrian motion tracking it can be somewhat misleading. This is because many systematic errors accumulated when travelling will be re-compensated on the return journey. For example, should the system consistently over-estimate the step length, then at the end of a straight line walk the error will be at a maximum. However, when the user walks back to their starting position the system will carry on over-estimating the step length and so the final error will be small.

3.5.3 Results Type 1 Walk

Three Type 1 walks were conducted and the results are shown in Figures 3.20 to 3.25. We can see that as the user walks forward a drift in position occurs. Figures 3.21, 3.23 and 3.25 illustrate how the drift accumulates, both in absolute terms and in relation to the distance travelled by the user. They show that the error, in absolute value, was largest at the "tip" of the walk. The average maximum error is 0.402 m from the three trials, at the tip of the walk. This can be contrasted with a average final error of 0.263 m. This is, as mentioned in 3.5, because of the system compensating for systematic errors on the return walk. Additionally we can see the effect of the EKF in the velocity data in Figure 3.19. Any non-zero velocity when the foot is detected as being stationary is taken as an error and used to correct for the system parameters of orientation, velocity, position and sensor bias.

Trial	**Absolute Errors** (m)		
	Final Error	Maximum Error	**Final Percentage Error**
1	0.191	0.332	2.11
2	0.326	0.400	3.55
3	0.271	0.475	2.92

Table 3.3: Summary of results for Type 1 walking. Note that the final error is consistently smaller than the maximum error due to the systematic errors being compensated.

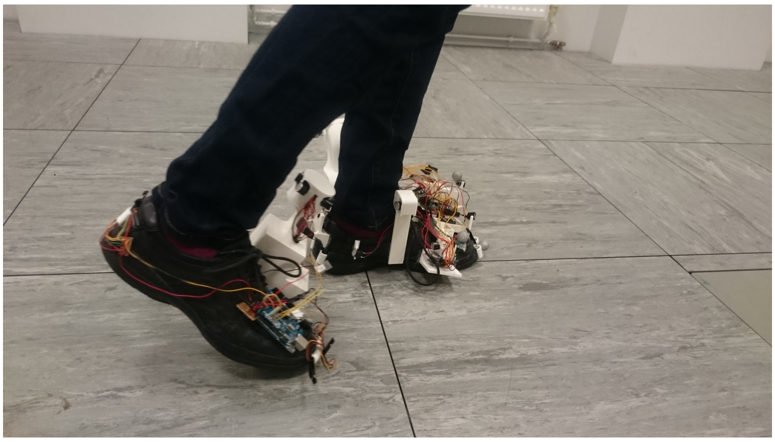


Figure 3.17: Image of the system undergoing validation. Here we can see the left foot is on the ground, signalling that ZUPT and HDR corrections should be applied.

Figure 3.18: Image of the system undergoing validation. Here we can see the left foot is on the ground, signalling that ZUPT and HDR corrections should be applied.

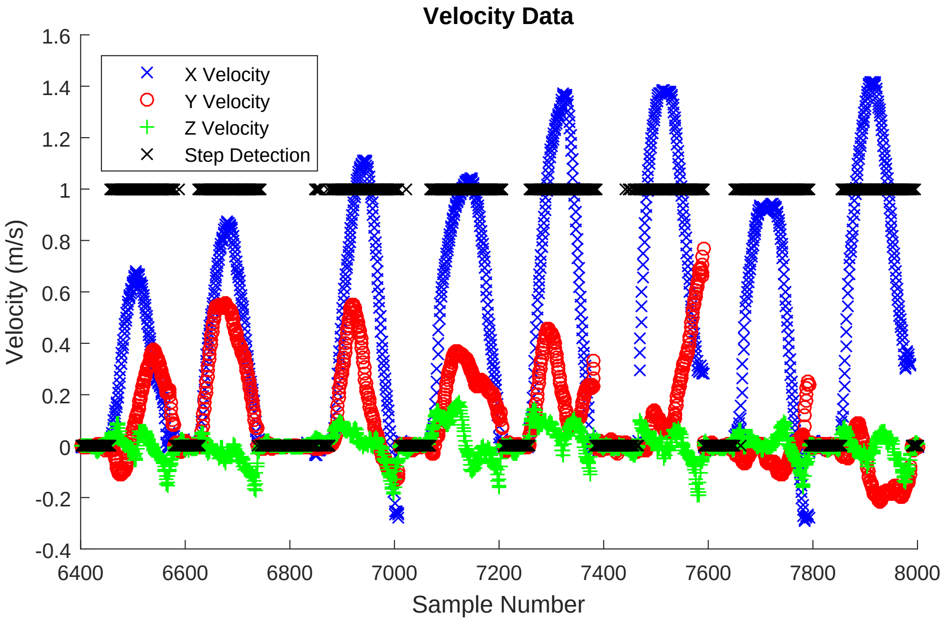


Figure 3.19: Velocity data from part way through a walk. This shows the effect of the EKF corrections on the underlying data. If the black markers labelled ‘Step Detection’ have a value of 0 it signals a EKF correction is to be made, whereas if they have a value of 1 it indicated the foot is in a stride phase. We can see that at the end of the stride phase if the velocity is non-zero it is taken as an error an corrected. Though the EKF algorithm described earlier the velocity errors are correlated to errors in orientation, position and sensor bias which enables these other system parameters to be corrected.

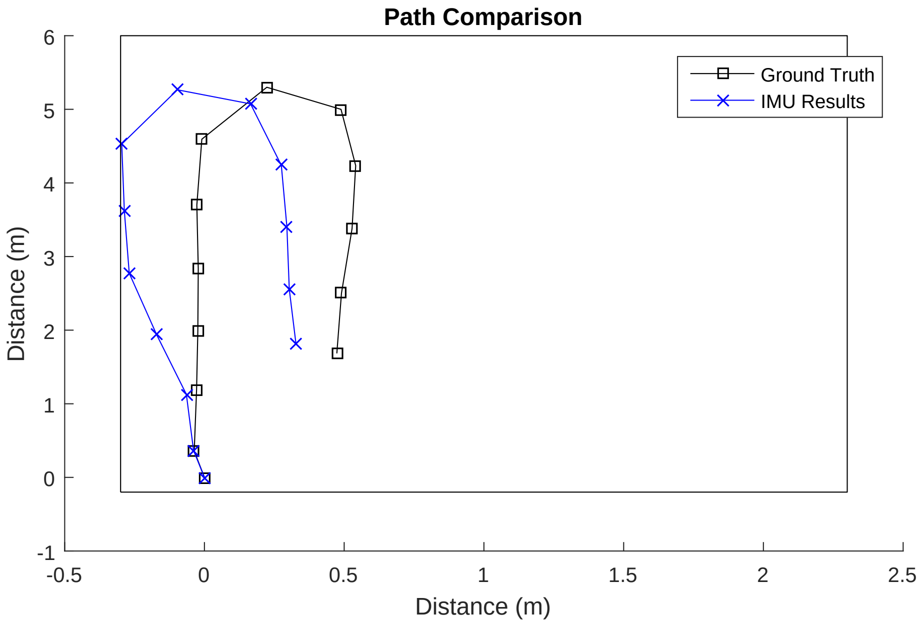


Figure 3.20: Path comparison between the IMU and Vicon systems for trial 1. The black border represents the constraints of the capture volume.

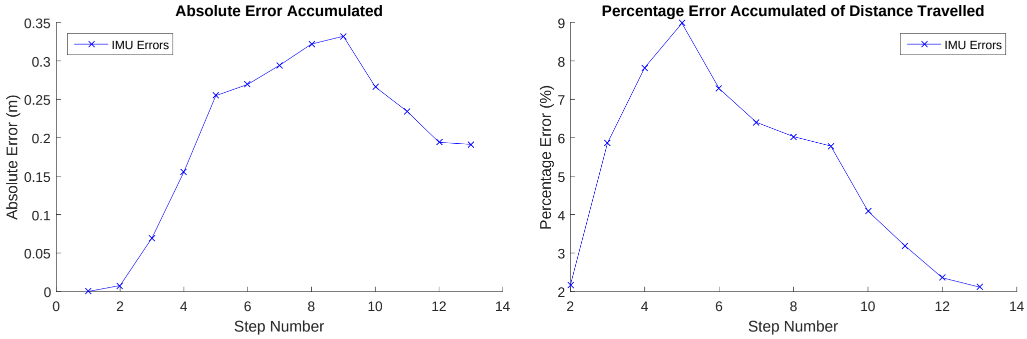


Figure 3.21: Percentage and absolute errors for the first Type 1 walk. Details of how the errors are computed are given in Figure 3.16.

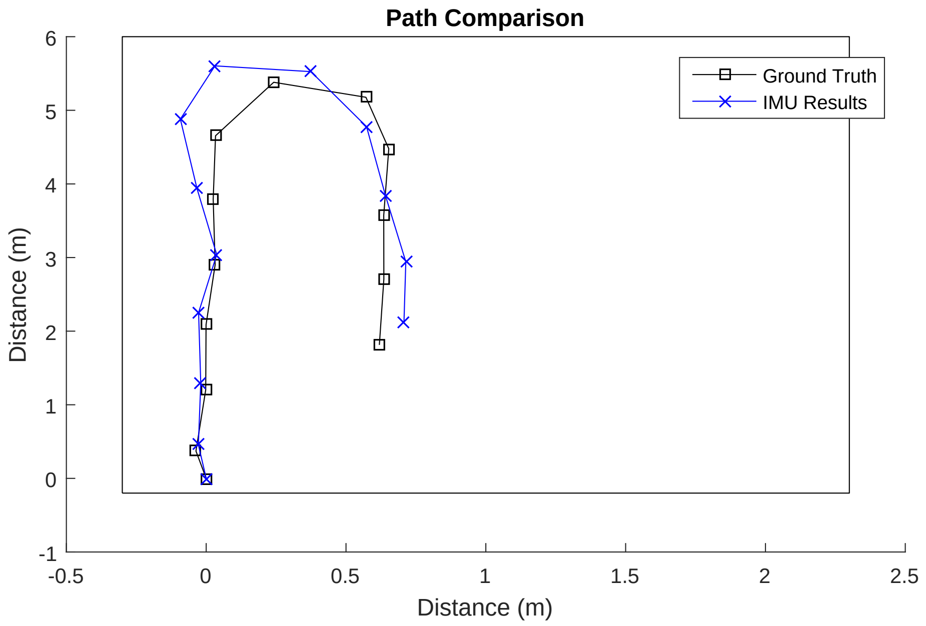


Figure 3.22: Path comparison between the IMU and Vicon systems for trial 2. The black border represents the constraints of the capture volume.

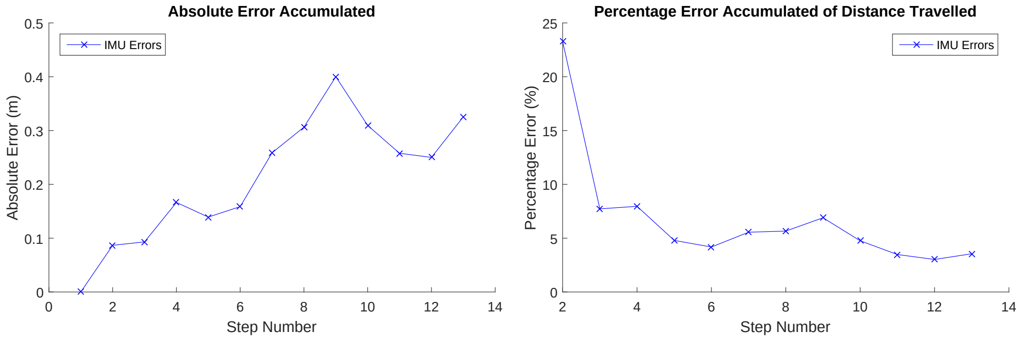


Figure 3.23: Percentage and absolute errors for the second Type 1 walk. Details of how the errors are computed are given in Figure 3.16.

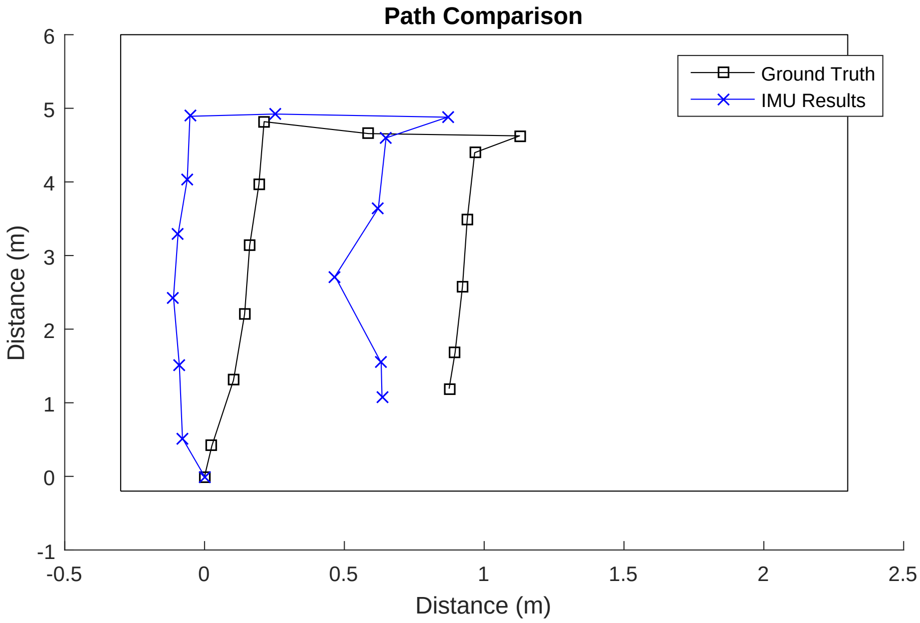


Figure 3.24: Path comparison between the IMU and Vicon systems for trial 3. The black border represents the constraints of the capture volume.

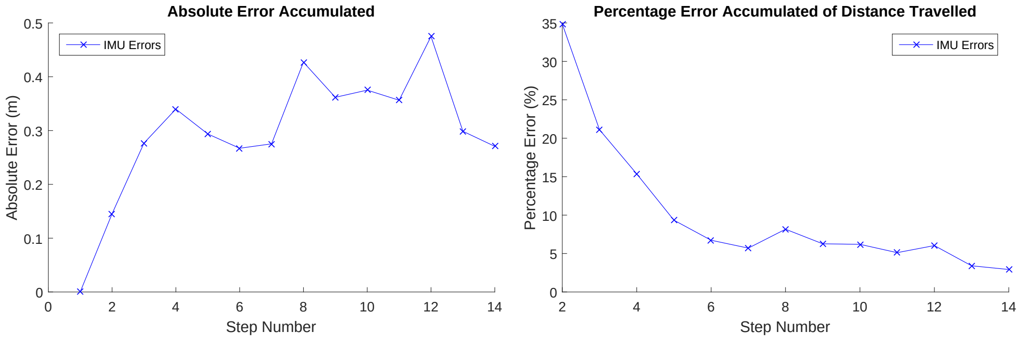


Figure 3.25: Percentage and absolute errors for the third Type 1 walk. Details of how the errors are computed are given in Figure 3.16.

3.5.4 Results Type 2 Walk

Type 2 walks were then examined. In total, three type 2 walks were conducted. Figures 3.26 to 3.31 show the results.

We can see from the figures the effect of walking in a closed loop path, and how it artificially reduces the error. When a loop is completed, or in some cases shortly afterwards, the error decreases rapidly. This suggests that the testing methodologies of many previous works, which evaluated the performance solely based on loop closure, were somewhat inadequate. This is due to the fact that the error will have been consistently measured at its lowest point, and so does not give a true indication of the accuracy of the various systems. This effect can be significant with the percentage error oscillating between 4% and 0.5% in a typical cycle.

We can also note from a visual inspection of the path comparison that the system displayed a systematic error where the position drifted to the user's left. This systematic error, which is present to a much lesser degree in the Type 1 walk, may be due to several sources. Clearly the initial heading error will be contributing. In addition, residual biases will be contributing to systematic yaw drift.

For Type 2 walks Tables such as 3.3 are not included as there are several loop closures and local maxima. Due to this, for Type 2 walks the results are better represented by considering Figures 3.26 to 3.31 directly.

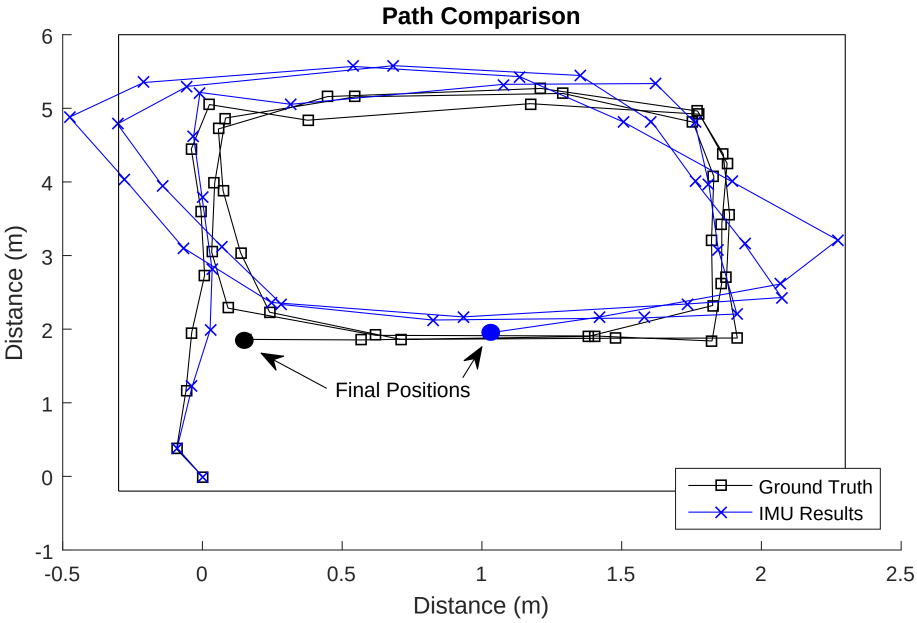


Figure 3.26: Path comparison for trial 1 of the ground truth position as given by a Vicon system and the motion as given by the IMU system.

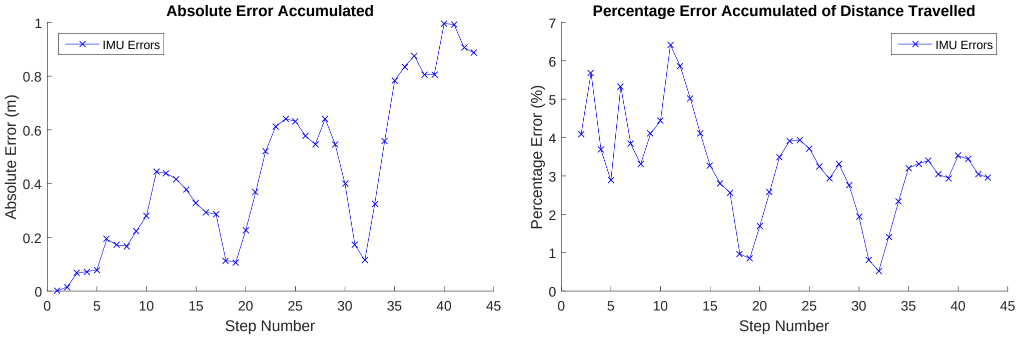


Figure 3.27: Percentage and absolute errors for the first Type 2 walk. Details of how the errors are computed are given in Figure 3.16.

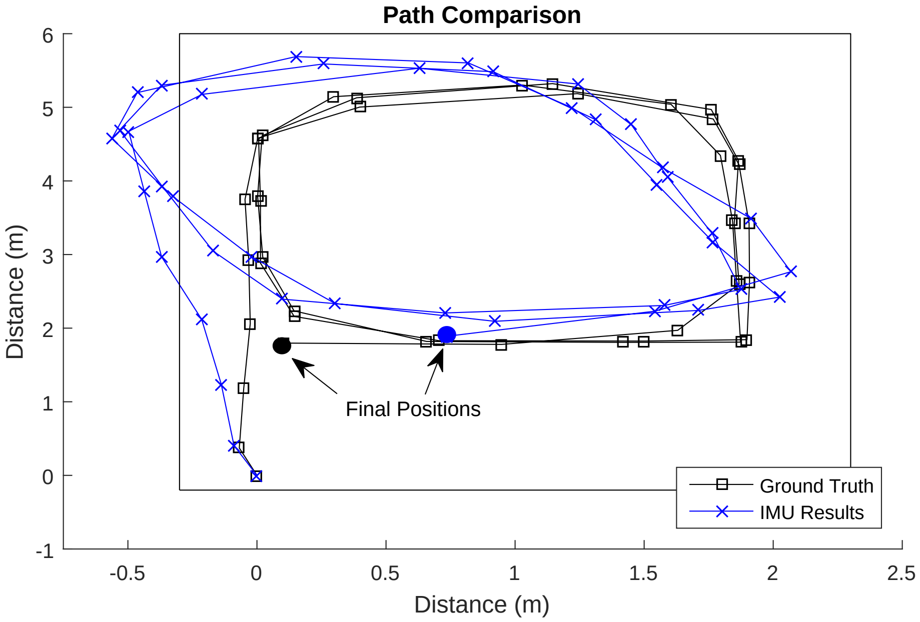


Figure 3.28: Path comparison for trial 2 of the ground truth position as given by a Vicon system and the motion as given by the IMU system.

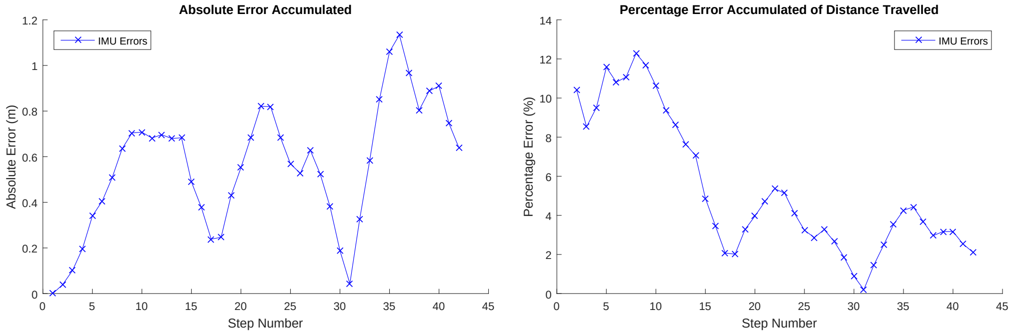


Figure 3.29: Percentage and absolute errors for the second Type 2 walk. Details of how the errors are computed are given in Figure 3.16.

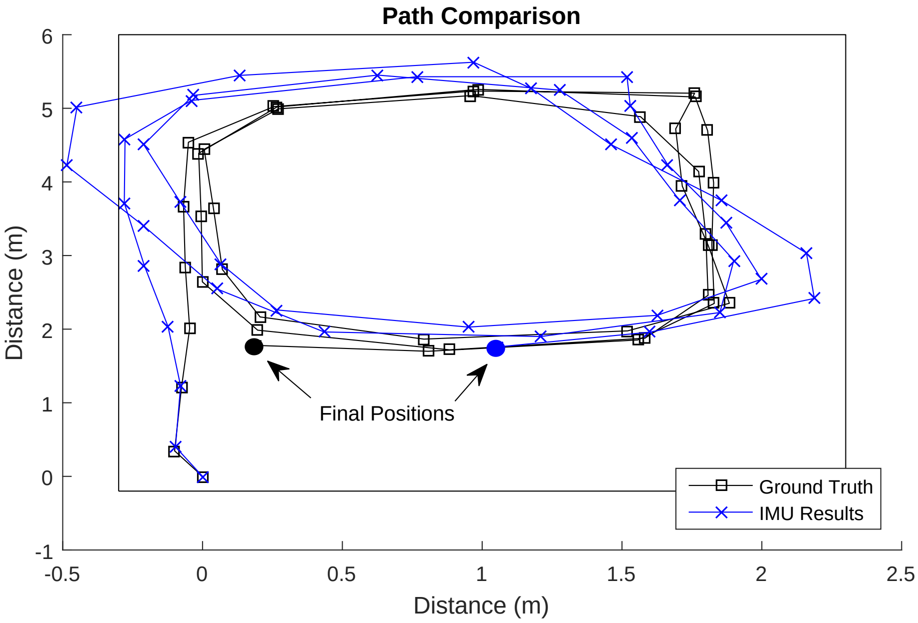


Figure 3.30: Path comparison for trial 3 of the ground truth position as given by a Vicon system and the motion as given by the IMU system.

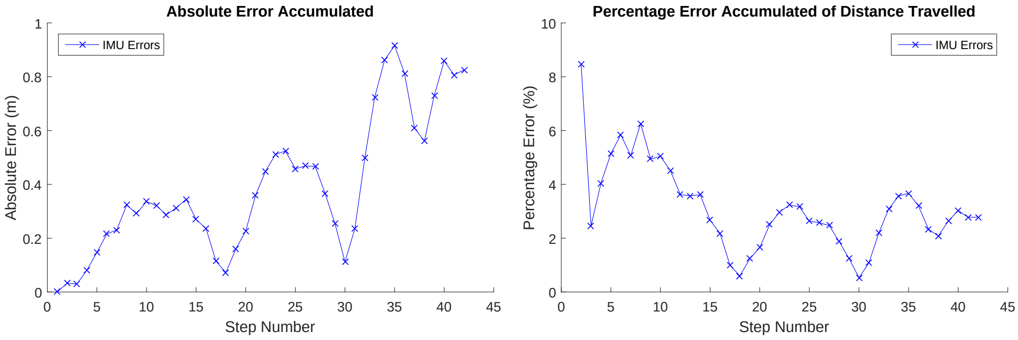


Figure 3.31: Percentage and absolute errors for the third Type 2 walk. Details of how the errors are computed are given in Figure 3.16.

3.5.5 Results Type 3 Walk

For the Type 3 walk we were unable to use the Vicon capture system as the walk went far outside the Vicon capture volume. The purpose of the walk was to examine the performance in a typical indoor environment and consisted of entering and exiting multiple rooms.

Hence, we use the previous testing methodologies of evaluating its performance in terms of loop misclosure. However, as previously discussed, this is a vague indication of performance, due to error cancellation in closed loop paths, and results in the system errors being understated. Therefore, we also make an estimate of the maximum error on each trial, but without accurate ground truth data it is only to be taken as a rough figure. This type of walk was repeated three times and the paths as measured by the IMU system are shown in Figures 3.32 to 3.33. As we can see, trial 3 showed more significant error with the heading estimate being incorrectly calculated. This caused the path to skew and end up rotated in an incorrect position. Again, we can observe why loop closure is an overestimate of system accuracy. In trial 3 the path went out of the building constraints by a maximum value of 1.035 m, implying that, at that particular instance there was at least a 1.035 m error. However, the final loop close error was only 0.553 m.

The loop close errors for all three walks are summarised in Table 3.4.

Trial	**Absolute Error** (m)		
	Misclosure	Maximum Error	**Percentage Misclosure**
1	0.176	0.47	0.32
2	0.319	1.19	0.58
3	0.553	1.78	1.01

Table 3.4: Table showing the comparison for the loop misclosure and maximum error for the different type 3 walks. In this case accurate ground truth data was unavailable and so the maximum error is an estimate. The percentage misclosure error is the misclosure expressed as a percentage of the total distance travelled.

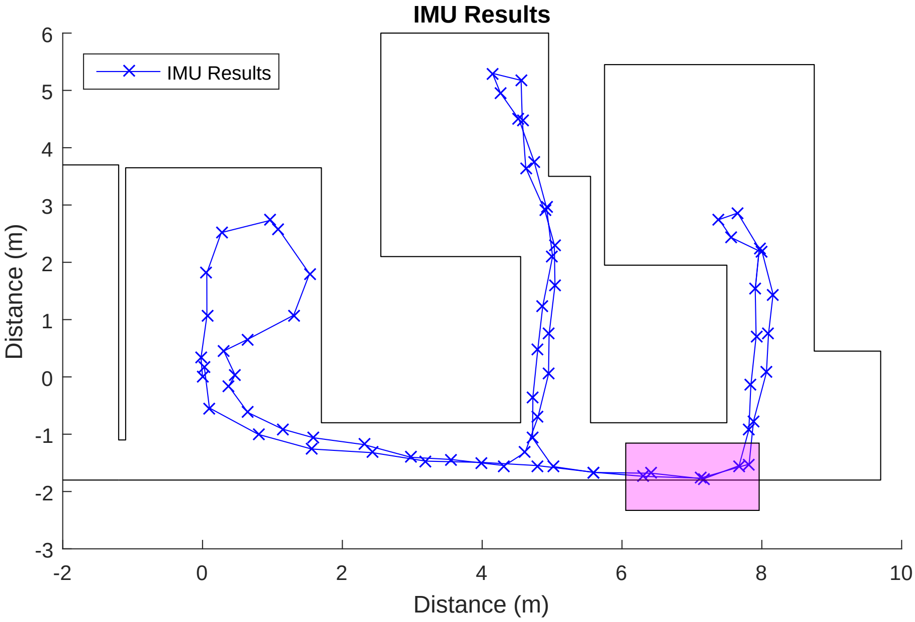


Figure 3.32: Predicted path trajectory for the first Type 3 walk. In this trial the system showed extremely strong performance The only notable error is that the path went too close to the wall in the highlighted area.

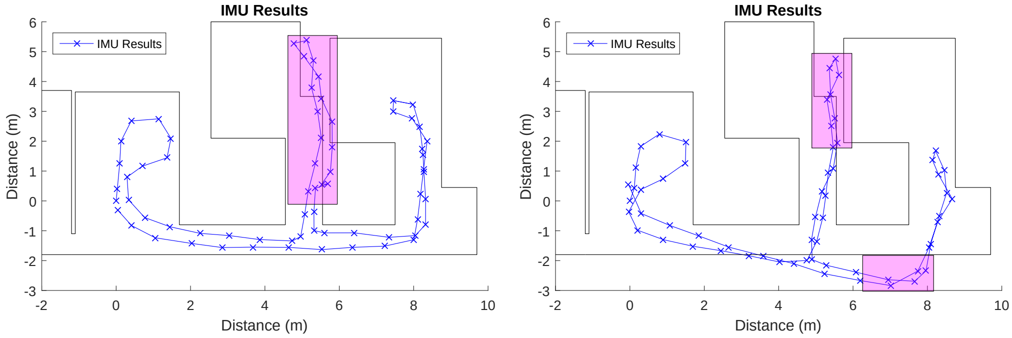


Figure 3.33: (Left) the second trial for Type 3 walking. The system showed generally good performance with the most significant error occurring in the centre room where the path want outside of the building constraints. This is highlighted in magenta. (Right) Shows the last Type 3 walk conducted. This was the weakest of the trials with the path being significantly skewed due to yaw drift. Although the final loop closure was rather small, we can see from a visual inspection of the path that significant errors occurred, with a maximum error in the top highlighted area, but also very significant error in the bottom highlighted area.

3.6 Magnetometers

Magnetometers could be a powerful tool for providing heading corrections. It is known that magnetic fluctuations indoors can case large heading errors. In literature there are several proposed techniques which can reduce the impacts of soft iron effects however, in the context of pedestrian motion tracking they have not been applied extensively. Thus the use of magnetometers was briefly investigated in this thesis as a potential supplementary sensor. Following the procedure as described in [33] a HMC5883L sensor was calibrated. However, using just a single magnetometer performed poorly indoors with the large magnetic fluctuations. Without developing the full magnetometer array, as was done in [33], the data was of too low quality to use effectively.

3.7 Conclusions

This chapter has presented a method to calibrate the IMUs without the need to use external equipment, enabling end user's to periodically re-calibrate their systems. Due to a lack of traditional IMU calibration equipment, it was not possible to compare the equipment free method to the results using standard approaches for the gyroscope. By using gravity as a reference signal the two accelerometers' calibration can be examined. The average error over 5 trials was found to be $0.0324\,\mathrm{ms}^{-2}$ and $0.0196\,\mathrm{ms}^{-2}$ for the first and second accelerometers.

An EKF scheme which uses ZUPT and HDR as error reduction techniques was implemented. In terms of loop closure it performed very strongly in a Type 3 walk with a maximum loop closure error of 0.553 m with an average of 0.349 m. This demonstrates that the relative motion of the user from a starting position can be tracked well for a walk of a few minutes in duration. However, due to the accumulation of error the quality of the information will degrade over time.

Furthermore, Type 2 walks showed similar levels of accuracy to Type 3 walks in terms of loop misclosure as the average loop misclosure in a Type 2 walk was 0.9055% against Type 3 loop misclosure of 0.636%. Additionally we have illustrated with the Vicon data for Type 2 walks that the testing methodologies of several previous works, most notably [17], [13] and [21], which evaluated the performance of their system in terms of a closed loop path, will give artificially accurate results. The cyclic nature of closed loop walks will lead to errors being compensated from the nature of the walk, rather than the accuracy of the system employed. This was a significant effect with the percentage error cycling between 0.9% and 4% over a typical walk.

The key contributions of this chapter have thus been to investigate and analyse the performance of low cost IMUs and see if they are comparable to higher end systems. Our low cost approach has performed better than certain works such as [13] obtaining 2-10% loop closure when running ZUPTS and HDR corrections. On the other hand it was slightly outperformed by other works such as [17] and [27] which had errors of 0.3% and 0.5% respectively. This indicates that lower end IMUs can, with proper treatment, provide comparable results to systems running more sophisticated sensors.

Chapter 4

Ultrasound

Ultrasound sensors are popular in localisation and tracking applications due to their low cost and power consumption while still offering high quality data. In certain environments, such as underwater, they are often the only viable sensor.

Having examined the performance of inertial navigation using an EKF we now turn to ultrasound as a potential means of gaining a more accurate measurement of step displacement. The operating principle behind ultrasound sensors is examined, before evaluating their performance.

4.1 Piezoelectric Effect

Sound waves at a frequency higher than that of human hearing are classified as ultrasound waves. Ultrasound devices operate from 20 kHz to several megahertz depending on their application. Ultrasound sensors work via the use of the piezoelectric effect, which is the phenomenon of electrical charge accumulating via the mechanical deformation of a material. It is a reversible process and substances that exhibit this phenomenon are referred to as piezoelectric materials. Piezoelectric materials work via the interplay of electrical, mechanical, and thermal effects thus requiring many parameters to be fully described.

However, we can give a qualitative description of their operation. First, the ceramic transducers are poled, that is, the dipole moments are aligned via an electric DC field giving the ceramic an overall polarisation. This polarisation gives the ceramics properties that can be exploited to produce ultrasound sensors. A mechanical deformation of the ceramic alters the dipole moment generating a voltage. Should the deformation be compression then a voltage, of the same polarity as the poling voltage, is created. Vice versa tension causes a voltage with polarity opposite to that of the poling voltage. In this scenario the transducer is acting as a receiver, detecting incoming ultrasound waves which deform the ceramic and thus induce a voltage.

Should the transducer function as a transmitter then a voltage must be applied to the transducer. If the applied voltage is of the same polarity as the poling voltage then the ceramic will elongate and its radius narrow. Similarly, if the voltage is opposite in polarity to the poling voltage the ceramic's length will shrink and its radius will increase, By applying an alternating voltage the rapid changes in dimension causes ultrasound waves to be generated and thus the transducer acts as a transmitter [69].

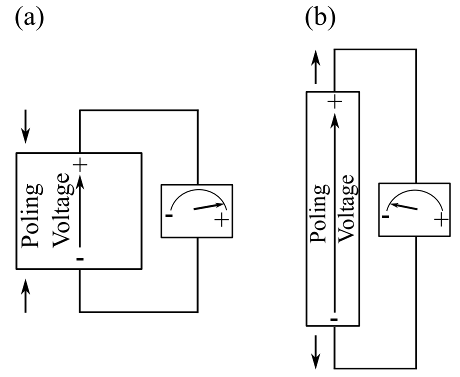


Figure 4.1: (a) Shows how compression of the piezoelectric material causes a voltage in the same direction to that of the poling voltage. (b) Demonstrates how tension creates a voltage in the opposite direction to the poling voltage.

The idea behind ultrasonic range finding is simple to conceptualise. An AC current is applied and thus generates ultrasonic waves. The ultrasound pulse then reflects off a distant object and returns to the ultrasound sensor. By deforming the piezoelectric material an electrical signal is generated that upon amplification can be readily detected. Thus, by measuring the elapsed time between the pulse being transmitted and then received the distance that the ultrasound pulse travelled can be calculated.

4.2 Ultrasound Propagation

Once the transducer has produced an ultrasound pulse it propagates through the air as a longitudinal wave. Being a mechanical wave, its speed of propagation is affected by the material it is travelling though. If we make the assumption that the medium follows the ideal gas law then the speed of sound, c, can be expressed as

$$c = \sqrt{\frac{\gamma RT}{M}} \tag{4.1}$$

were γ is the adiabatic index, R is the molar gas constant and M is the molar mass of the gas.

For the majority of cases we can take the speed of sound to be $343.2\,\mathrm{ms}^{-1}$, corresponding to the average molecular mass of air at a temperature of 20°C. Of potential concern is fluctuations in the temperature changing the speed of sound from our value at 20°C and thus rendering the time of flight calculations inaccurate. However, within a normal working environment this effect is minimal. Temperature variations will only begin to have an effect in very extreme cases, such as tracking firefighters or other emergency rescue workers due to the severe heat. Temperature corrections can be applied via the use of a thermometer however, for the operating conditions under which the tracking system will be tested in this thesis, such corrections are unnecessary.

4.3 Reflection

All ultrasound systems are affected to various degrees by noise interference and reflections. Noise effects are typically easier to deal with and can be minimised by operating at a frequency without much external interference and filtering around that frequency [70]. Reflection of the emitted pulse can cause the sound to travel via a longer route than a direct path and make the object appear further away than it actually is. Hard surfaces such as walls and flooring can reflect an ultrasound wave with sufficient amplitude to be detected by a receiver.

This only becomes a problem should line of sight between the transmitter and the receiver be lost. If line of sight is maintained then the first pulse to be picked up by the receiver will be the one that travelled the shortest path and therefore be the true distance. Any fraction of the emitted ultrasound wave which scattered off objects and arrives at a later time can be discarded. However, should line of sight be lost then it is possible for the ultrasound pulse to reflect off multiple objects and still be detected by the receiver as shown in Figure 4.2.

In the case of foot tracking, line of sight blockages are relatively rare. Typically, should line of sight be lost the signal is completely blocked, for example in the case of travelling up a flight of stairs. To reduce potential problems arising from reflections we can use prior knowledge of typical gait lengths and reject values which are significantly larger than the distance a person could travel in a single step.

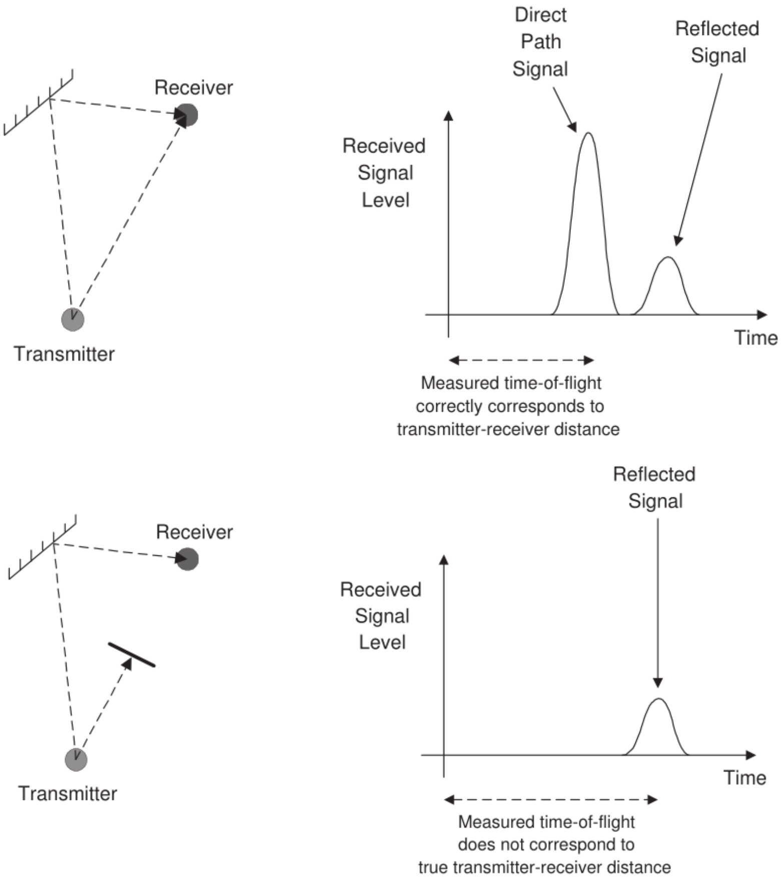

Receiver
Transmitter
Direct Path Signal
Reflected Signal
Received Signal Level
Time
Measured time-of-flight correctly corresponds to transmitter-receiver distance
Receiver
Transmitter
Reflected Signal
Received Signal Level
Time
Measured time-of-flight does not correspond to true transmitter-receiver distance

Figure 4.2: Illustration of how ultrasound reflections can cause the measured length to be much larger [70].

4.4 Ultrasound Navigation via Trilateration

In order to improve upon the results given by the inertial navigation algorithm a complementary ultrasound navigation system runs in parallel. It functions via trilateration with five receivers and one transmitter. Due to the limited angular width of the ultrasound receivers and transmitters there are two trilateration systems one functioning when the left foot is in front, and the other when the right foot is in front as demonstrated in Figure 4.3.

As stated in Section 2.6, the problem of trilateration can be expressed as that of determining an object's location, $O(x, y, z)$, which is the intersection of the spheres centred on a beacon's position, $B(x_i, y_i, z_i)$, via

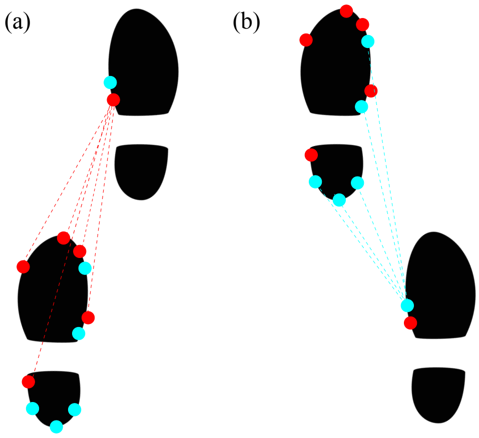


Figure 4.3: (a) Illustrates that when the right foot is forward the ultrasound sensors marked in red are active and are facing the correct direction. (b) Demonstrates when the left foot is forward the ultrasound sensors marked in blue are providing the trilateration data.

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_i^2 \tag{4.2}$$

in which r_i is the range measurement form the $i\,th$ beacon.

Through manipulation of the above equation we can express them in the following linear system

$$\mathbf{A}\hat{\mathbf{x}} = \mathbf{b} \tag{4.3}$$

where

$$\mathbf{A} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & x_3 - x_1 & z_3 - z_1 \\ \vdots & \vdots & \vdots \\ x_n - x_1 & x_n - x_1 & x_n - x_1 \end{bmatrix} \tag{4.4}$$

$$\hat{\mathbf{x}} = \begin{bmatrix} x - x_1 \\ y - y_1 \\ z - z_1 \end{bmatrix} \tag{4.5}$$

$$\mathbf{b} = \frac{1}{2}\begin{bmatrix} r_1^2 - r_2^2 + (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \\ r_1^2 - r_3^2 + (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 \\ \vdots \\ r_1^2 - r_n^2 + (x_n - x_1)^2 + (y_n - y_1)^2 + (z_n - z_1)^2 \end{bmatrix} \tag{4.6}$$

Theoretically without noise or interference only 4 beacons are needed to obtain $O(x, y, z)$ in a 3 dimensional case. However, when errors arise in both the range measurements and the positions of $B(x_i, y_i, z_i)$ then any solution calculated through the above would be inaccurate. Hence, a more sophisticated treatment is required and we turn to linear and non-linear least squares to deal with noisy measurements.

Considering a 2D case we define the vector R to be,

$$R = \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.7}$$

we can use the least squares method to give a good initial estimate of the foot location, R_1:

$$R_1 = (A^T A)^{-1} A^T b \tag{4.8}$$

To improve the results further we can use a non-linear least squares algorithm as presented in [49] and [71]. If $\hat{r_i}$ is the actual distance between the beacon and the object, then to minimise the square of the errors the function

$$F(x, y, z) = \sum_{i=1}^{n} (\hat{r_i} - r_i)^2 \tag{4.9}$$

must be minimised. To do this, an initial guess is obtained using the linear least squares algorithm and a Newton iterator is then used to find an optimal solution. Differentiating the above equation gives

$$\frac{\partial F}{\partial x} = 2 \sum_{i=1}^{n} f_i \frac{\partial f_i}{\partial x} \tag{4.10}$$

where f_i is

$$f_i(x,y,z) = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} - r_i \tag{4.11}$$

by defining the vector $\hat{\mathbf{g}}$ as

$$\hat{\mathbf{g}} = 2\mathbf{J^T}\hat{\mathbf{f}} \tag{4.12}$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \\ \vdots & \vdots \\ \frac{\partial f_n}{\partial x} & \frac{\partial f_n}{\partial y} \end{bmatrix}, \hat{\mathbf{f}} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \hat{\mathbf{g}} = \begin{bmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \end{bmatrix}. \tag{4.13}$$

Then Newton iteration gives

$$R_{k+1} = R_k - (\mathbf{J_k^T J_k})^{-\mathbf{1}} \mathbf{J_k^T} \hat{\mathbf{f}}_\mathbf{k} \tag{4.14}$$

Equation 4.14 is iterated 50 times to achieve convergence of the results.

Once the x and y displacement have been calculated for an individual footstep they must be transformed into the global co-ordinate frame. The total step length as obtained from the ultrasound is given by,

$$L_{US} = \sqrt{\Delta x^2 + \Delta y^2}. \tag{4.15}$$

where Δx and Δy are the displacements in the local frame of reference of a particular footstep. Therefore L_{US} is then rotated to the global co-ordinate system though,

$$x_k^{US} = L_{US} cos(\theta_k - \eta) \tag{4.16}$$

$$y_k^{US} = L_{US} sin(\theta_k - \eta) \tag{4.17}$$

where θ_k is the yaw at the end of the footstep and η is the difference between the user's final heading and the direction of the step [68]. These angles are illustrated in Figure 4.4.

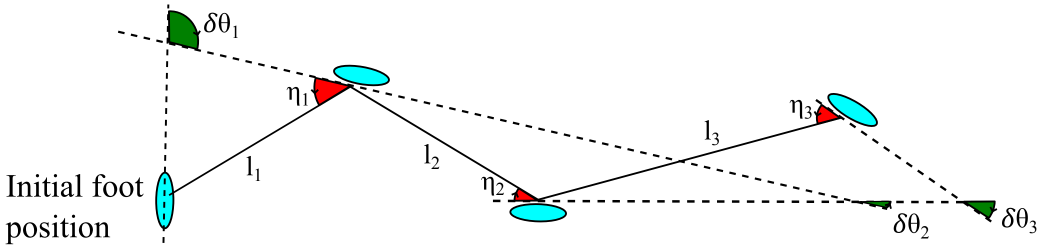


Figure 4.4: Diagram illustrating the parameters $\delta\theta$, the change in yaw for every step and η, the difference between the user's final heading and the direction of the step, for three separate steps l_{1-3}. θ_k is thus obtained by the sum of all the successive yaw changes. The blue ellipses represent the user's foot and the dashed lines indicate the user's heading after taking a footsetp. Therefore, to calculate the direction in which l_n must be projected the angle η must be obtained.

4.4.1 Sensor Weighting

Once x_k^{US} and y_k^{US} are obtained, which correspond to the displacement over a single step according to the ultrasound system, they are combined with the step displacement as given by the IMU via a maximum likelihood estimation (MLE).

To determine the relative weights we begin by stating that the error in the global co-ordinate frame for a single step is proportional to the displacements in the x and y directions.

Additionally, the accuracy of the ultrasound system was found to reduce as the change in yaw over a single footstep, $\delta\theta_k$, increased. Therefore, the errors assigned to the IMU data were multiplied by $cos^2(\delta\theta_k)$. In other words, we increase the weightings on the IMU data in relation to the ultrasound depending on the value of $\delta\theta_k$. It should be noted that, while several checks to ensure data validity have been implemented, due to the approximate nature of how the sensor weightings are carried out, if poor ultrasound data is fed into the algorithm it may have an undesired effect of degrading the results. In other words improvement due to using the ultrasound sensors is not a mathematical certainty, rather an experimentally derived observation.

4.5 Ultrasound Navigation Summary

The ultrasound system has several steps which enable it to offer an improved step length measurement. This is then combined with the IMU results as shown in Figure 4.5. The ultrasound system is an optional step, and can be skipped should the cost of the additional sensors be too much for a potential user.

The ultrasound algorithm is shown in a pseudo-code fashion in Algorithm 1, which illustrates the mathematics of the system and how it was implemented.

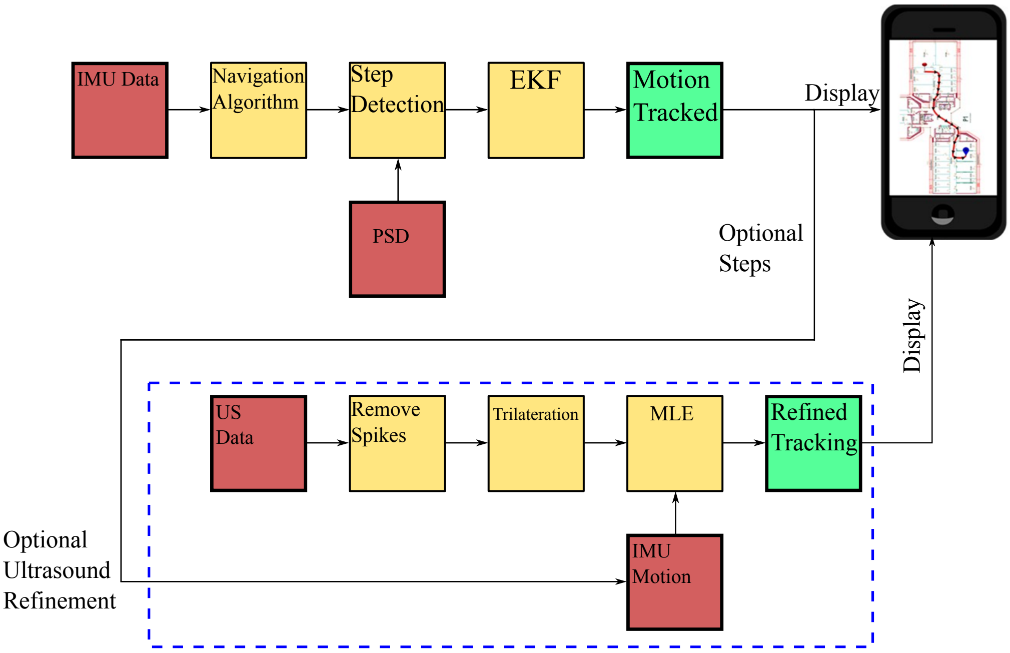


Figure 4.5: Block diagram illustrating how the optional ultrasound system interacts with the core IMU system [59].

Algorithm 1 Ultrasound Algorithm

```
procedure ULTRASOUND($\mathrm{US}_i$, $x_{i-1}, y_{i-1}, \eta, \theta_i, \delta\theta_i, x_i^{IMU}, y_i^{IMU}$)

    Description of input data


    $\mathrm{US}_i$                      ▷ Raw ultrasound range measurements for step $i$
    $x_{i-1}, y_{i-1}$                   ▷ Position calculated for previous step
    $\theta_i, \delta\theta_i$           ▷ Angle information for step $i$
    $x_i^{IMU}, y_i^{IMU}$               ▷ Position calculated for step $i$ according to IMU data.

    Start of algorithm


    Check that sufficient data is present to compute the step length
    $Valid \leftarrow Check\_Valid\_DataSet(\mathrm{US}_i)$
    if $Valid = true$ then
        Filter Range measurements for spikes
        $Filtered\_Data \leftarrow Filter\_Extremes(\mathrm{US}_i)$

        Obtain step length
        $L_{US} \leftarrow \mathrm{Trilateration}(\mathrm{US}_i)$

        Rotation
        $x_i^{US} = x_{i-1} + L_{US} cos(\theta_i - \eta)$
        $y_i^{US} = y_{i-1} + L_{US} sin(\theta_i - \eta)$

        Maximum Likelihood Estimation
        $(x_i, y_i) \leftarrow MLE(x_i^{US}, y_i^{US}, x_i^{IMU}, y_i^{IMU}, \delta\theta_i)$
    end if

    Return $(x_i, y_i)$
end procedure
```

4.6 Ultrasound Sensors and Setup

The ultrasound modules used are a modified HC-SR04 sensors. These ultrasound sensors have separate transmitter and receiver transducers and have a maximum range of 4 m with a 2 cm minimum measuring distance.

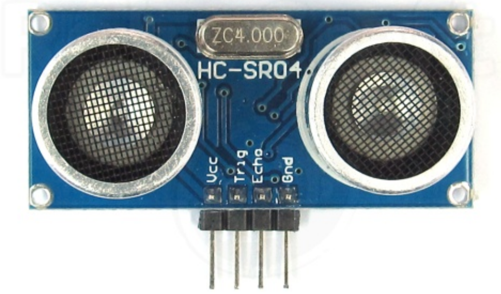


Figure 4.6: The HC-SR04 ultrasound sensor. The transducer to the right is the receiver while that on the left is the transmitter. Additionally, we can see the crystal clock mounted across the top as well as the trigger and echo pins.

The HC-SR04 modules have four pins: ground, V_{in}, trigger, and echo. The HC-SR04 operate by waiting for a signal on the trigger pin. When the pin is set to high for at least $10\,\mu s$ it signals a range measurement is to be made. The transmitter then emits 8 cycles of 40 kHz ultrasonic pulses. The HC-SR04 then raises its echo pin and waits for the reflected ultrasound wave. Upon receiving the returning ultrasound the echo pin is then brought to low. By timing the duration that the echo pin was set to high, the distance travelled by the ultrasound can be obtained. This cycle is illustrated in Figure 4.7.

The HC-SR04 modules are originally designed to have the transmitter and receiver on the same module. However, for our application we require separate transmitter/receivers. To achieve this the transmitter on a HC-SR04 was covered. This prevents the ultrasound pulse from propagating. The echo pin is still raised to high and awaits a ultrasound pulse to arrive at it. Hence, the HC-SR04 effectively acts as a standalone receiver. By employing another HC-SR04 to act as a standalone transmitter, a transmitter - receiver combination is achieved and the distance between them can be obtained.

The devices are synchronised via an infra-red (IR) pulse from a IR LED

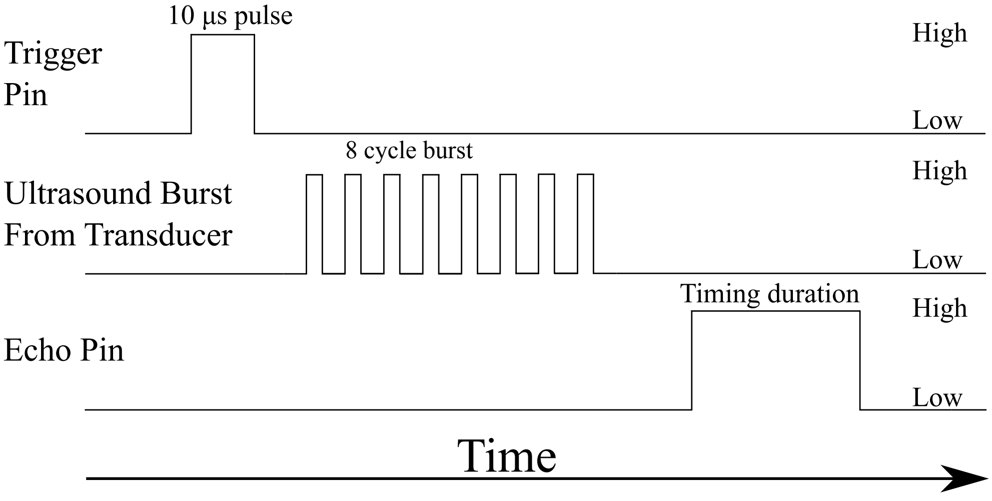


Figure 4.7: HC-SR04 Timing Diagram.

mounted on the left shoe which has the ultrasound receivers. As soon as the IR pulse is sent the receivers are set to begin timing. Likewise, as soon as the IR pulse is detected on the right shoe which has the ultrasound transmitters the ultrasound pulse is sent.

The IR pulse is picked up by a TSOP4038 IR receiver. The IR receiver is set to only receive signals which operate at 38 kHz. This has the advantage of filtering out ambient sources of IR radiation, such as filament light bulbs, and so provide a more reliable detection method.

In total 16 HC-SR04 modules are used; there are ten receivers and six transmitter types. They are arranged so that both the forwards and backwards directions on each foot are covered. The six transmitters are arranged into two clusters of three, one pointing forward and the other backwards. This was done so that the three transmitters would act as a single, larger transmitter with a much wider beam angle. Both the block diagram and the physical layout of the system are shown in Figures 4.8 and 4.9.

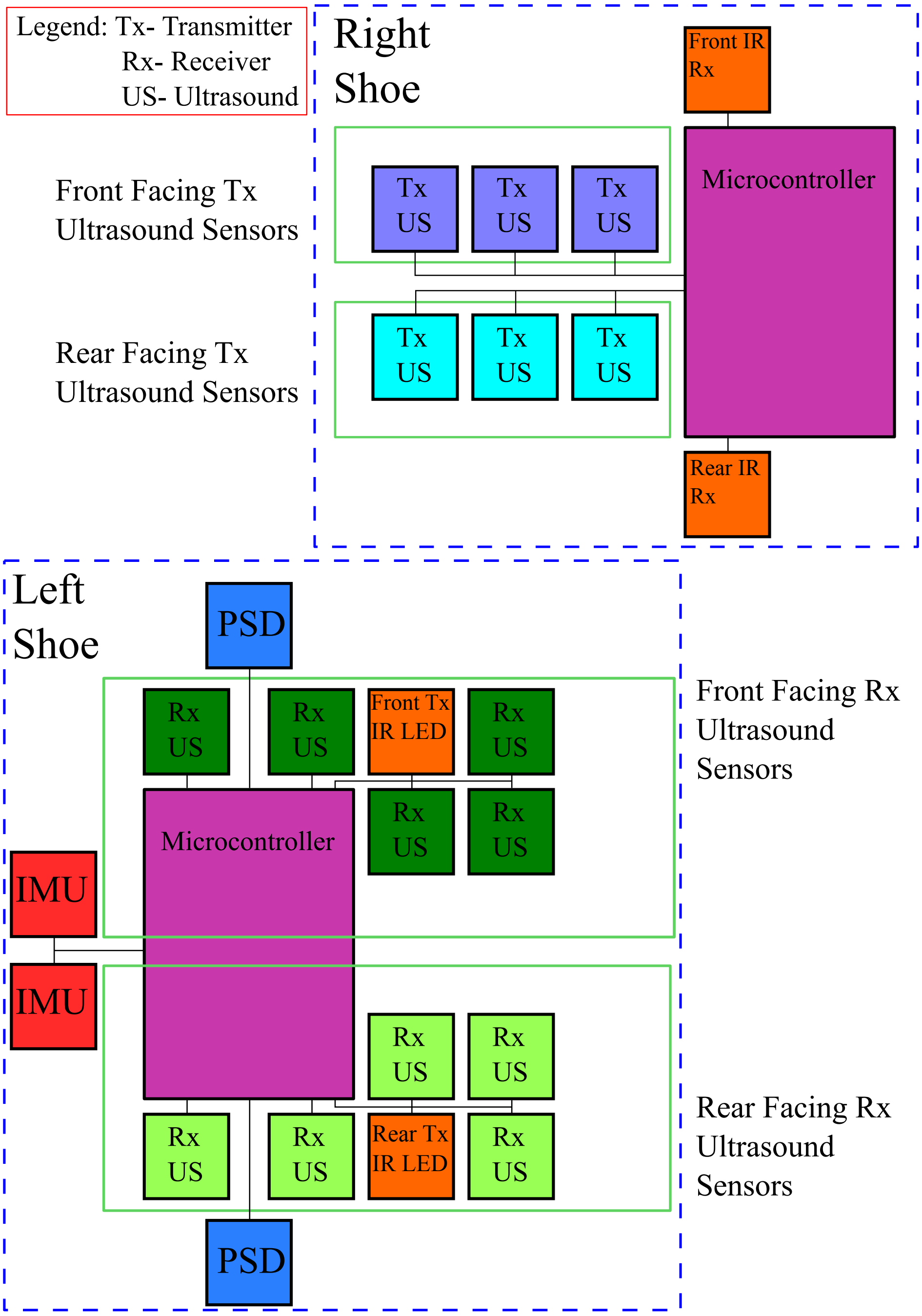


Figure 4.8: Block diagram illustrating the set-up of the ultrasound sensors.

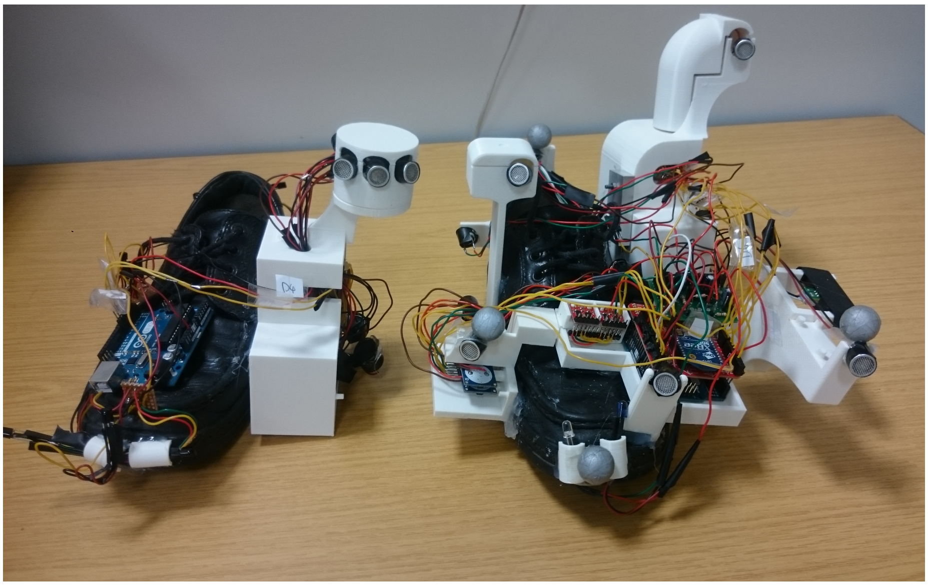


Figure 4.9: Physical layout of the ultrasound transmitters on the left and right shoes. On the right shoe we can see the 3 front facing ultrasound transmitters clustered together to act as a single transmitter with a wider beam angle.

The HC-SR04 had a further modification implemented to improve their use for motion tracking. The standard transducers which are mounted onto the HC-SR04 have a beam angle of 15°. This means that the beam is very focused and can be useful to detect distant objects. However, a 15° beam angle is too narrow for our purposes and the distances that need to be measured are well under a meter in length. Therefore, the transducers were removed and replaced with Prowave 400SR120 or Prowave 400ST120 for receiver and transmitter transducers respectively. The Prowave transducers have a beam width of 85° which made them far more reliable for our system.

An additional alteration had to be carried out on the HC-SR04 to enable them to function reliably as a transmitter-receiver pair. Should a receiver set its echo pin high and fail to detect an ultrasound pulse, it would frequently remain in a "locked" state with the echo pin stuck on high and be unable to receive range measurements. The exact cause of this problem was not fully understood as, according to the manufacturers specifications, the echo pin should fall to a low state after a brief interval if no ultrasound pluses are received. To solve this issue a transistor switch was installed as shown in Figure 4.10. At the beginning of every range measurement the HC-SR04 sensors would be powered down for $5\,\mu s$ to ensure that all the pins were in the low state. The full sequence of events required to make a range measurement is shown in Figure 4.11.

A minimum of three receivers had to successfully detect a range measurement for the trilateration system to function. However, five receivers were used to provide range measurements, both for redundancy and, if more than three range measurements are available then the position can be determined with more accuracy.

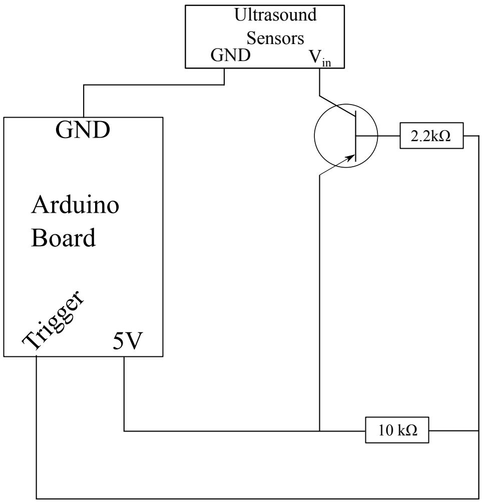


Figure 4.10: Circuit diagram for the transistor switch in controlling the power to the HC-SR04 modules. If a range measurement is to be made the trigger pin is set to high which cuts off power to the ultrasound sensors ensuring that all the pins are in a low state. After 5 μs the trigger pin is returned to low and power is restored to the ultrasound sensors.

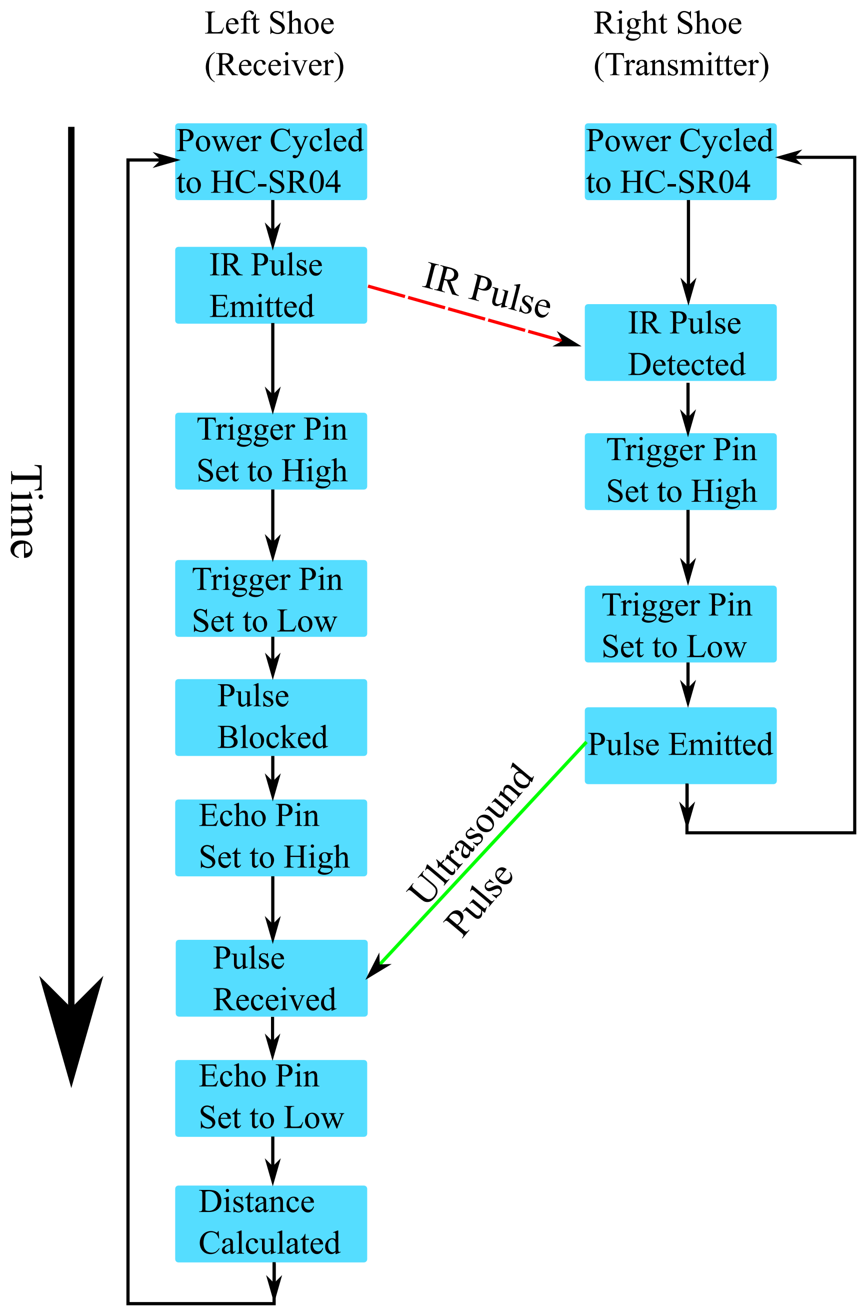


Figure 4.11: Block diagram illustrating the sequence of events occurring on both the left and right shoes

4.7 Evaluation

The ultrasound system was evaluated in the same manner as the IMU system. The same walks previously conducted had the ultrasound measurements added to the post-processing. The results obtained using the IMU and the IMU/ultrasound (IMU/US) systems are compared.

4.7.1 Results Type 1 Walk

The results shown in Section 3.5.3 had the ultrasound results included in the software and the comparison between the Vicon, IMU and IMU/US systems is conducted. As we can see from Figures 4.13 to 4.18 the ultrasound system resulted in a slight error reduction as the absolute and percentage error are consistently smaller than when using the IMU only. To gain a more quantitative measure of how the ultrasound improved the results Figures 4.14, 4.16 and 4.18 are integrated. Therefore, we compare the area under the error curve for the IMU and IMU/US systems. The results of carrying out the area comparison for the error curves is shown in Table 4.1.

Trial	**Total Absolute Error** (m)			**Total Percentage Error**		
	IMU/US	IMU	Ratio	IMU/US	IMU	Ratio
1	2.49	2.50	0.996	59.81	59.93	0.998
2	1.79	2.59	0.691	51.52	67.43	0.764
3	3.63	4.03	0.901	101.6	112.3	0.905

Table 4.1: Table to evaluate the performance of the IMU system and the IMU/US setup via integrating the error curves.

Trial	**Absolute Error** (m)					
	Final Error		Maximum Error		**Final Percent Error**	
	IMU/US	IMU	IMU/US	IMU	IMU/US	IMU
1	0.179	0.191	0.331	0.332	1.98	2.11
2	0.228	0.326	0.302	0.400	2.49	3.55
3	0.257	0.271	0.452	0.475	2.78	2.92

Table 4.2: Comparisons of key results for the IMU and IMU/US systems in Type 1 walks.

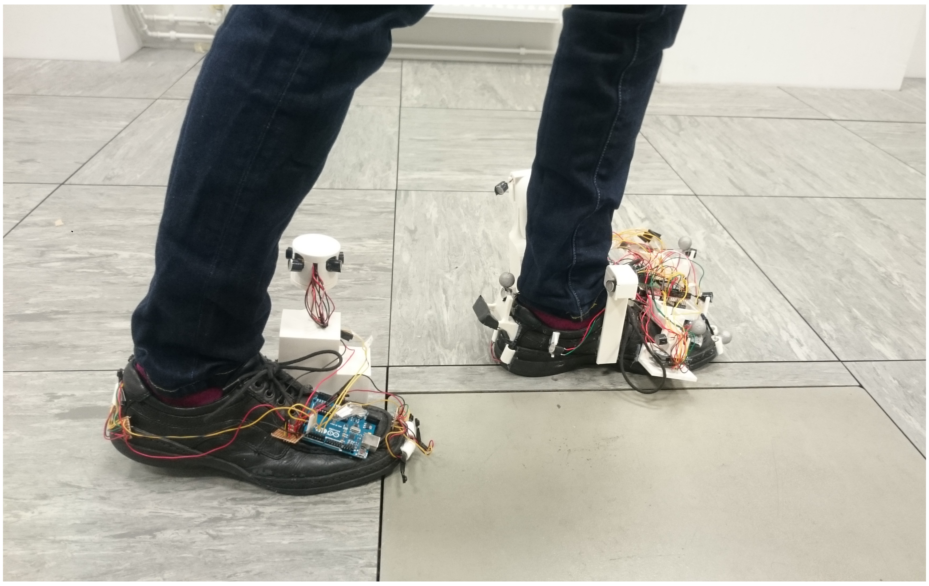


Figure 4.12: Picture showing both feet on the ground during walking. At this point ultrasound measurements are taken giving the displacements between the two feet.

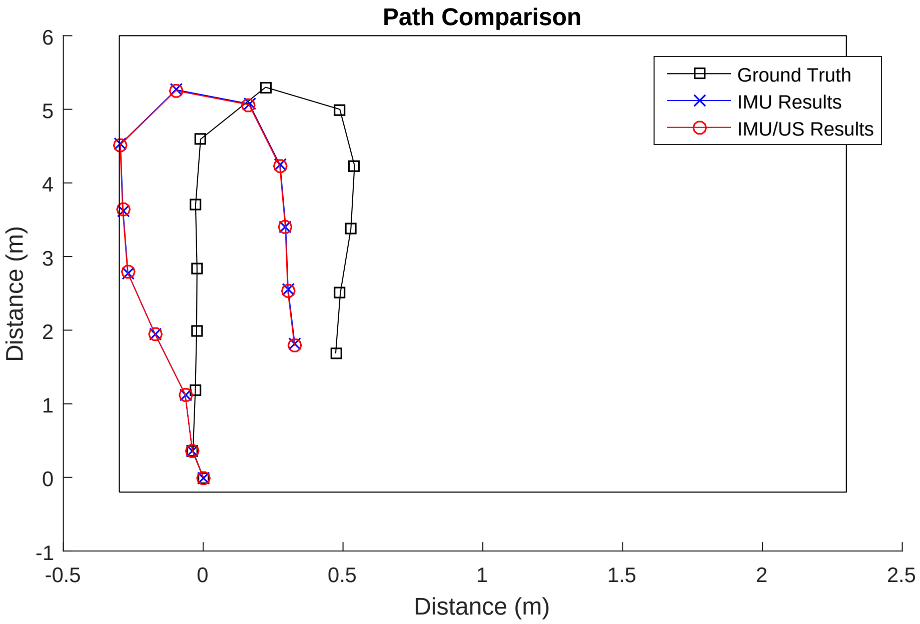


Figure 4.13: Path comparison for trial 1 of the ground truth position as given by a Vicon system and the motion as given by the shoe mounted systems.

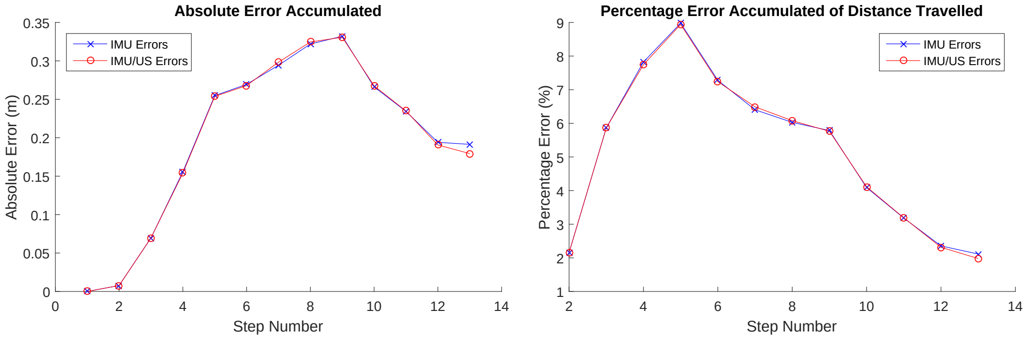


Figure 4.14: Percentage and absolute errors for the first Type 1 walk. For this particular trial the ultrasound offered only marginal improvement on the IMU results. Details of how the errors are computed are given in Figure 3.16.

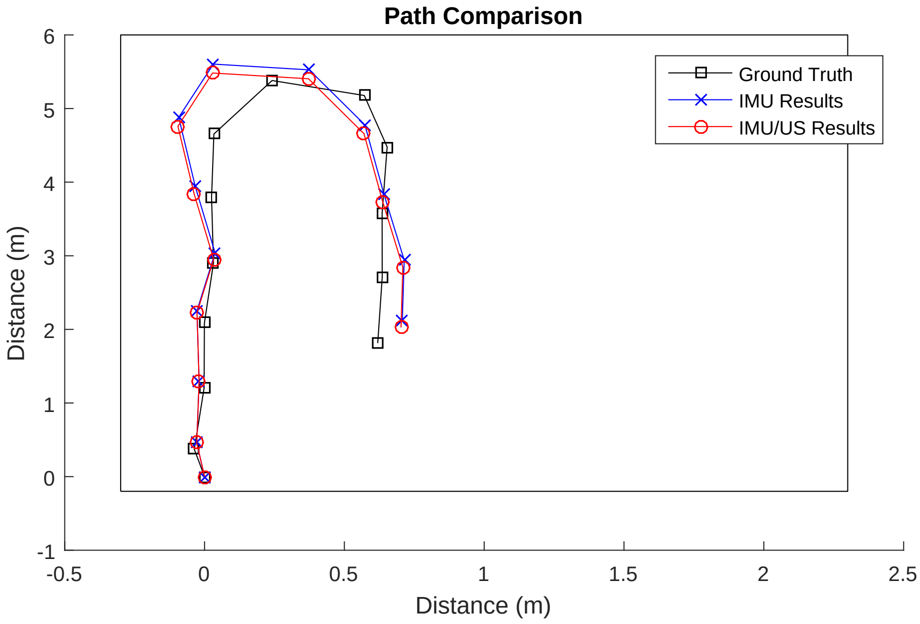


Figure 4.15: Path comparison for trial 2 of the ground truth position as given by a Vicon system and the motion as given by the shoe mounted systems.

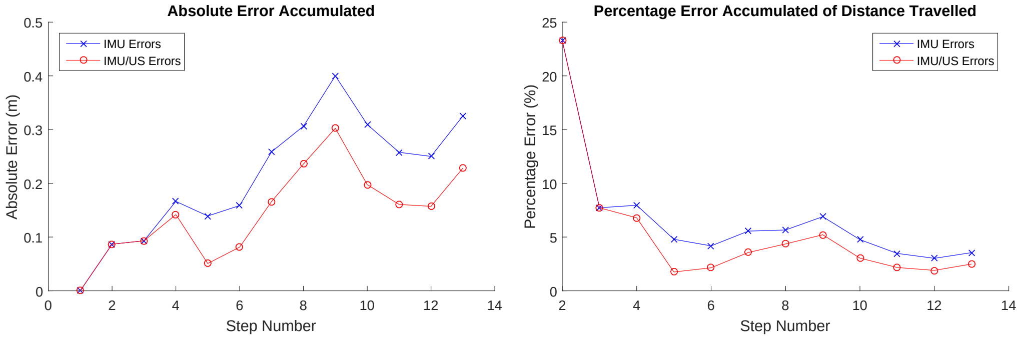


Figure 4.16: Percentage and absolute errors for the second Type 1 walk. We can see the improvement in accuracy due to the ultrasound measurements. Details of how the errors are computed are given in Figure 3.16.

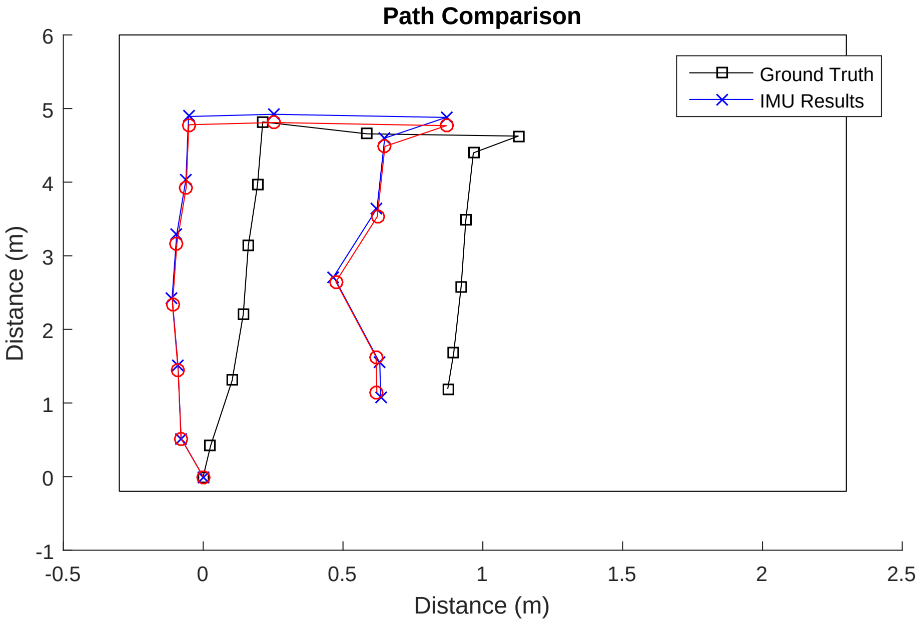


Figure 4.17: Path comparison for trial 3 of the ground truth position as given by a Vicon system and the motion as given by the shoe mounted systems.

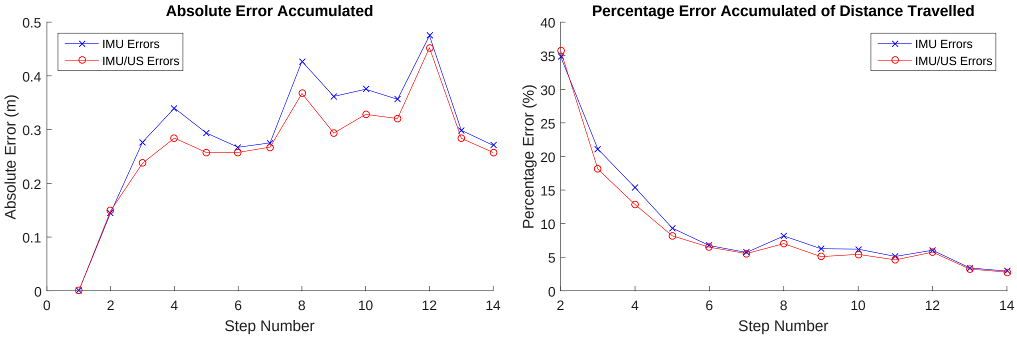


Figure 4.18: Percentage and absolute errors for the last Type 1 walk. We can see the improvement in accuracy due to the ultrasound measurements. Details of how the errors are computed are given in Figure 3.16.

4.7.2 Results Type 2 Walk

The same procedure is repeated for the Type 2 walks; the results as given by the IMU/US system are compared to both using the IMU in a standalone fashion and the ground truth data. Similarly to the Type 1 walk, the error when using the ultrasound data is reduced compared to using the IMU alone. The integrals of the error curves are shown in Table 4.3 to better quantify the effect of the ultrasound system. In addition we can see the cyclic effect of walking in a closed loop fashion in the results, similarly to the results presented in section 3.5.4.

Trial	**Total Absolute Error** (m)			**Total Percentage Error**		
	IMU/US	IMU	Ratio	IMU/US	IMU	Ratio
1	14.49	18.39	0.788	89.86	133.2	0.675
2	20.29	23.18	0.875	187.5	210.7	0.889
3	14.32	16.03	0.893	110.3	121.9	0.905

Table 4.3: Table to evaluate the performance of the IMU system and the IMU/US setup via integrating the error curves.

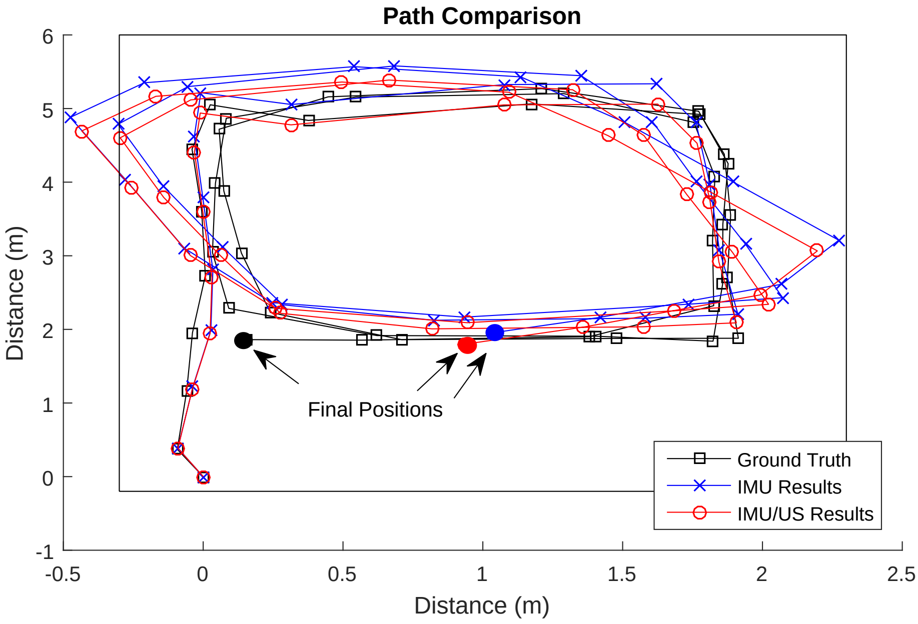


Figure 4.19: Path comparison for trial 1 of the ground truth position as given by a Vicon system and the motion as given by the shoe mounted systems.

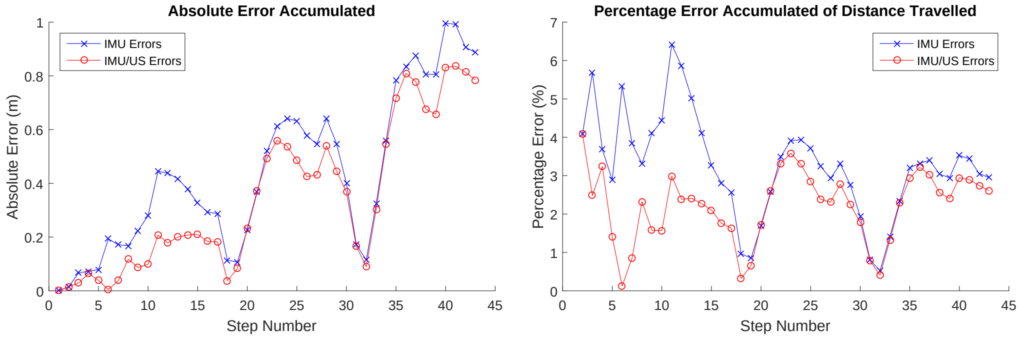


Figure 4.20: Percentage and absolute errors for the first Type 2 walk. We can see the improvement in accuracy due to the ultrasound measurements. Details of how the errors are computed are given in Figure 3.16.

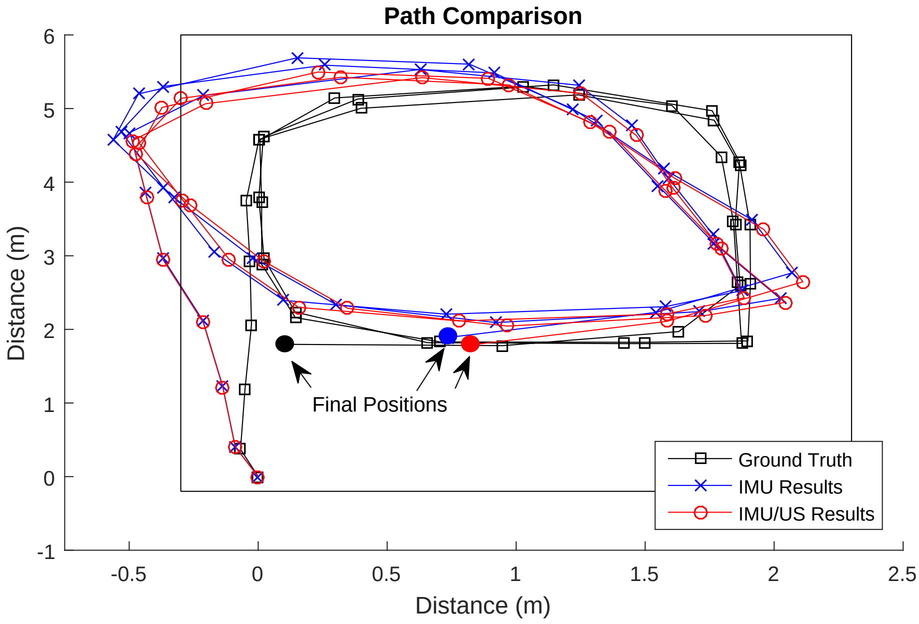


Figure 4.21: Path comparison for trial 2 of the ground truth position as given by a Vicon system and the motion as given by the shoe mounted systems.

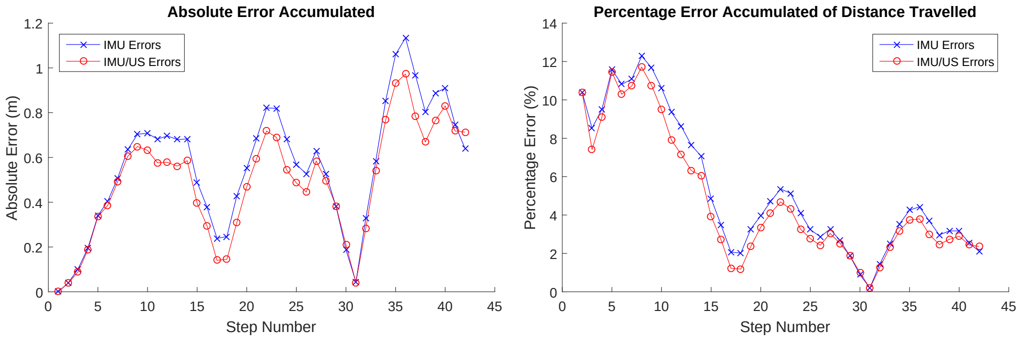


Figure 4.22: Percentage and absolute errors for the second Type 2 walk. We can see the improvement in accuracy due to the ultrasound measurements. Details of how the errors are computed are given in Figure 3.16.

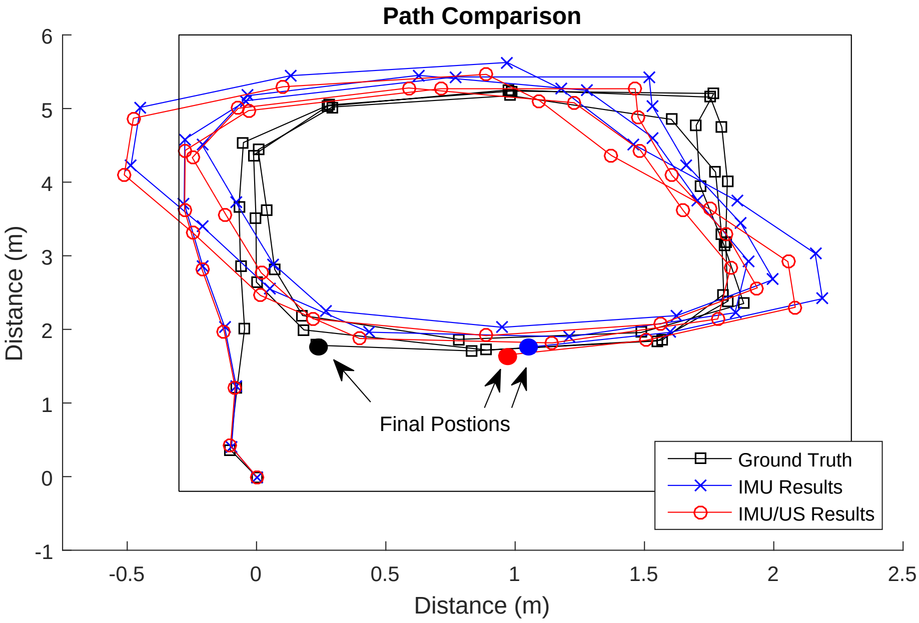


Figure 4.23: Path comparison for trial 3 of the ground truth position as given by a Vicon system and the motion as given by the shoe mounted systems.

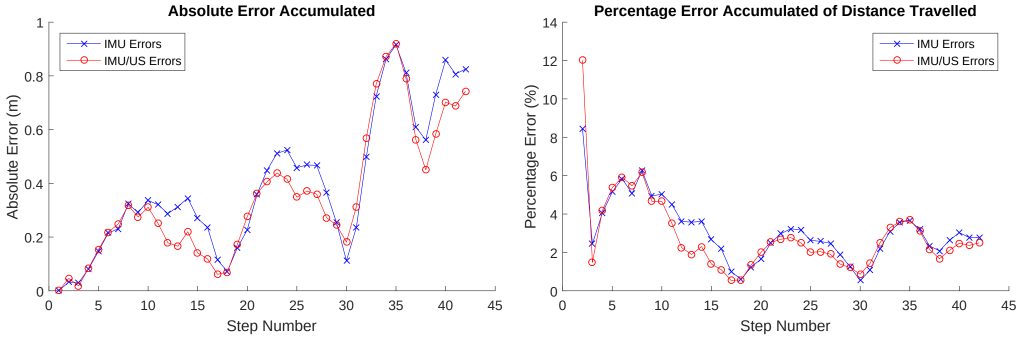


Figure 4.24: Percentage and absolute errors for the last Type 2 walk. We can see the improvement in accuracy due to the ultrasound measurements. Details of how the errors are computed are given in Figure 3.16.

4.7.3 Results Type 3 Walk

As the Vicon motion capture system was unavailable for the Type 3 walks, the effect of the ultrasound is considered in terms of loop closure and, rather more qualitatively, from a visual inspection of the path given by the IMU/US system. The results for including the ultrasound data are shown in Figures 4.25 to 4.26. Unlike the Type 1 and Type 2 walks the IMU/US system gave essentially equivalent performance to an IMU standalone system in a Type 3 walking scenario. The loop closures and maximum estimated error are summarised in Table 4.4. The lack of improvement in performance is due to the more "curvy" type of walking carried out, and so the ultrasound system suffers from more shadowing effects, therefore it fails to gather appropriate data for certain footsteps. This leads to the fact that the systematic effects are not cancelled out as effectively. In other words, due to the asymmetric walking trajectory the ultrasound system may function better on the outwards journey compared to the return journey. This means that systemic errors, which are always cancelled out in the case of an IMU functioning in a standalone manner, will not be fully removed in the case of an ultrasound system.

Trial	**Absolute Error** (m)					
	Misclosure		Maximum Error		**Percent Misclosure**	
	IMU/US	IMU	IMU/US	IMU	IMU/US	IMU
1	0.143	0.176	0.48	0.47	0.26	0.32
2	0.551	0.319	0.92	1.19	1.00	0.58
3	0.439	0.553	1.74	1.80	0.80	1.01

Table 4.4: Comparisons of key results for the IMU and IMU/US systems in Type 3 walks.

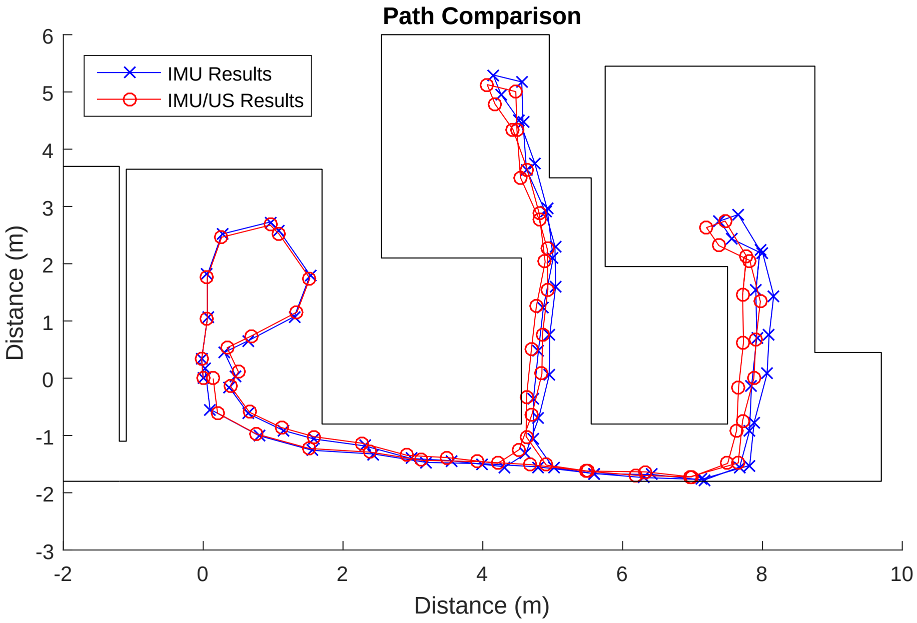


Figure 4.25: Path travelled using the IMU alone, and when combined with ultrasound corrections, for the first Type 3 walking trial.

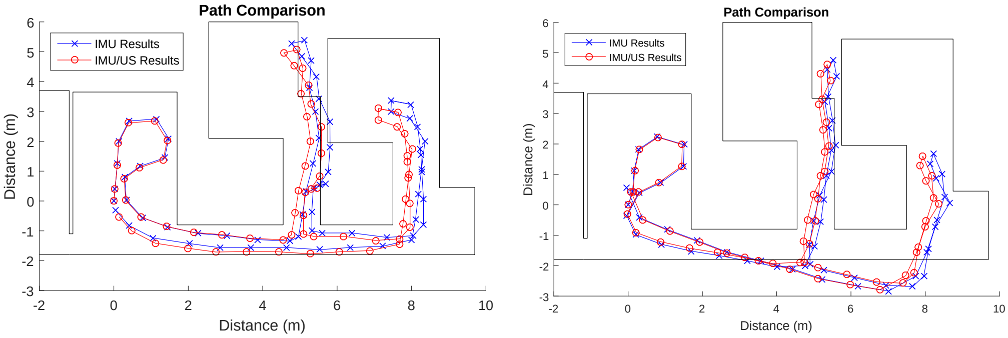


Figure 4.26: Path travelled using the IMU alone, and when combined with ultrasound corrections, for the second and third Type 3 walking trial.

4.8 Conclusions

When combining ultrasound trilateration with IMU system the additional ultrasound data gave a small, but noticeable, improvement in accuracy for Type 1 and 2 walks. The average reduction in the total error across both Type 1 and 2 walks was 15%. This improvement is due to the step length being measured to a higher degree of accuracy. The observed improvement was not uniform however, with some Type 1 and 2 walks benefiting far more from the ultrasound sensors then others. For example trial 1 in Type 1 walking had a ratio of absolute error accumulation between IMU/US and IMU of 0.996. In other words there was less than 1% improvement. On the other had trial 1 in Type 2 walking had a ratio of absolute errors of 0.788, i.e. close to 22% reduction in errors.

In Type 3 walking the IMU/US system resulted in equivalent with the average final loop misclosures being 0.636% with the IMU and 0.686% with the IMU/US system. This is due to the ultrasound system not fully cancelling out systematic effects in Type 3 walks, and if small improvements due to the ultrasound are present the vague ground truth data in Type 3 walks makes such small improvements impossible to measure.

Despite the ultrasound improving the results in Type 1 and 2 walking ultrasound provides no information about the orientation of the user. Hence, the uncertainty in the yaw, which was the main source of error, continues to dominate.

The contribution of this chapter has been to present a novel wearable ultrasound based system that can aid in pedestrian navigation. Very few works in literature explored this idea with ultrasound sensors generally being used as fixed infrastructure beacons. We have demonstrated that wearable ultrasound sensors are a viable technology to incorporate in navigation. However, the system shown here is simply a proof of concept with further work being required to make it feasible in everyday life. Improvements necessary include a better ergonomic design, more accurate error models for the ultrasound sensors, and making the system less susceptible to ultrasound signal blockage.

Chapter 5

Map Constraints

Until now we have not made use of any prior knowledge of environmental constraints such as walls and other impassable features. Although it represents a significant increase in the amount of information requested from the user, it can provide very strong error correction techniques.

As we have seen, the largest source of error in a IMU based navigation system is determining the user's heading. This chapter described the particle filter which is used to enforce the building constraints which corrects for both the step length, and crucially, the heading of the user.

5.1 Particle Filter Overview

Tracking the user's motion using the IMU/US system suffers the problem that the positional accuracy within a global co-ordinate frame degrades over time as the accumulation of errors results in unbound error growth.

A known map of the environment will limit the possible movement of the user to within known areas. Motion constraining obstacles, primarily walls, thereby can be used to correct for both positional uncertainty and heading errors.

To enforce environmental constraints a particle filter is employed. It receives data from the IMU/US system and by using environmental knowledge it corrects the calculated foot position. The accuracy of the particle filter scales with the number of particles used. In this case the number of particles was set to 2000. The drawback of using a large number of particles is that the algorithm is very computationally expensive, and in this current implementation, is too slow to be performed in real time. Scaling down the number of particles in conjunction with a more efficient implementation could reduce the computational time down sufficiently for real time tracking to be performed. However, that is left as future work.

The particle filter receives information from the IMU/US system at every footstep in the form of,

$$S = (l, \delta\theta, \eta) \tag{5.1}$$

where l and $\delta\theta$ are the step length and change in yaw respectively. η represents the difference between the step direction and the user's heading. Details of these parameters can be found in Figure 4.4. This data is passed though three stages which constitutes the particle filter algorithm. The stages are particle propagation, particle weighting and re-sampling, and clustering.

5.2 Particle Propagation

In the propagation step, the new particle's position is calculated from its previous state and the step information given by the navigation algorithm. The step length, l, and change in yaw, $\delta\theta$, are initially perturbed by l_{pert} and θ_{pert} drawn from our error model,

$$l' = l + l_{pert} \tag{5.2}$$

$$\delta\theta' = \delta\theta + \theta_{pert} \tag{5.3}$$

The values of l_{pert} and θ_{pert} are modelled as a normal Gaussian distribution with zero mean and standard deviation of 0.1 m and 0.1 rad respectively. The standard deviation is somewhat arbitrarily chosen to ensure that the particles are sufficiently spread, so that in the case of large IMU/US errors the particle filter is able to effectively function. Should the standard deviation be made too small then large errors could cause the entire particle cloud to cross a wall.

The new yaw, θ_k, and position, (x_k, y_k), are computed from the previous step parameters

$$\theta_k = \theta_{k-1} + \delta\theta' \tag{5.4}$$

$$x_k = x_{k-1} + l' cos(\theta_k - \eta) \tag{5.5}$$

$$y_k = y_{k-1} + l' sin(\theta_k - \eta) \tag{5.6}$$

where η is the difference between the user's final heading and direction of the step as previously defined in 4.16.

5.3 Particle Weighting and Re-sampling

The particle weighting stage updates the weight of a particle based on how likely it is to represent the true motion of the user. Here a simple weighting scheme is adopted: if the particle did not cross any walls then it represents a potential motion of the user and is given a weighting of $1/N$ where N is the number of particles. However, if the particle crossed a wall then it obviously represents an impossible motion and is given a weight of $1/10000N$. The reason for not assigning it a weight of zero is in the case of all the particles crossing a wall. In that case assigning a weight of zero would kill off the entire particle cloud. However by assigning a low weight it allows the particle filter to re-localise itself on subsequent steps.

Should we simply stop the algorithm at this stage it would work well in tracking the user over relatively short distances. However, as the number of steps increase, the number of high weight particles will gradually reduce as more and more particles cross walls and are assigned a low weight. This problem will lead to what is known as the weight degeneracy problem, in which the user's position is represented by a handful of high weight particles.

This has a two fold negative effect. Firstly, the belief distribution of the user's location is badly represented. For a particle filter to be effective a large amount of particles need to present and thus its accuracy is reduced. Secondly, the low weight particles are still passing through the particle filtering algorithm, leading to the bulk of the computational time being spent on particles with a negligible weight.

To solve this re-sampling can be implemented. A set of new particles are generated by randomly sampling the particles proportionally to their weights. This ensures high weight particles are duplicated and low weight ones are removed.

5.4 Clustering

A problem which may occur by using a particle filter in symmetric environments is a splitting of the particle cloud. In other words, there are two locations which are possible candidates for a user's true location. In this scenario, taking an average of all the particles would not make much sense as it is obvious that a large number of the particles have ended up in an incorrect position and furthermore the average location of all the particles is likely to place the user in a physically impossible position. Consider the two scenarios presented in Figure 5.1,

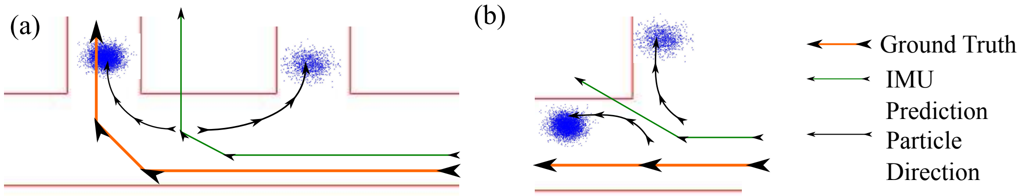


Figure 5.1: Both (a) and (b) demonstrate how the particle cloud can diverge. This clearly shows how computing a direct average will yield inaccurate results as it will give a location situated in an impassible terrain feature. By applying a clustering algorithm it is possible to exclude the smaller particle cloud from influencing the calculated position.

As we can see taking an average places the user in a physically impossible position. Hence, a more sophisticated approach is required. To do this we group the data into clusters to determine the user's position. The subtractive clustering algorithm proposed by Chiu is used [72]. The subtractive clustering algorithm is an extension of the mountain method first proposed by [73]. In the mountain method a grid is generated, and each grid point is assigned a potential value based on the number and distance of nearby data points. Therefore, should a grid point have many data points surrounding it, then that particular grid point has a high potential. The grid point with the highest potential value is taken as the cluster centre. The next step is to reduce the potential of all the grid points near the first cluster centre and repeat the procedure until all of the grid points have a potential reduced below a given threshold. The main drawback of the mountain method is that the computational load depends on the resolution of the grid, trading precision with computational complexity, and furthermore the higher the dimensionality of the problem the larger the computational time.

The subtractive clustering method overcomes these problems by considering each data point as a potential cluster centre. This eliminates the requirement of specifying a grid resolution and the computational complexity grows with the number of data points. The algorithm works as follows; firstly the potential P_i of a particle x_i is given by

$$P_i = \sum_{j=1}^{n} e^{-\alpha||x_i - x_j||^2} \tag{5.7}$$

where

$$\alpha = \frac{4}{r_a^2} \tag{5.8}$$

Therefore the potential of a particle is a function of its separation to other data points. The distance r_a determines how far particles contribute to the potential. This is one of the major differences of subtractive clustering compared to the mountain method, i.e. the potential is computed for a particle rather than a grid point. Additionally, the influence of data points decreases exponentially with the square of the distance, rather than the distance itself, as is the case in the mountain method. The data point with the highest potential is then declared as the first cluster centre.

Usually the subtractive clustering algorithm will reduce the potential of particles nearby the first cluster centre, and then repeat the procedure finding ever more cluster centres until the potential of all the particles is below a certain value. However, in this situation there can only be one location for the user. Therefore finding additional cluster centres is unnecessary, and we take the first cluster centre to be the true location of the user.

5.5 Particle Filter Summary

The particle filter algorithm is composed of a few key stages which allow it to take into account environmental information into the predicted user position. The stages which need to occur are shown in Figure 5.2. Figure 5.2 also shows how the particle filter algorithm integrates with the IMU and ultrasound based systems.

Below Figure 5.2 Algorithm 2 displays the particle filter in a pseudo-code manner illustrating the mathematics more clearly than a block diagram.

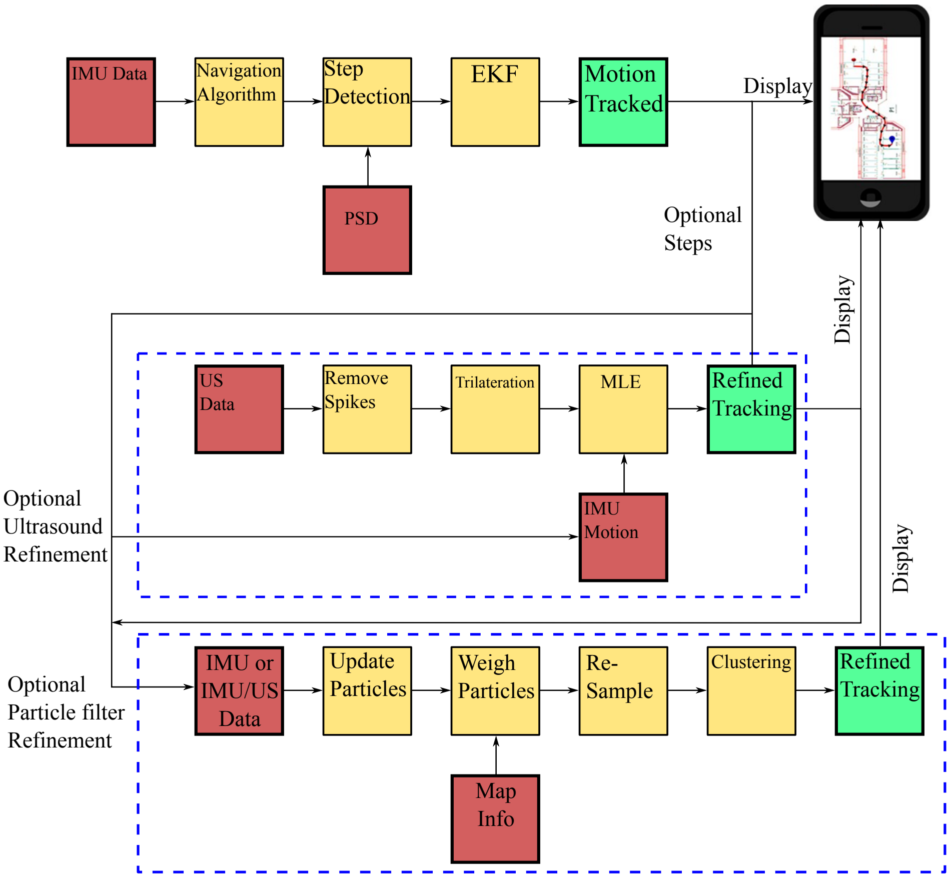


Figure 5.2: Block diagram illustrating how the different systems interact with each other as well as briefly examining their key stages [59].

Algorithm 2 Particle Filter Algorithm

```
procedure Particle Filter($x_{k-1}, y_{k-1}, l, \theta_i, \delta\theta_i, Poly$)

    Description of input data

    $x_{i-1}, y_{i-1}$                    ▷ Position calculated for previous step $i$.
    $\theta_i, \delta\theta_i,\ l_i$                              ▷ Data for step $i$.
    $Poly$                              ▷ 2D ploygon representing the map

    Start of algorithm

    Perturb the step parameters
    $l' \leftarrow l_i + l_{pert}$
    $\delta\theta' = \delta\theta_i + \theta_{pert}$

    Update particles position and heading
    $\theta_i = \theta_{i-1} + \delta\theta'$
    $x_i = x_{i-1} + l'cos(\theta_i - \eta)$
    $y_i = y_{i-1} + l'sin(\theta_i - \eta)$

    Determine if the particles has an invalid trajectory
    Intersect($x_i$,$x_{i-1}$,$y_i$,$y_{i-1}$, $Poly$)

    Weight the particles
    $w \leftarrow$ Weight(Intersect)

    Re-sample the Particles
    $x_i, y_i \leftarrow$ Re-Sample($x_i, y_i, w$)

    Determine Cluster Centre
    $x_i, y_i \leftarrow subclust(x_i, y_i)$
    Return $(x_i, y_i, \theta_i)$
end procedure
```

5.6 Evaluation

The same set of walks were examined when a particle filter was applied to them. The particle filter received step information from the IMU/US system and then imposed movement constraints. Therefore, this final set-up is referred to as IUP as it is a combination of **I**nertial+**U**ltrasound+**P**article filtering algorithms.

It should be highlighted that while the IMU and IMU/US systems could have been modified to generate results in real time (as discussed in section 3.4) the IUP system, in its current implementation, is too slow to run in a real time fashion. To run the particle filter algorithm with 2000 particles takes 1.75 s per step. As typical walking is at a rate of 1-1.5 steps per second the particle filtering algorithm would need to be able to run significantly faster for real time data to be generated.

5.6.1 Results Type 1 Walk

The IMU/US results as shown in Section 4.7.1 were then incorporated into the particle filter framework. The results are shown in Figures 5.3 to 5.8.

As we can see, the walk is well within the environmental constraints and so the particle filter has little way in which to provide assistance. Table 5.1 shows the integral of Figures 5.4, 5.6, and 5.8 as to have a more quantitative metric for measuring the different walks.

Trial	**Total Absolute Error (m)**			**Total Percentage Error**		
	IUP	IMU/US	Ratio	IUP	IMU/US	Ratio
1	1.02	2.49	0.410	30.51	59.81	0.510
2	1.25	1.79	0.698	35.91	51.52	0.697
3	2.88	3.63	0.793	86.41	101.6	0.850

Table 5.1: Table comparing the errors when using the IMU/US system with and without particle filtering for Type 1 walking via integrating the error curves.

Trial	**Absolute Error** (m)					
	Final Error		Maximum Error		**Final Percent Error**	
	IUP	IMU/US	IUP	IMU/US	IUP	IMU/US
1	0.070	0.179	0.151	0.331	0.78	1.98
2	0.230	0.228	0.230	0.302	2.51	2.49
3	0.237	0.257	0.390	0.452	2.56	2.78

Table 5.2: Comparisons of key results for the IMU/US and IUP systems in Type 1 walks.

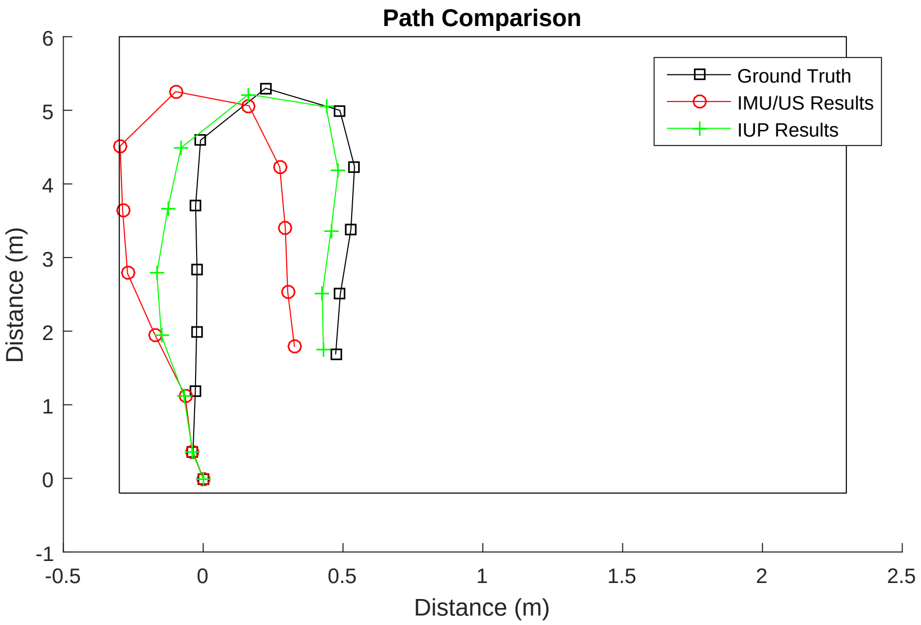


Figure 5.3: Path comparison for trial 1 of the ground truth position as given by a Vicon system and the motion as given by the IMU/US and the IUP systems.

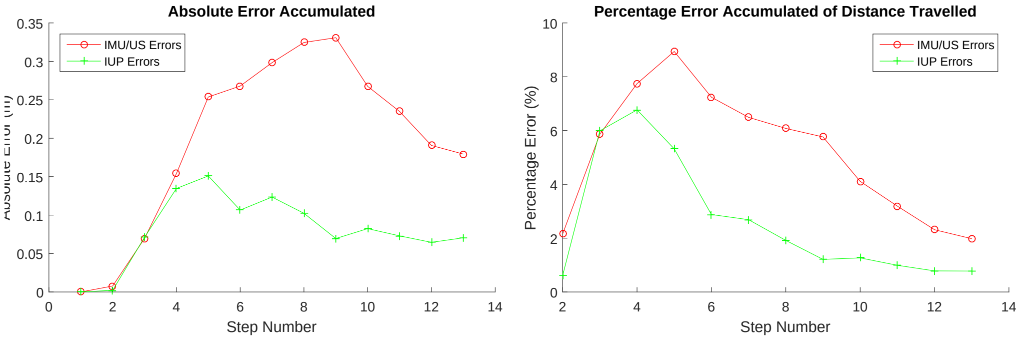


Figure 5.4: Percentage and absolute errors for the first Type 1 walk. Details of how the errors are computed are given in Figure 3.16.

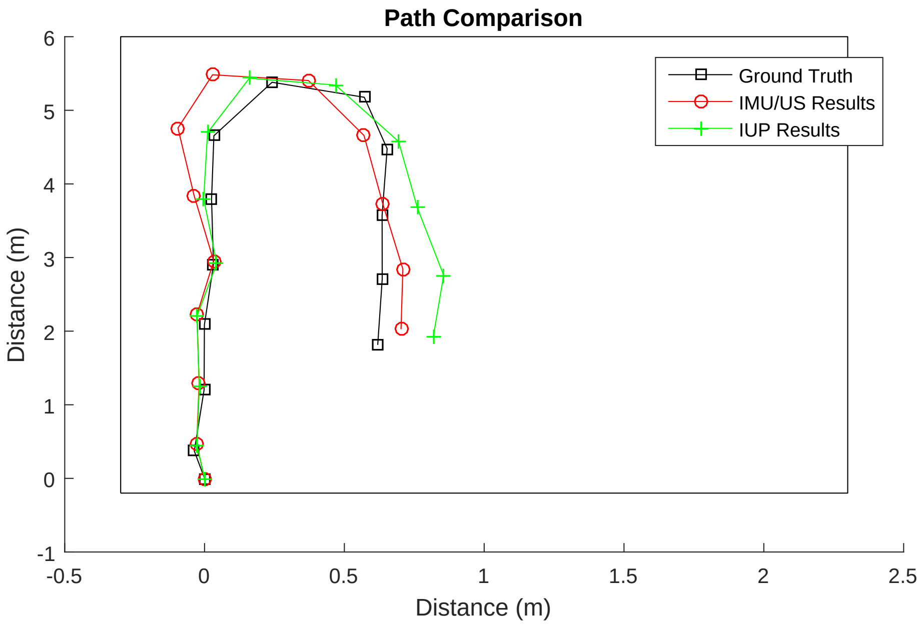


Figure 5.5: Path comparison for trial 2 of the ground truth position as given by a Vicon system and the motion as given by the IMU/US and the IUP systems.

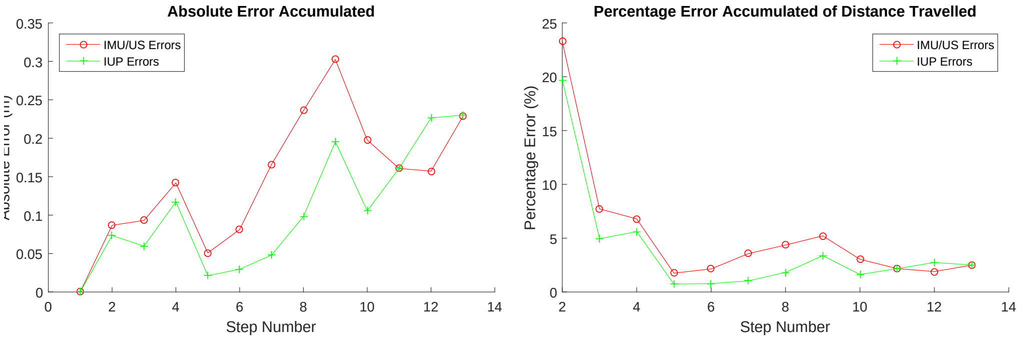


Figure 5.6: Percentage and absolute errors for the second Type 1 walk. Details of how the errors are computed are given in Figure 3.16.

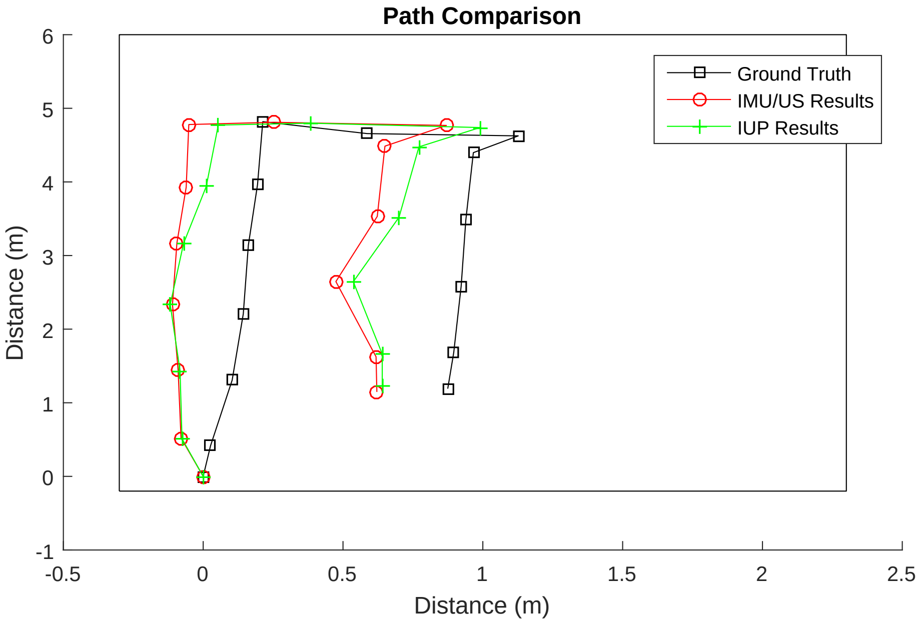


Figure 5.7: Path comparison for trial 3 of the ground truth position as given by a Vicon system and the motion as given by the IMU/US and the IUP systems.

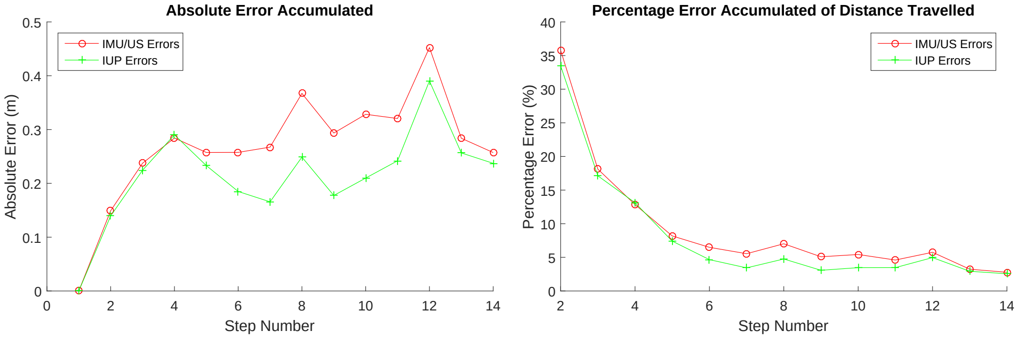


Figure 5.8: Percentage and absolute errors for the last Type 1 walk. Details of how the errors are computed are given in Figure 3.16.

5.6.2 Results Type 2 Walk

For Type 2 walks, environmental constraints play a much larger role as the user walked close to the wall edges. The results for the particle filter are shown in Figures 5.9 to 5.14. The particle filter provides a strong source of error corrections and keeps the maximum accumulated error under 0.3 m in the majority of footsteps.

Additionally, we can note that the particle filter removes a significant amount of systematic error accumulation, with all 3 walks showing an approximately constant absolute error of around 0.15-0.2 m.

Table 3.40 shows the integrals of the error curves and we can see that the absolute error accumulated when using the particle filter is much smaller than when using the IMU/US system, and more notably from Figures 5.10, 5.12 and 5.14, is that the error does not grow with the number of steps taken.

Trial	**Total Absolute Error** (m)			**Total Percentage Error**		
	IUP	IMU/US	Ratio	IUP	IMU/US	Ratio
1	6.18	14.49	0.427	56.46	89.86	0.628
2	5.20	20.29	0.256	65.80	187.5	0.351
3	4.95	14.32	0.346	51.11	110.3	0.463

Table 5.3: Table comparing the errors when using the IMU/US system with and without particle filtering for Type 2 walking.

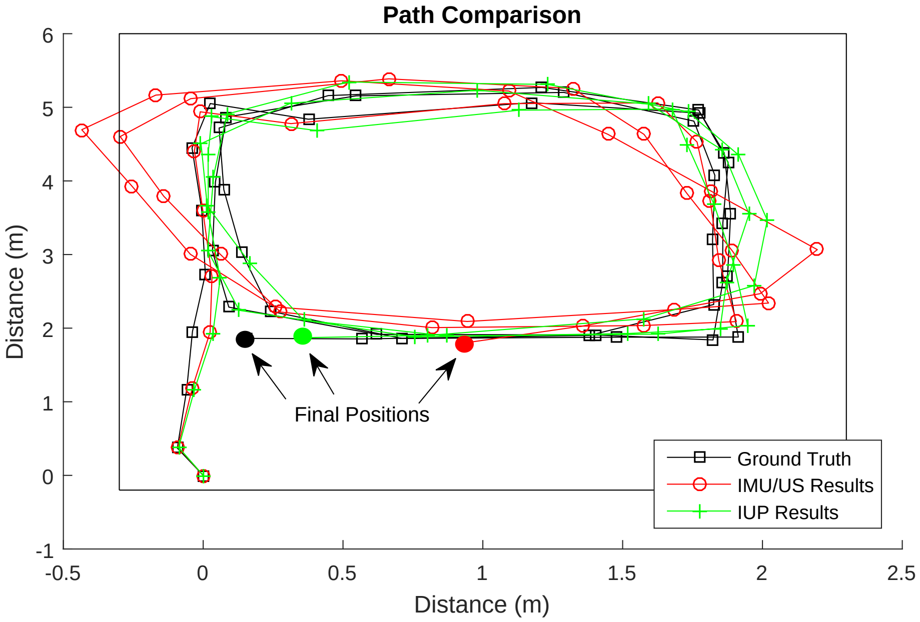


Figure 5.9: Path comparison for trial 1 of the ground truth position as given by a Vicon system and the motion as given by the IMU/US and the IUP systems.

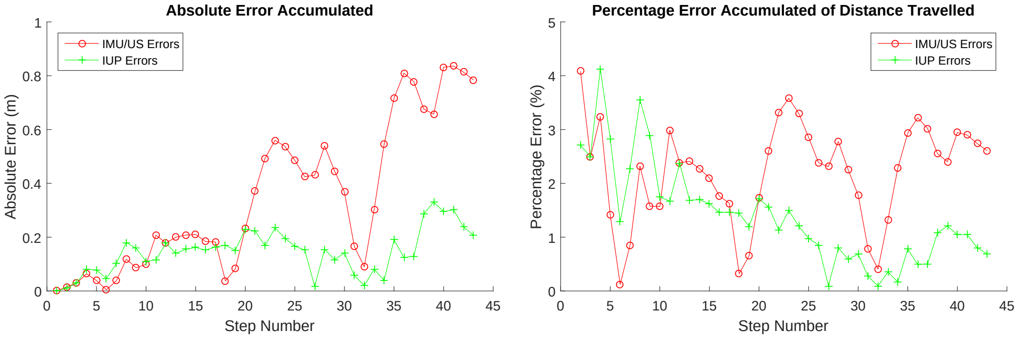


Figure 5.10: Percentage and absolute errors for the first Type 1 walk. Details of how the errors are computed are given in Figure 3.16

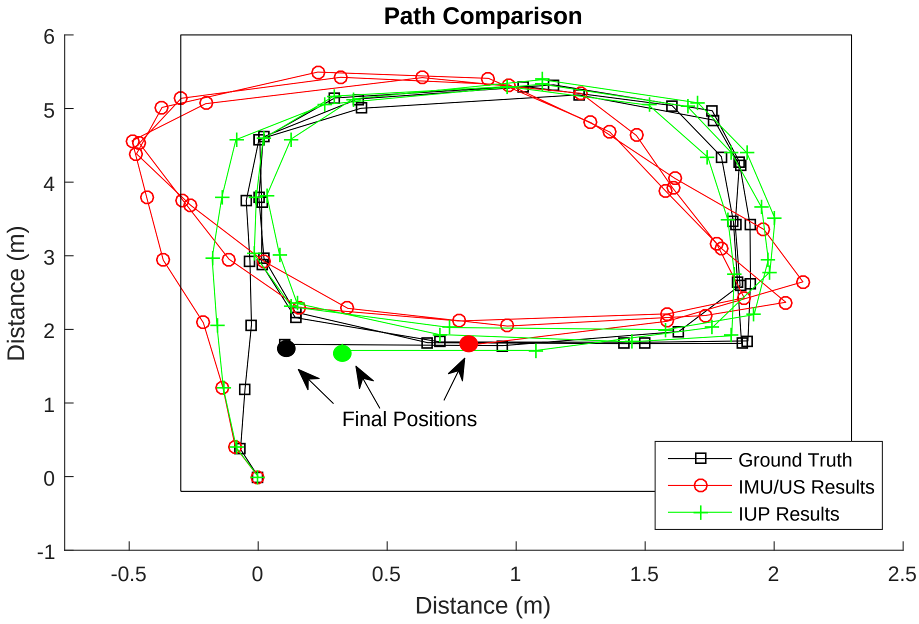


Figure 5.11: Path comparison for trial 2 of the ground truth position as given by a Vicon system and the motion as given by the IMU/US and the IUP systems.

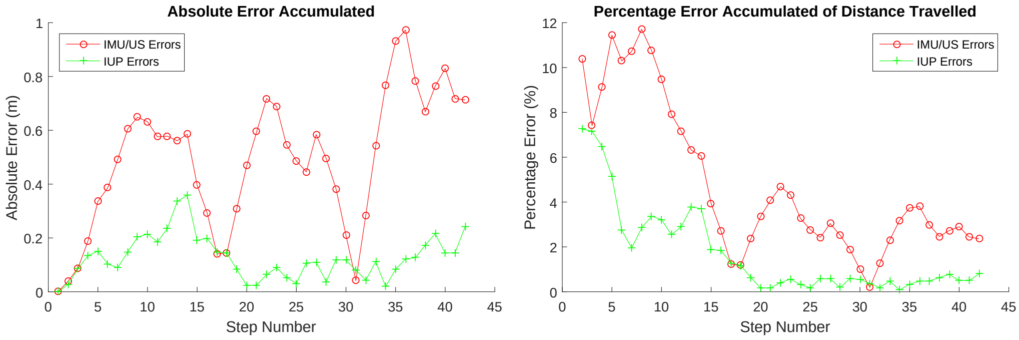


Figure 5.12: Percentage and absolute errors for the second Type 2 walk. Details of how the errors are computed are given in Figure 3.16

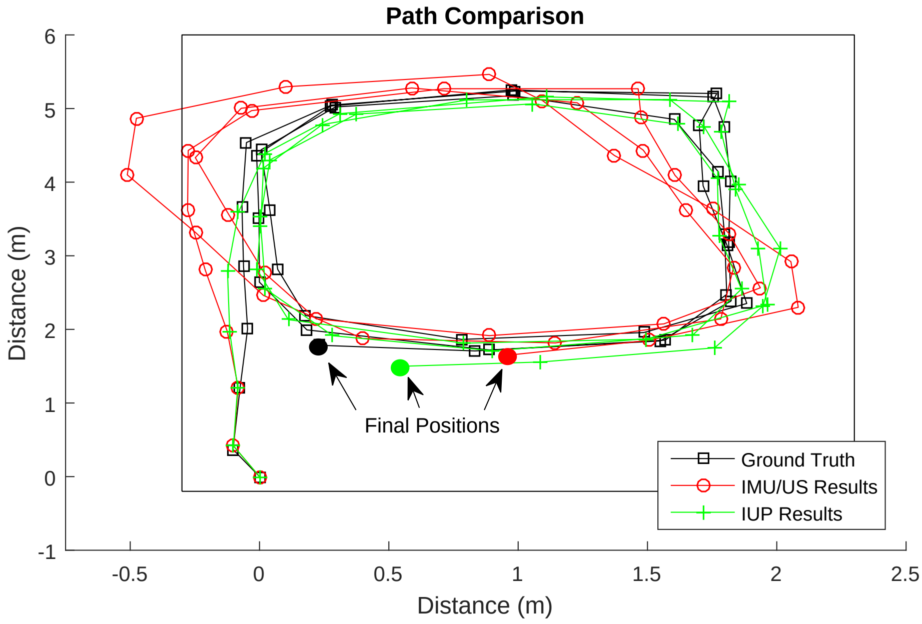


Figure 5.13: Path comparison for trial 3 of the ground truth position as given by a Vicon system and the motion as given by the IMU/US and the IUP systems.

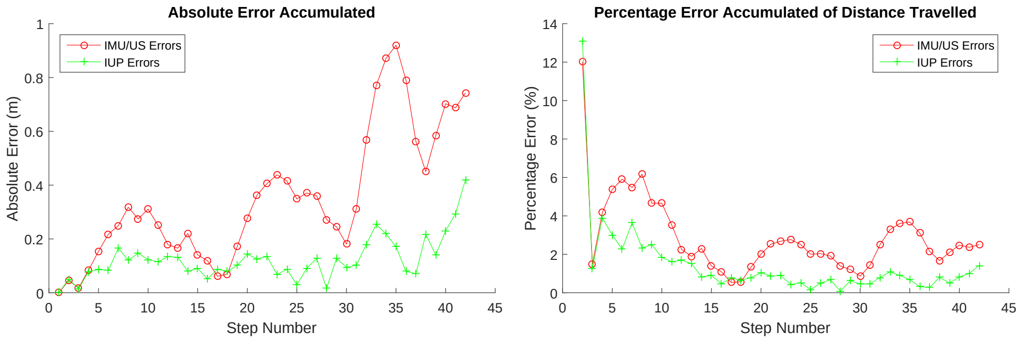


Figure 5.14: Percentage and absolute errors for the third Type 2 walk. Details of how the errors are computed are given in Figure 3.16

5.6.3 Results Type 3 Walk

The Type 3 walk has the largest amount of environmental interaction as the user entered and exited a number of rooms, as well as travelled repeatedly down a corridor. The results for the particle filtering algorithm receiving data from the IMU/US system are shown in Figures 5.15 to 5.16. As we can see the particle filter kept the walk within the building constraints at all times, and therefore the overall trajectory was far more accurate despite the loop closure being slightly higher. For Type 3 walking with the IUP system it was not possible to accurately determine the maximum error as no ground truth data was present. The IUP system gave results which where consistently very close to the path travelled with no large, obvious deviations.

Trial	**Absolute Error** (m)		**Percentage Error**	
	IUP Misclosure	IMU/US Misclosure	IUP	IMU/US
1	0.592	0.143	1.07	0.26
2	0.423	0.551	0.77	1.00
3	0.326	0.439	0.59	0.80

Table 5.4: Loop misclosure for Type 3 walking obtained using the IMU/US system with and without particle filtering. The particle filter gave a slightly higher loop misclosure on trial 1, however from a visual inspection of the path in Figure 5.15 we can see the overall trajectory is more accurate as the particle filter results centred the path to be in the middle of the corridors.

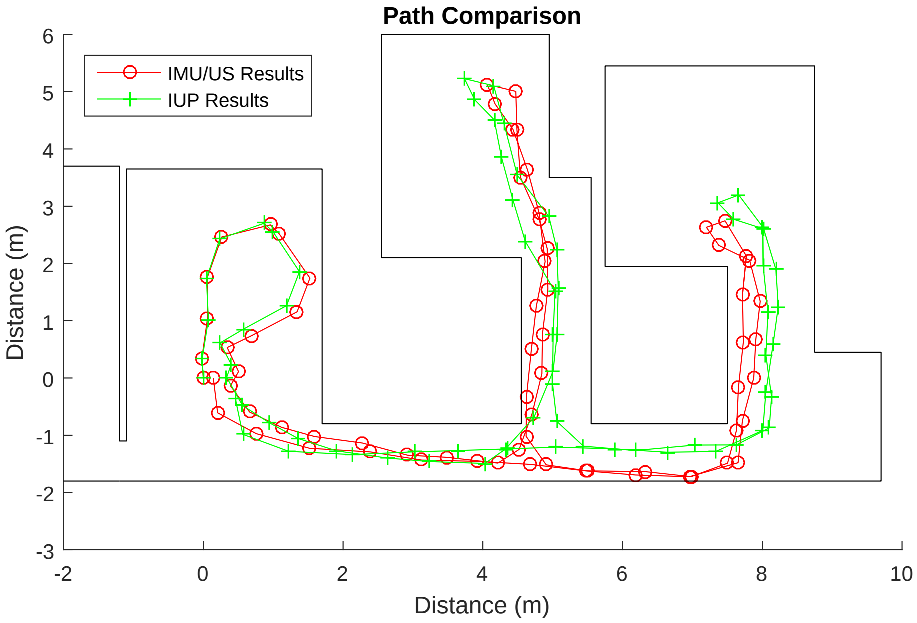


Figure 5.15: Figure showing the effect of including the particle filtering algorithm (IUP) on the IMU/US results.

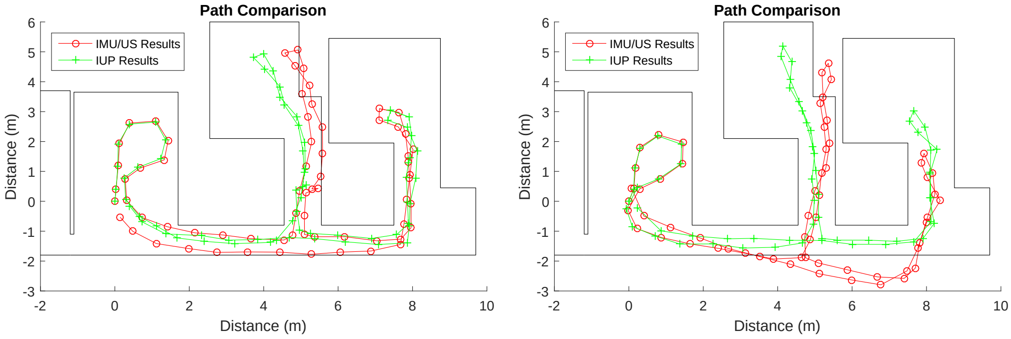


Figure 5.16: Figure showing the effect of including the particle filtering algorithm (IUP) on the IMU/US results.

5.7 Conclusions

The particle filter can provide very strong error reduction techniques. Of most importance it provides a way to correct for the drift in heading and so, should a map be available tracking over prolonged periods of time can be performed. However, this does come with drawbacks. Namely the inherent requirement for a map means that a user is constrained to only pre-known environments. Additionally there is a very high computational cost as for the filter to be effective a large number of particles need to be used. Hence, this represents a large jump in the computational time with each step requiring 1.75 s of computation.

For Type 2 and 3 walking the particle filter is shown to drastically reduce the error accumulated. The path trajectories in Type 3 walking was far better contained in the building constraints and, in two of three cases, also had a lower loop misclosure. For Type 2 walking the errors as shown in Figures 5.10, 5.12 and 5.14 was shown to be far smaller and exhibited a much smaller cyclic error pattern. Indeed the average total absolute error for the IUP system was just over a third (0.343) of the average total absolute error for the IMU/US system.

On the other hand, should the walk be well contained within the building parameters, for example if the user is walking near the middle of a large room, the particle filter does not provide as much of an improvement. This is due to the fact that there are no environmental constraints near the user that can be utilised by the particle filter. We can see this effect in a Type 1 walk with the average total absolute error for the IUP system being a factor of 0.6336 smaller compared to the IMU/US system.

Type 3 walking is more challenging to evaluate precisely without accurate ground truth data. The final loop misclosure was 0.447 m for the IUP system against 0.378 m for the IMU/US setup. However, despite the loop misclosure being slightly higher it is incorrect in saying that the IUP system performed worse in Type 3 walking as from a visual inspection of Figures 5.15 to 5.16 the overall path trajectory is far more accurate.

The innovation of this chapter has been to demonstrate that with a known map low cost sensing systems are just as effective at tracking compared to high end systems using a map. Our system had a typical tracking accuracy of ∼0.25 m while [39] tracked a user to under 0.75 m in 95% of cases and [38] achieving accuracy of 0.743 m. Additionally we have used a subtractive clustering algorithm which was able to prevent the particle cloud from splitting in Type 3 walking. The use of a subtractive clustering algorithm to account for highly symmetric environments has not been used in literature to our knowledge.

Chapter 6

Conclusion

6.1 Research Contributions

This work has shown the potential of using IMUs costing ∼£30 for tracking the motion of a pedestrian. This was done by firstly calibrating the IMU without the use of sophisticated equipment via a series of minimisations and constraints. However, the lack of traditional calibration equipment meant that it was not possible to compare the calibration parameters for the gyroscopes obtained in this equipment free manner to those acquired though a traditional calibration procedure. For the accelerometers, gravity can be used as a reference signal and the average error for the two accelerometers was $0.0324\,\mathrm{ms}^{-2}$ and $0.0196\,\mathrm{ms}^{-2}$.

Once the IMUs were calibrated, an Extended Kalman filter (EKF) which tracked the error states accumulated by a inertial navigation system was implemented. When the foot was detected as being stationary on the ground the EKF applied zero velocity and heuristic drift reduction updates. This enabled the user's motion to be effectively tracked.

A complementary wearable ultrasound trilateration system was developed which measured the user's step displacement. This information was then combined with the IMU data via a maximum likelihood estimator (MLE). This resulted in a small but noticeable improvement in the results.

Lastly, should the user be able to provide map information then environmental constraints could be enforced. This was done by having a particle filter which received data from the underlying IMU/US system. This was able to track a user's motion with a very high degree of accuracy, however it did come at a significant computational cost.

The three different layers (IMU, ultrasound, and particle filtering) were analysed in three different types of walks. Type 1 and 2 both involved travelling within the capture volume of a Vicon system. This allowed the performance of the systems to be evaluated on a step by step basis. The results highlighted that several previous testing methodologies, which discussed the system accuracy only in terms of loop closure, would have been over estimating the system accuracy. Type 3 walking, which did not use a Vicon system for comparison, was forced to be evaluated by using the loop misclosure and an estimated maximum error.

We now return to the research questions posed at the start of the thesis and discuss how the results have answered them.

Question 1: What performance can be achieved by using low cost IMUs?

To keep the same testing methodologies as previous works we look at the performance of the IMU system in terms of its loop closure in a Type 3 walk. The IMU system incurred loop closure errors of 0.32, 0.58 and 1.01 %. Its performance can be compared to other research groups systems: [13] had errors of 2-10 % when using ZUPTs and HDR; [68] achieved a worst case drift of 1.28 % and [17] had errors of 0.3 % even when considering vertical drift. This indicates that the performance of this system is comparable to those running higher end inertial measurement units, but can be outperformed.

Question 2: Can multiple low cost sensors compensate for each others weaknesses and match the performance of similar systems using state of the art MEMS?

The inclusion of the ultrasound system resulted in broadly more accurate tracking, as shown by the Vicon system results in sections 4.7.1 and 4.7.2. However, the largest limiting factor remains the drift in heading which is uncorrected by ultrasound. Further work could be done to incorporate a magnetometer system similar to that in [33] and [34]. If magnetometers could be effectively incorporated into the tracking system they could then provide an absolute reference in heading.

Question 3: To what extent is prior map knowledge required to provide accurate tracking?

Map constraints limit the possible motion of the user to areas bounded by walls. This is shown to give a very significant reduction in error and, of importance, is able to correct for the user's heading. This means that the errors do not grow significantly and can be used to track even long walks with a high degree of accuracy. However, as shown by the results given by the Type 1 walk, should the environment provide few constraints then the particle filter will not offer significant improvement.

The extent to which map knowledge is required depends on the length of walk being tracked. The longer the walk, the more errors accumulate and therefore the more important having motion limiting constraints become.

6.2 Limitations

Motion tracking is a wide multidisciplinary area with researchers from the medical field through to robotics, computer science and engineering. To create a comprehensive motion tracking system which can work effectively in any environment is a huge undertaking and so the following limitations were inherent to this thesis and can be approached in later research.

1. **The only mode of locomotion available to the user is walking.** More complex types of movement, such as running or jumping, increases the challenge substantially. Additionally, we assume that moving floor surfaces such as elevators, escalators and moving walkways are not used. The algorithms presented here could be adapted to account for these

more exotic forms of travel, however such extensions are left for subsequent research.

2. **Only indoor tracking is examined.** Outdoors the availability of GPS, the rugged terrain, and the lack of mobility limiting features makes the problem very different. Therefore, only the cases in which the user is indoors were presently considered.

6.3 Future Work

Many avenues exist for improving the work demonstrated in this thesis. There have been several works in literature which have showed interesting and novel approaches which could be adapted and merged together to potentially make a very accurate system.

Magnetometers

One of the dominating errors in unconstrained inertial navigation was the gradual degradation of the user's heading. HDR provided a way to limit this drift, however it is not possible using IMU technology alone to eliminate it completely. Magnetometers can provide an absolute heading reference which can provide a way to indefinitely give a drift-free heading. However, indoors the presence of large magnetic fluctuations makes them unreliable. Although the use of magnetometers was briefly examined in this thesis (Section 3.6) further work in this area could be one of the most promising avenues for future improvements.

Simultaneous Localisation and Mapping

As was shown in Chapter 5 map information can be a very powerful way of reducing the navigation errors. However, in Chapter 5 we used prior map knowledge to provide the environmental constraints. An extremely useful extension of this technique would be to incorporate more ideas from the area of robotics and perform simultaneous localisation and mapping (SLAM). This has been attempted to a degree in [53], however several improvements are possible and could provide a way to build up a map on the fly and therefore always be able to use particle filtering to enforce environmental constraints.

Smartphone IMUs

Smartphones contain within them an MEMS inertial sensor. As smartphones are extremely widespread in today's society, using the information from a smartphone IMU as an additional sensor to foot mounted IMUs could be an interesting avenue for further research. It would be very challenging to use smartphone IMUs in a stand alone fashion without the use of foot mounted IMUs as previous works on pocket mounted systems [74] have shown much larger errors compared to foot mounted systems.

The information from a smartphone IMU could be used in a similar fashion to [26] where a maximum separation between the various IMUs is declared and used to provide navigation constraints.

Multiple Users

Should multiple users be equipped with IMUs and have inter agent ranging sensors, then additional information between the distance which separates each user is obtained. This is of particular interest for firefighters and other emergency responders due to their team focused approach to operations. [75] looked into this area of investigation but lacked vigorous experimental validation. Therefore, further work into this type of system could yield promising results.

Chapter 7

Appendix

7.1 Bill of Materials

This section details all of the parts used in the system.

7.1.1 Microcontrollers

2x Arduino Uno microcontrollers, one for each shoe (∼£20 each depending on supplier).

7.1.2 Sensors

1. 2x ADXL345 and 2x ITG3200 mounted on two separate SparkFun 6 Degrees of Freedom IMU Digital Combo Board (∼£30 each depending on supplier).
2. 16x HC-SR04 ultrasound sensors, 10 mounted on the left shoe and 6 mounted on the right shoe (∼£1.5 each depending on supplier).
3. 6x Vishay TSOP38438 IR Receivers (∼£0.5 each depending on supplier).
4. 4x IR LED (∼£0.2 each depending on supplier).
5. 2x GP2Y0A41SK0F Position Sensitive Devices (∼£6 each depending on supplier).

6. 10x Prowave 400SR120 ultrasonic transducer (∼£5 each depending on supplier).
7. 6x Prowave 400ST120 ultrasonic transducer (∼£5 each depending on supplier).

7.1.3 WiFI Communication

1. 2x XBee Wi-Fi (S6B) PCB antenna (∼£45 each depending on supplier).
2. 1x Xbee explorer USB (∼£20 depending on supplier).
3. 1x Arduino Xbee shield (∼£11 depending on supplier).

7.1.4 Miscellaneous

All these items have a negligible cost.

1. 2x $10\mu F$ capacitors.
2. 2x BC 638 PNP transistor.
3. 2x $2.2\,\mathrm{k}\Omega$ resistors.
4. 2x $10\,\mathrm{k}\Omega$ resistors.
5. 1x $47\,\Omega$ resistor.

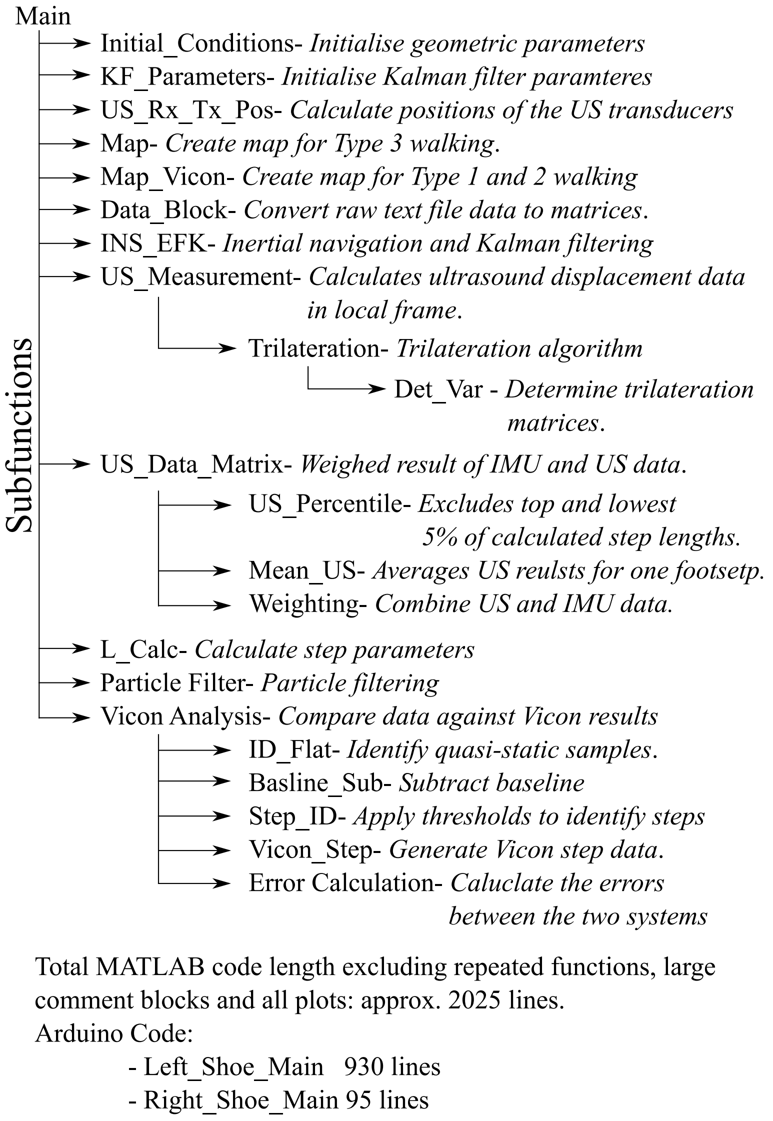
7.1.5 Overall System Costs

The minimum system cost for a one off prototype module composed of 2 IMUs, one Arduino microcontroller and two PSDs is ∼£92. Should wireless connectivity be included, which is highly desirable in many applications a cost of ∼£121 must be added. This setup of IMU system plus wireless connectivity (∼£213) is the most cost effective set-up considering the limited improvement offered by the ultrasound system. Of course this represents the cost of a single unit and several steps can be taken to reduce the cost further, for example the accelerometers and gyroscope sensors could have been purchased and a board developed for them in-house. Should the ultrasound system be included then an additional cost of ∼£127 needs to be included.

7.2 Code Overview

Here the flies and functions written to carry out the project are presented.

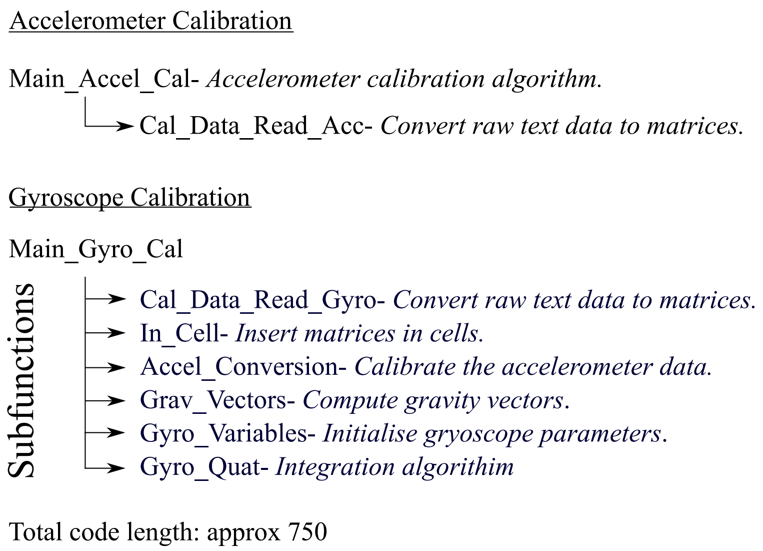
7.2.1 Motion Tracking Code


Main
Subfunctions
Initial_Conditions- *Initialise geometric parameters*
KF_Parameters- *Initialise Kalman filter paramteres*
US_Rx_Tx_Pos- *Calculate positions of the US transducers*
Map- *Create map for Type 3 walking.*
Map_Vicon- *Create map for Type 1 and 2 walking*
Data_Block- *Convert raw text file data to matrices.*
INS_EFK- *Inertial navigation and Kalman filtering*
US_Measurement- *Calculates ultrasound displacement data in local frame.*
Trilateration- *Trilateration algorithm*
Det_Var - *Determine trilateration matrices.*
US_Data_Matrix- *Weighed result of IMU and US data.*
US_Percentile- *Excludes top and lowest 5% of calculated step lengths.*
Mean_US- *Averages US reulsts for one footsetp.*
Weighting- *Combine US and IMU data.*
L_Calc- *Calculate step parameters*
Particle Filter- *Particle filtering*
Vicon Analysis- *Compare data against Vicon results*
ID_Flat- *Identify quasi-static samples.*
Basline_Sub- *Subtract baseline*
Step_ID- *Apply thresholds to identify steps*
Vicon_Step- *Generate Vicon step data.*
Error Calculation- *Caluclate the errors between the two systems*

Total MATLAB code length excluding repeated functions, large comment blocks and all plots: approx. 2025 lines.
Arduino Code:
- Left_Shoe_Main 930 lines
- Right_Shoe_Main 95 lines

Figure 7.1: Summary of functions written, with approximate code length.

7.2.2 Calibration Code



Accelerometer Calibration

Main_Accel_Cal- *Accelerometer calibration algorithm.*

Cal_Data_Read_Acc- *Convert raw text data to matrices.*

Gyroscope Calibration

Main_Gyro_Cal

Subfunctions

Cal_Data_Read_Gyro- *Convert raw text data to matrices.*

In_Cell- *Insert matrices in cells.*

Accel_Conversion- *Calibrate the accelerometer data.*

Grav_Vectors- *Compute gravity vectors.*

Gyro_Variables- *Initialise gryoscope parameters.*

Gyro_Quat- *Integration algorithim*

Total code length: approx 750

Figure 7.2: Summary of functions written, with approximate code length, for the calibration procedures.

Bibliography

[1] Martin Ford. *The Rise of the Robots.* Oneworld, 2015.

[2] Viacheslav Filonenko, Charlie Cullen, and James D. Carswell. Indoor positioning for smartphones using asynchronous ultrasound trilateration. *ISPRS International Journal of Geo-Information*, 2(3):598, 2013.

[3] C. Fischer and H. Gellersen. Location and navigation support for emergency responders: A survey. *Pervasive Computing, IEEE*, 9(1):38–47, Jan 2010.

[4] Mike Hazas and Andy Ward. A novel broadband ultrasonic location system. In Gaetano Borriello and LarsErik Holmquist, editors, *UbiComp 2002: Ubiquitous Computing*, volume 2498 of *Lecture Notes in Computer Science*, pages 264–280. Springer Berlin Heidelberg, 2002.

[5] Jaegeol Yim, Seunghwan Jeong, Jaehun Joo, and Chansik Park. Development of kalman filters for WLAN based tracking. In *Future Generation Communication and Networking Symposia, 2008. FGCNS '08. Second International Conference on*, volume 5, pages 1–6, Dec 2008.

[6] Farnel. Xsens MTI-20-2A5G4 MEMS module. http://uk.farnell.com/xsens/mti-20-2a5g4/vru-mems-motion-tracking-module/dp/2452307?MER=BN-2452307, April 2016.

[7] Greg Welch and Eric Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Comput. Graph. Appl.*, 22(6):24–38, November 2002.

[8] D. Fox, Jeffrey Hightower, Lin Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *Pervasive Computing, IEEE*, 2(3):24–33, July 2003.

[9] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[10] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(1):35–45, 1960.

[11] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[12] Sebastian Thrun et al. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, pages 1–35, 2002.

[13] A.R. Jimenez, F. Seco, J.C. Prieto, and J. Guevara. Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, pages 135–143, Dresden, Germany, March 2010.

[14] Sotgiu E Loconsole C Dehkordi MB, Frisoli A. Pedestrian indoor navigation system using inertial measurement unit. In *Sensor Netw Data Commun*, volume 3 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2014.

[15] M. Kourogi and T. Kurata. Personal positioning based on walking locomotion analysis with self-contained sensors and a wearable camera. In *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*, pages 103–112, Tokyo, Japan, Oct 2003.

[16] Thomas Judd. A personal dead reckoning module. In *Institute of Navigation (ION) GPS*, volume 97, pages 47–51, Kansas City, USA, 1997.

[17] E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *Computer Graphics and Applications, IEEE*, 25(6):38–46, Nov 2005.

[18] John-Olof Nilsson, Isaac Skog, and Peter Händel. A note on the limitations of ZUPTs and the implications on sensor error modeling. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN) 13-15th November*, Sydney, Australia, 2012.

[19] A. Peruzzi, U. Della Croce, and A. Cereatti. Estimation of stride length in level walking using an inertial measurement unit attached to the foot: A validation of the zero velocity assumption during stance. *Journal of Biomechanics*, 44(10):1991 – 1994, 2011.

[20] H. Martin Schepers, Edwin H.F. van Asseldonk, Chris T.M. Baten, and Peter H. Veltink. Ambulatory estimation of foot placement during walking using inertial sensors. *Journal of Biomechanics*, 43(16):3138 – 3143, 2010.

[21] J. Borenstein, L. Ojeda, and S. Kwanmuang. Heuristic reduction of gyro drift in IMU-based personnel tracking systems, 2009.

[22] Gabriel Girard, Stphane Ct, Sisi Zlatanova, Yannick Barette, Johanne St-Pierre, and Peter Van Oosterom. Indoor pedestrian navigation using foot-mounted IMU and portable ultrasound range sensors. *Sensors*, 11(8):7606–7624, 2011.

[23] John Elwell. Inertial navigation for the urban warrior, 1999.

[24] Hamza Benzerrouk, Alexander Nebylov, Hassen Salhi, and Pau Closas. MEMS IMU/ZUPT based cubature kalman filter applied to pedestrian navigation system. In *Proceedings of International Electronic Conference on Sensors and Applications*, www.sciforum.net/conferences/ecsa-1/, 2014.

[25] Raul Feliz, Eduardo Zalama, and Jaime Gmez. Pedestrian tracking using inertial sensors. *Journal of Physical Agents*, 3(1), 2009.

[26] Isaac Skog, John-Olof Nilsson, Dave Zachariah, and Peter Händel. Fusing the information from two navigation systems using an upper bound on their maximum spatial separation. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–5, Sydney, Australia, 2012. IEEE.

[27] John-Olof Nilsson, Amit K Gupta, and Peter Händel. Foot-mounted inertial navigation made easy. In *Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on*, pages 24–29, Busan, South Korea, 2014. IEEE.

[28] Kaiyu Tong and Malcolm H Granat. A practical gait analysis system using gyroscopes. *Medical Engineering & Physics*, 21(2):87 – 94, 1999.

[29] Shun-Yuan Yeh, Keng-Hao Chang, Chon-In Wu, Hao-Hua Chu, and Jane Yung-Jen Hsu. GETA sandals: A footstep location tracking system. *Personal Ubiquitous Comput.*, 11(6):451–463, August 2007.

[30] S.J.M. Bamberg, A.Y. Benbasat, D.M. Scarborough, D.E. Krebs, and J.A. Paradiso. Gait analysis using a shoe-integrated wireless sensor system. *Information Technology in Biomedicine, IEEE Transactions on*, 12(4):413–423, July 2008.

[31] I. Skog, P. Handel, J.-O. Nilsson, and J. Rantakokko. Zero-velocity detection - an algorithm evaluation. *Biomedical Engineering, IEEE Transactions on*, 57(11):2657–2666, Nov 2010.

[32] A.M. Sabatini. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *Biomedical Engineering, IEEE Transactions on*, 53(7):1346–1356, July 2006.

[33] V. Renaudin, M.H. Afzal, and G. Lachapelle. New method for magnetometers based orientation estimation. In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pages 348–356, Indian Wells, CA, USA, May 2010.

[34] Muhammad Haris Afzal Valrie Renaudin and Grard Lachapelle. Complete triaxis magnetometer calibration in the magnetic domain. *Journal of Sensors*, vol. 2010:10, 2010.

[35] Stacy J Morris. *A shoe-integrated sensor system for wireless gait analysis and real-time therapeutic feedback*. PhD thesis, Massachusetts Institute of Technology, 2004.

[36] Widyawan, M. Klepal, and D. Pesch. A bayesian approach for RF-based indoor localisation. In *Wireless Communication Systems. ISWCS 2007. 4th International Symposium on*, pages 133–137, Trondheim, Norway, Oct 2007.

[37] S. Beauregard, Widyawan, and M. Klepal. Indoor PDR performance enhancement using minimal map information and particle filters. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 141–147, Montereny, CA, USA, May 2008.

[38] Widyawan, M. Klepal, and S. Beauregard. A backtracking particle filter for fusing building plans with PDR displacement estimates. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pages 207–212, Leibniz University, Hannover, Germany, March 2008.

[39] Oliver Woodman and Robert Harle. Pedestrian localisation for indoor environments. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 114–123, Seoul, Korea, 2008. ACM.

[40] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241, Jun 2001.

[41] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598, Toronto, Ontario, Canada, 2002. AAAI.

[42] Michael J Milford, Gordon F Wyeth, and DF Rasser. RatSLAM: a hippocampal model for simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 403–408, New Orleans, USA, 2004. IEEE.

[43] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hhnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19:972–999, 2000.

[44] O. Stasse, A.J. Davison, R. Sellaouti, and K. Yokoi. Real-time 3D SLAM for humanoid robot considering pattern generator information. pages 348–355, 2006.

[45] D.E. Manolakis. Efficient solution and performance analysis of 3-d position estimation by trilateration. *Aerospace and Electronic Systems, IEEE Transactions on*, 32(4):1239–1248, Oct 1996.

[46] Alan L. Mackay. Generalised crystallography. *Computers and Mathematics with Applications*, 12(1):21 – 37, 1986.

[47] Herman Bruyninckx. Forward kinematics for HuntPrimrose parallel manipulators. *Mechanism and Machine Theory*, 34(4):657 – 664, 1999.

[48] F. Thomas and L. Ros. Revisiting trilateration for robot localization. *IEEE Transactions on Robotics*, 21(1):93–101, Feb 2005.

[49] William Murphy and Willy Hereman. Determination of a position in three dimensions using trilateration and approximate distances. *Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado, MCS-95*, 7:19, 1995.

[50] F. Izquierdo, M. Ciurana, F. Barcelo, J. Paradells, and Enrica Zola. Performance evaluation of a TOA-based trilateration method to locate terminals in WLAN. In *Wireless Pervasive Computing, 2006 1st International Symposium on*, pages 1–6, Phuket, Thiland, Jan 2006.

[51] Carl Fischer, Kavitha Muthukrishnan, Mike Hazas, and Hans Gellersen. Ultrasound-aided pedestrian dead reckoning for indoor navigation. In *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '08, pages 31–36, New York, NY, USA, 2008. ACM.

[52] Yongwon Jang, Seungchul Shin, Jeong Won Lee, and Seunghwan Kim. A preliminary study for portable walking distance measurement system using ultrasonic sensors. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 5290–5293, Lyon, France, Aug 2007.

[53] Jari Saarinen, Jussi Suomela, Seppo Heikkilã, Mikko Elomaa, and Aarne Halme. Personal navigation system. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 212–217, Sendai, Japan. IEEE.

[54] D. Weenk, D. Roetenberg, B.-J.J.F. van Beijnum, H.J. Hermens, and P.H. Veltink. Ambulatory estimation of relative foot positions by fusing ultrasound and inertial sensor data. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 23(5):817–826, Sept 2015.

[55] A.R. Jimenez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. In *Signal Processing, 2009. WISP 2009. IEEE International Symposium on*, pages 37–42, Budapest, Hungary, Aug 2009.

[56] E. Foxlin. *Handbook of Virtual Environments: Design, Implementation, and Applications.* CRC Press, 2002.

[57] David H. Titteron and John L. Weston. *Strapdown Inertial Navigation Technology.* The Institute of Electrical Engineers, 2004.

[58] Oliver J. Woodman. An introduction to inertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, August 2007.

[59] ET/IT & master AR project group WS. http://www.rst.e-technik.tu-dortmund.de/cms/de/Lehre/Projektgruppen/PG_20122013/English_version /index.html, 2012/13. Accessed: 27/04/206.

[60] Z.F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy. A new multi-position calibration method for MEMS inertial navigation systems. *Measurement Science and Technology*, 18(7):1897, 2007.

[61] Isaac Skog and Peter Händel. Calibration of a MEMS inertial measurement unit. In *XVII IMEKO World Congress*, pages 1–6, Rio de Janerio, Brazil, 2006.

[62] W.T. Fong, S.K. Ong, and A.Y.C. Nee. Methods for in-field user calibration of an inertial measurement unit without external equipment. *Measurement Science and Technology*, 19(8):085202, 2008.

[63] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[64] C. Jekeli. *Inertial navigation systems with geodetic applications*. Walter de Gruyter, 2000.

[65] Thomas F. Coleman and Yuying Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.

[66] Adrian Schumacher. *Integration of a GPS aided strapdown inertial navigation system for land vehicles*. PhD thesis, KTH Royal Instisute of Technology, Electrical Engineering.

[67] Eric Foxlin. Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, pages 185–194, Santa Clara, CA, USA, 1996. IEEE.

[68] Oliver J. Woodman. *Pedestrian localisation for indoor environments*. PhD thesis, University of Cambridge, 2010.

[69] SO Reza and Andrew J Fleming Moheimani. Piezoelectric transducers for vibration control and damping. *UK: Springer*, 2006.

[70] Andrew Martin Robert Ward. *Sensor driven computing.* PhD thesis, University of Cambridge, 1998.

[71] William Navidi, William S. Murphy, and Willy Hereman. Statistical methods in surveying by trilateration. *Computational statistics & data analysis*, 27(2):209–227, 1998.

[72] Stephen L. Chiu. Fuzzy model identification based on cluster estimation. *J. Intell. Fuzzy Syst.*, 2(3):267–278, May 1994.

[73] R. R. Yager and D. P. Filev. Approximate clustering via the mountain method. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(8):1279–1284, Aug 1994.

[74] U. Steinhoff and B. Schiele. Dead reckoning from the pocket - an experimental study. In *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*, pages 162–170, Mannheim, Germany, March 2010.

[75] John-Olof Nilsson, Dave Zachariah, Isaac Skog, and Peter Händel. Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging. *EURASIP Journal on Advances in Signal Processing*, 2013(1):1–17, 2013.