Towards Seamless Pedestrian Navigation

by Pekka Peltola, M.Sc. in Electronics

Thesis submitted to The University of Nottingham

for the degree of Doctor of Philosophy, May 2018



Abstract

The development of the Global Navigation Satellite Systems (GNSS) has offered the users the applications for car navigation, surveying and for other outdoor needs. While these outdoor services have now been available for over a decade, the users have started to demand more. Indoor positioning has thus emerged as a new field that challenges the old ways of doing things. Instead of the paper and the billboard maps, a demand for the digitized versions has become obvious. And together with the digital maps, the demand for the indoor positioning technologies has increased. The receivers of the GNSS systems function well in the open outdoor areas. But when indoors, in urban canyons or under the trees, the regular GNSS receivers are not able to reliably provide a positioning solution.

This thesis answers the questions on which of the alternative technologies and methods are available and can solve this challenge. The development and testing of a novel seamless positioning system helps to explore the current status. This study reveals how accurate the different positioning technologies are and which of

the technologies seem to be the most suitable.

The sensor technology types that can be used for positioning are many. The GNSS infrastructure maintenance is costly. GNSS positioning has been studied and developed for multiple decades. The focus shifted from military use into the public use after the change of the millennium. In comparison, the indoor positioning technologies have not yet received similar scale support. This is mostly due to the challenge that the indoor and urban areas pose.

Indoors, the RF signals attenuate due to the walls made of varying materials and with different thicknesses. The Wi-Fi technology has been the first potential help for indoor localisation. The existing infrastructure makes it the most attractive, in terms of deployment. According to the literature review study though, the Bluetooth low energy (BLE) and the Ultra-wideband (UWB) technologies are the better options for ease of deployment and accuracy, respectively.

A combination of outdoor and indoor positioning technologies demands inference processes for a reliable detection of the current environment. Recording the sensors and examining the change in quality and the current sensor measurement consistencies and availability, are the inputs for these context detection processes. A proven inference process is shown in this thesis that handles the basic indoor-outdoor switching scenarios for pedestrians.

When the user approaches an entrance of a building, the designed system changes its state into a preparatory state, enabling indoor and outdoor switch detection processes. This indoor/ outdoor switching scenario is handled by a context engine.

Using a context engine in the implementation represents a novel approach that will help future application designers to use and develop more reliable indoor and outdoor navigation systems.

The information fusion is the final phase in the positioning solution derivation process. The positioning technologies that have the deployed infrastructure within the areas of interest, will provide individual positioning solutions. It is the mobile devices' responsibility to check the available positioning technology options and to pick and to combine the best available information for the context, derived by the context inference engine.

The best methods for processing a positioning solution vary between the positioning technologies. The main methods are examined and the best are chosen and used in the final implementation. The path-loss modelling is the faster method when using BLE. It is slightly more advantageous, although slightly less accurate, when compared with the very laborious Fingerprinting alternative. For an UWB system, the accurate timing method is the better option over the signal strength measures.

These sensor subsystems produce the positioning solutions that are of different quality. The fusion filter is then the motor, where the information fusion of these subsystems happens. For the fusion, multiple methods exist. Accuracy and computational efficiency are the driving factors in this process of continuous development of an ultimate seamless navigation system.

A truly seamless navigation system uses the best available sensors and methods in the derivation of the most accurate positioning solution. This acquisition relies

on the context inference process. The system adapts to all contexts throughout the navigation process. The developed novel seamless positioning system in this thesis was implemented successfully and was proven to operate correctly. The accuracy of the implemented system reaches a level below 4 metres (95%), outdoors. Indoors, the accuracy reaches a level below 2 metres. The availability, the deployment and the existing infrastructure are an issue to be tackled in the future.

The thesis work was conducted as a part of the Marie Curie Initial Training Network project, named Multi-Pos. This project consisted of 15 researchers at different locations in Europe. Part of the thesis is a result of the cooperation between the project partners. Especially the parts of the work done on context detection and the literature review were completed with common effort.

Acknowledgements

Trials are always successful in the light of learning. It has been great, to be able to study and try out innovative approaches. I am happy that the future will hold many more interesting answers to our queries.

I would like to thank my supervisors Terry Moore and Chris Hill for their guidance throughout the research process. Special thanks for all the staff in NGI, Marie Curie organisation and the Universities for providing excellent support. Lukasz Bonenberg, Sean Ince and Stephen Fuller, Jari Nurmi, Alison, Jane and many more. Many thanks. And outside of the University, in work placements, Ramsey Faragher and Aled Catherall for their insights and tips into implementing smart methods.

I would like to thank my PhD colleagues at the NGI and the Marie Curie Multi-Pos project. Roberto, Melania, Peter, Mashuk, Jianlin, Jareer, Asyran, Laura, Bertrand, Francesco, Haiyun, Anahid, Ropo, Xiaolin, James, Jussi, Pedro and so many more. Thank you for making the time worth the while it was.

I would like to thank my family and friends for the invaluable support. And thank God, for reminding me of temporality.

July 2017

Contents

Abstract **1**

Acknowledgements **5**

List of Figures **14**

List of Tables **19**

Nomenclature **23**

1 Introduction **24**

1.1 Background 26

1.2 Research Problem 29

1.3 Research Hypotheses 31

1.4 Research Aim, Objectives and Methods 32

1.5 Contributions to Knowledge 34

1.6 Outline of the thesis 35

2 Navigation Sensor Review **37**

2.1 Maps and Infrastructure 38

2.1.1 Mapping the environment . . . 38
2.1.2 On infrastructure . . . 41
2.1.3 On geometry . . . 43
2.2 Tactile and Inertial . . . 44
2.2.1 Tactile probing . . . 44
2.2.2 Inertial dead reckoning . . . 47
2.3 Sound and Pressure . . . 51
2.3.1 Sound and ultrasound . . . 51
2.3.2 Pressure . . . 54
2.4 Broadcast RF . . . 54
2.4.1 GNSS . . . 55
2.4.2 Pseudolites . . . 57
2.4.3 TV, mobile and radio signals . . . 58
2.5 Local RF . . . 59
2.5.1 WiFi . . . 59
2.5.2 Bluetooth . . . 61
2.5.3 Ultra-wideband . . . 62
2.5.4 Other . . . 64
2.6 Magnetic sensing . . . 65
2.7 Light . . . 66
2.7.1 Image acquisition . . . 69
2.7.2 Feature extraction . . . 70

2.7.3 Match search 71
2.7.4 Camera calibration 72
2.8 Comparison and Sensor Technology Selection 73
2.8.1 Analytic Hierarchy Process for Technology Prioritisation . . . 79
2.8.2 Discussion of the results 86

3 Methods for Seamless Navigation Systems 89

3.1 Inertial Dead Reckoning 90
3.1.1 Coordinate Systems and Quaternion Representation 90
3.1.2 Mechanisation 92
3.1.3 Initialization and Alignment 98
3.1.4 Error Sources and IMU placement 99
3.1.5 Step and Heading Systems 101
3.1.6 Conclusions on IMU 102
3.2 Timing based Ranging using GNSS and UWB 104
3.2.1 GNSS Receiver 104
3.2.2 Ultra-wideband Ranging 108
3.3 Signal Strength Measures for BLE 112
3.3.1 Path-Loss Modelling 114
3.3.2 Fingerprinting 115
3.3.3 Conclusions on BLE 116
3.4 Map Features and Constraints 117

3.4.1 Conclusions on Mapping 118
3.5 Navigation Context Inference 118
3.5.1 Location Based 119
3.5.2 Behavioural 121
3.5.3 Conclusions on Context Inference 122

4 Sensor Fusion Design **124**

4.1 Configurational Flexibility 125
4.2 Environmental Adaptability 126
4.3 Classification 129
4.4 Kalman Filters 132
4.5 Particle Filters 141
4.6 Fusion Architectures 148
4.6.1 Main Architectures 148
4.6.2 Propagation and Observation Models for Closed Loop 15 Error State Kalman Filter 149
4.6.3 Sensor Data Abstraction and Heterogeneity 152
4.6.4 Alternative Architectures 154
4.6.5 Additional Methods and Summary 157
4.7 Conclusions 159

5 Reciprocity in Navigation Systems **161**

5.1 Transfer Inference 163

5.1.1 Trigger for a transfer from Kalman filter to Particle filter . . . 164
5.1.2 Trigger for a transfer from Particle filter to Kalman filter . . . 165
5.2 Common Parameters . . . 166
5.2.1 Accumulated position and yaw error updates . . . 173
5.3 Filter Transfer . . . 175
5.3.1 Kalman Filter to Particle Filter conversion . . . 177
5.3.2 Particle Filter to Kalman Filter . . . 178

6 Map and Context Engine Design **181**

6.1 Seamless Map . . . 182
6.2 Context Inference Engine Design . . . 184
6.2.1 Interface Agreement between Context Engine and Sensor Subsystems . . . 184
6.2.2 Interface Agreement between Context Engine and Navigation Filter . . . 188
6.2.3 Context Engine Inference . . . 189
6.3 Concluding remarks . . . 192

7 Particle Filter Design **194**

7.1 Design Basis . . . 194
7.2 Positioning Fixing and Dead Reckoning Sensor Subsystems Design and Tests . . . 197
7.2.1 BLE Subsystem . . . 197

7.2.2 UWB Subsystem . . . 205
7.2.3 Dead Reckoning with IMU . . . 207
7.3 Particle Filter Fusion . . . 210
7.3.1 Defining the Perturbation Characteristics . . . 211
7.3.2 Particle Filter Update using Tukey-style Distribution . . . 215
7.3.3 Resampling Strategy and Results Evaluation . . . 218
7.4 Map Matching . . . 224
7.5 Conclusions . . . 224

8 Kalman Filter and Pentagon Buffer 226

8.1 Design Basis . . . 227
8.2 Pentagon Buffer enhanced Kalman Filter . . . 228
8.2.1 Pentagon Buffer Propagation . . . 230
8.2.2 Pentagon Buffer Collision Logic . . . 231
8.3 Evaluation . . . 232
8.4 Track Form Matching . . . 239
8.4.1 Short Term GNSS Track Inaccuracies . . . 240
8.4.2 Longer Term inaccuracies within the IMU Track . . . 243
8.5 Using Pentagon Buffer properties in Initialisation . . . 248
8.6 Conclusions . . . 249

9 Seamless Navigation Filter 251

9.1 Design Basis . . . 252

9.2 Outdoor Performance 254
9.2.1 Particle Filter Characteristics 260
9.2.2 Track Form Matching Characteristics 262
9.3 Indoor Performance 268
9.3.1 Novel BLE positioning 271
9.4 Seamless Performance 273
9.4.1 State Machine 273
9.4.2 Novel Collision Logic for Pentagon Buffer 274
9.4.3 Particle filter with Pentagon Buffer 276
9.4.4 Seamless System Characteristics 278
9.5 Concluding Remarks 284

10 Communication and Conclusions **285**

10.1 On Context Inference 285
10.1.1 Filter transfer 286
10.1.2 On Map Matching 287
10.1.3 On Abstraction 288
10.2 On Sensor Fusion 289
10.2.1 Technologies and Methods 290
10.2.2 Characteristics Adjustments 291
10.2.3 Performance and Achievements 292
10.3 Seamlessness 293

Bibliography **294**

A Levelling using Quaternions **326**

List of Figures

2.1 Google Earth 3D view of London. 39

2.2 Infrastructure layer supports upper the Location Awareness, Service and Application Layers. 42

3.1 Quaternion rotation 92

3.2 Mechanisation process 98

3.3 GNSS receiver service levels 105

3.4 Decawave TWR 110

3.5 Inertial traces for stairs and cycling. 122

3.6 Summary of location and user behaviour context inference 123

4.1 Layer structure that enables navigation on mobile devices. 125

4.2 Three layers of adaptivity. 128

4.3 Support Vector Machine optimal hyperplane. 131

4.4 Kalman filter loop. 135

4.5 Particle filter pseudocode. 143

4.6 Closed Loop Kalman Filter. 150

4.7 Sensor measurement information abstraction levels. 153

4.8 Joint Directories of Laboratories model. 155

4.9 Blackboard model. 156

5.1 Particle and error state Kalman filter position parametre equivalence. 167

5.2 Direction of movement and yaw angle definitions. 168

5.3 Particle filter propagation distribution in pedestrian applications. . . 170

5.4 The test setup. 173

6.1 Map of the Surroundings of Nottingham Geospatial Institute with KML layer wall information. 183

6.2 System blocks. 185

6.3 Context engine implementation. 190

7.1 Top level particle filter system design diagram. 195

7.2 BLE anchor placement. 198

7.3 Quality of the positioning result acquisition using BLE fingerprinting. 201

7.4 The BLE resulting track using 2s measurement time window. 202

7.5 BLE track accuracy. 203

7.6 Second walk test with BLE fingerprinting. 204

7.7 Second test walk accuracy. 204

7.8 UWB anchors and test track. 206

7.9 UWB system accuracy with poor geometry especially along easting axis. 207

7.10 Inertial track using 62.5Hz sampling rate. 209
7.11 Particle filter structure. 210
7.12 Perturbation modelling setup. 211
7.13 Perturbation modelling results. 213
7.14 Perturbation modelling results 2. 214
7.15 Update timing for the Particle filter. 215
7.16 The Tukey-style window for the particle filter measurement update. . 217
7.17 The difference between Gaussian and Tukey-style particle filter measurement update. 218
7.18 Particle filter track. 220
7.19 Particle amount to accuracy examination. 222
7.20 The 100-particles filter accuracy indoors. 223

8.1 The Kalman filter with Pentagon Buffer. 227
8.2 The Kalman filter solution with Pentagon Buffer. 228
8.3 Error State Kalman Filter solution with Pentagon Buffer. 233
8.4 Single GNSS receiver and GNSS and IMU Kalman fusion accuracies. 234
8.5 Mechanisation solution. 235
8.6 Probable multipath condition on the used GNSS receiver. 236
8.7 GNSS receiver gap of 50s. 237
8.8 GNSS receiver gap of 100s. 238
8.9 GNSS gap accuracy compared to gapless. 238

8.10 Short Time Window Track Form Matching. 240
8.11 Short Track Form Matching comparison. 241
8.12 Angle difference between GNSS and inertial tracks. 241
8.13 The Effect of Multipath detection application. 242
8.14 The Effect of Multipath detection application. 243
8.15 7-point IMU track quality evaluation. 244
8.16 Result of corrective turn application on the inertial track. 245
8.17 Result of corrective turn application on the inertial track. 246
8.18 Resulting track when both short and long term corrections are applied. 247
8.19 The particle filter switch to pentagon buffer enhanced Kalman filter. . 248

9.1 Final design. 252
9.2 Final filter design accuracy outdoors. 255
9.3 Tripartite filter track. 256
9.4 Collision logic example. 257
9.5 Collision logic example 2. 258
9.6 Pentagon buffer reinitialisation. 259
9.7 Particle amount analysis. 260
9.8 Computational speed analysis. 261
9.9 Short Track Form matching only. 263
9.10 Right swerving inertial only track. 264
9.11 Long track form matching aid. 265

9.12 Long track form matching corrected inertial only track. 266
9.13 Indoor accuracies for Tripartite filter, BLE and UWB. 269
9.14 Indoor tracks for the Tripartite filter, BLE and UWB. 269
9.15 Indoor anchor placement for the BLE and UWB positioning fixing systems. 270
9.16 BLE positioning result using the alternative radiomapping method. . 272
9.17 State machine describing the Context Engine operation. 274
9.18 Enhanced Collision Logic. 275
9.19 Particle filter with pentagon buffer. 277
9.20 Revised state machine describing the Context Engine operation. . . . 278
9.21 Outdoor to indoor track using the novel design. 279
9.22 Indoor to outdoor track using the novel design. 280
9.23 Outdoor to indoor transition trails for IMU and GNSS. 281
9.24 GNSS horizontal accuracy when entering indoors. 281
9.25 Indoor to outdoor transition trails for IMU and GNSS. 282
9.26 GNSS horizontal accuracy when entering outdoors. 282
9.27 Typical UWB measurements when passing. 283
9.28 Typical BLE measurements when passing. 283

List of Tables

2.1 Tactile, inertial and compressible media based sensor comparison table. 74

2.2 Broadcast RF comparison table. 75

2.3 Local are RF technologies comparison table. 77

2.4 Magnetic and camera sensor comparison table 78

2.5 Analytic Hierarchy Process criteria comparison matrix. 80

2.6 Criteria weighing table. 80

2.7 Data rate comparison table. 82

2.8 Accuracy comparison table. 82

2.9 Reach comparison table. 83

2.10 Energy efficiency comparison table. 83

2.11 Device cost comparison table. 84

2.12 Infrastructure cost comparison table. 84

2.13 Analytic Hierarchy Process result table. 85

Nomenclature

AHP	Analytic Hierarchy Process
BLE	Bluetooth Low Energy
CEP	Circular Error Probable
CI	Consistency Index
CSR	Cambridge Silicon Radio
EKF	Extended Kalman Filter
FAST	Features from Accelerated Segment Test
FPGA	Field Programmable Gate Array
GDOP	Geometric Dilution of Precision
GNSS	Global Navigation Satellite System
IMM	Interacting Multiple Model
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IR	Infra-Red
JDL	Joint Directories of Laboratories
KML	Keyhole Markup Language
LDA	Linear Discriminant Analysis
LoS	Line of Sight

MAC	Medium Access Control
MCMC	Markov Chain Monte Carlo
MEMS	Micro-Electro-Mechanical System
NED	North East Down
NGB	Nottingham Geospatial Building
PCA	Principal Component Analysis
PRN	Pseudo-Random Noise
PVDF	PolyVinyliDene Fluoride
RANSAC	RANdom SAmple Consensus
RFID	Radio Frequency IDentification
RSSI	Received Signal Strength Indicator
SIFT	Scale-Invariant Feature Transform
SIG	Special Interest Group
SIS	Sequential Importance Sampling
SMC	Sequential Monte Carlo
TDoA	Time Difference of Arrival
ToA	Time of Arrival
UKF	Unscented Kalman Filter
UWB	Ultra-Wide Band
WAN	Wide Area Network
WLAN	Wireless Local Area Network

ZARU	Zero Angular rate Update
ZUPT	Zero UPdaTe

Chapter 1

Introduction

The Global Navigation Satellite Systems (GNSS) offer the users positioning for outdoors. As GNSS is not able to provide accurate positioning indoors, the replacement strategies are being researched. This thesis examines the challenges and answers questions on which of the alternative technologies and methods can work as a replacement.

One of the most promising alternative for an accurate short term navigation solution provision comes by using the inertial sensors. The inertial navigation is based on the dead reckoning. In the pedestrian navigation setups, the Inertial Measurement Unit (IMU) is often strapped down onto a boot. The sensor readings are used to construct the position trace by following a process called mechanisation. The turn and acceleration of the boot are recorded at each epoch. The higher the sampling resolution specification the more accurately the mechanisation can be calculated.

The short term accuracy is good in the boot-mounted setup. If the sensor is, for example in a pocket, the drift compensation becomes a greater challenge. The drift can be compensated using for example the zero velocity updates. On the foot, the zero velocity updates are more easily distinguished. In the pocket, the zero velocity updates can't be applied since the phone and the sensors in it are usually moving all the time. The movement does not stop, like it does stop in the boot-mounted setup.

The long term accuracy, using cheap GNSS receivers, is approximately a few meters, when walking in an open area. Unfortunately, within the urban canyons and forest paths, the position fix can't sustain a similar level of accuracy.

Distinguishable characteristics can be noticed between these two technology types (IMU & GNSS). The characteristics divide the two into the absolute and the relative positioning technologies. The sensors can be divided also according to the physical phenomena they measure. The chapter 2 of the thesis examines the sensors that have been used previously for pedestrian navigation purposes. The chapters 3 to 5 concentrate on the methods used for positioning, using the reviewed technologies. Only the methods for the technologies that were found the most fit for purposes of pedestrian navigation are examined.

The chapters 6 to 8 involve a design process exploiting the general building blocks for a typical navigation system. A novel approach that separates the sensor subsystem and the fusion process implementations from the top-level requirements is used. This approach enables plug-and-play -ability for the future navigation

systems. Separate sensor manufacturers can design very different sensor subsystems and still use them for the same fusion process design or vice versa, use the same sensor subsystem for multiple sensor fusion implementations. The timing and data interfaces between the sensor subsystems and the fusion design must have been agreed in an adaptable manner.

The chapter 9 of the thesis presents a seamless design implementation using the most fit sensors for the task, the most appropriate methods for the task and the most relevant fusion processes for the task. The implementation is tested and evaluated. Finally, in chapter 10 conclusions are made according to the behaviour of the tested system.

1.1 Background

Many reviews on indoor positioning have been made previously. The latest such as Brena et al. (2017), divide the sensor technologies by the physical features they measure. A similar division was used by Mautz (2012).

In this thesis, the maps are considered as virtual sensors since they represent the real-life features in the virtual equivalent. The static properties and features of the environment make up the basic contents of the information in a map. These are the walls, the doors and the possible furniture. In addition, the dynamic features bring additional temporal information on the map objects. Ferreira et al. (2012) introduces moving polygon structure. This brings interesting possibilities that can

only be represented with digital maps. For example, a main door of a building could have a characteristic that shows the time interval when it is passable.

The map represents objects in the environment. The tactile sensors are often a part of this infrastructure. A capacitive sensing technology based floor or a mattress that detects customers at the entrance can be placed into this map. Serra et al. (2016) introduce how location can be tracked passively by the infrastructure.

The inertial sensors can track the inertial motion track of the object that it is attached to. Harle (2013) examines the variety of location tracking methods using inertial sensors. An inertial sensor measures the sensor acceleration and turn rates around itself.

The CRICKET system is an example of a sound based localisation system. Walter & Schweinzer (2014) used sound waves for object form detection in a small constrained test setup area. Bai et al. (2013) measured the air pressure instead of sound waves to detect the floor level.

The Global Navigation Satellite System (GNSS) receivers record the RF signals sent by the GNSS satellites. Four main global systems exist. GPS, GLONASS, GALILEO and BEIDOU satellites send ranging signals. The receivers try to synchronise a local copy with these. Pulsed pseudolites were used in the study conducted by Kuusniemi et al. (2012). The pseudolites are ground based transmitters imitating the GNSS satellites.

TV, mobile, radio, Wi-Fi, Bluetooth, Ultra-wideband are few of the protocols that define the use of RF spectrum. Each has their distinctive properties. A signal

strength and a time of arrival are the two main variables used for positioning purposes. Many of the signals were designed for communications, although, nowadays navigation purposes have been taken into account more and more.

A magnetic north can be measured using magnetometers, unless surrounding infrastructure contains magnetically disturbing elements. Hao et al. (2010) used a magnetometer for heading adjustments.

The camera based tracking methods use either infrared or visible light. The process is often similar in between different imaging methods. First, the image is acquired. Second, the features are extracted, like image markers were in the test made by Mulloni et al. (2009). These features can be matched and analysed with previously collected database or by inference methods. Calibration or other necessary operations are then performed according to the current needs of the task.

A context Engine is the motor that controls the navigation solution derivation process. The sensor readings can be recorded and then analysed. The analysis is used in inferring the context based information. This means, for example, whether the user is near to a sensor indoors, which can then be used to trigger indoor settings for the filters. Kray & Kortuem (n.d.) lay down three different sources of context information. The raw sensor measurements are the first option. The parallel applications, like the calendar can also be used in provision of location information. For example, if the scheduled meeting takes place in a known location, then it can be used to initialise the first guess if the mobile device is opened without any previous position information. Interaction with the user is the third source of

context information.

Sensor fusion plays the final role in deriving the position solution. A Kalman filter and a particle filter are the commonly used choices for fusing varying quality sensor data. The Kalman filter is the faster option and it propagates a single state vector. The particle filter, on the other hand, is a sequential Monte Carlo method as described by Doucet et al. (2001). Both consist of the propagation and the update phases.

1.2 Research Problem

The goals for the research were set within the scope and targets of the Marie Curie Initial Training Network (ITN) project titled Multi-Pos. The stated research output was to develop novel methods for multi-sensor integration for navigation applications. This was further specified to cover the pedestrian navigation primarily for the following reasons.

On open outdoor areas, the Global Navigation Satellite Systems are able to provide a positioning solution that is accurate enough for the needs of applications like geocaching. For autonomous vehicles, the current receiver technologies and methods are not yet quite accurate enough. This covers only research on the GNSS receiver structure and used algorithms. Pedestrian navigation research, like friend finding as mentioned by Basiri et al. (2014), is a promising research area as is autonomous driving, in terms of market value. It is an obvious, next enhancement

for the pedestrian services within an urban setting. Very few mobile applications offer positioning aiding for shopping in malls or for searching a friend that is located in the vicinity. The biggest problem is that no infrastructure for indoors yet exist, as does GNSS for outdoors.

Trials on Wi-Fi are only good for proximity based applications. In order to take steps forward, either a completely novel way for positioning indoors has to be found or focus of effort be put on selecting the best technologies and methods using the existing technologies. In this thesis, the latter approach is followed, since, at present, no new technology seems to provide the "killer" solutions for indoor positioning. This gives us the problem statement and research questions.

1. Which technologies are the most appropriate for urban pedestrian navigation?

2. Which methods are the most fit for these technologies?

3. What choices are there for getting the best navigation solution for pedestrians in different environments?

After the initial study on existing systems and integration strategies, the research questions are as follows.

1. What are the best technologies and methods for different environments and contexts?

2. How to switch between technologies and methods for acquiring the best possible navigation solution?

3. Are there new ways and approaches that can be used in the fusion process?

Using these questions, the formulation of research hypotheses follows next.

1.3 Research Hypotheses

The research hypotheses are postulated here to answer the research questions. The first phase is to be able to assess the current technologies. The first hypothesis is then the following.

1. Compare current technologies for indoor localisation in terms of accuracy and select the most promising ones.

Related to this question are the methods that are used in the acquisition of the most accurate solutions.

2. Compare the methods for the selected technologies in terms of computational load.

The environment needs to be inferred by using all possible sensory information. This leads to the third hypotheses.

3. Development of a reliable inference of different contexts using the selected sensors. Preliminary tests need to be conducted for adequate operation of the first prototype.

After this follows a design process for creating the novel seamless navigation filter that would work indoors and outdoors without user intervention. Moreover, the system should be able to switch sensory subsystems independent of the number

of subsystems that are attached to it. This means that the plug-and-play style approach is to be followed. Hypotheses accordingly are the following.

4. Design an autonomous seamless navigation system structure that does not need user intervention while changing environment and context.

5. Design an autonomous seamless navigation system that enables plug-and-play -ability.

6. Design and test a novel seamless multi-sensor pedestrian navigation system for indoors and outdoors.

These hypotheses are the objectives of this research. The more detailed path towards the objectives is explained next.

1.4 Research Aim, Objectives and Methods

Following on from the research hypotheses, the main aim for this research is: to design, implement and test a novel multi-sensor fusion system that can offer a positioning solution for a pedestrian in a seamless fashion. This solution is derived using the most fit sensors for the current context. Seamlessness is defined as an autonomous navigation system that can operate both indoors and outdoors without user intervention. The navigation solution is also acquired by using methods that are the most computationally efficient for the current context.

Prior to the design of the novel seamless system, in chapter 2, the thesis starts by presenting a literature survey on previous technologies that have been used for

indoor positioning. GNSS is the obvious choice for outdoors. The information is collected into a tabular form and the technologies analysed using the Analytical Hierarchy Process method.

After this, in chapters 3 to 5, a further study is conducted on methods that are used in deriving position solutions using the selected technologies. The selection process is a straightforward elimination of the obviously weaker methods in terms of accuracy.

Using the literature and defining common characteristics often seen in designs, we then go forward in chapters 6 to 8, and create a novel multi-sensor fusion filter designs. It is a collection of the best working parts and methods of the literature survey. Moreover, promising ideas that are feasibly implementable are taken along into the general design.

The main key points in the design of the systems are plug-and-play -ability. This means that the sensor subsystems are interchangeable. Fusion design does not need to be changed, when the sensor subsystem implementations are changed. Fusion design is thus independent of the sensor subsystem implementation, according to the interface agreements between sensor subsystems and the navigation filter bank design.

An implementation using this novel general flexible and adaptive design is then made in chapter 9. The best selected sensor technologies, methods, context inference methods and fusion methods make up the novel implementation. The end design is built using the needs for real-time applications. This allows an easy further

development into possible products. The success of the research is explored in chapter 10.

1.5 Contributions to Knowledge

Seamlessness, offered by the design of the tested system is the primary novel aspect of this work. The work conducted shows the possibilities and challenges in seamless navigation.

The wide literature survey goes through the existing technologies. A comparison process results show the most prominent alternatives for achieving seamlessness in pedestrian navigation. Seamlessness indicates the ability of the navigation system to cope both indoors and outdoors without external help.

After this, the methods for the selected technologies are examined. Four technologies, the GNSS, the Ultra Wideband and the Bluetooth low energy positioning fixing systems and the foot-mounted inertial dead reckoning, offer a reasonable and implementable level of seamlessness. An implementation is constructed using these technologies. This implementation, with the use of context engine, advances the knowledge and gives guidance in improving future seamless navigation systems.

Novel aspects are raised and added to the common knowledge, when designing an adaptive and context sensitive seamless system. These are the interfacing of the context engine, the sensor subsystems and the navigation filters, where timing and data format agreements are essential. The flexible filter bank usage, controlled by

the context engine through a system status, is another novel aspect in this work. Moreover, the plug-and-play style loose coupled design offers insights in the future development of modular adaptive navigation systems.

The usefulness of Bluetooth low energy is examined and improved. Each subsystem is tested separately and also by fusing all together. The Tukey-style position update offers a new way to fuse positioning fixing measurements. The track form matching improves the inertial and the GNSS fusion process.

1.6 Outline of the thesis

Chapter 2 first examines the previous literature on the prominent technologies for indoor positioning. Then an analysis is conducted, to determine the most fit alternatives. Chapter 3 continues by comparing the solution derivation methods for each selected technology and introduces the context detection in navigation systems. Chapter 4 introduces the terms for information fusion within the positioning systems.

Chapter 5 is the beginning of the novel contributions of this work. The similarities between filtering approaches are identified and necessary switching characteristics are presented. Chapter 6 starts the novel design and development process that is the main goal of our research. The context engine is the control logic for the novel filter and is explained in this chapter. It is developed using the ideas cherrypicked from the literature review, especially the chapter 3.

Chapter 7 and 8 design, develop and test novel particle filter and Kalman filter approaches respectively. A particle filter is here designed for indoor purposes, while the Kalman solution with the novel pentagon buffer design is for outdoor environments. Chapter 9 presents the final general solution for seamless navigation and then continues by implementing this real-time capable structure into a real-life environment. Performance is tested.

Finally, chapter 10 concludes the research and offers insights into future research direction and for possible commercial products.

Chapter 2

Navigation Sensor Review

This chapter browses through the literature on sensors that have been trialled for positioning purposes. These are examined according to the physical phenomena they measure. A physical phenomena can be measured using the appropriate sensors. It is possible to acquire positioning information by harnessing the physical phenomena through smart measurement technology selection and sensor setup. The sensor deployment consists of placing the sensors into their positions and choosing the optimal geometry and the number of sensors for obtaining the best possible positioning information.

The abstract information in a map and the building and obstruction location properties are looked through first. The phenomena sensing sensor spectrum is then thoroughly examined from the motion to the electromagnetic light detection sensors. A conclusion is made in the end of the chapter on how the technologies compare against the requirements for pedestrian navigation, especially indoors. The

prioritisation is presented on the basis of this feasibility study.

2.1 Maps and Infrastructure

Maps are the first and the intuitive way to represent geospatial information. The paper maps are being replaced by the digital equivalents. Digital representation has allowed an easier way to update the maps and it opens up new possibilities on the uses of maps. The digital maps can be easily enhanced with additional information to offer more accurate and up-to-date location information. For example, the traffic information updates on the car navigator software enable the driver to avoid traffic jams.

2.1.1 Mapping the environment

Apart from architectural CAD drawings and construction designs there are not yet many widespread detailed floorplan standards for navigation purposes. Open Geospatial Consortium initiative aims to provide a common schematic framework for wider use in indoor navigation applications. This is accomplished via a minimal set of components defined in the indoorGML (Geography Markup Language) standard. In addition, the previous standards, CityGML and IFC (Industry Foundation Classes), partly define the ways to represent indoor components as explained in www.opengeospatial.org (Accessed 16.2.2017). Other existing related standards and examples include X3D, ESRI BISDM, Google, Bing and Here maps.

The Keyhole Markup Language (KML) is used to create a structured document which can be displayed in an Earth browser like Google Earth. The locations can be pinpointed and the areas selected for example by using the polygon tool. Zichar (2013) raises issues on aspects of geovisualization. The amount of information to be presented should match the cognitive capabilities of the user and the user need. For the application designer, the open interfaces ease the designer's workload. The visualization symbol choices offer artistic freedom in representing the geospatial data on the Earth browser. The passage of time is also to be considered when representing historical geospatial data. This is discussed with more detail by Ferreira et al. (2012), who add a moving object class into the kml language. As an example, an application could show only the shops that are open at the moment.

Figure 2.1 demonstrates a 3D city view of London in Google Earth.



Figure 2.1: 3D view of London in Google Earth.

The GNSS signals are blocked by tall buildings. The building height information

can be exploited in the computation of a signal blockage. Groves et al. (2012) developed a shadow matching technique which is an example that uses this information in an attempt to overcome the troubles with the GNSS signal blockage and multipath. The non-line of sight signal propagation from satellites is detected and analysed together with the three-dimensional mapping. The processing power needed for using 3D maps with many details is high. Many researchers including Groves (2013*b*), Walder & Bernoulli (2010), Pinchin et al. (2012*a*), Deng et al. (2013), Wirola et al. (2010), Torres-Solis et al. (2010) examine the challenges in mapping and in virtual modelling and search ways for more effective representation techniques, including the shadow matching, polygon wall representation, heuristic heading, location constraint model, crowd-sourcing and connected graphs respectively.

The access to floormap data of certain buildings is restricted. Thus, crowdsourcing the floormap information would concentrate on the available public areas. The use of images in a construction of a virtual crowdsourced map of buildings is mentioned by Wirola et al. (2010).

The walls and obstructions in a map can be thought as spatial classifiers. The polygons and lines divide two areas. This and the information coming from different sensors is combined. The pedestrian dynamics model and moving patterns are different from the way the car sensor data is combined. Since the dimensions indoors are generally smaller, the requirement for positioning accuracy is higher indoors. The walls, the streetways and parks or other open zones function as constraints for possible user locations. These constraints help the application to snap or steer

the location solution onto sensible map locations. Incorrect solutions, like walking through walls or driving the car over walkways can be corrected. The wall's logic function can be adjusted as in the case of a tree fence when compared to a brick wall. A tree fence can be passable at certain points. Map information can be enhanced with for example an information on the passability of certain areas, or with an information of which kind of service is offered at the location (restaurant, shop, garage...).

Pinchin et al. (2012*a*) took advantage of the cardinal headings of the building. The cardinal headings are the most probable directions of travel indoors, because building corridors are usually symmetrical consisting of only a few directions. The buildings are often symmetrical and walking trails are often parallel to the wall or the furniture layout. A cardinal heading adjustment aligns the walking track to sensible positions along passable pathways.

The measured user walking trail correction using the map data is called map matching. Lan & Shih (2014) used the track shape, angles and ratio in the map matching process. The map matching fuses the map information with the positioning sensor or the track information after which the final positioning solution can be inferred.

2.1.2 On infrastructure

The GNSS and the Wi-Fi infrastructures are well established. The GNSS was set up for navigation purposes while Wi-Fi for solely communications. Afterwards,

it was noticed that the WiFi signal strength could be used for positioning. The existing infrastructure enables services. In order to offer positioning solutions via other technologies, a proper infrastructure should be set up. Figure 2.2 depicts an example of the layers that a general infrastructure could support.

Figure 2.2 shows the necessary layer structure for offering a navigation solution. On the top of the infrastructure layer is the location layer where the position is known but it is not associated with any service. Deng et al. (2013) discuss that on the cloud service and the application layers the location information is presented in a meaningful way for the user.

Surveillance cameras can be used for positioning purposes. Other methods and technologies that exploit the existing infrastructural systems are discussed by Kivimäki et al. (2014).

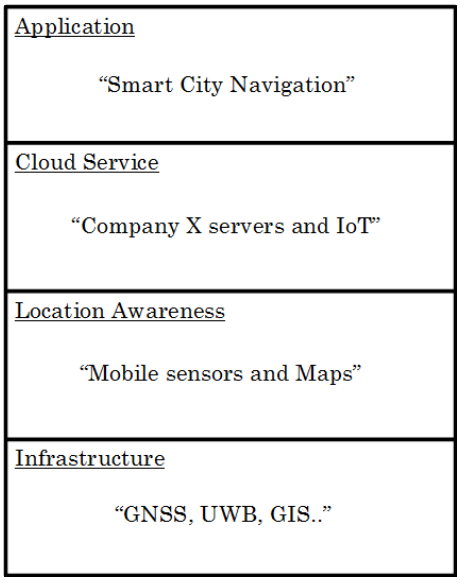


Figure 2.2: Infrastructure enables the services in the upper layers according to Deng et al. (2013).

The infrastructural localisation methods can often be implemented in a passive way. This means that the user doesn't have to wear tags or devices in order for the system to offer a positioning solution. Privacy issues may arise if the unaware user is positioned without permission. The floor pressure sensors, thermal sensors, sound and electric field change detection sensors are a few examples that are compared by Kivimäki et al. (2014).

2.1.3 On geometry

The sensor number and placement are the key factors in defining the quality of the positioning solution, in addition to the positioning sensor quality. The geometric dilution of precision is often used for analysing the quality of the GPS satellite geometry. Kirchhof (2013) studied the optimal sensor placement using the binary and the mixed integer programming methods and employing the geometric dilution of precision (GDOP) as the decisive factor on different sensor deployment setups.

The sensor deployment analysis goes through different deployment of anchors on the area map, taking into account factors such as mean node separation, and mean node population. Sharp et al. (2012, 2009), Feng et al. (2015) analyse further the use of GDOP in sensor deployment for indoors. When planning the deployment of the anchors these mentioned factors are good to be taken into account.

2.2 Tactile and Inertial

Sensing a physical contact or acquiring a measure of a direct physical distance or speed describes the techniques used in tactile and odometer sensing. A simple tactile sensor outputs binary data, either on or off. The force and odometer sensors output either transduced force measures or velocity information of an object.

The inertial sensors measure movement. There exists two kinds of movement of rigid objects, which are the rotation and the translation. The gyroscopes measure rotation whilst the accelerometers measure the object translation.

2.2.1 Tactile probing

The coordinate measuring machines are very accurate and expensive. They offer an exact way to position a product on a work bench but are not applicable for the pedestrian navigation scheme. A tactile sensor can be placed either on the user or within the environment. Not only can the tactile sensing prevent a robot from bumping into an object but the object or floor can send the location information when this collision event happens. Piezoelectric or capacitive touch surfaces, levers or buttons can be used in recognising activity at certain locations. The load cells on the floor may be used for sensing the position of a user within a building. Kivimäki et al. (2014) and Serra et al. (2016) list alternative techniques like the smart carpets and the calculation of the centre of pressure using load cell tiles.

Damian et al. (2015) analysed the artificial skin ridge formations for sensing

slipping objects on the artificial skin surface. A similar approach can be used for acquiring location information. The surface of the floor can give additional clues on the possible location if it is possible to detect between surface types. For example, when traveling on a forest path or on gravel or tarmac road the surface properties change. The speed of the car is different depending on the road and suspension activity depends on the road roughness and pedestrian crossing bumps. For the visually impaired a tactile or haptic feedback may be the only method to deliver information, as it is in the buttons of lifts without loudspeakers.

The vibration actuators on displays and on skin were tested by Jones et al. (2010) and by Filgueiras et al. (2016) respectively. The mechanical waves can be used to locate an object that is the source and is causing the mechanical waves. Vibratory controls can also be used to guide visually challenged people. Similar feedback as what Filgueiras et al. (2016) used is generated when a car is driving too close to the side of the road and the alarm stripe on the roadside makes the wheel whiz. Prete et al. (2012) tested a robot with artificial feeling sensors on limbs. The purpose of this research was to prevent harmful robot movements if an obstruction is detected next to a moving lever arm.

Bamberg et al. (2008) describe a wireless shoe sensor system that has three accelerometers, gyroscopes, four force sensors, two bend sensors, two pressure sensors and one electric field height sensor. The FSR-400 and the FSR-402 are the force sensors from Interlink Electronics. The applied force changes the resistance of the FSR-402 sensor. As a heal strike detection sensor a LDT0 piezoelectric PVDF

sensor was used. These sensors are small but will require additional signal processing hardware.

Dogruer (2014) examined the odometer parameters using a wheeled robot, while Reinstein & Hoffmann (2011) used an odometry model for their four-legged puppy robot. Compared to the wheel odometer model the gait model is more complicated. Slippage is a problem in both the wheel and gait tracking. The position error grows unless a slip can be detected and properly corrected. The system model and design for reading the odometer sensor data depends on the application. The odometer data needs to be matched with map data or otherwise combined with a GNSS or similar standalone positioning solution system for having a meaningful use. Jee & Song (2010) presented a heading filter measurement model that changes according to car movement and GNSS availability. The odometer measures the car speed and this is used with gyroscope angle information for the dead reckoning of the system when GNSS is not available. This is an example of a small context aware filtering selection.

Boll et al. (2011), discuss the sensory overload. When driving in the car, the audible information coming from radio, environment, navigator and passengers might sometimes be too much. The navigation is thus performed using a vibrating belt that tells the driver in which direction to turn next. The overall information content given to the driver using one technology channel in this method is less in comparison with plain audio guiding.

For pedestrian navigation, the environment can start sending location based

services when a user arrives at the scene. This approach would be energy saving since the service is provided only when needed.

2.2.2 Inertial dead reckoning

Harle (2013) conducted a literature review on inertial pedestrian positioning systems. The inertial sensor readings can be used to track a position via the position change information, using dead-reckoning or by statistical classification methods.

The gyroscopes measure the angular speed of the object. The accelerometers measure the force produced acceleration on the object. The gyroscope readings follow the attitude changes of the object. Double integrating the acceleration, according to the current attitude, the total movement of the object can be estimated. The systems that measure the inertial characteristics for dead reckoning are called the inertial navigation systems.

Perlmutter & Robin (2012) offer a helpful review on inertial MEMS and evaluate the future trends. The gyroscopes in the market appeared to offer the best performance to the price ratio in 2012. New integrated inertial sensors are in the markets. These include 10 degrees of freedom, three axis gyro, accelerometer and magnetometer sensors integrated with a pressure sensor.

Quinchia et al. (2012) divide the inertial sensor errors into the deterministic and the stochastic errors. The deterministic errors, as errors in manufacturing and mounting, can be calibrated out, but stochastic errors need modelling. These stochastic errors can be bias errors, orthogonal or scale factor errors. Inherent noise

and thermal changes while in operation also affect the error characteristics. Quinchia used Allan Variance analysis and Power Spectral Density techniques in analysing the random errors.

The zero updates can be used to reduce the accumulating drift errors in the double integration process. A drift means that the measured and derived position has an error increasing deviation from the true position. It is often described as the deviation from the true value over elapsed time or the covered distance. The inertial measurement units can be divided into three by accuracy as did Hide (2003). These three are the low-cost (least accurate), the tactical and the navigation (most accurate) grade sensors. The drift is hundreds of meters in several minutes even for the high quality tactical grade inertial navigation systems. If the IMU is mounted on a foot the step length can be tracked between two zero update periods.

The accelerometer and the gyroscope reading magnitude thresholds have been used in detecting the stance period in a gait cycle. Furthermore, a moving variance analysis on the signal acquired from the IMU on foot is considered the slightly better alternative. Multiple zero update detection methods have been tested for example by Brajdic & Harle (2013). A peak detection has been applied on heel strike detection. In the zero crossing method, the zero value crossings of the accelerometer data are collected. A template matching tries to identify the gait patterns within the collected data. Spectral analysis of the gait data gives out the peaks in the frequency pattern that can be interpreted as the step frequency. The peak-detection, the zero crossing, the template matching and the spectral frequency analysis are

ways to detect steps. Skog et al. (2010) compared four step detection algorithms. These were the acceleration moving variance, the magnitude, the angular energy rate detection and a likelihood method that combined all the previous three. Yin et al. (2014) deduce a step by using a logical AND function combining three different zero detection methods.

In a pocket, an IMU does not produce as clear signals as if placed on top of a shoe. The gait-cycle analysis is easier when the pedestrian walks at normal walking speed, because the zero velocity update period is easier to detect. Problems arise when the pedestrian stops and starts to shuffle his feet, ascends or descends stairs or runs. The zero-update detection is slightly different depending on the location context. IMU can be used in detecting this location context; the stairs, the elevator or a levelled floor.

The initialisation of an IMU when starting to navigate is another challenge. If the starting position and heading are slightly wrong in the beginning these errors will accumulate when navigating. Pinchin et al. (2012*a*) used the cardinal directions of buildings and built environments. This mapmatching technique adjusts the position by aligning it with the predefined cardinal directions that are typical in built environments.

Alternative ways for inferring the navigation solution using inertial sensors are for example by using learning methods like statistical model comparisons or artificial neural networks. Nguyen et al. (2010) implemented a learning statistical trajectory model for sensor displacement measurements at separate locations. Depending on

the estimated location a set of learned displacement measurements at that point were used to infer how the current measurement is used to get the new position. This functions as an alternative to the double integration. Another statistical learning method is presented by Adusumilli et al. (2014). The quantile regression forests from the sensory data are grown and are then used to determine the approximate position. Furthermore, Tejmlova et al. (2016) used context sensitive artificial neural network in parallel with a navigation filter to set the filter parameters. This IMU measurement readings learning helps to detect and apply the correct context mode (walk, run, stairs) which in turn enhances the overall filter performance. In addition, Le & Gebre-Egziabher (2016) adds inference for the step length constraining. They placed inertial measurement units on both feet. This inference process limits the step length by executing a range constraint update between the feet. Similar step length constraining update was applied by Weenk et al. (2015). Here the ultrasound sensors were measuring the range between the two feet. Ruppelt et al. (2016) applied finite state machine to infer the correct gait phase.

An additional positioning system is needed to keep the position and heading drift in limits. Hartmann et al. (2016) used an UWB system for additional position and attitude updates. Santoso et al. (2017) reviewed the inertial visual integration focussing on unmanned aerial vehicles. The visual tracking is very accurate and can correct IMU drift errors if line of sight conditions can be provided by the user.

Summarising, the inertial sensor, the gyro and the accelerometer readings can be derived into position information using either of the two.

- double integration or
- inference (using artificial learning)

The mechanisation process can be aided by for example these four updates.

- using zero updates
- range constraints
- heading corrections
- filter adjustments according to context detection

2.3 Sound and Pressure

The sound speed in air is approximately 346 m/s. This is obviously much less than the RF signal speed. In order to travel from source to target, the sound needs to travel through a medium like air or water. The properties of the medium need to be taken into account, when designing a localisation system based on sound waves. These properties are for example the temperature and the pressure.

2.3.1 Sound and ultrasound

Walter & Schweinzer (2014) used a sensor consisting of one transmitter and 5 microphones for detecting simple geometric objects in the environment. The object had no sensors on it. Instead a virtual 3D model was reconstructed analysing the echoes at the microphones using the transmitter. This is similar to what Torres-Solis et al. (2010) explains of multiple microphones being used for target pose estimation. The transmitters and microphones can be placed on the user or within the environment.

The two best known sound based positioning systems are the DOLPHIN and the CRICKET, also described by both Torres-Solis et al. (2010) and Kivimäki et al. (2014). Zhang et al. (2013) experimented so that the sound source was carried by the mobile user. The user carried a device with multiple ultrasonic (40kHz) transmitters that covered a horizontal 360 degrees range. The receivers were placed on known locations and each of them collected the timestamps of the ultrasound signal arrival. The position was then calculated by the network. On the contrary Rishabh et al. (2012) used barely audible signals which were played from the speakers in the environment and the mobile user carried a receiver that detected the played signals. The recorded signal detection was carried out by cross-correlation with reference signals.

In comparison to the camera systems the audio data throughput and amount is much smaller. The timing methods are also easier to be implemented since sound speed is much less compared to the electromagnetic waves. A radio frequency synchronisation is often used between the base stations because it is much faster than the sound signal. The systems combining Bluetooth with ultrasound are abbreviated as BLUPS. The location solution is triangulated from the synchronised sound signal ranging measurements. According to Torres-Solis et al. (2010), using the spread spectrum code division, multiple signals can be transmitted simultaneously.

The timing methods (ToA, Time of Arrival, and TDoA, Time Difference of Arrival) are used for timing the reception and sending of the sound signals. Zhang et al. (2013) used TDoA to calculate the position. Two filtering modes of the mea-

surement data were used for the detected LoS (line of sight) and the non-LoS cases. The Line of Sight case uses EKF (Extended Kalman Filter) while the non-LoS case uses robust EKF. This makes up a combined Interacting Multiple Model (IMM). It copes better than the two separate models by themselves. Moreover, the form of the signal plays a part in the localisation process. The audio signal composition and strength should be designed regarding all the external sound sources and properties of the target environment. Höflinger et al. (2014) exploited a chirp signal (18kHz) and envelope detection on the mobile device for location determination. The different signal forms and modulation options perform differently in varying environments. Höflinger et al. (2014) also lists multiple audio positioning trials conducted using smart phones.

The cooperative underwater vehicle scenarios were reviewed by Paull et al. (2014). A GNSS fix of one vehicle on the water surface was used to get the absolute position. This absolute position was then used for deriving the absolute positions of the other vehicles underwater by using the relative sonar distances in between each vehicle.

The noise and echoes are troublesome for sound system localisation. Unlike within the Ultra-wideband RF systems the correct echo peaks are harder to distinguish in the sound systems. Although this feature can be exploited. Bisio et al. (2015) exploited the sound echoes in detecting the indoor/outdoor context. Laverne et al. (2011) estimated the step length by having ultrasound transceivers on both feet. A commercial off-the-shelf ultrasonic transceiver, like the MB1010 ultrasound

sensor, can be used for tracking and proximity detection. The output data rate is 20Hz. The ranging capability reaches up to six meters.

2.3.2 Pressure

The atmospheric pressure sensors can be used to detect the floor level within a building. Pressure decreases as altitude increases. As the weather conditions changes the parameters, the reference pressure, the measured pressure and the temperature are slightly more challenging to deduce in real time. Bai et al. (2013) used a real-time calibration method with a barometer to detect the current floor where the user is. The hardware system that was built takes into account the three most relevant parameters, the temperature, the reference pressure and the measured pressure in order to solve the correct floor.

For a pressure altimeter, like the BMP085, the pressure conversion time is 25ms and it uses i2c for data transfer. This sensor also offers the temperature measurement.

2.4 Broadcast RF

The electromagnetic systems consist of the positioning systems using the radio frequencies with varying transmission power. Included here in this section are also both the magnetic sensors and the camera sensors. Metals and water reflect and absorb electromagnetic signals as wood or concrete merely attenuate. The electromagnetic

RF signals reflect and scatter indoors. A receiver suffers from a multipath condition, when the signals arrive via more than one path to the receiver. Torres-Solis et al. (2010) discuss that a multipath environment can be modelled by mapping the materials in the environment and by estimating the attenuation.

2.4.1 GNSS

The GNSS satellites broadcast positioning signals. These signals reflect from buildings and ground before reaching the GNSS receiver. The multipath environment can be modelled by mapping the materials in the environment and by estimating the attenuation. He et al. (2011) examined the multipath performance level characteristics of the receiver. This level information cannot only indicate the quality of the signal but also helps in sorting out the receiver location context between the rural and the urban operating environments.

The GNSS signal availability in urban areas and indoors is limited. Therefore, alternative methods need to be used for indoor and urban navigation. GNSS signal strength indoors as stated by Mautz (2012) (-170 to -190 dBW) is 5 to 30 dB weaker than outdoors (-158dBW). If the user stands on an open field with good satellite coverage, for a receiver, like the U-Blox AEK-4T GPS receiver that was used by Pinchin et al. (2012*b*), the cold start takes approximately 34s. The receiver's cold sensitivity is stated to be -142dBm. The raw data output rate is 10Hz. Mautz (2012) mentions that some high sensitivity receivers can cope with weak signals by using parallel correlation and longer correlation times.

A GNSS receiver compares a locally generated pseudorandom code with the ones received from satellites. A software defined radio receivers implement the whole signal processing in software. GPS and GLONASS are fully operational navigation satellite systems. In addition, Galileo and Beidou are being set up. Multi-constellation receivers offer more robust navigation solution. Integrity is better when there are more satellites and pseudoranges to choose from. The best satellite signals that are in view can be selected more easily.

The processor that is used in mobile devices, like the Snapdragon 400 processor chip, has an inbuilt IZat block for GNSS processing. The module provides position output. A cold start usually takes approximately two minutes or more. Compared with FPGA or hardware integrated circuit implementations the software defined radio implementations are not yet sufficiently fast. These software implementations would offer more configurability. Interesting tests have been made by Pesyna et al. (2014) to achieve decimetre level positioning by using carrier phase aided differential measurements on lightweight mobile devices. The data was sampled using a smartphone-quality GNSS antenna. The weak multipath rejection with the low-quality GNSS patch antennas was found to be the main cause for not achieving an integer fix via the phase ambiguity solving. And consequently, a centimetre level accuracy can't be reached if integer fixing fails. The study also found that a wavelength scale random antenna motion helps to reduce the time for achieving an integer fix using the low-quality antenna.

GNSS signal processing on mobile platforms require processing power. An inter-

esting approach is presented by Lucas-Sabola et al. (2016) where the processing is partly transferred to the cloud. This will, if the data transfer is efficient, make the computing load on the mobile device to be less.

2.4.2 Pseudolites

Pseudolites are transmitters on the ground and buildings that imitate if as transmitting GNSS signals. Kuusniemi et al. (2012) used pulsed pseudolites. Here a high sensitivity GNSS receiver was used together with a particle filter for positioning and an accuracy of few meters was achieved inside an office building. Pulsed pseudolites allow receivers to cope better with the near-far problem that arises when weaker signals can not be detected because of the presence of a very strong signal. When using pulsed pseudolite signals the interference with true satellite GNSS signals is smaller. As the pseudolite signals imitate the original satellite system signals the availability of these signals must be carefully planned. Mautz (2012), Kuusniemi et al. (2012), Basiri et al. (2014) state that in UK, same frequency bands cannot be used for pseudolites as are used for GNSS signals.

Locata is a pseudolite system using GNSS like signals where the nodes use accurate clock synchronisation signal. Slaves in the locatanet synchronise themselves with a master transceiver.

The Locatalite positioning accuracy is approximately 18cm on open areas according to Deng et al. (2013). The long-term performance of seven locatalites were tested with GPS by Bonenberg et al. (2013). Bonenberg et al. (2011) had

previously integrated GPS and Locata together for better availability.

2.4.3 TV, mobile and radio signals

Torres-Solis et al. (2010) mentions that television broadcast and cellular signals penetrate indoors better than GNSS signals. According to Deng et al. (2013) the positioning accuracy that has been achieved with these opportunistic signals is not very accurate (>50m). To enhance positioning coverage, TC-OFDM (Time & Code Division-Orthogonal Frequency Division Multiplexing) systems are being designed. These signals share the cellular communication network spectrum capabilities but contain the navigation message along with communication data. This helps the receiver to act more seamlessly and have a faster time to first position fix.

Popleteev et al. (2013), Navratil et al. (2016) analyse the FM radio signal feasibility for positioning. In the fingerprinting method, a feature like the signal strength is attached to a position. In the learning state a fingerprint map is generated that contains these signal strengths for the selected location. In the online state the measurement strengths are then compared with the interpolated signal strength maps and a position estimate can be derived. Accuracy for the FM fingerprinting trials were reaching approximately the level of Wifi fingerprinting indoors because of the small-scale variation of the signal indoors. But over larger areas the fingerprinting resolution decreased to few tens of metres. In addition, they tested GSM signal usage for positioning which had as bad accuracy as the FM radio fingerprinting approach.

2.5 Local RF

Liu et al. (2007) conducted a survey listing the possible methods for indoor RF positioning. Timing methods for WiFi or Bluetooth protocols are generally unable to reach an accuracy below few metres. Instead, the fingerprinting methods are preferred. Kolakowski & Djaja-Josko (2016) state that for Ultra-wideband the timing approach works well.

New RF communication standards take into account the needs for positioning. Previously localisation had to be implemented using protocols that were designed only for communication purposes. Groves (2013*b*) expects that modelling the signal propagation medium will be a bigger challenge in the future than the new standards. As an example, the multiwall model is a path loss model depicting the effect of the environment structures on signal propagation. A ray tracing analysis is another way to model the RF propagation within the environment. Faragher et al. (2012*a*) indicates that the transmitter power is limited for the devices operating on ISM frequencies. In practice, this means that the usable positioning distance indoors is few tens of metres.

2.5.1 WiFi

The WiFi availability is the best among the local RF protocols. There are many existing WiFi access points. The signal strength and the flight time are usually the measured attributes. A signal strength from an access point depends on the

distance and obstacles in between the access point and on the mobile unit. The measured signal strength is represented with the unique WiFi access point MAC address. Other method for ranging is to use accurate timing between the access points and the mobile user.

The current wireless network standards like the 802.11v consist also of a positioning protocol. Ciurana et al. (2011) assesses the 802.11v standard for ToA positioning. As well, Schatzberg et al. (2014) used WiFi timing integrated with INS. Woodman & Harle (2008) used WiFi fingerprinting with INS. Khan & Syrjärinne (2013) similarly integrate fingerprinting of WiFi access points with INS and maps. The tracking using fingerprinting involves offline and online phases. The RSS values are assigned to location fingerprints in the offline phase. In the online phase, the user measurements are compared and matched with these assigned values.

Furthermore Sendra et al. (2010) compares the coverage and interference of the different protocols in the 802.11 family. the WLAN network adapter that they used was Linksys WUSB600N. It works with the WiFi standards a, b, g and n and has 11Mbps data rate. The receiver sensitivity is -84dBm at this speed. One of the four types of access point routers used was Linksys WRT54GL. It has a RF power output of 18dBm. The data throughput varies between technologies because of the environment and the equipment used.

Crowdsourcing WiFi access points and signal strength fingerprints is a promising way to implement proximity extension for GPS positioning within the near future. The WiFi access point locations and the signal strength maps are already commer-

cially available. In addition, a cooperative position scheme offers another dimension into real time tracking. Sharing the location of nearby vehicles could for example smoothen the traffic flow. Yang & Soloviev (2014) raises the issue of the reduction of excess information. In collaborative ranging and in localisation in general, the number of sensors and ranging information should be kept in an optimal level by selecting the best sources. Lohan et al. (2016) compare Bayesian approach to Dempster Schafer data fusion for WiFi fingerprinting finding that the Dempster Schafer method does not bring additional advantages.

Hejc et al. (2014) used WiFi with GPS receiver and INS. The transition from indoor environment to outdoors was more problematic, because the GPS needed time to achieve first fix. Thus it is necessary to identify these transition region characteristics between the technologies used. The observation of availability of a technology is the first phase. In practise scanning the WiFi access points for received signal strength values takes multiple seconds and as such is not practical for dynamic positioning.

2.5.2 Bluetooth

The Bluetooth low energy (BLE) on the other hand offers 20ms advertising interval, which makes it more suitable for more dynamic applications. The BLE advertising uses three narrow bands for beaconing preselected data. This can be turned on and off. In addition, ease in deployment of BLE anchors increases interest in using it for a range of positioning services.

Many portable devices use Bluetooth for communication purposes as defined by the https://www.bluetooth.org/en-us/specification/adopted—specifications (2017). Bluetooth smart ready adapters on mobile devices are able to record the received signal strength of the advertising anchors. The advertising anchors are independent and send the advertising messages unsynchronised independently. This creates a race condition between multiple beacons on the receiver. While a message from one beacon is received, the others are discarded. The distance conversion and fingerprinting of the RSSI operate faster than the corresponding WiFi equivalents. Palumbo et al. (2015) and Faragher & Harle (2015) examine the BLE fingerprinting, which is the most accurate positioning method setup within the scope of positioning using WiFi and BLE.

Bluegiga WT41-A evaluation kit is able to receive signal strength indicators from Bluetooth access points in the environment. It has a long line of sight range of up to 1000m. The transceiver sensitivity is -90dBm and transmit power can be 20dBm. Kuusniemi et al. (2011) used this as one sensor in a multi-sensor setup. Bluegiga AP 3201 and AP 3241 were used as access points.

2.5.3 Ultra-wideband

The ultra-wideband (UWB) characteristics offer advantages for coping with multipath. Especially the UWB impulse radio short pulses makes it easier to detect the multipath components. Repeatability is a strong advantage in the ultra-wideband approach. According to Meng et al. (2012) this means that the positioning result

stays consistent over a time period. Zampella, Angelis, Skog, Zachariah & Jimenez (2012) placed an UWB tag on shoe and helmet. The tag measurements on the shoe had more outliers due to non-line-of-sight conditions. Although high time resolution of UWB signals makes it easier to distinguish between original and multipath signals, the non-line of sight and obstruction conditions are still a challenge.

Alarifi et al. (2016) examine the ultra-wideband technology in detail. They list five main characteristics for a navigation system - accuracy/precision, coverage/resolution, latency, impact on infrastructure and random errors. SWOT (strengths, weaknesses, opportunities and threats) analysis shows the UWB very fit for indoor positioning. Issue is raised on the antenna design of the UWB antennas. Bras et al. (2012) also agrees the good sides using UWB being, the robustness to interference and jammers and high positioning accuracy due to large bandwidth. Again, antenna design challenge for UWB is raised as an issue. Designing an antenna for an UWB system is not as easy as for narrowband systems. Kok et al. (2015) used UWB receivers with inertial sensors. As a future study, they proposed researching positioning using multiple IMUs and UWB receivers for more accurate pose and navigation estimation.

Decawave UWB microchip based test kits, used and explained in Mikhaylov et al. (2016) and Peltola et al. (2016) offer an accurate indoor position solution (∼15cm). This accuracy is desirable and achievable with the UWB which makes it overpower the WiFi and Bluetooth alternatives. The reach of an anchor is approximately 30m indoors. The current draw and consumption of the Decawave UWB chip is around

100mA while for Kontakt BLE chip it is around 20mA. Although 5 times higher the accuracy in comparison makes this technology more useful. This of course depends on the application requirements and moreover, whether the application needs only proximity information rather than exact (∼15cm) positioning. The UWB real-time location system used by Zampella, Angelis, Skog, Zachariah & Jimenez (2012) was Ubisense.

Laaraiedh et al. (2013) compares fingerprinting methods with a model based method using UWB technology. These are less accurate than the timing based methods. The support vector machines, the K-nearest neighbour and the artificial neural networks are tested on the fingerprinted data. These are compared with a path loss model using maximum likelihood. The SVM was found the best among the position estimation from the fingerprint learning methods. Method using unique signal strength to distance conversion model for each separate anchor was found better than general model for all anchors.

2.5.4 Other

Wang, Bauer, Kirsch & Vossiek (2012) deployed ZigBee nodes. The article discusses the unreliability of RSS measurements because of the changing environment. The environment poses a challenge for the measurements if there are pedestrians walking or passing through the target area or furniture and reflections change places. In addition, the anchor battery level and the possible automatic gain control of the measuring device cause additional challenges.

Seco et al. (2010), Pateriya & Sharma (2011), Hasani, Peixeiro, Skrivervik & Perruisseau-Carrier (2015) explain the RFID systems consisting of RFID readers and transceivers or tags. In the active approach, the user carries the reader and scans the tags in the environment. The reach of an active RFID interrogator and transponder system is approximately 30 m. In passive approach, the user carries the tag and the environment has readers set up for positioning. The passive RFID detection range is very short (∼2m) and in practice a stand-alone passive system would be costly to set up. Privacy is of concern especially in passive RFID tag systems where the computation capability of the tag is not able to support necessary cryptographic data protection. RFID systems are implemented generally as proximity detection systems.

2.6 Magnetic sensing

Afzal et al. (2011) used Honeywell 3-axis magnetoresistive sensor HMC5843. The magnetic field strength vector data is sent through an i2c interface. The sensor has a measurement period of 10 ms and the data can be output at 116Hz rate. Jimenez et al. (2010) use the 3-axis magnetometer which is inbuilt in the integrated IMU sensor MTi Xsens. Fuzzy Inference System (FIZ) was designed by Jimenez et al. (2010). This used four magnetic field parameters to detect whether the magnetic field was disturbed inside a mall. The detector that read the magnetic field parameters used likelihood ratio testing. By following the magnitude and angle changes

of the magnetic field parameters it was possible to detect disturbances. A magnetic map can be constructed from the characteristics. This map can be used as a fingerprint map. Then while navigating the measurements are compared with recorded magnetic maps and a location estimation is derived.

The environmental magnetic anomalies and perturbations indoors, like metallic objects or radio devices, often make orientation determination difficult with magnetometers. Magnetic north measurements indoors do not work well because of this. This is the same inside a building or a vehicle where the changing magnetic fields disturb the measurements. Alternative way to take advantage of the magnetic field measurements is described by Zampella, Khider, Robertson & Jimenez (2012). Here a stable magnetic field during the stance phase was measured. If there was any angular rate detected during the stance it was used to correct the yaw drift and gyroscope bias. The magnetometer was also used to detect the swing period of the foot. The magnetic north measurement error update outdoors is helpful in keeping the yaw drift in control. Like in Hao et al. (2010) the magnetometer is most often integrated to adjust the heading.

2.7 Light

Cheap digital cameras are nowadays easy to find anywhere. These can be used for navigation in at least three separate ways. The camera images can be compared and matched with a database of crowdsourced images. The camera pose can be

resolved through analysis of multiple image features and by comparing feature descriptors of known objects. In addition, by using visual odometry, the location can be tracked from the image flow by comparing patterns in sequential images. A stereo vision setup can be applied for a more accurate camera movement estimation. Laser scanners offer even more high-resolution ranging data.

Using optical tracking technology, the demanding accuracy requirement (few centimeters) is achievable. This approach is though relatively clumsy and computationally expensive when considering applications for pedestrians and depends on the lighting conditions. The requirement of continuous camera recording on a mobile device is not the most attractive alternative. On the other hand, embedded augmented reality navigation information on smart glasses is already reality for example by using the Google glasses.

Santoso et al. (2017) discuss the visual inertial navigation technologies and methods with a special focus on microaerial vehicles. The issues common to visual and inertial navigation are considered. These include the brightness of the image, the technologies and methods used in acquiring the position solution, whether using single camera and structure from motion approach or stereo vision using multiple cameras. The time of flight cameras are mentioned, in which additional depth information is available. An example of such is the camera on the Google Tango device.

A Google Tango device was used by Laskar et al. (2016) to capture structure from motion images. Google uses area learning method for reconstructing the en-

vironment and in localising the camera. Laskar et al. (2016) managed to enhance this process to be more accurate by adding a database for structural recognition of the environment, aiding the spatial solution dislocation. CLIPS by Mautz (2012) is a similar system where laser spots are projected onto a wall to form a pattern that the camera recognises. It can then derive its position using this projected laser marker pattern.

A simpler method was used in Mulloni et al. (2009), where markers were placed around the conference venue. A mobile phone camera was used to detect unique markers and to solve the positions according to the acquired marker image. The artificial markers work well under different lighting conditions and are flexible for small local scale. Still, it requires preparatory work in mapping the marker locations to the area map application. For the detection to be successful, an element in the marker image on the camera image should be at least two pixels. The power consumption of these snapshot images is not too high either.

The database collection phase is called the offline phase. The actual navigation with the help of this collected database is called the online phase. There are two ways in setting up the navigation scheme. Either the camera can be carried by the user or a user feature or marker can be detected by a camera network that is set up around in the environment.

Generally, the first phase is the image acquisition in camera based positioning. The second step is segmentation and feature extraction. The third is the search for matches where features are compared with pre-existing information. This step

involves scale rotation and luminance analysis. The last step is to compute the camera location and pose. Methods are now examined for each step.

2.7.1 Image acquisition

Ruotsalainen et al. (2011) used a Nokia N8 mobile phone. It has a 12 Megapixel camera. Resolution of 1280x960 pixels was used. Solving the heading change took on average 1.1 seconds using Matlab on a desktop PC. Calculations that were conducted on images included contrast adjustment, canny edge detector and Hough transformation filtering.

Aufderheide & Krybus (2010) used a greyscale Unibrain Fire-i digital camera. The camera can output 30 frames per second. The resolution is 640x480 pixels. Hide et al. (2010) used a Canon IXUS 750 camera. Also, here the resolution was 640x480 pixels with 30 frames per second. A 3GHz desktop PC took 125ms to compute a frame pair. From this computation, the speed of the pedestrian could be derived. The camera recorded the ground while the pedestrian was walking.

In Wang, Virrantaus, Pei, Chen & Chen (2012) a regular 3D model map was compared with a 3D model map using geocoded images. The image based approach occupied 53 % of the processor execution time and could run 2.5 frames per second and was faster than the regular 3D model. The frame rate of 2.5 is much less than the rate that the camera can output frames. The platform was a Nokia 6710 handset with 15 fps camera.

Honorato et al. (2008) discusses the Devantechs TPA81 Thermopile array. The

array consists of 8 sensors each having a 6 degrees field of view. Thermopile response time is approximately 20 to 50ms. In Yoshiike et al. (1999) a pyroelectric infrared sensor (PIR) array was fabricated to detect occupants inside a room. The sensor rotated 360 degrees with a refresh rate of 20Hz. Kivimäki et al. (2014) lists infrared sensor technologies. Microbolometer and Golay cell based IR cameras are very expensive and are not applicable for pedestrian navigation. Thermopiles and pyroelectric sensors, although less accurate, are very affordable. These could be preferred in low lighting conditions where conventional image processing is impossible.

2.7.2 Feature extraction

SIFT (or SURF), Canny edge detector, FAST or Harris corner detector are typical examples of feature detection algorithms. In Frontoni & Zingaretti (2006) the colour and spatial information of an object was recognized in the image. The extracted features can be compared either with natural estimated features or with created synthetic patterns in the environment. In Yoshiike et al. (1999), thermal images were acquired. The hot spots above a threshold that were detected were interpreted as persons in the room. In Aufderheide & Krybus (2010), a general feature detection processor is described. SIFT, SURF and corner detection algorithms are used for feature detection. In Mulloni et al. (2009) the physical marker image is detected on the captured image. Likewise, in the CLIPS system by Mautz (2012), the laser projected pattern is recognised. In Hide et al. (2010) FAST corner detection method was used. Ruotsalainen et al. (2010) lists detection of vanishing points, epipolar

lines and hue histogram invariance as alternatives. As a summarising statement, the feature extraction is a pattern recognition or a classification problem.

2.7.3 Match search

Bundle adjustment, RANSAC, GOODSaC and BaySaC are algorithms that can be used in the refinement and the outlier rejection of the matched features. Frontoni & Zingaretti (2006) uses colour sets and polar weighted walkthrough representation to match the images. PCA and LDA metrics are mentioned. Yoshiike et al. (1999) has four stages in localising a human from the captured thermal image. From the measurement image, the information goes through a binary filter where each pixel is either above or below a threshold. Dust treatment removes separate pixels because the human body is considered as a wider area of pixels in the image. Aufderheide & Krybus (2010) discusses area based approaches on successive images. Intensity patches are compared around the detected features' positions. The hungarian method was used in the bipartite graph matching of 3D correspondences. In Mulloni et al. (2009) the captured image of the marker is compared with a database containing the location data where the marker was placed. Hide et al. (2010) implemented sum of squares differences (SSD) between pixel values on successive images. Kd-tree is mentioned as a more efficient approach instead of SSD. Nearest neighbour search is made afterwards to search the closest match within the kd-tree. BaySAC that is a RANSAC derivative, is used for outlier rejection. Homogeneity of the corresponding points are calculated using discrete linear transform (DLT).

Ruotsalainen et al. (2010) raises up the troubles in image matching. These are the differences in image lighting conditions, orientation and scale.

2.7.4 Camera calibration

In Frontoni & Zingaretti (2006) the location is updated using Markov's belief distribution using measurements. The location and pose are determined by comparing the captured images with the database snapshots collected during the offline stage. Yoshiike et al. (1999) labels the pixel groups that are filtered from the thermal images and represents the group as a tracked person. Aufderheide & Krybus (2010) divides the camera calibration into motion and structure recovery. Five point matches in the consecutive images is the minimum necessity for calculation. In order to minimise the computational load not all of the frames are computed. The key frames are selected where the corresponding point amount is still sufficient in consecutive images. Triangulation is then used to derive the new location of the tracked object.

In Mulloni et al. (2009), the pose of the camera was calculated using a database of known markers. Hide et al. (2010) computes a homography between the successive images and furthermore uses Levenberg-Marquardts algorithm to derive the rotation and translation. The vision motion information was integrated with the IMU and GPS measurement data. The camera measurements were integrated into the INS error state Kalman filter. Ruotsalainen et al. (2010) describe the location calculation by deriving the change within the vanishing point information.

2.8 Comparison and Sensor Technology Selection

An extensive literature review was conducted on indoor sensor technologies and methods. The main characteristic features for positioning systems were compared against the wide range of sensor technology-method combinations. The positioning system characteristics included the following.

- Physical property the sensor measures
- Positioning method type
- Technology role, either absolute or relative positioning
- Data type, describing the format of the data
- Data rate, for how fast the data is output
- Accuracy, can be either absolute or relative
- Reach, for how far from an anchor positioning is possible
- Energy consumption and computational load to estimate the efficiency
- User device cost, for estimating cost on the user device
- Infrastructure cost, is the cost for the entire system to function anchorwise

For each positioning system characteristic, the typical value or value range of the sensor technology-method combination is depicted in the following four tables 2.1, 2.2, 2.3 and 2.4. The references for the extracted tables are marked with the associated number among the references in the back of the thesis (marked as **LR-referendeID**).

Table 2.1: The tactile, inertial and compressible media navigation technologies comparison table.

Physical property	Tactile			Inertial		Compressible media	
Positioning method	tactile device on user	tactile environment	odometer	inference	estimation	sound	barometer
Technology role	relative	absolute	relative	relative	relative	absolute	relative
Data type	binary/ force level	binary/ force level	counter	reading classification	dead-reckoning	timing/ synchron./ signal pattern	difference
Data rate	50, 100, 500 Hz	22, 60 Hz	4 pulse/ rotation	100 Hz	2, 10, 40, 50, 100, 250, 400 Hz	1, 3, tens of Hz	2 Hz
Accuracy	-	4cm – 40cm/ 28.3cm	-	~3%/ 60m per km	4% & 16% (with map)/ 5%(ZigBee)/ 90% success inclination detection/ 4,62m in 3km/ 2%	1cm-1m/ 9cm/ 10cm/ 5cm/ 20cm	33cm/ 0.2m
Reach		implemented structure	dead reckoning	dead reckoning	dead reckoning	~3 – 10m/ 10m	
Energy consumption/ Computational load	150mV per control element	low	low	low	~350mW Microstrain & MTi Xsens/ ~200mW Kistler 8395a / low for separate components	>20mW (Devantech)/ 100mW for HC-SR04	5mW
User device cost	-	low	low	Motorola Droid/ Crossbow IMU 300CC	Microstrain 3DM, MTI-G Xsens ~1000£/ phone sensors ~5£	~few hundred £ (Neumann microphone)/ Devantech US module(10£)	£10.00
Infrastructure cost	-	Expensive/ 100£ per 3x2m mat/ 80€ per 3x0.5m mat	-	low	low	Devantech sonar module (10£)/ 3D-Locus 100£ per node	£10.00
References	81, 82, 83	18	84, 108	72, 78	1, 36, 38, 41, 42, 53, 56, 58, 60, 62, 65, 69, 71, 80, 85, 92, 98	1, 16, 17, 35, 39, 66, 86, 87, 99	35,53,89

The tactile devices on user can be touch sensitive elements to indicate for example zero updates for the inertial mechanisation. The environmental tactile devices are for example mats with smart sensors measuring the user location. The odometers are familiar from cars and bicycles that count the travelled distance. Inertial dead-reckoning is usually implemented by using double integration of the gyro and accelerometer outputs. An alternative method is to map the sensor reading into next position by using classification and learning methods related to each individual location. The sound and ultrasound localisation is accurate but suffers from multipath. A barometer as a relative measure can help in determining the current height.

Table 2.2: The broadcast RF navigation technologies comparison table.

Physical property	Broadcast RF		
Positioning method	GNSS	Pseudolites	Wide area network
Technology role	absolute	absolute	absolute
Data type	synchronisation	synchronisation	synchronisation/ fingerprinting/ timing
Data rate	~1 Hz	~1 Hz	~1 Hz
Accuracy	4m – 7m	2cm – 3m/ ~7m (pulsed pseudolite)	lambda ~1m/ 4-25m/ 3m (FM)/ 10 – 23m/ >100m
Reach	outdoor/ satellite availability	~tens of kms (50km)	~kms
Energy consumption/ Computational load	~500mW/ >150mW/ 1.5W receiver consumption	~1W transmit power	~1W (TI OMAP)
User device cost	ublox LEA5H ~50£	Locata receiver ~5000£/ IFEN NavX	>10£ (OMAP)
Infrastructure cost	>expensive	>expensive ~100000£/transmitter (CNET.com/news)	expensive
References	1	1, 40, 46, 4	1, 8, 37

The global navigation satellite systems emit their specific signals. GPS, GLONASS, Galileo and Beidou satellite signals are different and thus the receiver needs to have the ability to synchronise with each using different methods. The pseudolite signals imitate the satellite signals but the pseudolite anchors are located on the ground. The Locata is perhaps the most advanced pseudolite alike device that has been developed. Wide area network opportunistic positioning is challenging since these signals have not been designed to be used for positioning. These are the radio and the television signals.

The received signal strength fingerprinting is perhaps the most trialled method for acquiring position indoors. Time resolution for WiFi protocol is not yet fast enough for enabling timing approaches to work as well as the fingerprinting. The angle of arrival method needs sophisticated antenna design on the receiver side. The Bluetooth low energy anchors are easily deployable. The advertising frequency is also much higher than with the WiFi. Similar methods work faster when using BLE. For Ultra-wideband transceivers, the timing methods have been proven successful. It is perhaps the most feasible technology for line of sight indoor positioning. RFID methods demand more setting up effort compared to the UWB technologies.

Table 2.3: The local area broadcast RF navigation technologies comparison table.

Physical property	Local RF						
Positioning method	WiFi RSS	WiFi ToF/ AoA	BLE RSS	UWB ToF	UWB RSS	RFID active user	RFID passive user
Technology role	absolute	absolute	absolute	absolute	absolute	absolute	absolute
Data type	finger-printing/ range conversion	range/ angle	finger-printing/ range conversion	range	finger-printing/ range conversion	pattern	proximity
Data rate	0.2, 1, 3 Hz	1, 2, 10 Hz	0.2, 1, 2, 30 Hz	0.75, 10, 25 Hz		0.2, 0.5 Hz	20, 80 Hz
Accuracy	2 – 4m/ 1.53m	1.7 – 10m	~2 – 5m/ 1.88m	15cm/ 50cm/ 2cm/ 20cm/ 30cm/ 1 – 7cm	2.5m	~1-3m/ 2m	~15 – 50cm
Reach	~45m/ 10m/ 25 - 50m	~25m	Modifiable 1-25m/ 150m open field	~5m/ 50-100m/ 175m (sounder Zetik)/ 30m(in tunnel/ 72m with los)	-	30 – 100m	~2m
Energy consumption/ Computational load	>1W/ 700mW WSN802GX/ >500mW transmit and 200mW for receiver	>1W/ 100mW	<50mW/ 30mW nRF51882/ 25mW	>1W/ 500mW transceiver/ 300mW receiver 600mW transmitter	-	~250mW	~200mW reader/ <50mW tag/ 300mW reader
User device cost	HP Ipaq 77£	>5£	~5£ BT receiver	expensive laboratory equipment/ 60£ for ubisense tag IP63 slim	-	~300£ I-Card III interrogator/ M220 >500£ reader	>10£ tag
Infrastructure cost	>50£ (AP Prices)/ 20£	>50£ (AP Prices)	~10£/beacon, 5£	expensive laboratory equipment	-	>10£ tag	~200£/ >1000£ reader
References	1, 7, 9, 10, 19, 20, 21, 22, 57, 64, 91, 92, 114	1, 19, 22, 76, 77, 9	1, 2, 3, 4, 5, 6, 23, 24, 25, 26, 27, 28, 35, 100, 11	1, 11, 12, 29, 30, 31, 32, 36, 68, 6	35,74	1, 15	1, 13, 14

Table 2.4: The magnetic and camera navigation technologies comparison table.

Physical property	Magnetic		Light		IR-light		
Positioning method	magneto-meter	electromagn. transmitter-receiver	image marker	image feature	image marker	image feature	PMD/ToF
Technology role	relative	absolute	relative when marker in environment/ absolute when marker on user	absolute with database/ odometer relative	relative when marker in environment/ absolute when marker on user	absolute with database/ odometer relative	relative
Data type	angle/ fingerprinting	Nfer-ranging/ Complex Image Theory ranging	pattern detection	pattern detection	pattern detection	pattern detection	ranges
Data rate	5, 75 Hz	1 Hz	5, 15, 30 Hz	0.1, 4, 30 Hz	50 Hz	20 Hz	
Accuracy	1mm for permanent magnet/ 20cm for fingerpr.	1% of range/ 5cm (resonance coupling)/ 21cm/10cm	1mm/ 30cm	~10cm/1% drift for odometer/ 20cm/ 50cm/ 15cm	6m(Active Badges CoO)/ 10cm	0.2 – 0.8m	<decimeter
Reach	1m/ magnetic fingerprint map	~ 50m/ 2m	~6m/ scalable and resolution dependent	~6m/ scalable and resolution dependent	(tag placement)/ ~6m	~6m/ 10m	0.1 – 4m
Energy consumption/ Computational load	<50mW	>1W	>300mW/ ~2W/ snapshot 700mW	<50mW for odometer and >200mW/ >1W for cameras/ 684mW	<50mW for markers and 3,3V*50mA= 165mW for camera(OV9625) + processing consumption	<50mW (thermopile)	300mW
User device cost	>2£/ 10£	>1000£	~10£ for camera/ 500£ for CLIPS	~1£ for odometer/ 10£ and 100£ for camera modules	~1£/marker and ~10£/camera	~1£/ thermopile	-
Infrastructure cost	>2£*n	>1000£/ 16€ per mm2 (tiletrack)/ Xsens Mti	>10£ for marker amount	~10 – 100 £ per camera	~1£/marker and ~10£/camera	~1£ per thermopile/ 8000€ microbolometer camera	-
References	1, 49, 53, 63, 94, 96	1, 39, 93, 95	1, 33, 34	1, 33, 34, 35, 43, 44, 45, 51, 61, 70, 79	1	1, 39, 101	117

Positioning exploiting the magnetic fingerprinting method is relatively accurate within a designed room sized space with a known magnetic map. For pedestrian purposes, the most useful way to use magnetometers is to measure the north while outdoors and when no disturbing nearby magnetic fields are present. The camera based stereo vision or structure from motion methods need computational power and line of sight for the user camera. The infrared light methods could be used in the dark for specific applications. The time of flight cameras offer depth information in addition to the 2D image information.

2.8.1 Analytic Hierarchy Process for Technology Prioritisation

The AHP, analytic hierarchy process was selected for technology comparison and evaluation. Navigation application design is subjective to the psychology and needs of the people as well as to the exact mathematics of the different navigation technologies. The AHP method combines the two, the hard figures with the subjective expert opinions. It is a structured technique for organising complex decisions. The overall goal is divided into subsets. These subsets are simple enough for a person to be able to evaluate. They can be subjective or defined simply by mathematics. The technique is explained as the technologies are being evaluated in the following pages according to Ye et al. (2008) and Basiri et al. (2015).

Pairwise Comparison Matrix

The first step in the process is to acquire the pairwise comparison matrix for the selected criteria. Then the the criteria is chosen by keeping in mind the research goals which are for seamless pedestrian navigation. The following list shows the chosen criteria; data rate, accuracy, reach, energy consumption, user device cost and infrastructure cost. The pairwise importance matrix is shown in table 2.5 below. Here the ratios show the importance given pairwise for each pair. For example, accuracy is taken as 5 times more important than energy consumption.

Table 2.5: Pairwise comparison matrix of the selected criteria.

	data rate	accuracy	reach	energy consumption	user device cost	infrastructure cost
data rate	1	1/9	1/5	1/3	1	1/5
accuracy	9	1	3	5	5	3
reach	5	1/3	1	3	5	3
energy consumption	3	1/5	1/3	1	3	3
user device cost	1	1/5	1/5	1/3	1	1
infrastructure cost	5	1/3	1/3	1/3	1	1

The columns are then normalised, after which the average of each row is taken. These average values are now the weights for each criterion. They are shown in table 2.6.

Table 2.6: Weights for the criteria. Indicates how much importance is given to each criteria.

data rate	0.04
accuracy	0.42
reach	0.24
energy consumption	0.14
user device cost	0.06
infrastructure cost	0.10

Consistency Index

For checking whether the subjective pairwise evaluation was consistent we can derive a consistency index and ratio which indicate this. First, the weighted sum of the two matrices in tables 2.5 and 2.6 is derived.

Then a consistency vector is formulated by taking a dot product between the weighted sum matrix and the reciprocal of the weight matrix. The average, which is the eigenvalue of the consistency vector is then denoted as the lambda λ. The Consistency Index (CI) can then be derived by the following formula and is in our study 0.097. The Consistency ratio is the consistency index divided by 1.25 which is called a random index. This divider is a random index with which the acquired consistency index is compared. This particular value is for the amount of six different criteria and is taken from a look-up table in the AHP literature.

$$\begin{gathered} CI = \frac{(\lambda - n)}{(n-1)} \\ CI = 0.097 \\ CR = CI/1.25 = 0.078 \end{gathered} \tag{2.1}$$

If the consistency ratio is below 0,1 as it is, we can say we have been consistent in our pairwise evaluation. If it was not so, the pairwise subjective evaluation step should be retaken.

Pairwise technology comparison matrices for each criterion

Then next step is to repeat the pairwise comparison and consistency check between each technology. First we estimate the data rate between importances on each technology and then accuracy and so forth. These subjective evaluations are shown in the tables 2.7, 2.8, 2.9, 2.10, 2.11 and 2.12.

Table 2.7: Data rate.

	tactile environ-ment	sound	GNSS and Pseudolites	WAN	WiFi	BLE	UWB	RFID	User cameras	Network cameras
tactile environ-ment	1,00	1,00	5,00	5,00	3,00	0,33	3,00	2,00	3,00	1,00
sound	1,00	1,00	5,00	4,00	3,00	0,33	2,00	3,00	5,00	2,00
GNSS and Pseudo-lites	0,20	0,20	1,00	1,00	0,50	0,14	0,33	0,50	0,50	0,20
WAN	0,20	0,25	1,00	1,00	0,50	0,14	0,33	0,50	0,50	0,20
WiFi	0,33	0,33	2,00	2,00	1,00	0,33	0,50	1,00	1,00	0,33
BLE	3,00	3,00	7,00	7,00	3,00	1,00	3,00	5,00	4,00	3,00
UWB	0,33	0,50	3,00	3,00	2,00	0,33	1,00	2,00	1,00	0,50
RFID	0,50	0,33	2,00	2,00	1,00	0,20	0,50	1,00	0,50	0,33
User cameras*	0,33	0,20	2,00	2,00	1,00	0,25	1,00	2,00	1,00	0,25
Network cameras	1,00	0,50	5,00	5,00	3,00	0,33	2,00	3,00	4,00	1,00

Table 2.8: Accuracy.

	tactile environ-ment	sound	GNSS and Pseudo-lites	WAN	WiFi	BLE	UWB	RFID	User cameras	Network cameras
tactile environ-ment	1,00	1,00	5,00	5,00	4,00	4,00	1,00	3,00	1,00	0,50
sound	1,00	1,00	5,00	5,00	4,00	3,00	1,00	2,00	0,50	0,33
GNSS and Pseudo-lites	0,20	0,20	1,00	1,00	0,50	0,33	0,14	0,25	0,14	0,11
WAN	0,20	0,20	1,00	1,00	0,50	0,33	0,17	0,25	0,14	0,11
WiFi	0,25	0,25	2,00	2,00	1,00	0,50	0,14	0,20	0,17	0,14
BLE	0,25	0,33	3,00	3,00	2,00	1,00	0,20	0,33	0,20	0,17
UWB	1,00	1,00	7,00	6,00	7,00	5,00	1,00	4,00	0,50	0,33
RFID	0,33	0,50	4,00	4,00	5,00	3,00	0,25	1,00	0,20	0,14
User cameras*	1,00	2,00	7,00	7,00	6,00	5,00	2,00	5,00	1,00	0,50
Network cameras	2,00	3,00	9,00	9,00	7,00	6,00	3,00	7,00	2,00	1,00

Table 2.9: Reach.

	tactile environment	sound	GNSS and Pseudolites	WAN	WiFi	BLE	UWB	RFID	User cameras	Network cameras
tactile environment	1,00	0,33	1,00	0,33	0,20	0,25	0,33	1,00	0,20	0,50
sound	3,00	1,00	1,00	0,33	0,33	0,50	0,50	2,00	0,33	1,00
GNSS and Pseudolites	1,00	1,00	1,00	0,50	0,25	0,33	0,33	0,50	0,25	0,50
WAN	3,00	3,00	2,00	1,00	0,33	0,33	0,50	1,00	0,50	2,00
WiFi	5,00	3,00	4,00	3,00	1,00	1,00	2,00	4,00	1,00	2,00
BLE	4,00	2,00	3,00	3,00	1,00	1,00	1,00	3,00	1,00	2,00
UWB	3,00	2,00	3,00	2,00	0,50	1,00	1,00	3,00	0,50	1,00
RFID	1,00	0,50	2,00	1,00	0,25	0,33	0,33	1,00	0,33	0,50
User cameras*	5,00	3,00	4,00	2,00	1,00	1,00	2,00	3,00	1,00	2,00
Network cameras	2,00	1,00	2,00	0,50	0,50	0,50	1,00	2,00	0,50	1,00

Table 2.10: Energy efficiency.

	tactile environment	sound	GNSS and Pseudolites	WAN	WiFi	BLE	UWB	RFID	User cameras	Network cameras
tactile environment	1,00	1,00	5,00	2,00	1,00	0,33	1,00	2,00	6,00	1,00
sound	1,00	1,00	3,00	3,00	2,00	1,00	2,00	3,00	5,00	1,00
GNSS and Pseudolites	0,20	0,33	1,00	1,00	1,00	0,20	0,33	0,50	2,00	0,33
WAN	0,50	0,33	1,00	1,00	1,00	0,25	0,50	1,00	2,00	0,33
WiFi	1,00	0,50	1,00	1,00	1,00	0,33	0,50	1,00	3,00	0,33
BLE	3,00	1,00	5,00	4,00	3,00	1,00	2,00	3,00	4,00	1,00
UWB	1,00	0,50	3,00	2,00	2,00	0,50	1,00	2,00	3,00	0,50
RFID	0,50	0,33	2,00	1,00	1,00	0,33	0,50	1,00	2,00	0,33
User cameras*	0,17	0,20	0,50	0,50	0,33	0,25	0,33	0,50	1,00	0,25
Network cameras	1,00	1,00	3,00	3,00	3,00	1,00	2,00	3,00	4,00	1,00

Our subjective evaluation of technologies concentrates on a buildingwise solution. Data rate comparison concentrates on the data rate of the user device. Similarly, the accuracy is thought to be the accuracy capable of achieving indoors rather than outdoors since this is the key problem to solve.

Table 2.11: Device cost.

	tactile environment	sound	GNSS and Pseudolites	WAN	WiFi	BLE	UWB	RFID	User cameras	Network cameras
tactile environment	1,00	1,00	3,00	1,00	1,00	0,50	2,00	3,00	2,00	1,00
sound	1,00	1,00	3,00	3,00	1,00	1,00	2,00	4,00	3,00	1,00
GNSS and Pseudolites	0,33	0,33	1,00	1,00	0,33	0,20	0,25	0,50	1,00	0,25
WAN	1,00	0,33	1,00	1,00	1,00	0,20	0,50	3,00	3,00	0,50
WiFi	1,00	1,00	3,00	1,00	1,00	0,33	2,00	3,00	2,00	1,00
BLE	2,00	1,00	5,00	5,00	3,00	1,00	2,00	5,00	5,00	2,00
UWB	0,50	0,50	4,00	2,00	0,50	0,50	1,00	3,00	3,00	0,50
RFID	0,33	0,25	2,00	0,33	0,33	0,20	0,33	1,00	1,00	0,33
User cameras*	0,50	0,33	1,00	0,33	0,50	0,20	0,33	1,00	1,00	0,25
Network cameras	1,00	1,00	4,00	2,00	1,00	0,50	2,00	3,00	4,00	1,00

Table 2.12: Infrastructure cost.

	tactile environment	sound	GNSS and Pseudolites	WAN	WiFi	BLE	UWB	RFID	User cameras	Network cameras
tactile environment	1,00	0,20	0,50	0,11	0,14	0,17	0,25	0,50	0,13	1,00
sound	5,00	1,00	6,00	1,00	0,33	0,50	1,00	3,00	0,25	5,00
GNSS and Pseudolites	2,00	0,17	1,00	0,33	0,17	0,20	0,25	1,00	0,14	1,00
WAN	9,00	1,00	3,00	1,00	2,00	2,00	3,00	5,00	1,00	7,00
WiFi	7,00	3,00	6,00	0,50	1,00	1,00	2,00	4,00	0,33	2,00
BLE	6,00	2,00	5,00	0,50	1,00	1,00	2,00	4,00	0,50	3,00
UWB	4,00	1,00	4,00	0,33	0,50	0,50	1,00	2,00	0,33	2,00
RFID	2,00	0,33	1,00	0,20	0,25	0,25	0,50	1,00	0,25	1,00
User cameras*	8,00	4,00	7,00	1,00	3,00	2,00	3,00	4,00	1,00	6,00
Network cameras	1,00	0,20	1,00	0,14	0,50	0,33	0,50	1,00	0,17	1,00

Reach is not only how far a user device is capable of getting ranging information from but also current existing availability and the ease in deployability. Efficiency is plainly thought on the user side. How efficient the end user product could be. Likewise, the device cost only concentrates on the device itself. For example, if only communication link on the phone is necessary to acquire solution from the

environmental camera positioning system, then in this case the user device side is very cheap. Infrastructure cost depicts then how expensive it would be to set up a system on a normal commercial building block.

The preliminary elimination of few approaches was executed in the selection of the technologies for the comparison process. The first obvious elimination was all of the relative positioning methods since it is more feasible to examine the absolute positioning systems first. Relative positioning systems will be considered afterwards as the inference is easier after we know which of the technologies can provide absolute positioning reasonably well. An absolute position for indoor is emphasized in our evaluation since this is the decisive factor for the whole hierarchic process study for achieving seamless positioning.

AHP Result

Table 2.13 lists the importance for each absolute technology method that was selected for the comparison.

Table 2.13: Result of the AHP analysis. Higher is better. Total sum is 1.

tactile environment	sound	GNSS and Pseudolites	WAN	WiFi	BLE	UWB	RFID	User cameras	Network cameras
0.09	0.11	0.03	0.06	0.09	0.12	0.12	0.06	0.15	0.17

When ordered the list looks like following

1. Network cameras
2. User cameras

3. BLE and UWB
4. Sound
5. WiFi and Tactile
6. WAN and RFID
7. GNSS and Pseudolites

2.8.2 Discussion of the results

The surveillance cameras are already around cities and especially within commercial centres. Other reasons why network cameras came first in the evaluation were the necessity of only a data link in the position processing. This study did not take into account the difficulty in identifying the correct person within the surveillance cameras in order to offer an individual the correct localisation information. But the accuracy of the camera systems is undoubtedly very high, although there are challenges remaining with the luminance and the multi-user identification.

Usability and privacy were not taken into account in this study. If the user has to pinpoint their camera while walking, the usability of the navigation application would not likely reach the acceptance level of a wider audience. Camera on the user also demands battery power when processing through image matching or structure from motion steps. Still again the accuracy is good. Accuracy along with the snapshot approach for snapshot positioning made the camera on the user to come second.

Both the Bluetooth low energy and Ultra-wideband technologies scored similarly. Bluetooth is very fast and easily deployable while UWB is very accurate. Compared to cameras, both technologies can function without line of sight although multipath and indoor environment makes the position derivation challenging. Fingerprinting locations partly compensates this for the BLE. However, changes in environment like moving furniture will require recording the required radiomaps again. Ultra-wideband ranging reaches accuracy of few decimetres when line of sight conditions apply. Usability of both of these RF techniques is much better compared to cameras. There is no need for the user to do extra adjustments for acquiring their position.

Sound based methods scored close to the BLE and UWB. Reach and noise resistance are slightly worse.

WiFi is very slow and the tactile environment will be very expensive which makes these two not to fare well. Even more so for WAN and RFID. The wide area network signals were not designed for positioning and are thus very inaccurate. RFID security issues and expensive readers and tags make these two the second last. The pseudolites are expensive. The GNSS signal carrier frequencies tend to attenuate in urban areas and especially indoors. Only more expensive high sensitive receivers are able to operate indoors.

On relative positioning technologies and conclusions

Inertial sensors are often used in assisting a navigation system. For cars, if a GNSS signal gap occurs the inertial system can follow the car movements by dead reck-

oning. Barometer is small and offers an additional estimate of the height change. Likewise, magnetometers can indicate the north and aid with heading error. Tactile event switches can offer constraints.

According to what the study indicates the following technologies are chosen.

- UWB (for accuracy)
- BLE (for deployability)

These technologies for indoor positioning offer absolute positioning information. In addition,

- GNSS

for outdoor absolute positioning.

- Foot-mounted IMU

was chosen as an additional relative navigation aid. These agree with the study and also with the usability requirements and are now stated here as the most promising alternatives in order to achieve the goal of seamless navigation. The rest of the thesis will focus on these technologies and seamless navigation trials conducted using them.

Chapter 3

Methods for Seamless Navigation Systems

The plethora of sensors offer a wide range of raw data to be processed. Unlike the raw data, the location information is at a higher level of abstraction. The raw data needs to be processed using appropriate methods to get the best out of it. In this chapter, the focus is on acquiring position and context information out of the raw data. The focus is on the methods for the selected technologies of the previous chapter. The comparison and the selection of the most fit methods follows, after an overview of the methods for each technology is presented. The context derivation from the raw data is examined in the end of the chapter.

3.1 Inertial Dead Reckoning

The Inertial Measurement Units include the accelerometer and the gyroscope sensors. The accelerometers measure the current acceleration of the sensor body axis. The reading of the current acceleration is shown for each axis of the 3-dimensional space.

The gyroscopes sense the rotation of the body around the three axes. Current angular speed reading is given for each axis.

We look first into the coordinate frames used in this study. The mechanisation is the core component of this chapter that depicts the process of dead reckoning with associated formulas. The initial value derivation is essential for the mechanisation process to succeed. As well, the errors need to be tracked. A compensation of these errors have to be done either by using zero updates or by other means. Alternative ways for dead reckoning are shortly examined before making conclusions in order to proceed.

3.1.1 Coordinate Systems and Quaternion Representation

Four common coordinate systems are introduced. The definitions according to Groves (2013*a*), Hide (2003), Rahim (2012), www.navipedia.net (Accessed 20.9.2017) for the Inertial, Earth, Local Tangent-Plane and Body coordinate systems follow.

In an *Inertial coordinate system*, the origin is at the Earth's centre of mass. According to the conventional celestial reference system like the J2000, the z-axis

points towards the earth mean rotation pole. The z-axis also points towards the north. The x-axis is the direction from earth to sun at mean vernal equinox at exact time epoch defined in the J2000. The y-axis completes the right-hand coordinate system. The coordinate system does not rotate with the earth.

Similarly, in the *Earth frame earth centered coordinate system*, the origin is at the Earth's centre of mass and z-axis points towards the north. According to the ECEF coordinate system WGS84, the z-axis points towards the IERS Reference Pole, which is also the Conventional Terrestial Pole (CTP). The x-axis is fixed to point towards the intersection of equator and IERS Reference Meridian (IRM) that is the 0-degree meridian. The y-axis points towards the intersection of the equator and 90-degree east meridian.

Local-Tangent Plane coordinate system origin is a selected location with a distinctive feature within the environment on the earth's surface. The z-axis points down towards the centre of Earth according to the normal to the ellipsoidal model. The x-axis points to the north and the y-axis to east.

Body Frame coordinate system origin is a point on the object or user that is tracked. The x-axis points usually towards the direction of movement. The y-axis points to the right and z-axis down.

The quaternions represent the orientation of the tracked object w.r.t. the reference frame. A quaternion by itself can be thought of to be depicting a rotation from the zero-rotation state. It is a hyper-complex number with four components. The pure imaginary vector part consists of three numbers that define the rotation axis.

The scalar part is always real and defines, in part, the amount of rotation around the rotation axis. Figure 3.1 depicts quaternion rotation.

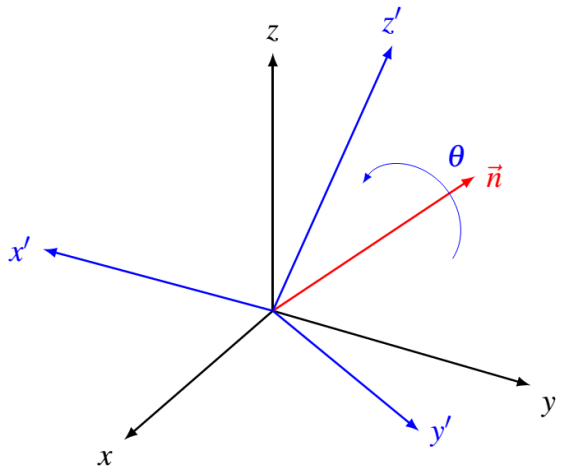


Figure 3.1: Example of rotating object body coordinates in the local tangent plane coordinate system orientation using a quaternion. Rotation axis is marked with n and θ shows the amount of rotation.

3.1.2 Mechanisation

The mechanisation is the process of acquiring the orientation, the velocity and the position changes using the gyro and accelerometer values. An estimate of the position, velocity and orientation of the object are the required initial information. If the Inertial Measurement Unit (IMU) sampling rate is 100Hz we get 100 accelerometer and gyro measurements per second.

Attitude

Mechanisation starts with the gyroscope readings. A prior attitude is used as the input for the new attitude determination process epoch. The orientation of the

object body is then updated using the latest gyro measurement. The angular rate reading for each axis shows how the object has turned during the epoch.

The rotation matrix is another tool for representing object rotation. A quaternion can be converted into rotation matrix and vice versa. The small angle approximation can be used with the rotation that is defined by the gyro measurement if the sampling rate is high enough. Then the new rotation matrix at the end of the attitude update can be presented by the following equation (i is inertial and b is body frame).

$$C_{new} = (I_3 + \delta t \Omega^b_{ib}) C_{old} \tag{3.1}$$

Here C is the rotation matrix and Ω is the skew-symmetric matrix of the angular rate vector as described by Groves (2013*a*). Moreover, the angular rate is considered constant over the sampling period. This is also the first order approximation of the true angular motion change. Using the higher order terms for the more precise derivation of the rotation change, increases the computational load and complexity. Equation 3.2 is the power series representation of the rotation derivation.

$$C_{new} = \sum_{r=0}^{\infty} \frac{[\alpha^b_{ib}\wedge]^r}{r!} C_{old} \tag{3.2}$$

Here $\alpha = \delta t \Omega$, is the angle change for the current epoch. The so-called exact solution can be used instead of the first order approximation. Equation 3.3 shows this in power series form according to Groves (2013*a*).

$$C_{new} = (I_3 + \left(\sum_{r=0}^{\infty} (-1)^r \frac{\left|\alpha_{ib}^b\right|^{2r}}{(2r+1)!} \right) [\alpha_{ib}^b \wedge] + \left(\sum_{r=0}^{\infty} (-1)^r \frac{\left|\alpha_{ib}^b\right|^{2r}}{(2r+2)!} \right) [\alpha_{ib}^b \wedge]^2) C_{old} \quad (3.3)$$

Here r is the order. The Rodrigues' formula, mentioned by Groves (2013*a*), can be used to take advantage of sine and cosine, but the higher order the approximation is the more computation is required.

The above equations are for the body to inertial coordinate system transformations. To transform to the Earth fixed reference coordinates, earth rotation needs to be taken into account. The equation 3.4 includes the earth rotation rate Ω_{ie}^e. This rotation rate has a very small value in comparison and has a very small effect (i is inertial, e is earth and b is body frame).

$$C_{new} = (I_3 + \delta t \Omega_{ib}^b) C_{old} - (\delta t \Omega_{ie}^e) C_{old} \quad (3.4)$$

When transforming into local tangent plane coordinates, transport rate is an additional characteristics to be taken care of. Transport rate is the turn rate of the local tangent plane centre with respect to the Earth. This is depends on the latitude. The equation 3.5 includes the transport rate Ω_{en}^n (i is inertial, e is earth, n is navigation and b is body frame).

$$C_{new} = (I_3 + \delta t \Omega_{ib}^b) C_{old} - (\delta t (\Omega_{ie}^n + \Omega_{en}^n)) C_{old} \quad (3.5)$$

An alternative way is presented by both Rahim (2012) and Hide (2003). Here the Earth turn rate and local navigation coordinate system transfer rate is first extracted using the reverse conversion of the previous attitude matrix. The turn rates are thus converted from the Earth fixed coordinates to the sensor body coordinates for subtraction from the gyro measurement. Equation 3.6 shows this. ω is the angular velocity.

$$\omega_{nb}^{b} = \omega_{ib}^{b} - C_{n}^{b}(\omega_{ie}^{n} + \omega_{en}^{n}) \tag{3.6}$$

The attitude is then updated using this compensated turn rate.

Velocity

After the object attitude change has been applied the acceleration measurement can now be integrated into velocity, if the initial velocity is known. First the acceleration measurement needs to be converted from the body coordinates to the coordinate system where the solution is resolved. As the measurement is an average of the whole epoch, equation 3.7 shows the formula for the epoch when the angular rate is assumed to be constant.

$$f_{ib}^{i} \approx \frac{1}{2}(C_{new} + C_{old})f_{ib}^{b} \tag{3.7}$$

Here f_{ib}^{b} is the accelerometer measurement vector. The same formula applies for Earth fixed and local tangent plane coordinate system transformations. A more

precise transformation, with less approximations, can be found again in Groves (2013*a*).

Again, the velocity in the inertial coordinate system is simply as is shown in equation 3.8.

$$v^i_{new} = v^i_{old} + a^i_{ib}\delta t \tag{3.8}$$

Here,

$$a^i_{ib} = f^i_{ib} + g^i_{ib}(r^i_{ib}) \tag{3.9}$$

where $g^i_{ib}(r^i_{ib})$ is the gravitational acceleration expressed using the inertial coordinate system position r.

Centrifugal and Coriolis terms need to be included when resolving the solution in the Earth fixed coordinate system. According to Groves (2013*a*) the Coriolis term can be neglected except in space applications for being relatively small. Equation 3.10 shows the realisation formula in the Earth fixed coordinate system.

$$v^e_{new} = v^e_{old} + (a^e_{ib} - 2\Omega^e_{ie}v^e_{old})\delta t \tag{3.10}$$

For the local tangent plane coordinate system, the transport rate of the local plane needs to be added. This is shown in equation 3.11 (l is the local tangent plane replaceable by n for the navigation frame).

$$v^l_{new} = v^l_{old} + (a^l_{ib} - (\Omega^l_{en} + 2\Omega^l_{ie})v^l_{old})\delta t \tag{3.11}$$

Position

Position after one epoch is the old position added with the distance travelled according to the average velocity during the epoch. Equation 3.12 shows this.

$$p^i_{new} = p^i_{old} + (v^i_{old} + v^i_{new})\frac{\delta t}{2} \tag{3.12}$$

This applies for the ECEF coordinate system also when using the Earth fixed coordinate system velocities. We used this representation in the local tangent plane derived from the local tangent plane velocities. A curvilinear position representation can be used instead of the local tangent plane. The equation 3.13 shows the height, latitude and longitude for the local tangent plane. This is also the order they are solved, first height then latitude and longitude. The h, L, λ are the ellipsoidal height, geodetic latitude and longitude respectively.

$$\begin{aligned} h^b_{new} &= h^b_{old} - \frac{\delta t}{2}(v^{l,Down}_{old} + v^{l,Down}_{new}) \\ L^b_{new} &= L^b_{old} + \frac{\delta t}{2}\left(\frac{v^{l,North}_{old}}{R_N(L^b_{old})+h^b_{old}} + \frac{v^{l,North}_{new}}{R_N(L^b_{old})+h^b_{new}}\right) \\ \lambda^b_{new} &= \lambda L^b_{old} + \frac{\delta t}{2}\left(\frac{v^{l,East}_{old}}{(R_E(L^b_{old})+h^b_{old})cosL^b_{old}} + \frac{v^{l,East}_{new}}{(R_E(L^b_{new})+h^b_{new})cosL^b_{new}}\right) \end{aligned} \tag{3.13}$$

Here R_N is the meridian radius of curvature and R_E is the transverse radius

of curvature of the Earth. Constant velocity and acceleration were assumed for the above equations during the epoch. The precision equations and more detailed explanation is found in Groves (2013*a*).

Figure 3.2 sums up the mechanisation steps.

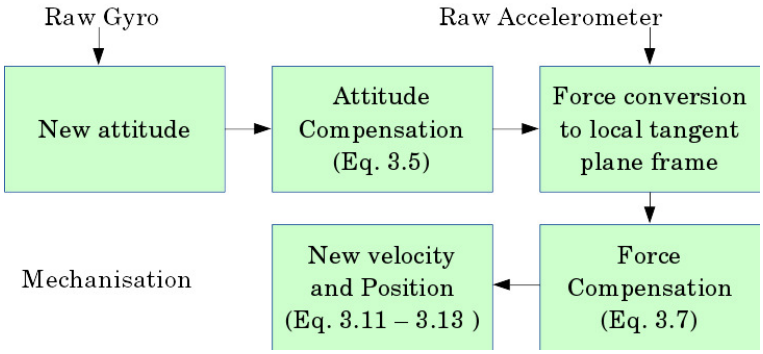


Figure 3.2: The mechanisation process.

3.1.3 Initialization and Alignment

In the beginning the initial attitude, velocity and position of the object are unknown. The initialisation is the process where the position and the velocity of the object are determined. This can be done using external systems like GNSS or UWB positioning system. The velocity and position estimates are acquired using subsequent measurements of the absolute positions.

The roll and pitch components of the alignment can be acquired using the inertial measurements and the knowledge of gravity when the object is stationary. The roll is the angle around the x axis while pitch around the y axis. This is called levelling or horizontal alignment. Equation 3.14 shows the roll and pitch derivation from the accelerometer measurements, according to Groves (2013*a*), Hide (2003), Rahim

(2012). The x corresponds to the Northing and y to the Easting and z Down in the NED coordinates.

$$\begin{aligned} \varphi &= atan2(-f_y, -f_z) \\ \theta &= arctan(\frac{f_x}{\sqrt{f_y^2 + f_z^2}}) \end{aligned} \tag{3.14}$$

If the inertial measurement unit accuracy is sufficient, gyrocompassing can be used to get the yaw angle. In gyrocompassing, the earth rotation rate can be distinguished. This can be used to derive the Earth rotation axis and object heading after levelling reveals the down direction. For MEMS sensors used for pedestrian navigation this is not yet possible, since they are not accurate enough.

3.1.4 Error Sources and IMU placement

Groves (2013*a*) lists three error sources for inertial navigation systems. These are the inertial sensor measurement errors, the initialisation errors and the processing approximations. The processing approximations include the discrete time, the gravity model, the computational and the timing approximation errors. The object trajectory and dynamics influence the mechanisation errors. For example, if the object trajectory consists of sharp turns the attitude error will be bigger, because of the approximation of a constant force acting on the object during one epoch. This approximation is more invalid the more there are fast sharp turns.

Furthermore, the errors can be divided into short, medium and long-term errors. The short term errors include attitude errors, which will propagate into velocity

direction and position errors. The scale factor and cross-coupling errors also have an effect. Errors in gravity model and Schuler oscillation are mentioned as longer-term errors. The Schuler tuning compensates for the inertial navigation system's changing north, east and down vectors while the platform moves on the earth.

Rahim (2012) mentions four different perturbation models for the inertial accelerometer and gyroscope errors. Phi-angle, Psi-angle, modified error and large heading uncertainty models are the four mentioned error models. The Error model can be used to estimate the accelerometer and gyro based error propagation in the output values. During an epoch, the error model predicts the development of the error during this epoch. In addition to the accelerometer and gyro error propagation, sensor bias, temperature bias, scale factor and axis misalignment errors can be included in the error propagation model.

For the pedestrian applications, the placement of the inertial sensor has a notable effect on the mechanisation process. If the used sensor is a mobile phone that will be carried on hand or in the pocket the mechanisation process will be challenging, because the zero velocity updates can't be applied as within the foot-mounted setup. Hardegger et al. (2015) mentions examples on hip and belt mounted trials. He names these as step and heading systems since instead of full mechanisation only step detection with inference is applied for the user tracking. A similar naming is in the review by Harle (2013).

The zero velocity updates can be used in the mechanisation process. This is suited for the foot-mounted pedestrian systems. When the IMU is placed on the

foot, each step a short stationary period can be detected. On the contrary when the inertial unit is on the hand this detection is not possible since the unit is in constant motion. This applies for the whole of the upper body IMU placement. Error modelling and zero velocity updates will be further discussed in the following chapters while examining the ways of integrating multi-sensory data.

3.1.5 Step and Heading Systems

If full mechanisation is impossible, estimation techniques can be used in determining steps and heading. Nguyen et al. (2010) uses machine translation techniques for mapping inertial sensor data readings into movements. Steps are calculated and the readings are recorded. These inertial sensor readings are then compared with external positioning system position outputs, like that of GPS, to learn the realised user inertial track mechanisation in order to construct a database for IMU data to walk trail translation. Neural networks can be used in the learning process. Furthermore, Adusumilli et al. (2014) analyses the use of regression forests for the same scenario.

Compared to Kalman approach the statistical error estimation requires a great deal of effort from the system. At first the system has to learn how to translate the IMU records into position updates. This gathered data needs to be saved, which requires storage space. Adaptation to a new user, processing and organising stored data requires computation power. Although, a simple system uses just a step detection method with a fixed step length. Attitude determination requires gyros.

With just one gyro on the user's belt for example, the heading change can be tracked with low accuracy.

3.1.6 Conclusions on IMU

Three accelerometers and three gyros make up an inertial measurement unit. Placing one accelerometer onto a user is enough for step detection. One gyro is enough for rough estimation of direction change. Statistical learning and inference of heading and step length is thus possible with only the two sensors. Placement of these sensors is crucial. Placing only one gyro on the foot may work as well as placing it onto user's belt. Generally, the acceleration change of foot makes the stationary period detection much easier compared to if the accelerometer was placed on belt, pocket or user's hand.

As inference and machine learning techniques for high sampling rate data demand a lot of storage space and processing, the implementation selection of inertial sensor dead reckoning technique for mobile devices would benefit of being less resource demanding. In this thesis, the mechanisation process that is the double integration is used.

The POINT software is a Nottingham Geospatial Institute's navigation software package. It can do precise point positioning and double difference solution derivation for GPS and GLONASS data. Currently Galileo and Beidou processing are being added. In addition, it does precision double integration for the inertial data. The software package is implemented for post processing the recorded measurement text

files. It can fuse the GNSS and inertial data by both loose and tightly coupled architectures (see 3.2.1).

The UWB and BLE positioning subsystems were implemented in MATLAB. In order to be able to debug and visualize the multi-sensor fusion tracks, a port of the inertial double integration of the POINT software was made into MATLAB. POINT is written in C++ and can be compiled for Windows and Linux environments. The POINT MATLAB port includes the inertial double integration process previously described in this chapter. It uses first order approximations for saving computational power. The MATLAB port is also a friendly way to learn how the POINT software package works. For analysing the accuracy differences between POINT and the MATLAB remake of the POINT software, multiple inertial tracks were processed on both platforms. Same Zero velocity update algorithm was used for both versions. Zero velocity detection compared the measured total gyro vector magnitude and total accelerometer vector magnitude with a pre-set threshold. The results indicate that the approximation errors are small in comparison with errors produced by change in ground type, change in the strapdown technique of the IMU or the success of the zero-velocity detection. The remake of the POINT software for MATLAB is described later at the sensor fusion section along with the Kalman filter and the zero-velocity update detection explanation.

3.2 Timing based Ranging using GNSS and UWB

Ranging when using RF technologies requires very high-resolution timing. One nanosecond error in timing, when ranging, is equivalent to an error distance of approximately 30cm. Anchor locations, for GNSS the satellites and for UWB the beacons, need to be known. Time synchronisation between anchors has to be exact. Time recording of the sending and receiving of the ranging signal must be exact. We look first shortly how the GNSS receiver functions. After that the ranging using UWB is discussed.

3.2.1 GNSS Receiver

The basic function for a GNSS receiver is to offer position information outdoors. As explained previously, in urban and indoor areas this task is challenging because of multipath and attenuation due to obstructions on the satellite signal paths. High-sensitivity receivers try to extend the operation into urban areas, but no breakthrough has been yet made that would make GNSS receivers work deep indoors. Longer correlation times help in the synchronisation process.

Receivers constitute three abstraction levels. The top level offers position, velocity and time information. This information is derived from the inner level, which calculates pseudoranges for each available satellite. Further, correlation loops adjust the pseudo-random noise sequences (PRN) in the deepest receiver level to match with the captured satellite sequences.

If fusing GNSS receiver data with other sensors, loose coupling is a term used when applying the top-level position information in the fusion process. Tight coupling means that pseudoranges are used and deep coupling when correlation loops are involved in the sensor fusion process. Main points considering sensor fusion applications are now investigated. Figure 3.3 depicts the GNSS receiver service functions.

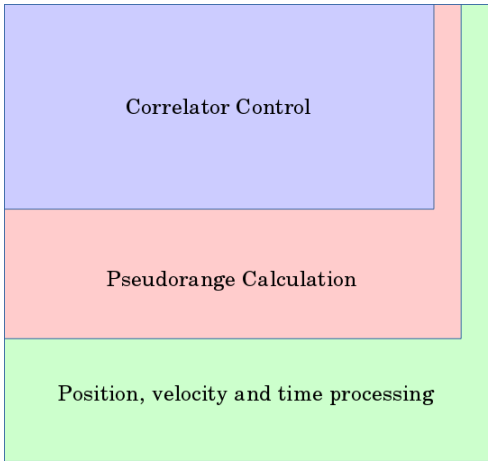


Figure 3.3: GNSS receiver service levels.

Previously, only position domain has been accessible for regular users of mobile phones and the chips included. Android Nougat OS is possibly the first large scale mobile phone product where the system can offer pseudoranges. A very interesting development could enable centimetre level positioning for handsets is examined in Humphreys et al. (2016). In any case, as Fernandez et al. (2011) emphasises, the presumption is that the receiver has access to at least four line of sight satellite ranges. Fernandez et al. (2011) use only three satellite ranges with additional information from a WLAN anchor.

Positions for Loose Coupling

A reference frame uniquely points a location on the earth. As an example, we examine the U-Blox manual and the receiver output that is built in the Microstrain 3DM GX4-45 sensor. The reference frame for the sensor output is WGS-84. Position reference format options are LLH and ECEF. These are latitude, longitude and height or the position in the Earth fixed frame. Dilution of precision, velocity, clock information and further, the native options for U-Blox receiver communications are available in different sensor modes.

Latitude and longitude in decimal degrees and height in metres outputs a position on the earth. In addition, the sensor is able to offer horizontal and vertical accuracy estimates. The accuracies are given in Circular Error Probable (CEP), which is the 50% circular confidence limit for half of the probability that the measured point is inside the circle.

Pseudoranges for Tight Coupling

Instead of positions from the GNSS system, pseudoranges can be used in the final solution derivation. Here range measurements from the GNSS receiver and for example UWB system can be used simultaneously in the position solution derivation. Wang et al. (2016) applied this in their cooperative vehicle scenario with Doppler difference measurements. UWB infrastructure anchor ranges were combined with GNSS satellite pseudoranges in a robust Kalman filter, which improved the accuracy.

Furthermore, for deep integration, the reader is guided to check the excellent

technical report by Schmidt & Phillips (2010).

High-Sensitivity Receivers

High sensitivity GNSS receivers are able to track the user even indoors. These receivers are 1000 times more capable of resolving the GNSS signals. Ayaz et al. (2010) used longer coherent integration times for static user indoors. GPS and Galileo satellite signals were compared for indoor positioning and GPS L5 was found to function best. GLONASS and Beidou signal study was left for future.

Lin et al. (2013) extended the high sensitivity functionality for software receivers. The used software receiver was the C++ based GSNR, which is developed in the University of Calgary. More in depth explanation of this software receiver can be found by Tamazin (2011).

Conclusions

In this thesis, the fusion research concentrates on the loosely coupled approach. The position and position accuracy measurements will be used from the GNSS receiver. This simplifies the plug-and-play adaptability and modularity of the fusion design in the first phase. The obvious next step continuing from this study would be to extend the adaptability and modularity into the next inner abstraction level. This means using pseudorange and Doppler difference observables with similar quantities from the other sensors that are included in the sensor fusion process.

3.2.2 Ultra-wideband Ranging

Partly because GNSS does not work well indoors, Ultra-wideband technology has got recent attention as a technology that works indoors. UWB is defined as a signal that covers 20% of the centre carrier frequency. Alarifi et al. (2016) reviews the UWB technology and extracts three main reasons for recent interest to use UWB for indoor positioning. High data rate (100 Mbps) allows fast communication, high bandwidth makes it easier to deal with multipath and the low frequency used to send the UWB pulses enables the signal to travel through obstacles that can be detected on the other side. This means that line of sight condition is not necessarily needed for positioning. Alarifi et al. (2016) also raises the modulation of the signal as a decisive factor in successful enhancement of localisation. Pulse position modulation (PPM), on-off keying (OOK), pulse amplitude modulation (PAM), pulse width modulation (PWM), time-hopping spread spectrum (TT-SS), pseudo-random time modulation (PR), binary phase shift keying (BPSK), time-hopping binary phase shift keying (TH-BPSK), time hopping pulse position modulation (TH-PPM) and minimum shift keying (MSK) are mentioned as alternative modulation options.

Alarifi et al. (2016) presents also algorithms that are used in UWB positioning. These are time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) and received signal strength (RSS) based algorithms. The angle of arrival method is difficult on the tag side because of the bulky antenna requirements while RSS doesn't use the advantages of the large bandwidth. Bellusci (2010) and Bras

et al. (2012) agree that antenna design in general is one of the bottlenecks in UWB performance. Thus, the timing approach seems the most promising approach. UWB has a high time resolution although clock jitter is a problem. Out of TOA and TDOA, TDOA is the more stable approach if the UWB anchors are not synchronised.

Bellusci (2010) analyses the statistical path-loss model used in UWB ranging. The path loss model accuracy of the signal strength was found to be between decimetre to metre. Bellusci (2010) ended up in similar conclusion to that of Alarifi et al. (2016) that timing based methods are more accurate. The accuracy can be below decimetre down to 1cm. Moreover, Bellusci (2010) found that using bandwidth larger than 2GHz improves the detection between line of sight and non-line of sight conditions, although high bandwidth requires more from the hardware. His main conclusions are that timing based methods are better, with an accuracy of below a decimetre can be achieved under line of sight conditions and that fading increases and accuracy decreases when used bandwidth is decreased.

Two Way Ranging by Decawave

In this thesis the Decawave TREK1000 ranging kit was chosen for the implementation of the UWB subsystem. The following is an excerpt from the Decawave application notes, the white papers and documents found at the Decawave.com (12/2016) website. A 450m line of sight reach and a 45m non-line of sight reach are stated for the kit. The kit consists of three anchors and one tag. The range between each anchor and the tag is calculated using a two-way ranging time of flight

approach. The master anchor is sending a proprietary timing synchronisation signal that is used at the slaves for time synchronisation.

Figure 3.4 depicts the two-way ranging sequence between the tag and an anchor.

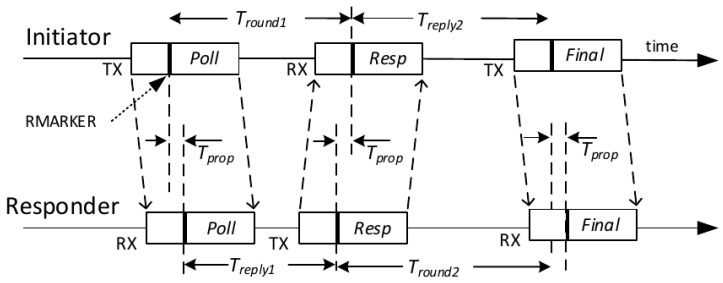


Figure 3.4: Decawave Two-Way Ranging sequence between the tag and one anchor.

The equation 3.15 shows the time derivation for double-sided two way ranging between the tag and one anchor. Initiating anchor sends an UWB impulse signal which reaches the responder and which sends a response. Then the responder sends a signal back for which the initiator sends a response and we have two round-trip measurements as seen in the Figure 3.4.

$$T_{prop} = \frac{T_{round1}T_{round2} - T_{reply1}T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}} \tag{3.15}$$

The above equation is not exact as two main error sources affect the timing. These are reported as clock drift and signal level at both ends.

Ideally the signal level at the receiver or the transmitter side should not have an effect in the ranging result timing. But unfortunately, this happens and to compensate for this the microcontroller code does a tabular correction depending on the measured range.

For decreasing the effect of clock drifts at the measured ranging times the whole two way ranging sequence should be as fast as possible. It is stated that if the reply times on both the tag and the anchor are programmed to be 10ms the ranging due to the clock drift should be around 2cm. The wireless clock synchronisation algorithm is based on Kalman error tracking of the time of arrival and the clock skew. For the Kalman filter the measurement variance and the process noise were optimised. The measurement variance is declared to be $1.5 * 10^{-20} s^2$ for the TOA measurements. Taking into account clock jitter, temperature variations and other device dependent non-idealities, a measurement noise of $3.0 * 10^{-20} s^2$ was used. Process noise was manually adjusted and found to be $5.0 * 10^{-20} s^2$. For more detailed information, the reader is directed to register for the website and browse through the application notes. The range measurement code for the kit is openly available and enables customizing the decawave chip functionality for the target application purposes.

Conclusions on UWB

The accuracy of the UWB system makes it the most attractive positioning technology indoors. The TREK1000 kit, which provides the most accurate results of the available testing kits, achieves an accuracy of few decimetres even with slight non-line of sight conditions and with human body shadowing effects.

The timing approach is held as the most accurate method for UWB. The IR-UWB multipath rejection capabilities are better than the sound based technologies. Multipath compensation using sound based systems is challenging indoors. These

are the two main reasons for using the selected kit.

Along with being a relatively mobile and light weight kit, the selected UWB kit is very customizable. More in-depth study by changing the microcontroller code enables researchers to try out their own algorithms if necessary. For example, a system design similar to combined GNSS code and phase ranging by Wan et al. (2017) or double sided two-way ranging by Kwak & Chong (2010) could even further enhance the current design the decawave system uses. Limitations like the algorithm complexity and storage space on the embedded device limit the possible tests.

When using ranging and when the anchor locations are surveyed as accurately as possible, position acquisition methods can be used. These include triangulation, trilateration or multilateration techniques.

An iterative least squares trilateration was used for position derivation from the ranges produced by the selected kit. The output of this algorithm was the position and a standard error deviation estimate for this position and for each anchor the residuals at the end of the iteration. A fixed number of 10 iterations was used with the initial value as the previous value from the last UWB subsystem epoch. The UWB subsystem offers a new measurement each second that is at 1Hz update rate.

3.3 Signal Strength Measures for BLE

Bluetooth low energy is another wireless technology standard. It operates within the ISM 2.4GHz band. The low energy standard concentrates on saving the energy

consumption in comparison to the regular Bluetooth which concentrated more on the personal mobile device data streaming.

The main application of BLE for localization is proximity based. As Cerruela Garca et al. (2016) mentions, compared to WiFi, BLE signal strength has more rapid fading, which makes the BLE to produce few more erroneous readings compared to the WiFi. But the sampling rate being much higher compared to the WiFi compensates for this drawback. In addition, BLE beacons are more deployable than WiFi anchors. Channel separate polynomial regression and fingerprinting models are mentioned. BLE consists of 40 2MHz narrow-band channels. Contreras et al. (2017) describes that 37 channels are data channels while 3 channels, 37, 38 and 39 function as advertising channels. Being narrow-band the timing resolution and multipath detection with BLE is much more challenging than with UWB. This is why signal strength measurements are the better approach for BLE over the timing based methods.

The three advertising channels can be set to advertise at 20ms interval. The advertising channel can be set to use only channel 37 or to use all three channels. When using all three advertising channels the advertising hops using the sequence 37, 38 and 39 at the defined interval. The advertising message can be set and even the advertising power can be controlled when using class 1 BLE devices.

In the work conducted for this thesis, two Raspberry Pi's with Cambridge Silicon Radio adapters were used. These were set to advertise at 20ms interval. In addition, four Kontakt.io beacons were used at 100ms advertising interval.

Similarly, the scanning period can be set to 20ms using BLE adapter at the laptop for example. The beacons advertise according to their own clocks and possibly overlap when transmitting simultaneously and thus race over who gets heard by the scanner. In general, the advertising channel signal strength is then used for proximity positioning.

Zhuang et al. (2016) divides methods for BLE positioning into Polynomial Regression Model (PRM) and fingerprinting (FP). Next these are examined, the PRM, which indicates the RSS value into range conversion model and the fingerprinting approach which uses the location based anchorwise RSS fingerprints.

3.3.1 Path-Loss Modelling

Path-loss models estimate the signal strength at a given distance. Lohan et al. (2015) tested BLE and WiFi access points and their path loss models. A general phenomenon was seen between 2.4 and 5GHz frequencies. Higher frequency signals have higher path losses. Viswanathan & Srinivasan (2015) lists possible path loss models. These are Log distance, Rappaports pathloss model and Friis transmission model. In addition, Figueiredo e Silva et al. (2015) present an enhanced ITU-R model. Equation 3.16 shows the ITU-R pathloss model.

$$P_r(d) = P_T + C - 20log_{10}(\frac{4\pi f}{c}) - 20\eta log_{10}(d) + v \tag{3.16}$$

Here the received power at distance d is $P_r(d)$. P_T is the transmission power.

C models additional system losses, while η is a constant modelling additional losses in the path of the signal. v is log-normal distributed random variable presenting fading. Fading describes the wireless channel attenuation due to multipath and electromagnetic interference. Shadowing similarly describes the signal strength fluctuation at a point in the channel. Multipath can be estimated by using ray-tracing which requires knowledge of the walls and reflections and is then included in the fading measurement variance for the channel.

3.3.2 Fingerprinting

Signal strength fingerprinting is a pattern recognition method. First, in the training phase a pattern is recorded for each selected location. Secondly, in the measurement phase, a measurement vector is compared with the database created in the training phase. The best match can be derived from the matching process between the measurement and the database.

Signal strengths can be recorded at different map locations. This way the wanted locations can be associated with the signal strengths of each BLE anchor and with the signal strength fluctuation at that map point. If available, other distinctive sensor measurements could be used at the same time as fingerprints. The points can be picked by using a grid structure or other applicable point node network structure. The measurements conducted by Hao (2015) indicate that the Gaussian Processes Regression can be used to model the WiFi signal strength and its variance at a certain location. The signal strength area or path between recorded points can

be interpolated using for example Gaussian Processes regression. In Lohan et al. (2015) the k-nearest neighbour fingerprints were selected and an average of these were taken as the location. Shadowing coefficient was the standard deviation of the measurements from the average. Faragher & Harle (2015) achieved an accuracy around approximately 3m when applying mitigation techniques for the shadowing.

3.3.3 Conclusions on BLE

When comparing with the UWB, the effort that is input in creating fingerprinting maps and the low accuracy of this makes Bluetooth less attractive. But the ease in deployment of BLE anchors and energy saving features, makes BLE the more robust option.

A Bluetooth Special Interest Group (SIG) is investigating the ranging and angle of arrival methods for standarsisation. The pathloss model deployment is fast unlike the heavy fingerprinting method. Fingerprinting on the other hand is considered slightly more accurate especially indoors where walls and changing obstructions limit the usability of the pathloss models.

Both the pathloss and the fingerprinting were chosen to be tested. The constructed systems are explained in detail in the following chapters and the test result indicate which approach is better when using BLE.

3.4 Map Features and Constraints

Davidson & Piche (2016) lists three ways to match map and positioning subsystems data. These are using wall constraints, road link-nodes or by adjusting user heading according to the environmental features. 2D maps come naturally in mind when talking of maps. Nowadays, when more accurate measuring is possible, 3D representation is conquering the field.

Particle filter has been the favoured filter for indoors because of its robust nature. Mansour & Waters (2013) successfully applied map matching for Kalman filter. Here the map matching result was used as a correction update measurement for the Kalman filter.

Further Quddus et al. (2007) reviewed map matching applied for vehicular traffic. He divided the methods into geometric, topological, probabilistic and advanced map-matching algorithms. Point and curve matching describe the geometrical approach, while usage of polygons and network geometry were described as being topological. An example of a probabilistic approach was used for crossing or link node proximity decision by comparing the link node structure with the position solution error region. Graphs with links and nodes by Koivisto et al. (2015) applied probabilities to the nodes. Context inference, like a known destination, made some paths more probable for the user to take. Advanced systems can for example use an inference engine or fuzzy logic in deciding the correct travelled path.

3.4.1 Conclusions on Mapping

The test walks in this thesis are conducted around and in the The Nottingham Geospatial building (NGB). The NGB building walls have been surveyed beforehand. Also, the location of the doors is known. Thus, using walls is an obvious choice. In addition, each sensor subsystem produces unique tracks. The application of track form matching in the final fusion process is investigated.

3.5 Navigation Context Inference

The previous sections focussed on deriving the positioning information out of the selected sensor technologies. This section gives an insight to how the sensors can be used for deriving contextual information.

This contextual information for navigation is roughly divided into two by Groves (2013*b*). The sensors at different locations or worn by different users produce typical output values and patterns that can be detected. These two main context subsections are the location based and the behaviour dependent contexts. First, we look at the location based context derivation and then through the behavioural context derivation trials. The conclusions section sums up the major points to be noted for the pedestrian navigation applications.

Rivero-Rodriguez et al. (2014) summarises that context can be determined using physical sensors, virtual sensors (applications) and logical sensors (inferred from physical and virtual). The contextual information changes according to location.

Context also differs according to different behavioural features. As well, user interaction is involved in the determination of the context as stated. Moreover, navigation target and route preferences define the context in part. Yus et al. (2014) tested an adaptive location based node network system called SHERLOCK. This multi-user system was tested for two use cases. These were firefighter communication during mission and transport vehicle search for tourist who has just arrived to his/her destination and does not know the surroundings. The main feature of the adaptive system was the possibility to add multiple users whose knowledge of the situation could be combined and shared for the advantage of every other node user.

Mobile applications use the context detection information. The context inference algorithms try to figure out the user context. Either for commercial use or user's personal chores, in any case, defining the correct context offers enhanced information for predicting user patterns and adapting to these possible upcoming needs. The navigation filter can use this information to use the most fit sensors and corresponding data to offer the best possible positioning solution.

3.5.1 Location Based

Rivero-Rodriguez et al. (2014) tested five classification methods for location detection. Out of the Naive Bayes, Decision Tree, Bagged Tree, Neural Network and K-Nearest Neighbour methods the Bagged Tree method gave the best results. In the Bagged Tree method, multiple decision trees are trained using different subsets of the same larger dataset. Each tree is assigned a weight. Using a voting system,

more weight is given to a tree in the region where the classification rate is better. The locations used in the study were in the abstraction levels of home, work and so on.

Perhaps the first and the most relevant detection is between the indoor and outdoor environments. Daniel et al. (2016) developed a method for simple indoor/ outdoor detection. An indicator derived from GNSS receiver properties managed to differentiate whether a person was walking outside or inside.

Bluetooth low energy (BLE) beacons send an advertisement message to the environment at fixed intervals. If GNSS receiver measurements are not available or the quality is bad, these alternative proximity sensors may be used to indicate the current location context and proximity. Camera based image matching using online image database is another alternative.

A map is the most obvious representation of location. Most of the navigation applications need a preloaded or online map database in order to present the location to the user. For the blind or the deaf, touch and audio based navigation databases are the better choice. In both cases, the location context is very much defined using the map, depending on how successful the acquired positioning solution has been.

The map updates are relevant. Especially if the sensor infrastructure, like in the case of Bluetooth low energy anchor deployment, is changing, the changes need to be informed for the users map at latest when arriving to the proximity of the system. GNSS signals report rough satellite locations within the navigation message structure. Similar approach for the Bluetooth low energy devices could ease the map

update process. BLE anchors capable of sending their locations would free the map database of this responsibility.

3.5.2 Behavioural

Indoor spaces consist of stairs, hallways, corridors, offices, rooms, lifts and so on. The user movement changes in these differing spaces. Restricted spaces usually require the user to walk instead of running. In open areas, especially if the calendar alerts of a missed meeting, the user might make a dash and run for a certain amount of time. In taxis and trains, the vehicle vibration can be sensed.

Especially the inertial sensor can detect changes defining the behavioural context. Khalifa et al. (2013) examined this and classified the pedestrian motion into activities relative to the location. Walk, run and climbing stairs are examples of the changing behavioural context. This behavioural activity information is related to the vehicle type as well according to Groves et al. (2014). Vehicles have their individual features that can be sensed. For example, if a smart shoe detects the round motion of the foot while cycling, this is a clear indication of the user's vehicle. This is depicted in Figure 3.5.

An inertial trace, for climbing the stairs and when cycling, has distinctive characteristics. The context engine can detect these differences and give instruction to for example the navigation filter or another application. When cycling, an additional odometer on the bike could connect to the user's mobile phone to enhance smart-boot and GNSS position solution.

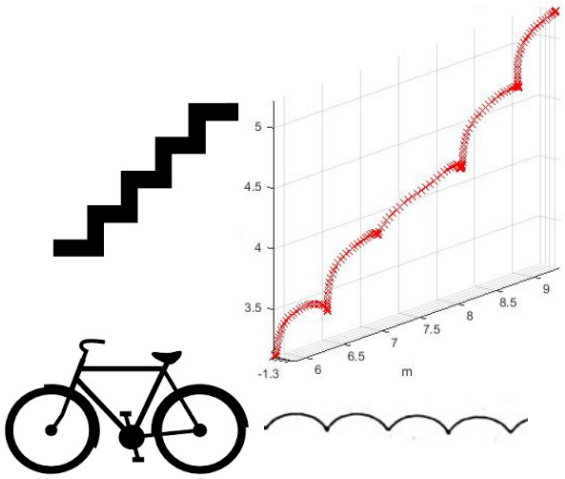


Figure 3.5: Inertial traces for stairs and cycling.

The behaviour detection can be implemented at various levels of abstraction. When cycling and the user is wearing a smart boot which senses inertial movement; If the application needs the round motion information fast, the detection can be implemented within the sensor itself and the sensor can have an output for the cycling detection. This saves the processes on the mobile device side for something more relevant, like the navigation solution derivation. The interface design between the sensor and the mobile device has then an impact on the overall performance of the system.

3.5.3 Conclusions on Context Inference

The context inference can include machine learning techniques. The user can be identified by the smart shoe from the distinctive personal walking styles. Figure 3.6 summarises the location and user behaviour context inference for a mobile applica-

tion.

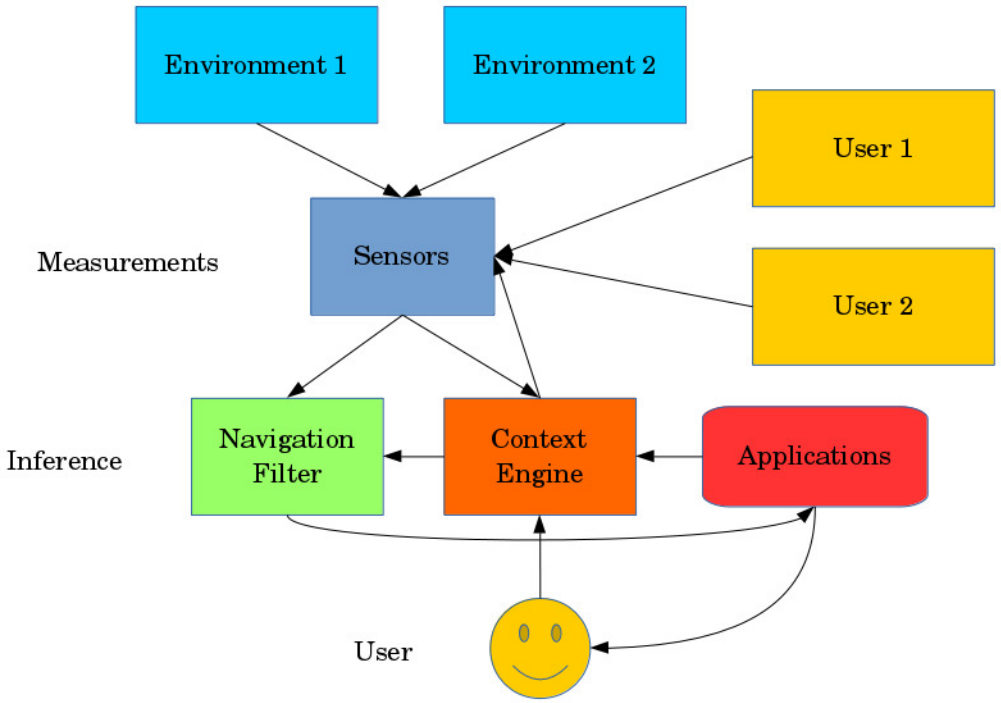


Figure 3.6: An example block structure of a location and user behaviour aware mobile application.

Locations, environments and user behaviours produce individual sensor outputs that can be distinguished. The depth of the context inference can vary from single step detection to a more complex gait style/ user identification/ gait phase capable detection for medical purposes as described for example by Tunca et al. (2017). Our context engine uses the knowledge acquired from this chapter and from the literature review. The design is presented in the later chapter dealing with map and context engine design.

Chapter 4

Sensor Fusion Design

So far, adaptability in navigation has solely been the user's responsibility. A simple example of this is the use of GPS while driving and then checking the billboard map within the commercial centre for finding the wanted shop. In order to increase the level of seamlessness, the mobile phone that was used for car navigation should be able to offer the same information as the billboard map. Even a further increase could be possible if the application on the phone would be able to assist in navigating to the correct shop while walking.

Two kinds of novel features of an adaptive device can be distinguished in the previous example. The device that is capable of navigating the user to the shop, has the configurational flexibility to do this. In addition, it has the environmental adaptability to adapt to car navigation as well as to indoor navigation.

We look first through these two characteristics of an adaptive system. Secondly, a review of filters presents filters like the Kalman filter and the Particle filter. Before

concluding remarks and comparisons we examine the alternatives for sensor fusion architectures.

4.1 Configurational Flexibility

Configuration for navigation purposes consists of the infrastructure, mobile device sensors, mobile device hardware and of the navigation software. The infrastructure and the sensors define the technology choices that the navigation solution derivation can use. This affects strongly to the availability of the navigation solution. That is if only GNSS is used and available there will not be a solution for all indoor environments. Hardware aspects, such as the type and capability of the battery and processor define the power configuration which also limits the software methods that can be used. Software methods that will be implemented will have an effect on the efficiency and accuracy of the final navigation solution. Figure 4.1 depicts this novel division and lists the features that the corresponding level affects.

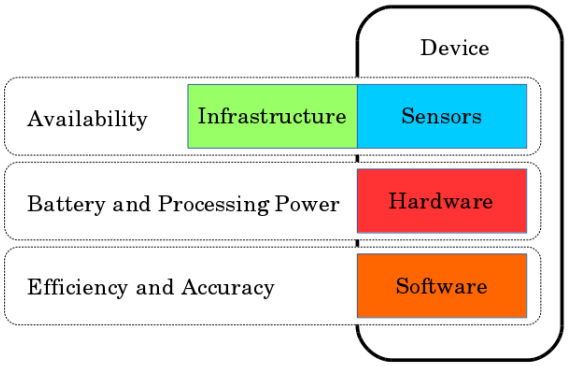


Figure 4.1: Layer structure that enables navigation services.

The diversity of sensors range from sensing the temperature to recording the acceleration and measuring the deformation. Plug-and-play sensor modules all using a well-defined unified serial interface would ease the hardware designer's workload. A common sensor interface would reduce the cost and complexity in connecting the sensors to the system.

The interface consists of hardware and software components. Xi et al. (2006) developed an adaptive multisensory interface. To reduce the cost, the same port components were used for multiple sensor type interfacing. This increases the efficiency in the size and power when less components are needed. Eid et al. (n.d.) own a patent for an Universal Sensor Interface model for aeronautic applications.

In addition, a smart sensor would be capable of telling the system of which type it is. Dynamic calibration of the sensor is furthermore a feature that increases the flexibility of the system. Smart sensors could be set up and used in altering ways depending on the final application. To save space the sensor hardware could be implemented on a reconfigurable FPGA (Field Programmable Gate Array). Then the sensor hardware could be able to reprogram its FPGA accordingly.

4.2 Environmental Adaptability

For a truly seamless navigation system the changing conditions in the environment should not affect the availability of the position solution. A seamless system has an adequate number of sensors, the processing hardware and software for deriving the

correct good quality position solution. This adequate sensor configuration is able to offer enough positioning information for the navigation application. The application is also smart enough in selecting the best quality measurements to use, independent of the environment. In other words, the system adapts to the surrounding environment.

The application uses the available hardware resources and possible parallel application information (e.g. calendar). Kray & Kortuem (n.d.) lists three resources. The sensor measurements are the first obvious source of positioning information. Secondly, the inference based positioning information from other applications and position history help in deriving the position. Third, the interaction with the user can be taken along to the solution derivation. The combination of information happens in the Context Engine as depicted in Figure 3.6.

The applications at present are not very aware of all the possible resources that are available in handheld devices. User's context is hardly exploited in applications nor even tried to be extracted. Apart from battery level bars the battery saving or memory allocation information is not much shared between applications.

Kray & Kortuem (n.d.) combined measurements and inference based position information with interactive positioning information coming from the user. The user information can be often used for example as the first initialisation information. Figure 4.2 depicts this procedure of measurement and user defined context enhanced data combination in determining the context.

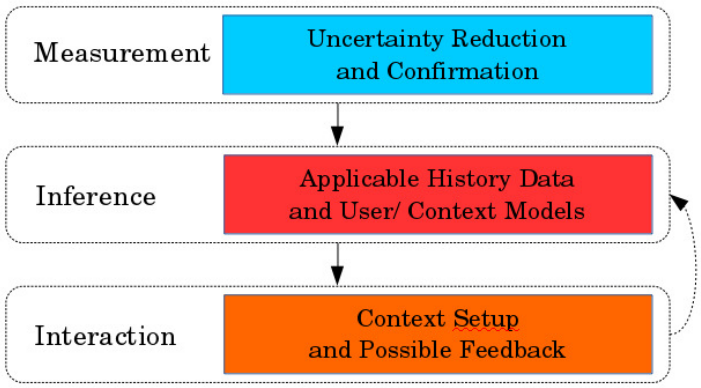


Figure 4.2: Adaptive system uses and combines the information from multiple sources and reiterates a solution when necessary.

It consists of three layers. For the actual positioning motor, like Kalman filter, an adaptive feature is the monitoring of the residuals and the control of the noise in the system and measurement models. These adaptive additions to conventional Kalman filter prevent the divergence of the filter. Sasiadek et al. (2000) used a Takagi-Sueno singleton fuzzification logic. It prevents the Kalman filter from diverging. This is advantageous especially if the tracked target has sudden dynamic movements.

Similar divergence control was implemented by Hu et al. (2003). Here two algorithms were tested of which the first one applied fading memory where the latest measurements are given more weighting over a time period. The second one dealt with the variance factor of the dynamic model. Ladetto (2000) lists that stride length and gait characteristics should be taken into account when determining the pedestrian steps. Step frequency, signal variance, road inclination or the stairs give

hints for adjusting the filter to function correctly. For cars and odometers, the same applies with suspension and the circumference of the car tires. Hide et al. (2007) raises the question on how to use one filter in different environments. Synchronisation of these different measurements and events is possibly the most important challenge to overcome.

4.3 Classification

The dictionary definition for classification is the "comparison of the measurement with pre-recorded databases". The inputs are mapped into the outputs. For continuous output, the mapping can be called regression whereas for discrete it is called classification.

Alpaydin (2014) lists machine learning methods. These offer alternative ways for navigation. The positioning systems are characterised by a set of parameters like position and velocity etc. These parameters can be optimised through learning methods. For example, if the same person walks at the same speed through checkpoint every day, this information can be used to identify that person.

There are two ways to learn, by prediction using example data or by description acquiring knowledge of the data and measurements. Machine learning has often a large space and time complexity. That is why efficiency and predictive accuracy compete between applications. Nguyen et al. (2010) used classification by mapping sensor readings into displacements using a statistical model. In the following few

classification and regression methods are examined.

Gaussian Processes: Rasmussen & Nickisch (2010) explains that Gaussian process is a stochastic process. It depicts a gaussian probability distribution at each point of the output function. Gaussian probability governs the properties of the mapping function between test or the training data and the output values. Faragher et al. (2012*b*) used Gaussian processes for RSS fingerprint heat mapping.

Principal Component Analysis: Mitchell (2007) considers that to save storage within a multi-sensor system an approach like PCA can be applied. A common set of vectors is built to represent the raw data. The measurements can be transformed into, for example, common representation of a posteriori probabilities.

Support Vector Machines: SVM is also called the kernel machine. The support vector machine is a classifier. This classifier is usually a separating hyperplane that distributes new data into either of the sides. The hyperplane is called optimal if the marginal between the data points and the hyperplane on both sides is maximal. Using multiple kernels additively the new data can be separated by the learning criteria within these kernels or support vector machines. Figure 4.3 depicts the optimal hyperplane.

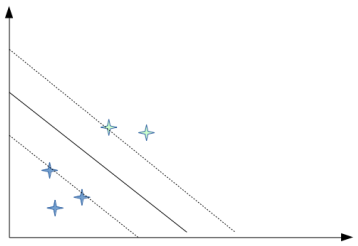


Figure 4.3: SVM divides data into two using a kernel. Multiple kernels can be used additively to classify the data further.

Decision Trees: Decision trees consist of decision nodes. The training input data defines the tree. When test data is input, it goes through the decision nodes. Decision nodes implement a test function with discrete outputs that further classify the test input. The regression tree is different from the classification trees in that the node condition is inherently of separate type. The random forest derivative was used by Adusumilli et al. (2014). In random forests, the decision trees are grown randomly using a training set. There is no pruning of trees.

Neural Networks: Neural networks imitate and examine the function of brains. These models help in building computer systems for specific tasks. Many neurons or perceptrons gather information. They pass it on to neurons with different tasks or higher level of information. The interconnections can change and the weights between connections can be recalculated depending on the input values. Specific output neurons are activated in the end of the network process.

4.4 Kalman Filters

Kalman filters are estimation methods. The Kalman filter can be categorised further into the linear, analytical linear (the Extended Kalman Filter, where the nonlinearity is linearised locally), Sigma-Point and Monte Carlo Kalman filters. Haug (2012) raises an interesting aspect that until present only Gaussian densities have been tested, although the basic Kalman filter framework does not only assume or condition for this.

A common factor for these estimation methods is the use of a model for approximating the position from the measurement data. The raw input data has noise or an error distribution depending on the wanted derived or measured parameter. In estimation methods, the sensory data input is usually closer to raw data. That is for example accelerometer readings.

Deriving Weighted Recursive Least Squares: The least squares method, according to Figueiras & Frattasi (2010), minimises the squared error between the actual measurements and the predicted measurements derived using a model.

The model (h) that relates the measurements (z) with the estimated position (X) is

$$z_i = h_i(X) + \eta_i \tag{4.1}$$

where η is noise and i the epoch. The squared cost function is

$$\Sigma\eta_i^2 = \Sigma(z_i - h_i(X))^2 \tag{4.2}$$

where the minimum for X is searched by setting the gradient to zero.

$$\nabla(\Sigma\eta_i^2) = -2[Z - H(X)]^T \nabla H(X) = 0. \tag{4.3}$$

In weighted least squares the noise in the measurements of different stations can be modelled to be different. When the noise vector is

$\eta = [n_1, ..., n_n]^T$ we get

$R = E[\eta\eta^T]$

and the cost function is then

$$\Sigma\eta_i^2 = \Sigma R_{i,i}^{-1}(z_i - h_i(X))^2 \tag{4.4}$$

and the gradient function becomes

$$\nabla(\Sigma\eta_i^2) = -2[Z - H(X)]^T R^{-1} H(X) = 0, \tag{4.5}$$

where R is usually defined as a diagonal matrix, because the measurements are assumed to be independent.

The weighted linear least squares solution is

$$X = (H^T R^{-1} H)^{-1} H^T R^{-1} Z \tag{4.6}$$

where the covariance $P = (H^T R^{-1} H)^{-1}$ represents a measure of certainty in the position estimate.

The weighted recursive least squares solution is

$$X_{k+1} = P_{k+1} H^T_{0:k+1} R^{-1}_{0:k+1} Z_{0:k+1} \tag{4.7}$$

which can be rewritten as

$$X_{k+1} = X_k + K_{k+1}[Z_{k+1} - H_{k+1} X_k] \tag{4.8}$$

where the

$K_{k+1} = P_k H^T_{k+1} [H_{k+1} P_k H^T_{k+1} + R_{k+1}]^{-1}$

is the update gain.

Kalman Filter: Figueiras & Frattasi (2010), Siciliano & Khatib (2008), Chui & Chen (2009) describe a Bayesian filter consisting of two steps. These are the prediction and the correction steps. The Bayesian filter has Markovian properties meaning that the position estimate depends on position knowledge of the current and one previous step.

Van-Keuk's evolution model described by Koch (2014) and the Interacting Multiple Models used by Zhang et al. (2013) are examples of process propagation functions. Sensor likelihood functions or sensor models correct the predictions with measurements. In a Kalman filter the prediction and correction models are assumed

to be linear. Figure 4.4 shows the loop.

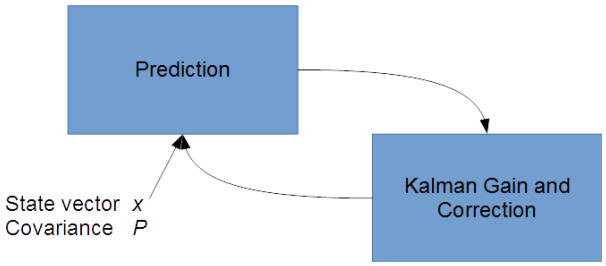


Figure 4.4: The Kalman filter loop consists of the propagation and update blocks which affect the state and covariance matrices.

The noise is Gaussian. Kalman filter state can be initialised with the best guess at time t_0 and the covariance with the system noise Q. The prediction equations of a Kalman filter are

$$X_{k|k-1} = A_k X_{k-1|k-1} + B_k U_{k-1|k-1} \tag{4.9}$$

and

$$P_{k|k+1} = A_k P_{k-1|k-1} A_k^T + Q_k \tag{4.10}$$

where X is the state, A is the state propagation model, B is the model to integrate previous external input U into the new state, P is the error covariance and Q is the propagation model noise.

The correction equations are

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{4.11}$$

$$X_{k|k} = X_{k|k-1} + K_k(Z_k - H_k X_{k|k-1}) \tag{4.12}$$

and

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}) \tag{4.13}$$

where K is the Kalman gain, H is the measurement model, R is the measurement noise and Z are the measurements. The measurement Z and the predicted state HX difference in the equation 4.12 is called as "the innovation process". This innovation is important for characterising the filter data association confidence.

Extended Kalman Filter: For being able to cope with weakly nonlinear processes the linear Kalman filter is modified by allowing the propagation (A) and measurement (H) models to be nonlinear in the Extended Kalman Filter. In the EKF the nonlinearity is approximated by a first order Taylor expansion, meaning that the A and H are Jacobian matrices. These are not usually constant and must be recalculated at every iteration. This adds computation load. Haug (2012) mentions that a second order EKF can be derived by computation of the Hessian matrix. The EKF prediction equations then are

$$X_{k|k-1} = f(X_{k-1|k-1}, U_{k-1|k-1}) \tag{4.14}$$

$$P_{k|k+1} = A_k P_{k-1|k-1} A_k^T + Q_k \tag{4.15}$$

where f represents the nonlinear function of state and control. The correction equations are

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{4.16}$$

$$X_{k|k} = X_{k|k-1} + K_k (Z_k - h(X_{k|k-1})) \tag{4.17}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}) \tag{4.18}$$

where h is the nonlinear measurement function.

Unscented Kalman Filter: Candy (2008) explains that in UKF a set of sample points (sigma-points) are selected. The unscented transformation approximates the posterior mean and covariance through the filter using these sample points. These sigma-points represent the mean and covariance that are propagated through the nonlinear process and measurement models. The nonlinearity is proven to be captured in the posterior mean and covariance to the 3rd order Taylor series expansion. Unlike the EKF, UKF does not require the derivatives of the propagation nor the measurement model.

The state vector prediction equation is

$$X_{k|k-1} = \Sigma w f(\chi_{k-1|k-1}) \tag{4.19}$$

where w is weight and χ are the sigma-points. And the state vector covariance prediction is

$$P_{k|k-1} = \Sigma w[f(\chi_{k-1|k-1}) - X_{k|k-1}][f(\chi_{k-1|k-1}) - X_{k|k-1}]^T + Q \tag{4.20}$$

where Q is the process noise. The observation equation is

$$X_{k|k} = X_{k|k-1} + K_k(Z_k - \Sigma w h(\chi_{k|k-1})) \tag{4.21}$$

where

$$\begin{gathered} K = P_{xz}P_{zz}^{-1}, \\ Pzz = \Sigma w[h(\chi_{k|k-1})(1-w)][h(\chi_{k|k-1})(1-w)]^T + R, \\ Pxz = \Sigma w[f(\chi_{k-1|k-1})(1-w)][h(\chi_{k|k-1})(1-w)]^T \end{gathered} \tag{4.22}$$

and the covariance is

$$P_{k|k} = P_{k|k-1} + K_k P_{zz} K_k^T \tag{4.23}$$

The Sigma-point filters suffer from a “curse of dimensionality”. This means that

the integration points increase exponentially with the dimension of the state vector. Briers et al. (2003) brings up the topic of choosing the number of sigma points as an ongoing debate. Further, Julier & Uhlmann (2002) defines the minimum amount of sigma points to be $n + 2$ for n-dimensional space. In central difference Kalman filter the sigma-points are redrawn or resampled twice for time update and the measurement update (Hao et al. (2010), Figueiras & Frattasi (2010), Candy (2008).

Kalman Filter and variants: Koch (2014) mentions that the prediction and correction steps can be linear or nonlinear. Kalman filter is flexible enough to allow the most efficient models to be used for the current estimation task at hand. Sensor data from multiple sensors can be fused together in the correction step. This is a centralised approach.

Another two alternative ways are to correct all sensor measurements separately or first fuse the results from the separate sensor measurements, described by Mitchell (2007), Haug (2012), Koch (2014). The two methods are called the measurement fusion and track to track fusion respectively. Bar-Shalom-Campo fusion algorithm can be used to fuse tracks that correlate. A few Kalman Filter variants are listed here that are described further for example by Mitchell (2007). Robust KF, Kriged KF, Augmented KF, Switching KF, Mixture KF, Spherical Simplex KF, Gauss-Hermite KF, Monte Carlo KF and Ensemble KF which all vary in implementation details.

Data Association: Mitchell (2007), Chui & Chen (2009), Candy (2008), Koch

(2014) examine how to handle false measurements, for which there are various methods. Not all the measurements are correct. Hard data association although assumes this. In soft data association, multiple measurements are analysed. The nearest or strongest of these measurements can be picked or multiple hypothesis weighting can be used to select good quality measurements. Kalman gain itself adjusts the weight between the prediction and measurements.

Hard data association assumes that every measurement that comes to the system is true. In soft approach, multiple measurements are examined. Nearest neighbour assumes that the closest of multiple measurements to the predicted value is the true measurement. Strongest neighbour takes the strongest measurement. In probabilistic data association, each measurement is given a weight and multiple updated state predictions are then combined into one. In multiple hypothesis tracking/testing all hard measurements are calculated and the least probable are then discarded. For multi-target tracking a separate Kalman filter for each target can be used.

The frequency of new measurements is connected to convergence. With higher frequency, the trace of covariance P becomes lower. The trace of P gives confidence for the estimator. Certain threshold value can be defined for converging estimator. If the values of P stay below the threshold the received measurements can be said to be consistent.

Error ellipses of the propagated covariance matrices can be used to represent the performance of the tracker. Root mean squared errors are another method in comparing performance. Cramer Rao lower bound can be used to show the minimum

achievable RMS error and thus show also the performance of the filter as described by Jing (2013).

Luo et al. (2011) states that covariance union and intersection are methods for fusing multiple sensor information that are not independent. Covariance methods combine the means and covariances of the multiple sensor estimates.

4.5 Particle Filters

Doucet et al. (2001) describes that in Sequential Monte Carlo (SMC) methods recursive sets of Monte Carlo points and weights are generated. SMC methods can be used for modelling nonlinear and non-Gaussian processes. The particle filter implements the Bayesian framework using a Monte Carlo approach. This is similar to the Sigma-point Kalman filter except usually there are many more particles in particle filters. In addition, sigma-point filter particles are deterministically selected. Particle filters are suitable for systems having not only Gaussian noise but also any other kind of distribution of noise.

The sequential importance sampling (SIS) defines the core of the particle filter. The filter is initialised by generating samples and weights. The samples are drawn first from a distribution of known probability density. Amount k samples X_k that belong to a prior belief distribution $Bel(X_k)$ are propagated through the system model. Measurement likelihoods can be used to obtain weights. The samples' weights are recalculated and normalised. The state vector mean and covariance can

be calculated from the progressed particles.

The bootstrap particle filter can be considered as the basic particle filter. Here the particle set is resampled so that the weaker weighted particles die and the stronger ones are duplicated. Jing (2013) lists different resampling strategies that have been implemented. Resampling is usually made using the number of effective samples still alive. If the sequential measurements do not contain characteristic differences for location estimation the particles may diverge. The regularisation can be made after resampling. This is to prevent sample impoverishment. Often the stronger duplicates concentrate on certain values and are repeated. Regularisation moves the particles slightly separate in a random fashion so that the sample impoverishment is prevented. In an optimal SIS particle filter care is taken in choosing the importance density that is the weights. The optimal choice is to make it to minimise the variance of weights.

Siciliano & Khatib (2008) views the particle filter approach to be well suited for deriving multi-modal or multiple hypothesis density functions. Here if the state vector is large then the number of samples required is large as well. The computation becomes lighter if some of the states can be modelled without sampling and instead by using a Kalman filter. This is called as Rao-Blackwellisation according to Haug (2012). Figure 4.5 depicts the particle filter pseudocode.

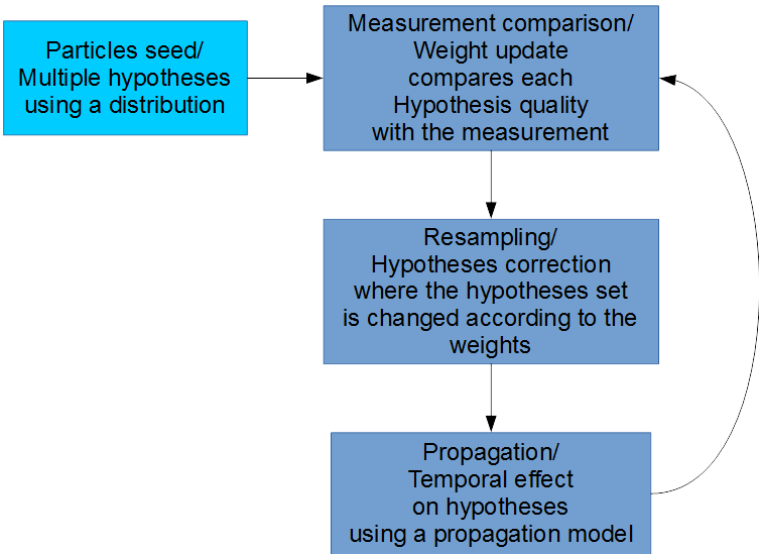


Figure 4.5: Particle filter loop structure.

The following paragraphs describe the particle filter phases.

Initialisation: N particles x_k^i, where $i = 1, .., N$, are created from a prior proposal distribution $p(x_k|x_{0:k-1}, z_{0:k})$ to represent the system state. Particles consist of the state and in addition of the associated weights w_k^i.

Prediction: The state x_k is propagated using a prediction model. This is derived using the Chapman-Kolmogorov equation as described by Arulampalam et al. (2002).

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \tag{4.24}$$

Here $p(x_{k-1}|z_{1:k-1})$ is the a posterior from the previous epoch. $p(x_k|x_{k-1})$ is the state propagation model including the process noise v_{k-1}.

Update: The particle weights are updated using the latest measurement. The weight update is the following according to Arulampalam et al. (2002).

$$w_k^i = w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{0:k-1}^i, z_{1:k})} \tag{4.25}$$

If only the filtered estimate from the previous epoch is needed then the equation becomes

$$w_k^i = w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)}. \tag{4.26}$$

Optimal importance density function, q, has been found to be the following (Arulampalam et al. (2002)).

$$q(x_k|x_{k-1}^i, z_k)_{opt} = p(x_k|x_{k-1}^i, z_k) \tag{4.27}$$

Then the new weight of each particle is proportional to

$$w_k^i = w_{k-1}^i p(z_k|x_{k-1}^i). \tag{4.28}$$

The weight of each particle is then adjusted by normalising so that

$$\sum_i w_k^i = 1. \tag{4.29}$$

Resample: Different resampling strategies have been listed by Hol et al. (2006). The four most common are named as Multinomial, Stratified, Systematic and Residual resampling. Systematic resampling is stated as the most friendly solution in terms of computation and quality.

Resampling is often done after an effective sample size N_{eff} is below a threshold number. Here too few particles have high enough weight for the rest of the particles to be considered having an effect in the final solution. These particles are "killed" and a new set of particles are generated according to the distribution defined by the remaining higher weight particles.

Epoch solution: The resulting prior probability density function (pdf) from the propagation step is updated with the latest measurement z_k. Bayes rule is used to obtain the posterior state.

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{4.30}$$

This posterior distribution can be approximated using the empirical Dirac measure according to Arulampalam et al. (2002).

$$p(x_k|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i) \tag{4.31}$$

Measurement model and measurement noise define the conditional pdf of z_k.

$$p(z_k|x_k) = \int \delta(z_k - h_k(x_k, n_k))p(n_k)dn_k \tag{4.32}$$

Degeneracy is an unwanted drawback in particle filters. This means that only a few particles have high weight while most particle weights are below an effective threshold. Resampling compensates degeneracy with regularisation by generating and perturbing particles so that the set is more favourably spread and the weights are reweighted appropriately. Also by using suitable importance density function helps as was mentioned in the update stage of the example particle filter procedure.

Impoverishment of particles means that same or nearly same particle copies exist. This happens especially when the process noise is small. After few iterations only a couple repeated points remain and the particle cloud loses diversity. Arulampalam et al. (2002) and Doucet & Johansen (2011) list methods for preventing the loss of diversity. Markov Chain Monte Carlo approach and resample-move, regularization and block resample methods are mentioned that aid in preventing the loss of diversity. In a Markov Chain quality can improve step after step.

Additional aids: Logical deduction of higher level abstractions is typical for inference processes. Fuzzy logic is an example of this type of a process. In fuzzy logic, the output variable is divided into weighted regions. The combinations of weights and different input variables define the inference output. Fuzzy logic or Dempster Schafer theory, Liu et al. (2012), can be used in describing the uncertainty. In fuzzy logic systems, the parameter values can be divided into sets. Using these sets a

logical inference can be conducted.

Liu et al. (2012) and Siciliano & Khatib (2008) introduce Dempster-Schafer theory that further extends the manipulation of these sets or beliefs into a more complex belief choice ranges. The Dempster Schafer is considered a robust alternative for noisy or uncertain measurement and prior data processes. Instead of a probability distribution the Dempster Schafer theory represents a degree of belief as a belief function in the context of the transferable belief model. This increases complexity of the system. Multiple model estimates can be implemented simultaneously. This is common in tracking moving objects.

When the space is discrete and finite a grid can be constructed. The grids can be hierarchical or irregular like triangular shaped grids. Siciliano & Khatib (2008) explains that the grid cell or node can be labelled with a property, for example with temperature at that location. Sensor grid models can be constructed for updating the grid with the sensor grid using for example Bayes' law. In hidden Markov model or the grid mass filter the grid points states are recalculated for all cells.

Precalibration is done before the positioning stage. This means that true positions provided by an alternative reliable system are compared with the positioning system estimations that are to be used. These two are compared and the system parameters are adjusted to be optimal. Fingerprinting is an example of a precalibrated system. Signal strengths are measured and heat maps are constructed for use in the positioning stage. The dynamic stability of the heat map can vary over time which is a challenge in fingerprinting systems. Trends or seasonal variations in the

data could be extracted using time series techniques. Figueiras & Frattasi (2010) describes that in a joint state and parameter estimation, both, the calibration and the tracking can be conducted simultaneously. The parametres of the system are usually considered independent of time or can gradually change.

4.6 Fusion Architectures

Both the Kalman and the particle filters can be used in fusing data from multiple sources via the update method. Whether the source will be used in propagation of the system state or in the update stage depends on the agreement made between all the involved sensors. The abstraction level of the data that will be fused and the procedural location in which different data can be fused are examined next.

4.6.1 Main Architectures

Mitchell (2007) offers three ways of multisensor fusion; centralised, decentralised and hierarchical. Groves (2013*a*) divides integration architectures into four main types. These are the cascaded and the centralized single-epoch and the cascaded and the centralised filtered architectures. In the single-epoch architecture, the information from the previous epoch might not necessarily be used. In the cascaded architecture, separate navigation solutions (position, velocity, time) are output from sensor subsystems and the fusion is done using these separate solutions. This is usually weighted least squares solution. Here the sensor subsystems should provide the

solution with error covariance information. If error information is not provided it should be estimated. Furthermore, in a decentralised, or cascaded, system, there are multiple local fusion stations. This reduces the capacity requirements of a single fusion node, but the redundancy and control of the information flow are more challenging to implement.

In a centralised system, all sensor data are input to the central fusion node. This option demands a lot of capacity of the fusion processing node. In centralised architecture, instead of navigation solutions, raw measurements are fed into the fusion filter with measurement error estimates. The loose coupling is often cascaded while the tight coupling is using centralised architecture. We use the loose coupling, where the subsystems provide position information, which is fused with the PDR. Hybrid solutions are also possible, or in other words, hierarchical fusion is a mixture of the centralised and decentralised architectures.

4.6.2 Propagation and Observation Models for Closed Loop 15 Error State Kalman Filter

Figure 4.6 depicts the closed loop Kalman filter.

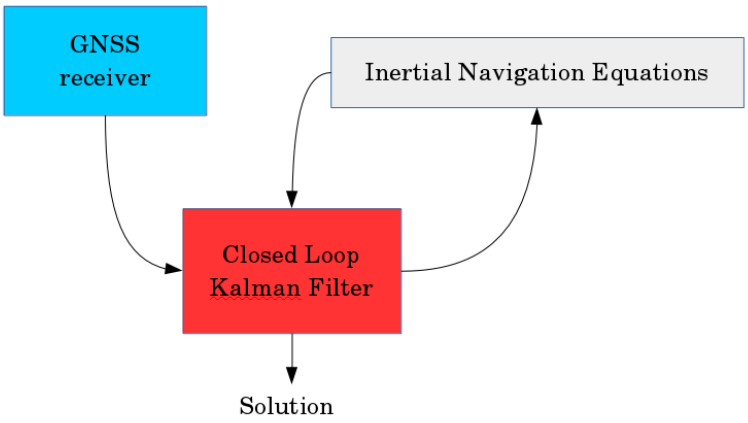


Figure 4.6: Closed Loop Kalman Filter for integrating inertial sensor data with systems outputting fixed positions like GNSS.

The 15 error states in the navigation frame n are

$$\delta x = (\delta\psi_n \, \delta v_n \, \delta r_n \, b_a \, b_g). \tag{4.33}$$

Here the variables are attitude, velocity and position errors and accelerometer and gyroscope bias errors respectively.

Other inertial sensor errors like scale factor error, cross-coupling error and gravity dependent error can be taken into account by increasing the accelerometer and gyro process noise in the Kalman filter. Thus, measurement noise will have more impact on the Kalman loop.

When consumer grade inertial sensors are used the error propagation matrix according to Groves (2013*a*) is

$$\phi = \begin{bmatrix} I_3 & 0_3 & 0_3 & 0_3 & C_b^n\tau \\ -(C_b^n f_{ib}^b)\tau & I_3 & -\frac{2g_0(L_b)}{r_e(L_b)} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tau & C_b^n\tau & 0_3 \\ 0_3 & \begin{bmatrix} \frac{1}{(R_N(L_b)+h)} & 0 & 0 \\ 0 & \frac{1}{(R_E(L_b)+h)cosL_b} & 0 \\ 0 & 0 & -1 \end{bmatrix} \tau & I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & I_3 \end{bmatrix} \tag{4.34}$$

Here C_b^n is the attitude conversion matrix from the body to the navigation frame. This conversion matrix defines the attitude of the object. It is the rotation matrix found in equations 3.1 to 3.5. τ is the sampling time, f_{ib}^b is the accelerometer measurement vector, r_e is the earth radius at the current latitude and R_N and R_E are the earth radius components at the latitude L_b.

By constructing a measurement matrix any position, velocity or attitude sensors can be used to update the inertial errors of the error state Kalman filter. This update is also called dynamic alignment. The measurement matrix can include from one type of measurement up to all the three measurements together.

$$v_k = z_k - h(k)\delta x(k) \approx z_k - H_k \delta x_k = z_k^{observed} - z_k^{predicted} \tag{4.35}$$

The forced zero velocity update measurement vector for example is

$$z_k = [0\,0\,0]. \tag{4.36}$$

And this is combined with a velocity measurement covariance that is put to be very trustworthy. This means that the measurement has a very small error.

$$R_k = \begin{bmatrix} 0.1^2 & 0 & 0 \\ 0 & 0.1^2 & 0 \\ 0 & 0 & 0.1^2 \end{bmatrix}. \tag{4.37}$$

The measurement update matrix includes the velocity error component of the state and is the following according to Groves (2013*a*).

$$H_k = \begin{bmatrix} 0_3 & -I_3 & 0_3 & 0_3 & 0_3 \end{bmatrix}. \tag{4.38}$$

4.6.3 Sensor Data Abstraction and Heterogeneity

Multi-sensor fusion strategy depends heavily on the application. It is challenging to evaluate the wide scale of sensor technologies and integration methods, because it is usually the application that determines the desired objectivity. Rein & Biermann (2013) depicts that the application defines the "world views" for the filter in

handling the different sensor data. Snidaro et al. (2013) agrees that this processing of context is slightly similar to integrity monitoring. It is done in parallel with the actual navigation solution extraction. It is possible that similar sensors use the same interfaces on higher abstraction levels. Figure 4.7 depicts the abstraction levels as the vertical axis and sensor heterogeneity in the horizontal axis.

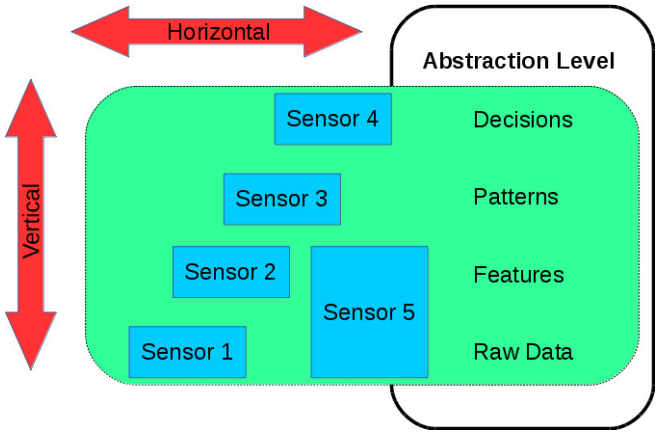


Figure 4.7: Sensor measurement information abstraction levels.

Snidaro et al. (2013) and Rein & Biermann (2013) discuss the abstraction level of the sensor data. Mitchell (2007) divides the sensory data into cognitive, informative and physical. The physical is the sensor data, the informative is the encoded data used in software and cognitive is human inference based logical data. The raw sensor data is called hard data and its abstraction level is low. In contrast, the soft data means high abstraction. Data becomes information after it has been analysed. The filter should be able to produce information using the different abstraction level input data. The challenge is in how to handle or fuse the heterogeneous data provided by multiple types of sensors and subsystems.

Soloviev (2014) agrees that processes usually benefit from the combination of heterogeneous data, although the complexity increases if there is too much data. Data integrity monitoring detects changes in the data, tries to offer correction control and limits the amount of outlier data used in producing relevant information. Along with complexity, Groves et al. (2014) lists also context, ambiguity and environment as the challenges in advanced multi-sensor navigation and positioning. It is hard to achieve high integrity and availability simultaneously.

4.6.4 Alternative Architectures

Luo et al. (2011) and Esteban et al. (2005) lists advantages of multisensor systems. These include higher signal to noise ratio, increased robustness, improved resolution and reduced uncertainty. Also 7 strategies for data fusion systems are listed. The most well-known is the JDL (Joint Directories of Laboratories) process model.

JDL divides the data fusion into three levels. These are the object refinement, situation assessment and threat assessment or inference. These go in accordance with the previously mentioned abstraction levels from lower abstraction to higher. Figure 4.8 depicts the abstraction levels in JDL.

There is also a feedback loop for internal process refinement.

Thermopoulos architecture, Luo et al. (2011), is another model with three different ways for the data integration. In signal level, the fusion in Thermopoulos model is executed using correlation and learning methods, while in dynamics level a mathematical model is applied. Evidence level fusion uses statistical models.

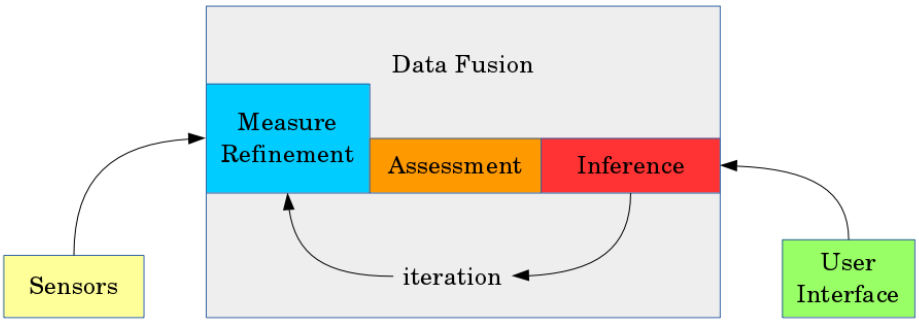


Figure 4.8: Joint Directories of Laboratories model.

Multi-sensor integration fusion model defines the terms "multi-sensor integration" as task-oriented and "fusion" as the actual data combination process. Here multisensor fusion is executed in fusion centres and the centres are used in the final integration output.

A behavioural knowledge-based data fusion model clarifies the derivation process from raw data features into higher level formalism. Again, raw data feature extraction is followed by association with known patterns like in the JDL model. In the highest level, the system behaviour is analysed and represented.

The waterfall model describes yet a similar process of low level data derivation into higher level symbols. A slightly different idea is presented in the distributed blackboard model, Luo et al. (2011). Here different parallel sensors that operate with multiple transducers use the same shared memory and algorithm. The sensor supervisors act as integrity controllers for the complete system together with the shared algorithm. System output will be the more stable of the two in the example

Figure 4.9.

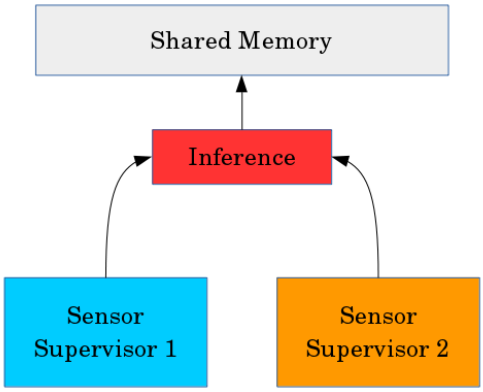


Figure 4.9: Blackboard model where the more stable sensor is selected.

The omnibus model, Luo et al. (2011), consists of a loop with the sequence of observation, orientation, decision and an action. An action could mean moving a robot arm or some other hardware according to the sensor measurements.

Choosing the right architecture for the application is a key aspect in designing a working multi-sensor navigation system. All the abstraction levels should be identified and listed out before starting to design a system. This is especially true for adaptive systems which should be exactly defined also in the higher levels of abstraction.

Diving lower in the abstraction levels of the vertical sensor data spectrum we are faced by the actual data fusion method choices in the different fusion levels. Estimation methods are used more for the raw data. These include Kalman filter variants and covariance based estimation strategies. Classification methods are used more for data in the middle of the vertical abstraction and inference methods for high

level data. Support vector machines etc. are used in the medium level and particle filters within the high-level fusion methods. In high abstraction level, the context is closer to human language. Context is thus often defined by weighing uncertainties. The low-level sensor data (e.g. accelerometer) has to be interpreted correctly for determining the exact context (pedestrian/car/submarine/indoor/outdoor/space). There are two domains affecting the context. One is the environment and the other one is the host vehicle or pedestrian dynamics. Signal availability, context and integrity should be determined correctly by the positioning system by excluding the forbidden wrong data.

According to the JDL model the three levels can also be called as data, feature or decision fusion levels. The dominant uncertainty should be identified. The greatest concern in designing a system is the fusion accuracy. This comes often with a trade-off between computational speed. Achieving the best performance demands knowledge in control theory, signal processing, artificial intelligence, in probability and statistics.

4.6.5 Additional Methods and Summary

Another aspect for determining the context as well as the location is the help from other peers. Sharing data and relative position between adjacent users can enhance the accuracy. Mutual ranging and location information gained from the peer that knows his position helps others deriving theirs.

Groves et al. (2014) states that this exchange can be done through pre-loading or streaming and through a central server or from peer to peer. The nodes in these schemes should all be cooperative. Cooperation can take place in signal processing, measurement or estimation levels of the navigation systems. Yang & Soloviev (2014) describe vehicles first using their own initial estimates and then updating these with the mutual inter-nodal measurements. There exists also a critical size for the node count after which more information does not enhance the result. It is then a choice between bandwidth and node count to be accounted in the same inter-node calculation. Hao (2015) uses inertial dead reckoning with WiFi fingerprinting and cooperation between nodes. It was noted that cooperation reduces the dependency on WiFi hardware. The number of users and integrity of the cooperative data will have to be further investigated in this scheme. All cooperation algorithms should also be designed in a plug-and-play style.

The plug-and-play approach applies for the integration methods also. There is a clear advantage to have reusable method blocks in different levels of abstraction in the integration process. The application designer would then be able to organise these blocks optimally for the specific application.

The common goal for the research community is to aim for a plug and play modular navigation framework, where the communication between different partakers is flexible. Usually one manufacturer is responsible of one type of sensor technology. The manufacturer interests and subsystem designers need standards for information exchange. The optimal extraction and integration of the data from various sensors

are the two main questions. Adaptability and the plug and play idea should be considered. This means that the filter can use the same type of sensors independent of the manufacturer and that it will use the best integration method in the right situation at the right time.

4.7 Conclusions

The tightly coupled architecture is heavily centralised and all the processing concentrates on the same single unit. In this respect, the loose coupling serves better for modular plug-and-play design.

On the other hand, the accuracy of tight coupling is better since the made approximations in the sensor subsystems using the cascaded design are not involved in the centralised fusion. Thus, the accuracy, complexity and efficiency triangle should be examined. But for this study the complexity should be kept lower because of the lack of resources. The tight integration involves more complex design and as a first novel context sensitive seamless navigation system design it is practical to start with the loosely coupled scenario.

The configurational flexibility needs to meet the needs for both indoor and outdoor navigation. This means that we need to have the sensor infrastructure set up for both. This defines the availability. The available platforms need to be evaluated and the most sensible one chosen. The efficiency and complexity of the algorithms need to be evaluated for the chosen application scenario.

The environmental adaptability is enabled by the context engine design. Reading the sensors and inferring the correct context will enable more efficient navigation solution derivation for the current situation. The sensor interfaces need to be designed for the navigation and the context inference purposes. In addition, user interruptions should be handled.

The classification techniques tend to demand more storing space, which the mobile devices try to save. The Kalman filter is more efficient than the particle filter, although for pedestrian application the particle filter is the more robust option. Moreover, the sensor data abstraction levels and formatting for the navigation solution and context inference demand separate data association processing blocks.

These questions listed here above will be answered in the remaining chapters. Design choices, in creating the first novel seamless navigation system, are evaluated in their respective chapters.

Chapter 5

Reciprocity in Navigation Systems

This chapter introduces the concept of filter switching from particle to Kalman filter and vice versa. The particle filter is more robust and the Kalman filter is faster. Depending on the situation either is preferred. The first section answers the question of how the switch is initiated. Second, the common parameters that need to be transferred are discussed. And third, the methods are presented for transferring these parameters.

The previous chapters presented the existing relevant literature. The reciprocal procedures between different filter types, on the other hand, have not been examined extensively previously, to the extent of our knowledge. Reciprocal procedures are defined here to mean the handling of equivalence of attributes between, for example, Kalman and Particle filters. Rao-Blackwellisation reflects a similar idea, but it is more of a mixture of the properties of one entity rather than an equivalence inspection. The few studies found are listed here, and a model is created in this

chapter, for the purpose of this study.

A particle filter and a Kalman filter are chosen for the first implementation of the seamless adaptive navigation system. The Kalman filter is chosen for the computational speed and the particle filter because of its robust nature. Dean et al. (2010) first tests the "Gaussianness" of the particle filter distribution and then initialises an unscented Kalman filter. A very good study on filter replacement, according to the context, is shown by Zaugg et al. (2003). Here, a particle filter is initialised when the Kalman filter solution can no longer function by itself. Further, Won et al. (2015) used both the extended Kalman and the particle filters in parallel for a docking application. Raitoharju et al. (2015) examined the reinitialisation of a particle filter using a Kalman filter. Our case study combines these properties in a novel design for seamless pedestrian navigation.

The term "Reciprocity" means here the methods, the equivalent attributes and the characteristics when replacing or swapping a one type of navigation filter into another. And the novel design approach assumes this switch being executed in real-time. In later chapters, parallel processing or multiple model inference (MMI) -style filtering is explored, where parallel hypothesis run concurrently and the best one is inferred by a context engine.

The term "Transfer inference" describes the general conditions for different filters to be switched into another. An example is a unimodal Kalman filter encountering an obstruction where the more robust particle filter would easily survive. Moreover, the common parameters are derived and listed. The parameter initialisation is dis-

cussed for both filters, the Kalman and the particle. The transfer characteristics are then discussed and what to take into account when switching between the estimation methods for navigation purposes.

A few helpful terms are listed here that are repeated in the chapter, which help to understand the system design presented in the following chapters.

Direction of Movement is taken as the direction indicated by two points in the local tangent plane. The first point is the start position of the step and the second position is the end of the step. This is different to the yaw angle, although the error for these two variables, yaw and direction of movement, both describe an error in the xy-plane. This is an angle error in the local tangent plane. **yaw angle** is the heading angle, the body-frame x-axis of the foot mounted inertial unit.

Tripartite filter is a term used for the three parallel filters used in the final design. The inertial-only mechanisation, using a Kalman filter, is running in all test runs. Additionally, another Kalman filter or a particle filter is initialised for multi-sensor fusion purposes.

5.1 Transfer Inference

The transfer inference section defines the conditions that are used throughout this thesis for initiating a parallel filter or switching completely from one filter to another. Going through the wide range of existing filters would take a long time. This is why we concentrate on the two most common and basic filters, the particle filter and the

Kalman filter.

First, a reason is needed for a switch between Kalman and Particle filters that are being used for sensor fusion. This is very much application specific like the example of target tracking that Zaugg et al. (2003) describes. A need for a switch is a necessity. This is called a trigger. When considering pedestrian applications, the particle filter is more robust when the environment consists of many walls. The "wrong particle estimates" can be propagated until the correct particles either survive in the wall-rich area or can be inferred to be the correct particles.

Zaugg et al. (2003) used a four state filtering scenario. Either the particle filter or the Kalman filter were used alone or then both the Kalman filter and the particle filter in parallel or two parallel slightly different particle filters. In the tests made for this thesis, and for simplicity, the two basic filters are used separately for sensor fusion. These are the particle filter and the 15-error-state Kalman filter estimating the user position and heading and the errors.

5.1.1 Trigger for a transfer from Kalman filter to Particle filter

A simple condition for this switch is used in our study. If the Kalman filter solution bumps into a wall, then the control is switched for particle filter. Depending on the Kalman filter solution map matching result, the failed iteration step is retaken. The Kalman filter step is discarded. A particle filter is instead initialised at the previous

Kalman filter solution and the failed step retaken using the particle filter instead.

5.1.2 Trigger for a transfer from Particle filter to Kalman filter

The particle filter can represent a non-unimodal solution. In order to initialise a Kalman filter from the non-unimodal particle filter solution, an increase in uncertainty is nearly unavoidable.

If the position updates exist and they are very accurate, the survived particles tend to focus and to be resampled on a smaller area. This means that the deviation of each particle from the estimated solution (weighted average position), will be short. Also, the particle scatter is then small and indicates an unimodal solution. A threshold value was used for the standard deviation of particles in the particle cloud.

This threshold is environment dependent. Mainly, this is because different positioning fixing systems are used. In indoor environments, the dimensions, like passageway width or corridors, are smaller and thus the threshold is shorter as well. Different threshold values were used for indoors and outdoors. These values will be examined further in the presentation of the whole context engine design.

5.2 Common Parameters

An inertial-only Kalman filter solution is run constantly in our design. This is to have a clear option to use and analyse the inertial only track. For application development phase of the designed and implemented system this inertial only solution could be an excess load, but for the first version and purposes of analysis it is essential. This gives us the estimated mechanisation result that is the dead reckoning position, velocity and attitude changes. Alongside, either an additional Kalman filter or a particle filter for fusing the positioning fixing system information is used. In this section, the equivalent parameters for the state and covariance representation in both the particle and Kalman filters are examined.

Position: The Kalman filter is a 15-error-state Kalman filter where we have the position error as a state. This position error is compensated every epoch using closed loop feedback. This means that the error is reset to 0 every epoch. Thus, a separate position solution is propagated alongside the error state Kalman filter. In a similar manner, an accumulated error in northing, the x, and easting, the y, directions is followed. Each epoch, the Kalman filter resets its error to 0 and thus we take the deviation, each epoch, from the covariance matrix. This indicates the epochwise deviation of the position error estimate. The position error is tracked in the Kalman filter. The error estimate of the error state Kalman filter is typically too optimistic. This was noticed in the measurement campaign, when comparing the Kalman behaviour with the Leica reference system. Alongisde, the accumulating

error is estimated. This was noticed to describe an epochwise accumulation of error that is closer to the true increase of the error. Thus the epochwise error value is accumulatively summed using the consequent deviation values of the covariance matrix as an additional less optimistic position error estimate.

In the particle filter design we have the 2D position in x- and y- directions as the particle states. This position is proportional to the three-dimensional position solution propagated alongside in the sensor fusion Kalman filter. The position error in the particle filter is described by the particle position scatter. This particle scatter, or when describing in more detail, the deviation in both axis from the weighted average position is proportional to the x- and y- axis error estimates in the sensor fusion Kalman filter. Although, the Kalman filter error estimates are too optimistic and the additional accumulated error estimates are used instead. Figure 5.1 depicts the position parametres in question. In this thesis the x-axis corresponds to the northing, y-axis to the easting and z-axis to the down in the NED frame.

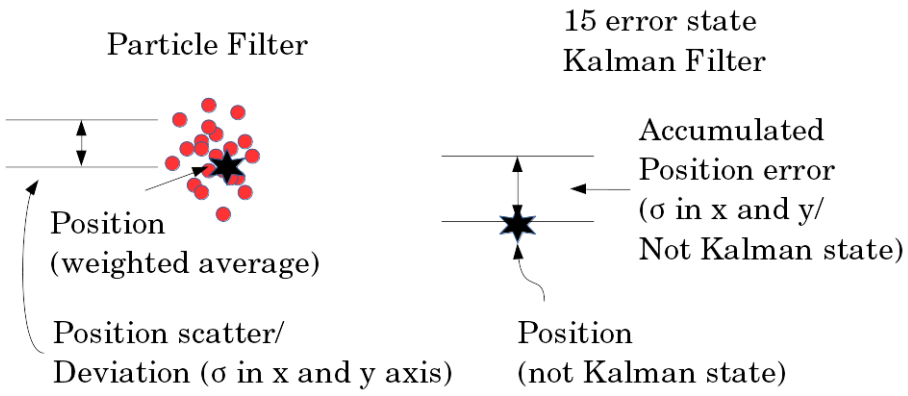


Figure 5.1: Particle filter position is the weighted average of particles. This corresponds to position archived alongside Kalman filter. The accumulated position error alongside the error state Kalman filter corresponds to the particle filter particle scatter.

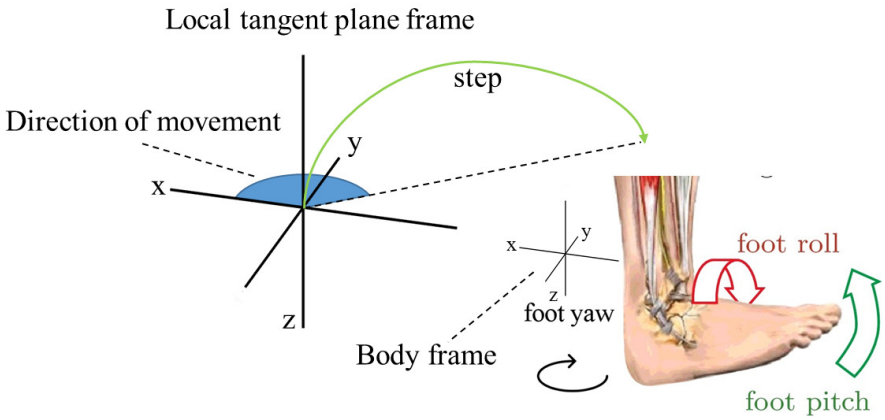


Figure 5.2: The direction of movement is the angle of one step. The yaw angle is the current heading of the foot.

Yaw and Direction of Movement: In addition, the attitude error estimate is estimated in the Kalman filter of which the yaw angle error is of interest to us regarding the particle filter design.

In a similar manner to the accumulating position error, the accumulating yaw angle error is followed alongside the Kalman filter yaw angle error. Each epoch the yaw angle error is compensated in the closed loop Kalman filter. Thus, again the deviation from the covariance matrix P is taken and this, epochwise error estimate, is accumulated into the additional accumulated yaw angle error estimate.

The direction of movement is the third particle attribute that is used in the particle filter. The Figure 5.2 shows the difference between the foot yaw and the direction of movement. The direction of movement is defined to be the angle of a line in the local tangent plane formed by two positions that are the beginning of the step and the end position of the latest step. Each particle direction of movement value is an individual estimate of the true direction of movement. By definition,

the direction of movement is different to the yaw angle. This is clearly understood if considering sidestepping or strafing user. The yaw of the foot remains in one direction, while the direction of movement may change. But the errors tracked in both filters follow the same one dimensional error estimate value. This is the user attitude error in the xy-plane. In a particle filter, this consists of multiple direction of movement guesses and the deviation within these values. For a Kalman filter this is the yaw error or the additional accumulating yaw error estimate that is not too optimistic.

Since the inertial-only solution is run concurrently, we can use this information when we convert the additional Kalman filter into a particle filter and vice versa. For each step we know the relationship that is between the direction of movement and the yaw angle.

Proportionality: The covariance of the Kalman filter represents the estimated magnitude of the position and yaw error variances. Taking a square root, we have the deviations of the errors in all the x, y and z -directions, of which we consider using only the x and y for a planar position solution. The additional accumulated error deviation that is followed, is the sum of the subsequent error covariance values rooted. This is written for the x-axis component of the position in the following manner. The additional accumulated deviation estimate using the consequent Kalman filter error covariance values is defined here to be proportional to the standard deviation of the particles from their weighted mean value.

$$r_{x(t)} \sim r_{x(t-1)} + \sqrt{P_{pos,x}} \sim \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)^2} \tag{5.1}$$

Here r_x is the accumulated position error along the x-axis (or the North using NED) at time t and μ_x is the weighted mean value of the particles' position in the x-direction. N is the particles amount and x_i is the position of the particle in the x-axis. $P_{pos,x}$ is the x (or the North using NED) component of the position variance. The level of proportionality is higher when the Particle filter follows a Gaussian distribution. In pedestrian applications, it is not. The loss of information has to be accepted if we are to convert from Particle filter representation into a Kalman filter approximation. Same applies for the y-axis.

If a person walks straight ahead and particles are propagated at every step, the distribution that the particle cloud depicts is like that exaggerated in Figure 5.3.



Figure 5.3: Typical Particle filter distribution profile for pedestrian applications when the person walks in a straight line. Only IMU measurements with zero updates are applied in this figure.

When the pedestrian turns and the particles interact with the walls, the distri-

bution that the particles depict even further transform into a multimodal depiction of a distribution. For now, this phenomenon is simply accepted, which is the loss of information when converting particle filter into a Kalman filter.

Each particle guess consists of a position (x,y), direction of movement (angle) and weight values. For the particle filter the definition, "direction of movement", is the angle of the line of two subsequent positions during the last step, the position in the beginning of the step and at the end of step. For the multiple guesses of the direction of movement angle in the particle filter we can write proportionality with the Kalman filter yaw angle error. This is the user heading angle error in the local tangent plane. The yaw angle and the accumulating yaw angle error estimate are tracked slongside the Kalman filter. The covariance matrix presents the very optimistic angle error variance in the local tangent plane. The yaw angle error in the Kalman filter describes the equivalent attribute to the direction of movement uncertainty described by the deviation of multiple guesses in the particle filter. Both track the angle error characteristics in the xy-plane.

Alongside the very optimistic Kalman filter yaw angle error, the accumulating yaw angle error is estimated. The deviation of multiple guesses of the direction of movement is considered in the particle filter to describe the accumulating yaw angle error that is tracked outside of the Kalman filter as well as the Kalman filter yaw angle error when switching between filters. It has to be again emphasized here that the yaw of the foot is different to the direction of movement (particle filter variable). But we know the angle and the relation between the two variables,

because the inertial only solution is propagated in parallel. The angle error estimate is a one-dimensional estimate. Thus, this proportionality between the yaw error in the Kalman filter and the particle filter direction of movement error is used in the transfer process. In the particle filter the direction of movement error is described by the multiple independent guesses and by the deviation in these values. For the particle filter we need a circular mean of the particles' direction of movement angles. This can be derived by taking the mean of the x and the y axis -components of each particle's direction of movement angles according to the unit circle and deriving the arctan of the two difference components.

$$\alpha(t) \sim \alpha(t-1) + \sqrt{P_{yaw}} \sim \sqrt{\frac{1}{N}\sum_{i=1}^{N} atan2(y_i - \mu_y, x_i - \mu_x)^2} \tag{5.2}$$

Here the μ_y is the mean of y-components derived from the particles' angles using the unit circle. In this thesis the y-axis corresponds to the East axis in the NED frame. y_i is the unit circle component of each particle's direction of movement. The same applies for the x-axis. α is the accumulated error angle at time t. P_{yaw} is the angle error variance around the z-axis (Down axis in the NED representation). When we have the circular mean of the direction of movement, we can take the difference between the mean and each particle and derive the standard deviation of the particles' direction of movement.

5.2.1 Accumulated position and yaw error updates

The setup used for the trials throughout in this thesis is shown in Figure 5.4. The strapdown quality of the foot-mounted IMU has an effect on the accuracy of the dead reckoning. If too loose, the Kalman filter system propagation noise should be set higher to be able to compensate for the random errors due to loose attachment.

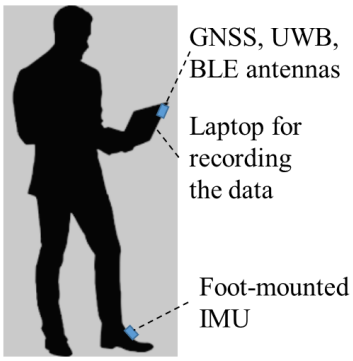


Figure 5.4: The sensor setup that is used throughout this thesis for recording.

The position and yaw errors increase if the Kalman filter advances using only inertial data. Generally, the position drift has exponential characteristics while the yaw drift is linear. The position drift is limited by the Zero velocity updates. But without positioning fixing system updates, and because of the double integration, the position drift is still more significant than the yaw drift. Both should be followed. The position updates can be used to limit and adjust the position error. The position error is adjusted using the system covariance (P) and the measurement covariance (R) matrices defined weighting between the propagated and measured error (update). Zero velocity updates can't limit the growing error in the yaw angle. In this thesis, the yaw angle error is limited by moderately weighted angle updates

using the consequent GNSS or UWB subsystem measurements.

In addition, the accumulated position and yaw errors are followed. These are followed alongside the Kalman filter. The position fixing system and the angle updates from the GNSS and UWB systems for the accumulated errors are executed using the covariance intersection method as explained by Julier & Uhlman (2009). This is demonstrated in the following equation. Again, the x, y and z -axis correspond to the N, E and D in the NED representation.

$$e_{x_accumulated(t)} = \frac{P_x}{R_x + P_x} e_{x_accumulated(t-1)} + \frac{R_x}{R_x + P_x} e_{x_meas} \tag{5.3}$$

$e_{x_accumulated}$ is the accumulated error in x-axis at time t. e_{x_meas} is the measurement error estimate in the x-axis. P_x is the variance and R_x is the measurement variance in the x-axis. Same applies for the y-axis (or the easting in the NED representation). This update of the accumulated error resembles the sample importance step in particle filter when updating the particle weights using a position update.

The positioning fixing antennas (GNSS, UWB, BLE) were attached to the laptop. The laptop moves according to the continuous upper body motion while inertial unit according to the foot. The gait phase planar lever arm estimation, this is estimating the xy-plane lever arm between GNSS antenna phase center and IMU origin, would increase the position update fusion accuracy, but this is left for future study.

For updating the accumulating yaw angle error magnitude, consequent positioning fixing measurements were used in a similar manner. This was applied only when

the user was moving (step length threshold). In addition, a Zero Angular Rate Update was applied at the same time with the Zero Velocity UPdates. During ZARU period the yaw angle error was not increased. Here we assume that the pedestrian walks so that the x-axis aligns mostly with the direction of the subsequent positioning fixes. This is a reasonable assumption, although we do not have an indicator or a measure to detect whether the pedestrian advances straight or whether the pedestrian starts sidestepping. This could be checked stepwise, by comparing the attitude solution yaw direction with the step direction indicated by the consequent position solutions. The yaw direction indicated by the consequent positioning fixing system positions is fused once in the Kalman filter update during the ZUPT period together with levelling the roll and pitch while the foot is stationary.

The used accumulated values are less optimal than the error variance within the Kalman filter and have not been thoroughly tested. But for the test scenarios in this thesis the error behaviour behaved more close to the real situation than the optimal error variance within the Kalman filter. This is left as future research.

5.3 Filter Transfer

Now that the triggers for the filter switching have been derived and also the parameters that need to be switched, we can continue to complete the transfer event both ways between particle and Kalman filter. In this thesis, the Kalman filter tracks the position errors and the position solution is updated outside of the Kalman filter

using this error estimation information. The initial information passed to the particle filter is contained within the additional accumulated position and yaw errors followed alongside the Kalman filter.

Particle filters on the other hand track the total state position. The scatter of the particles depict the position error. The weighted mean of the particles' positions is thus the equivalent to the position solution that is outside of the error state Kalman filter. The particle scatter properties then represent the same information that the Kalman filter error and error covariance hold and the additional accumulated estimates outside of the Kalman filter. The Kalman filter error estimate and covariance and the additional accumulated error describe error ellipses, while the particle filter scatter can represent a more complex multimodal distribution of error. Thus, when converting the information that the particle filter holds, some of this information is lost because of the Kalman filter linear Gaussian approximation.

Naturally, if the particle filter consists of for example two separate clusters and is thus a bimodal representation of the position, then the particle scatter is bigger and it can't be transformed into a unimodal Kalman solution for not exceeding the trigger threshold. Moreover, the threshold is approximately in the neighbourhood of 1 metre, which makes two clusters inside a small area still indicate the user body position well enough to let the transfer happen. This is, since the body itself consists of a circular area of approximately of half a metre radius especially when carrying a laptop or mobile device in hand.

5.3.1 Kalman Filter to Particle Filter conversion

As stated, the position, the velocity and the attitude solutions together with the additional accumulated error estimates are propagated alongside the Kalman filter error estimates. If the particle filter is to be updated at every step, and the inertial solution is derived for a foot-mounted inertial navigation system, the velocity will be zero at each update. Moreover, if the particle filter only follows a 2-dimensional position, the velocity does not need to be transferred and so only the yaw component of the attitude.

The weighted average position of the new particle filter would then be at the position indicated by the position solution that is tracked alongside the Kalman filter error estimates. The particle scatter should follow the Gaussian ellipse defined by the error and the covariance of the Kalman filter and also according to the additional accumulated error estimates. Particle positions are thus created using a Gaussian random generator function using initial values from the accumulated error information. The accumulated error e_{init_x} on x-direction (or Northing in the NED representation) is shown here.

$$\sigma_{init_x} = e_{init_x} \tag{5.4}$$

This initialisation deviation can now be used to initialise the particles according to the distribution at the position tracked alongside the Kalman filter. In Matlab the initial position guesses of the particles can be written as follows.

$$\begin{aligned} x_i &= x_{Kalman} + normrnd(0, \sigma_{init_x}) \\ y_i &= y_{Kalman} + normrnd(0, \sigma_{init_y}) \end{aligned} \tag{5.5}$$

Here normrnd returns a value according to the normal distribution and a given mean ("x") and deviation ("P"). x_{Kalman} and y_{Kalman} are the initial position information. The more particles there are the more accurate is the measure of the particle cloud in describing the initial Kalman filter distribution.

Similarly, for the angle of the direction of movement for each particle, we get.

$$\alpha_i = \alpha_{Kalman} + normrnd(0, \theta) \tag{5.6}$$

α_{Kalman} is the direction of movement derived using the yaw angle solution and the last step motion information. θ is the additional accumulated yaw error estimate. i is the particle index.

5.3.2 Particle Filter to Kalman Filter

The particle filter can represent multimodal distributions and is a very robust option for highly dynamic applications like the pedestrian tracking scenario. Computational demand is higher, which is why Kalman filter is preferred. When converting from particle filter representation of a distribution to a Kalman filter, a loss of information and the distribution accuracy is nearly unavoidable.

With turns and stepwise propagation characteristics the true distribution of the

particle filter can be very complex. Although, when the particle filter particle positions scatter is small enough, a conversion from the heavy computation demanding particle filter can be attempted. In this thesis, the mentioned scatter properties were used in starting the conversion from particle filter to Kalman filter representation.

The position that the particle filter indicates is the weighted average of all the particle positions according to the associated weight values.

$$r_x = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i}, r_y = \frac{\sum_{i=1}^{N} w_i y_i}{\sum_{i=1}^{N} w_i} \tag{5.7}$$

This is the transferred position. Velocity is zero if the conversion is made when the Zero Velocity Update is true. The attitude is the circular mean of the particles' direction of movements. Moreover, in the tests in this thesis, a parallel inertial only mechanisation is processed. This mechanisation result is used in propagating the particle filter. With this we can also adjust the new Kalman filter yaw solution in case the user has taken a sidestep which indicates a difference between foot heading and the direction of movement. Direction of movement was defined to be the direction indicated by two positions before and after the latest step.

The position error distribution conversion is where information is lost. The standard deviations (σ_x, σ_y) are used here that are measured from the mean position in the x and the y directions of the particle scatter. The Kalman filter additional accumulated position error estimates and the Kalman filter error estimates are initialised using these values. The Kalman filter yaw error and the additional accumulating yaw

error are derived from the particle filter direction of movement deviation similarly to the position derivation. This was presented in the Equation 5.2.

Chapter 6

Map and Context Engine Design

The context engine is an inference engine that aims to set the navigation filter parameters optimally for the current context. Like the navigation filter it reads the sensor data. The interpretation though, is completely different. As navigation filter tries to derive the most accurate positioning solution, the context engine tries to infer the exact context from the input data. Thus, the purpose of the context engine is to infer, from the input data, the correct state for the context parameters. These context parameters are called as flags in this work. These flags define how the navigation filter is set.

In this chapter, the solution to the context control inference is presented that is used in the resulting seamless navigation system. Both the behavioural and location based inference are examined. The University campus acted as the test scenario. First, an UAV survey map is presented. This is used to depict the results.

6.1 Seamless Map

A high resolution unmanned aerial vehicle survey map was used for depicting the results. The image was further georeferenced in QGIS. Four survey points, measured with Leica equipment with 4cm accuracy setting, were used for georeferencing the image onto WGS84 coordinates. The map represents the surroundings of the Nottingham Geospatial Institute in the Jubilee Campus of the University of Nottingham.

The image was further transferred into MATLAB. It was fitted into the local tangent plane coordinate system so that the image coordinates matched with the surveyed points. Figure 6.1 shows the image.

Impassable objects were listed in a KML file. The information content was limited as suggested by Zichar (2013) to only include necessary information on the map. In this thesis only wall information is used. Doors were presented as walls but as being continuously passable. 3D representation would have been too laborious for the first prototype. Moreover, for our purposes 3D representation would have contained unnecessary extra computational load. A surveyed and detailed floormap of the first floor of the Nottingham Geospatial Building was also listed in the KML file. A polygon structure was used for representing the wall information. Each room layout was presented using a polygon structure where the room corner coordinates are listed as is shown here.

```
<Polygon>
        <outerBoundaryIs>
                <LinearRing>
```

```
                <coordinates>
                        -1.18412858303779,52.9519706312043,0
                        -1.1840127174419,52.9519829666691,0
                        -1.18401169447041,52.9519809783207,0
                        -1.18399284326467,52.95190719057071,0
                        -1.18410870886055,52.951894855106,0
                        -1.18412858303779,52.9519706312043,0
                </coordinates>
            </LinearRing>
        </outerBoundaryIs>
</Polygon>
```

The bushes, pools and fences were also added for outdoors as impassable areas. Figure 6.1 shows the walls. The Nottingham Geospatial Building was cleared at the image for clarity.

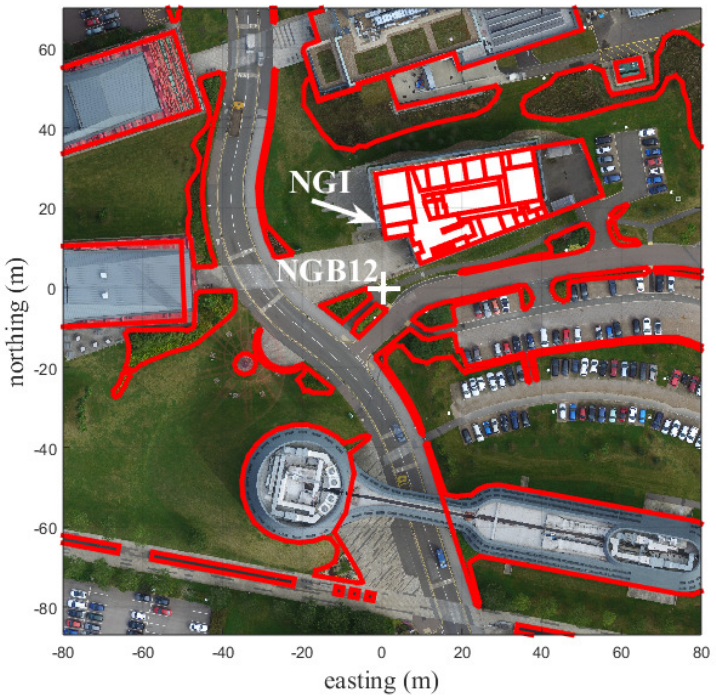


Figure 6.1: Nottingham Geospatial Building and the surrounding area in the Jubilee Campus with the wall information (red) defined in KML polygon structures. NGB12, the origin point is shown in the map.

The origin is a survey point NGB12 in front of the main doors of Nottingham

Geospatial Building. This is shown in the Figure 6.1. The used doors were marked in the code. The doors were the few selected KML walls. When matching the map, the particles or the Kalman solution were allowed to walk through these walls which were marked as doors. No temporal effects were included neither in the KML file nor the code as described by Ferreira et al. (2012). This would be useful for example when shops are open or closed and whether a shortcut could be taken through a building.

6.2 Context Inference Engine Design

The complete system consists of three distinguishable entities with different responsibilities. This follows a similar layout as presented by Groves (2013*b*). These are the sensor subsystems, the navigation filter and the inference processor or the context engine. Interfaces between context engine and sensor subsystems and the navigation filter need to be agreed before the context engine can be designed. Figure 6.2 shows the three blocks and interfaces.

6.2.1 Interface Agreement between Context Engine and Sensor Subsystems

The data flow is mostly only towards the context engine from the sensor subsystems. This is from the sensor subsystem blocks towards the context and control block referring to the Figure 6.2. The selected sensor setup was introduced in the Figure

5.4. The sensors, inertial, UWB, BLE and GNSS, data is formatted and feed to the context engine for context inference.

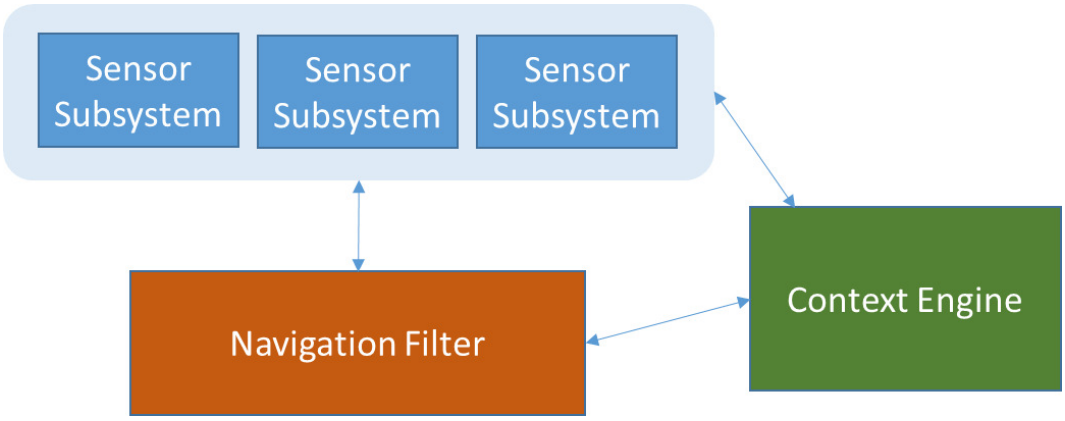


Figure 6.2: The system consists of the three main blocks and interfaces between them.

The data needed in the context inference process is stated next. This is the data format structure agreement for the sensor subsystems. The sensor subsystems thus follow this data format agreement towards the context engine. Five sensor subsystems are included in the designs of this thesis.

1. **Inertial sensor:** The inertial sensor is mounted on the user's foot. The selected sensor outputs are from the gyroscope in rads per second. These measure how the sensor turns. The values are output in the body axis. The fusion engine follows the orientation of the body axis in the local tangent plane coordinates. The accelerometer data is output in the body axis as well, in g value, which is the gravitational constant and that can be caluclated more exactly according to the position on the earth. The selected sensor type output format provides the data values with 6 decimals accuracy. The sensor type is detected when the sensor is

attached to the software. The inertial data format is the following.

```
sensor timetag: seconds.milliseconds,
gyroscope values: rad/s using 6 decimals,
accelerometer values: g using 6 decimals.
system timetag: seconds.nanoseconds.
```

A C-program was written for recording the Microstrain 3DM-GX4-45 inertial sensor. The highest sampling rate is 500Hz, but 100Hz was used for saving processing power for the three other sensors. The sensor timetag is the timetag that the sensor provides. The system timetag is the timetag that is attached to a measurement when the system writes it into the file. The output file contains inertial values with two timetags. This is passed onto the context inference block.

2. GNSS receiver: A U-Blox GNSS receiver is embedded into the same casing with the 3DM-GX4-45 inertial sensor. The antenna was placed onto the laptop cover that was used for the recording. The same C program recorded also the output from the GNSS receiver. Only the horizontal accuracy was passed onto the context engine. The GNSS receiver data format for the context engine is the following.

```
sensor timetag: seconds.milliseconds,
horizontal accuracy: in meters CEP (50% error bounds) using 6 decimals,
system timetag: seconds.nanoseconds.
```

3. BLE sensor subsystem: One BLE anchor was placed on the side of the main door. The received signal strength of this anchor is passed onto the context engine if and when available. The data format for the BLE sensor subsystem is the following.

```
system timetag: seconds.nanoseconds.
anchor RSS: dBm, no decimals.
```

4. UWB sensor subsystem: The UWB subsystem was not used for context detection.

Map: The wall information is readily available when needed. The context engine can do inquiries to the map sensor. The context engine will inquire the map sensor information regarding the location of the walls and compare this with the track information provided by the navigation filter. The call sequences are examined in more detail in the next chapter.

The timing agreement between the context engine and the sensor subsystems is asynchronous. Every measurement will be processed at an arrival to the context engine. No priority scheme is used if two sensor subsystems happen to send a measurement simultaneously. This is highly unlikely because of the system timetag resolution. First measurement is served first.

6.2.2 Interface Agreement between Context Engine and Navigation Filter

The context engine needs position information from the navigation filter. The context engine can make queries to the navigation filter. In addition, the context engine uses the particle scatter value and can ask for the saved previous sensor subsystem values. The scatter value is the standard deviation of the particle cloud from the particle filter mean solution when the particle filter is used. Also state flag statuses can be asked from the navigation filter. These are provided immediately when queried.

The output from the context engine towards the navigation filter consists of commands. These commands change the state of the navigation filter. The state machine in the navigation filter is discussed in the following chapters.

The output commands from the context engine are listed here that affect the state of the navigation filter. The context engine outputs these commands for the navigation filter to switch its status flags. The navigation filter then uses different code for deriving the positioning solution. For all of the commands there exists the code section that is used after the command has been issued. To switch back to using the previous code section a reverse command needs to be sent. This list of commands depends on multiple sensors and is thus common for all the sensor subsystems. The relation to the sensor outputs is shown later in the Figure 6.3.

turn on indoor flag ,

```
turn off indoor flag,
turn on entrance flag,
turn off entrance flag,
turn on stairs flag,
turn off stairs flag,
ZUPT flag on,
ZUPT flag off,
ZUPT flag waiting,
turn on particle filter flag,
turn off particle filter flag
```

The following derivation shows how and when each of these commands is sent to the navigation filter.

6.2.3 Context Engine Inference

Th navigation Filter has five status flags. They are the indoor, entrance, stairs, ZUPT and particle filter flags. These affect the internal state of the navigation filter. Figure 6.3 shows the logic of the context engine.

The BLE anchor at the main entrance door (inside) acts as a proximity sensor telling the system that the user is definitely indoors if a threshold value so indicates.

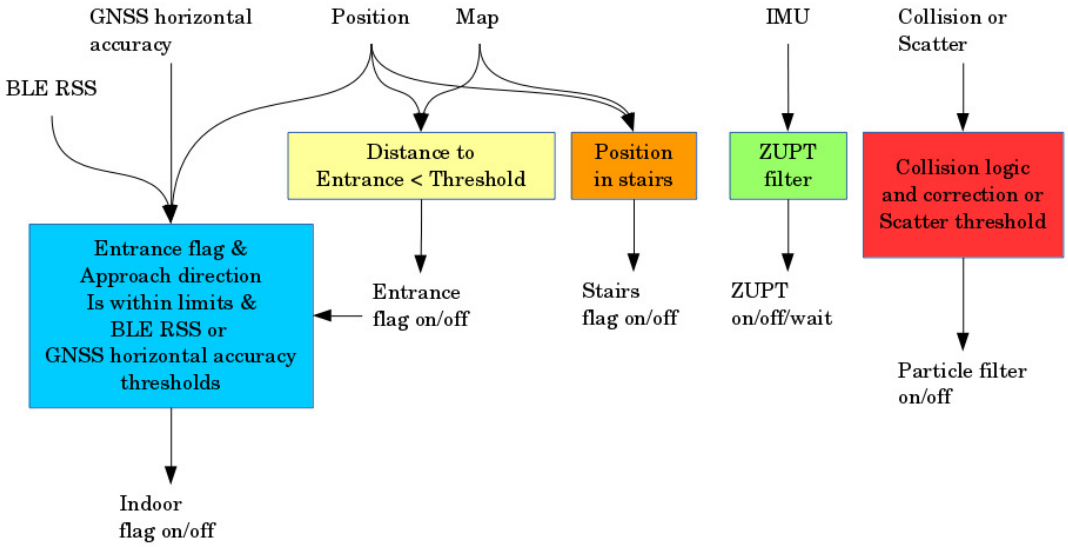


Figure 6.3: Context engine logic.

The threshold for switching the indoor flag on was set to -65dBm. This means approximately less than 3 metres proximity to the anchor. Inspecting a window of previous anchor values indicated that this condition was true if any of the values were below the threshold.

The threshold for GNSS receiver horizontal accuracy was set to 2.5m CEP. This was tested to be a reasonable level for the quality of the GNSS to produce trustworthy position output outdoors. Again, a number of previous GNSS quality values were inspected to indicate this condition to be true if the value was below or above the threshold depending of the current status of the indoor flag.

An approach direction was calculated from the latest 3 step position updates. This means the mean angle of approach during two last steps. Two thresholds were used which covered 170 degrees approach angle to be accepted for switching the

indoor flag either on or off depending on whether going outdoors or indoors.

Two entrance distance thresholds were used for turning the entrance flag. 5m threshold was used when user was initially outdoors and going indoors. 3m threshold value was used when vice versa. Entrance flag turns on the indoor-outdoor detection code.

The stairs flag was turned on simply if the navigation filter position was assumed to be in the stairs.

Zero update filter used a 7-tap low-pass filter. Both the magnitude of the total acceleration and total turn rate were used for detecting zero movement. If the acceleration was beyond 0.92g or 1.08g or the turn rate was more than 0.2rad/s the zero update flag was changed. This was implemented with an AND function between the two total values. A similar approach was used in the previously mentioned work by Yin et al. (2014). Once ZUPT was triggered in the following epochs wait state of the flag was used to tell the navigation filter to be ready for the end of the step. When the foot was not moving, ZUPT on message was sent to the navigation filter. If the ZUPT method is the only method applied into the mechanisation process and the Kalman filter following the foot mounted IMU, the yaw estimate of the foot will drift, because it is not observable. Additional moderately weighted yaw measurements can be used to limit the drift by using for example the subsequent positioning fixing measurements.

Collision logic and scatter properties are discussed in detail after a detailed introduction to the used particle filter and Kalman filter in the following chapters.

6.3 Concluding remarks

The navigation filter state can be considered to be of either the Kalman or Particle filter solution and the state flags. The particle filter flag function depends heavily on the wall structure. If there are many walls in the environment, more collisions are likely to happen, which in turn would indicate more usage of the particle filter. The other way around, having less walls and accurate position updates will make particle scatter smaller and a transfer to the Kalman solution easier. The ZUPT filter thresholds can be adjusted. These thresholds could be replaced by machine learning attributes that adjust the threshold values according to the position updates and navigation success.

The stairs flag is triggered by the user entering the location according to the navigation solution. Other areas could be implemented like walking on grass or the gravel on the parking area. Settings could be then adjusted according to the surface of the ground and how it will affect the inertial errors. The entrance flag or the distance to the entrance is used for enabling the indoor-outdoor transition check-up functions. While far away we don't need to unnecessarily check for the current location.

The main function of our small context inference is the indoor-outdoor context check. Weights for the indoor positioning systems can be increased and GNSS measurements ignored when going indoors. Moreover, filter settings for walking in a wall-rich environment are different than those for the open area walks. This

first version is designed for testing walks which are determined walks from indoor to outdoor and vice versa. Tests are shown in the later chapters. More inference design is needed or additional sensors added to cover exceptional cases where the user shuffles feet or stays long at the entrance turning around often. A possible addition could be a more accurate approach sensor like UWB sensor for measuring the accurate distance to the entrance.

No behavioural context function was added other than the adjustment of the threshold values while in stairs. The inference in this work is based on location. The GNSS receiver function would ease the indoor detection by using the method described by Daniel et al. (2016). No pure virtual sensors were used as described by Rivero-Rodriguez et al. (2014). Indoor flag inference is closer to logical sensing than raw measurement sensing because of location and measurement history usage. A more behavioural aware context engine could identify stairs by checking the stepwise z position change, but this and run detection are left for future research.

As a summary, the context engine derives the current status by reading the history information of the position solution and the individual sensor subsystem solutions and combines this information with the current sensor measurement indicators. The filter state and manner of operation is controlled by this context inference.

Chapter 7

Particle Filter Design

Particle filter is a robust filter for presenting multimodal probability distributions as mentioned by Siciliano & Khatib (2008) in chapter 4. This chapter presents the design, implementation and tests on the particle filter properties. Particle cloud properties, propagation characteristics and ways to correct the particle filter are examined from the seamless navigation positioning system perspective. First, we go through the general design of the particle filter. After this follows the sensor subsystems designs for the particle filter implementation. Finally, we evaluate the sensor fusion implementation.

7.1 Design Basis

First, the top-level design of the system is examined. Figure 7.1 shows the top-level design.

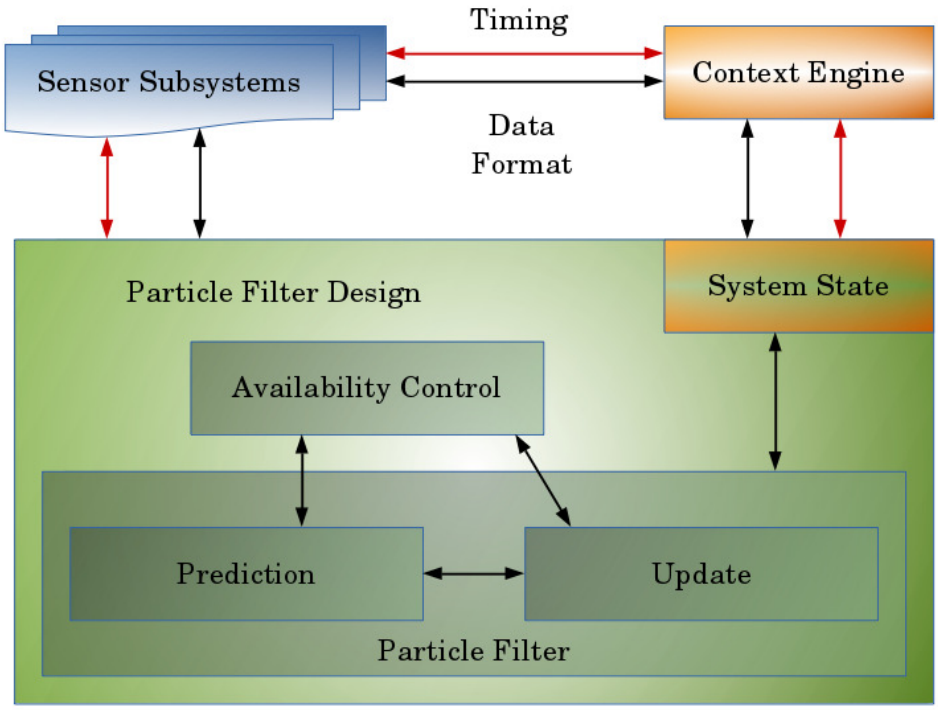


Figure 7.1: Top level system design for a particle filter.

The timing and data format agreements from and to the context engine were explained in the previous chapter. No feedback goes from the particle filter to the sensor subsystems. The timing agreement between the particle filter and sensor subsystems block has one condition. For one measurement, the context engine path has to be completed before submitting the measurement to the particle filter.

For the particle filter the data formats are the following. The inertial sensor interface is the same as for the context engine block.

GNSS receiver: For the GNSS receiver the data format includes also the converted latitude and longitude values in the local tangent plane coordinates. The

reference point for the local tangent plane is the point NGB12 described previously in Figure 6.1.

```
sensor timetag: seconds.milliseconds,
converted latitude: meters in the MATLAB local tangent plane
                    coordinate frame,
converted longitude: meters in the MATLAB local tangent plane
                     coordinate frame,
horizontal accuracy: in meters CEP (50% error bounds)
                     using 6 decimals,
system timetag: seconds.nanoseconds.
```

BLE sensor subsystem: Six BLE anchors were placed inside around the first floor of the Nottingham Geospatial Building. A fingerprinting method, first mentioned in chapters 2 and 3, was used to produce an estimate of a position within a defined time interval. In addition to the position estimate, an error bound estimate was also attached for both x and y directions. These are explained in the next section concentrating on the sensor subsystems. The data format of the BLE sensor subsystem for the navigation filter is the following.

```
system timetag: seconds.nanoseconds.
position estimate: northing, that is x, and easting, y in the
                   MATLAB local tangent plane coordinate frame.
position error estimate: northing, x and easting, y
```

standard deviation error estimates.

UWB sensor subsystem: The UWB sensor subsystem produces a position result with an error estimate. This is a similar format to the BLE subsystem.

7.2 Positioning Fixing and Dead Reckoning Sensor Subsystems Design and Tests

The BLE and UWB subsystems are presented here in more detail. Moreover, the accuracies of both systems are evaluated.

7.2.1 BLE Subsystem

The Bluetooth low energy beacons are more robust than the WiFi access points when it comes to the deployment for positioning system purposes. WiFi covers wide areas of the urban environment but its deployment is not optimised for positioning purposes. Easy deployability makes the BLE approach very attractive. The design is similar to the one presented by Faragher & Harle (2015). The fingerprinting scenario was constructed for positioning.

In fingerprinting the BLE beacons are deployed within the area of interest. Then in the offline phase fingerprints at strategic locations are recorded. Radiomaps can be constructed using these points.

In the online stage, the radiomap database is compared with measurements and position is inferred.

Offline Radiomap Recording

As presented previously, six BLE anchors were used. Four anchors were the Kontakt.io tags advertising at the rate of 10Hz. Two anchors were the Raspberry Pis with CSR BLE adapters advertising at the rate of 50Hz. These are shown placed in the A floor of the Nottingham Geospatial Institute in the Figure 7.2.

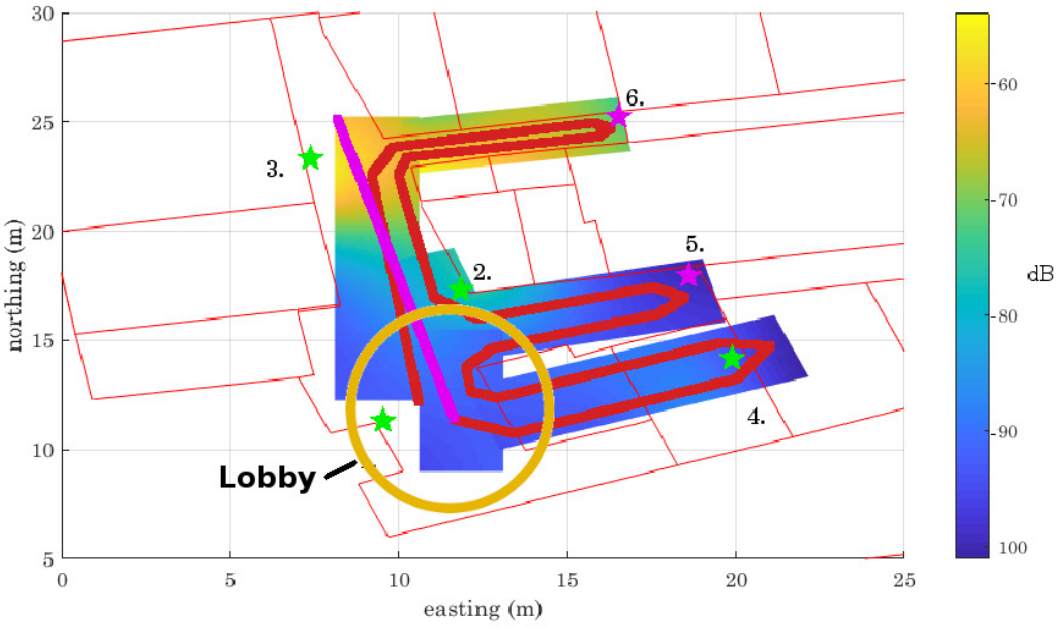


Figure 7.2: BLE anchors placed in the Nottingham Geospatial Building A-floor. Green stars indicate Kontakt.io anchors: Magenta stars are Raspberry Pi anchors.

The green stars mark the locations of the Kontakt.io anchors. The magenta stars present the Raspberry Pi anchors. The signal strength map shown is for the anchor 3. Similar map was constructed for all six anchors. Altogether 30 points

were recorded in the A-floor. Each point recording time was 3 minutes. During the 3 minutes interval, the person who recorded was turning around. This was to take the effect of body shadowing along into the fingerprints. The resulting variance was thus greater than if only the receiver would have been at the spot. As Faragher & Harle (2015), the Gaussian processes regression was used to construct the square radiomap areas with upper and lower standard deviation limits. The resulting grid was interpolated at every 10 cm.

The red track in Figure 7.2 shows the approximately 80m test track. The reference track was marked on the floor and the inertial measurement unit was used to mechanise the walk to get the travelled distance. No shadowing effects were compensated for in this study. Especially the effects of body shadowing were evident, although this was taken into account, as mentioned, when recording the fingerprints.

Online Test Setup 1

In the online phase, the measurements were averaged over a time period. Faragher & Harle (2015) used 1s window.

The following percentual nearest neighbour grid method was developed for position and position error acquisition.

1. Measurement mean during the averaging period for each anchor is taken.

2. In addition, the deviation during the measurement window was recorded.

3. Each anchor measurement mean was differenced with the radiomaps. Now the

most likely locations were the grid points where the difference was smallest.

4. If the measurement deviation was greater than the average fingerprint deviation within the whole area, we used this deviation value for our measurement. If not, the average fingerprint deviation value was used for the current measurement.

5. Each grid cell was assigned with a probability according to $p = exp^{-\frac{difference^2}{2\sigma^2}}$.

6. The difference maps were normalised for each anchor to get six probability masses for the whole area. Now there was six or less (depending on availability) normalised anchor radiomaps presenting the probabilities for all the available anchors that had a measurement value.

7. These six or less probabilities were multiplied and normalised to get the combined measurement probability for the whole area.

8. Now 5% of grid points were selected according to the height representing the most probable locations. This is the nearest neighbour and maximum a posteriori.

9. The weighted average of these collected points was the measurement solution.

10. The deviation for x and y directions were taken using the rest of the 5% grid points from the measurement solution.

Figure 7.3 depicts the results in a good and bad case.

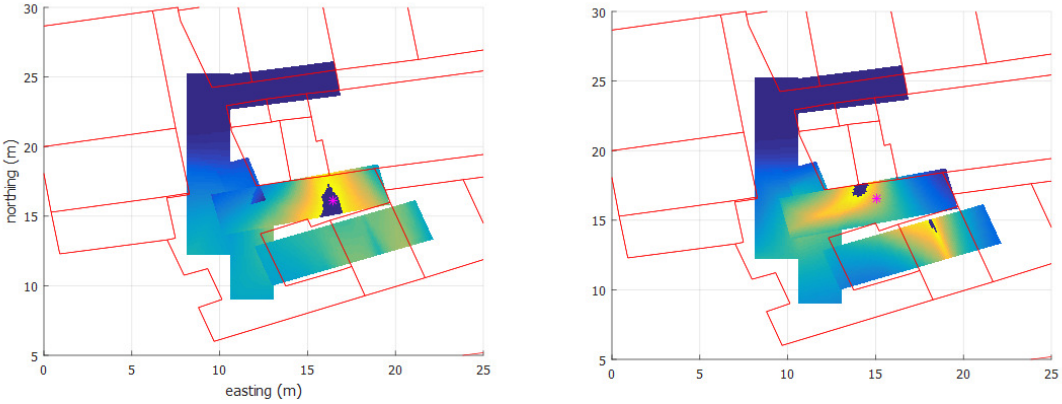


Figure 7.3: The image shows a good quality position result using the BLE fingerprinting on the left. On the right is shown a bad quality measurement.

The good quality position was close to the reference track and consisted of a unimodal location. The bad quality measurement result consisted of two or more peaks. Anchorwise measurement quality check-up could eliminate or deweight bad quality anchorwise measurements.

The position measurement error estimate was taken as the standard deviations on both x and y directions of the weighted result. All the points that were inside the probability mass that was over 95% (can be adjusted easily, 68% was tested) were involved in the standard deviation calculation.

Walking speed was 1.3 m/s. The magenta section in Figure 7.2 was measured running. The section where anchor number 4 is located was in stairs. The deployment of the six anchors produced trouble in identifying whether the result was in the lobby (shown in Figure 7.2 near anchor 1) or in the stairs (near anchor 4). Body shadowing had a notable effect in the corridors and was partly involved in the bad quality results in the lobby. In the corridors, the anchor, shadowed by the body,

gave results indicating a smaller value than was supposed. A user location based shadowing model could help in raising up the anchor measurement values when being shadowed by the user body.

Figure 7.4 depicts the track using 2s time window for measurements.

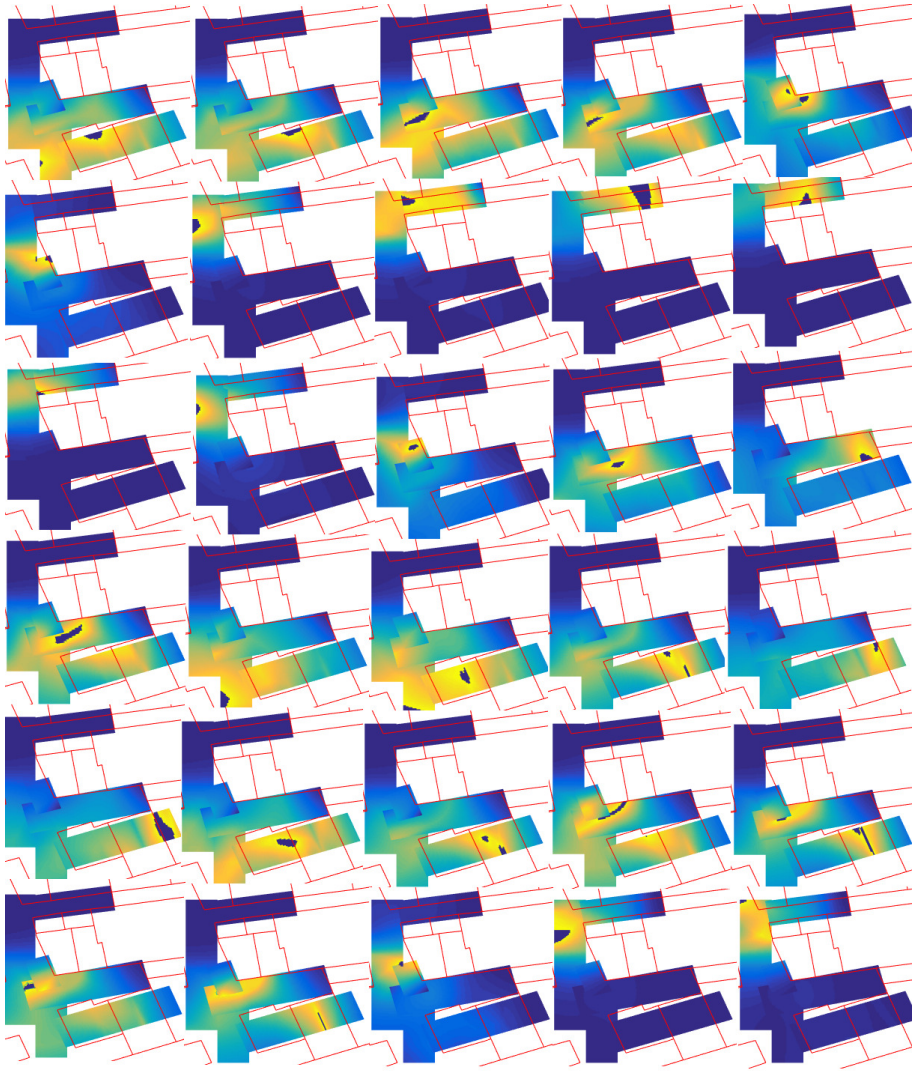


Figure 7.4: The BLE track using 2s measurement time window (left to right).

Figure 7.5 depicts the accuracy for the BLE track using a 1s time window.

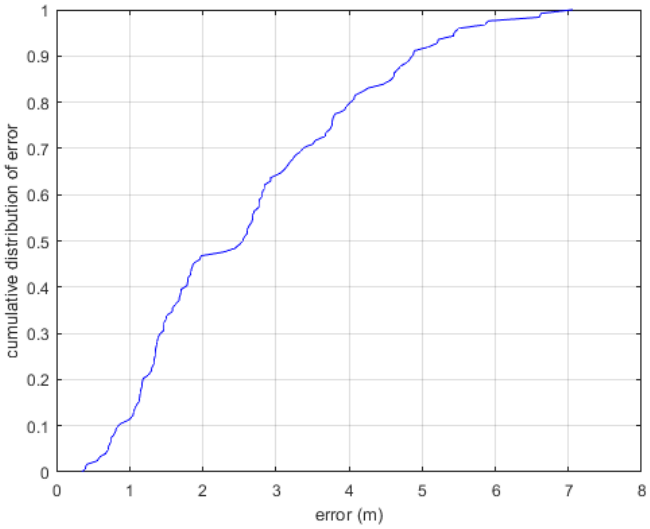


Figure 7.5: The cumulative distribution of error for the BLE track using 1s measurement time window averaging.

Most of the largest errors come from the lobby measurements. In corridors, the accuracy is enhanced by the constraining walls.

Online Test Setup 2

A new trial was conducted using 40 training points of which a positioning result is shown in Figure 7.6.

The first 50 metres of the test track until the start of the stairs was walked three times. The accuracy of four different measurement time windows is shown in Figure 7.7.

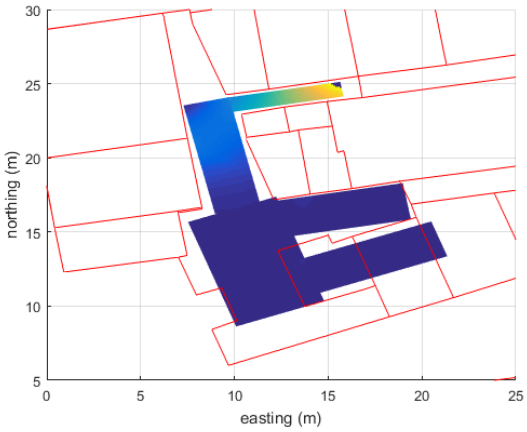


Figure 7.6: Second walk test with BLE fingerprinting. A measurement result probability distribution is shown in the figure.

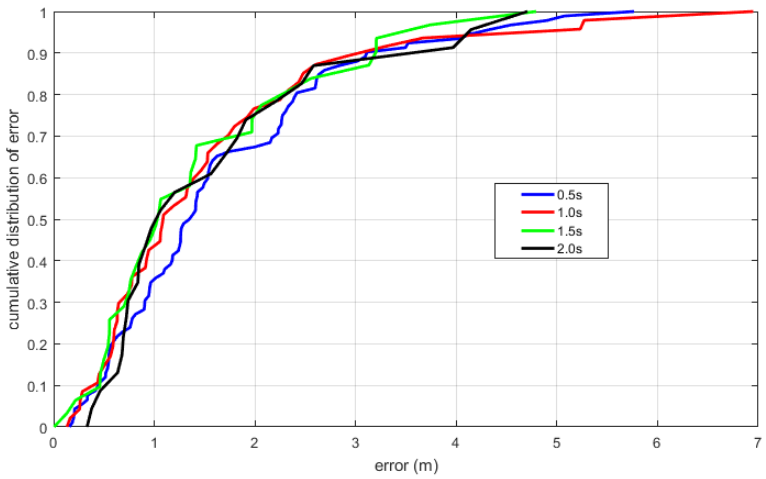


Figure 7.7: Second test walk accuracy using four different measurement time averaging windows with BLE fingerprinting.

The biggest errors are reduced when using a longer time window. Otherwise the accuracy indoors and in corridors is approximately within 4 metres (96% of error). The stairs were not included in the second test walks. We can see here the effect of the troublesome positioning results in the lobby area that were included in the first test case. Four metre error in a corridor of five metres is relatively large. It is

nearly impossible to distinguish perpendicular to the corridor on which side of the corridor the user is. That is, how close the user is to either of the side walls of the corridor. But BLE fingerprinting can distinguish on which end of the corridor the user is located.

This tells us that with body shadowing included, the BLE function cannot help us in achieving sub-metre level positioning. As a rough proximity and presence beacon it functions well but for applications demanding even more accurate positioning it does not work. For initialisation it can be used, especially for particle filter initialisation in case of a filter fail.

7.2.2 UWB Subsystem

For deriving a 2D position from the three anchor ranges, an iterative least squares method was used with a constant of 10 iterations. Figure 7.8 depicts the UWB anchor positions and the resulting track for the same test walk described in the first BLE test.

The previous position was used as an initial value for the least squares iteration. The geometry is very narrow for the three anchors, especially along the easting axis. Body shadowing effect is present in the UWB system. Tests indicate up to a half a meter change in range values with body shadowing. This effect is amplified in the least squares derivation with small distances between anchors and a poor geometry as can be seen in the resulting track.

The non-line of sight function in the north corridor was surprisingly good. In

the stairs though the northernmost anchor's ranges were too great. Combined with the ranges affected by the glass fence in the staircase on the middle anchor, the resulting track in the stairs is very wrong (south of the outside walls). Also, the turn in the middle west to east corridor was mostly missed by the system since no readings were able to be acquired from the northernmost anchor.

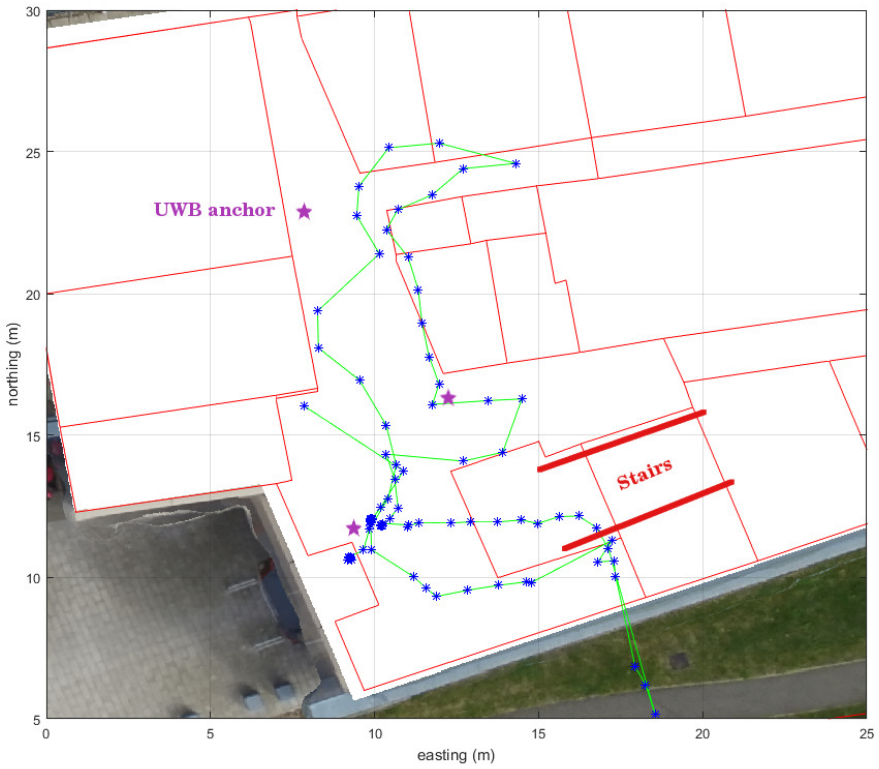


Figure 7.8: UWB anchors placement and resulting test track.

Figure 7.9 depicts the accuracy of the UWB system with the poor geometry placement of the anchors.

Non-line of sight measurements are worse the more obstructions there are between the target and anchors. This happens in the staircase where the results are

the worst. UWB position measurement is the resulting position from the least squares iteration. The position error is taken as the standard deviation of the error residuals between the three range measurements from the anchors and the position measurement.

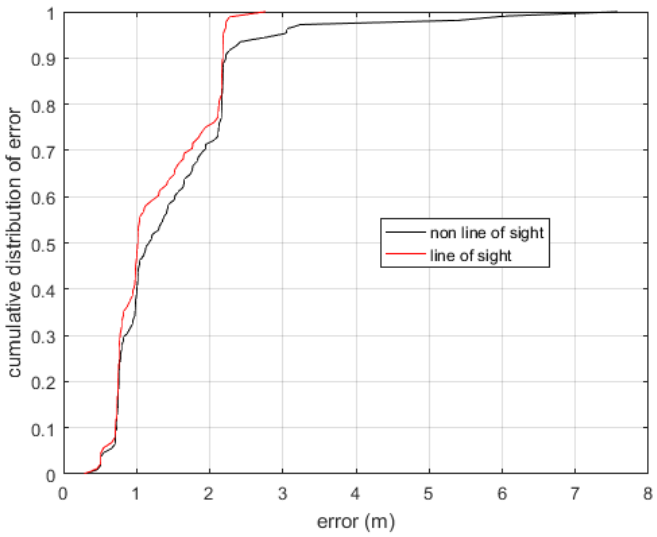


Figure 7.9: UWB system accuracies for line of sight and non-line of sight cases with poor placement of anchors.

7.2.3 Dead Reckoning with IMU

A closed loop, 15 error state Kalman filter was used for dead reckoning with the inertial sensor as was described in Chapter 4. Position, velocity and attitude solutions are propagated outside of the Kalman filter. The Kalman filter tracks the position, velocity and attitude errors. Every epoch, these errors are compensated for the position, velocity and attitude updates. A closed loop approach requires that the error is reset at each epoch. Covariance thus tracks the epochwise error bounds

of the error estimate.

The state vector was declared in Equation 4.33. The state that is these error values, are propagated according to Equation 4.34. Zero velocity, zero angular rate and attitude levelling correction updates that were used are presented here, with the small virtual measurement error for velocity and approximated maximum measurement errors for the attitude error estimate.

$$H = \begin{bmatrix} 0_3 & -I_3 & 0_3 & 0_3 & 0_3 \\ -I_3 & 0_3 & 0_3 & 0_3 & 0_3 \end{bmatrix},$$

$$R = \begin{bmatrix} 0.005^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.005^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.005^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.08^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.08^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.12^2 \end{bmatrix} \tag{7.1}$$

The zero velocity updates are stated here to be very accurate. In addition, levelling is performed. This is attitude fixing expecting that when the shoe is on the ground and zero acceleration is assumed, the g vector should be pointing down. If it is not, attitude is corrected epoch by epoch towards the correct attitude using the given degree error values in the measurement error matrix. Quaternion based levelling was used while ZUPT condition was on. This is presented in the appendix A.

Figure 7.10 shows a typical track using 62.5Hz sampling rate on the Microstrain sensor. The error distribution is not shown because we compare a dead reckoning system and this is not informative. The sampling rate is relatively low for inertial dead reckoning. Moreover, loose attachment of the IMU to the foot table produces wrong zero updates with flimsy sensor backswings and missed turns as can be seen in the figure. This result, though characterizes errors that should be accounted for in the navigation filter on the upper fusion level. These errors are further discussed when testing the fusion with particle filtering.



Figure 7.10: Inertial track using 62.5Hz sampling rate.

7.3 Particle Filter Fusion

It is not computationally efficient to iterate a particle filter at each epoch of the IMU. The particle filter is propagated when a step is completed or a measurement is ready from either the BLE or UWB sensor subsystem. Figure 7.11 depicts the particle filter structure.

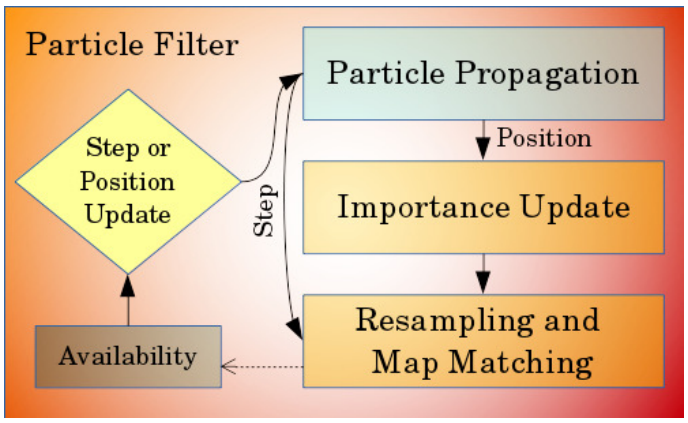


Figure 7.11: Particle filter block diagram.

The sensor subsystem measurement formatting is the first phase. The sensor subsystems are shown on the top left in the Figure 7.1. After this, the context engine infers how to set up the navigation filter settings. Third, the navigation filter is processed. The 15 error state Kalman filter runs in the background for the inertial solution. This solution is used in the propagation stage of the particle filter as seen in Figure 7.11. Each particle is propagated according to its angle and position states and the inertial propagation model. The new position is derived using the previous particle filter update, taking the distance travelled according to the IMU and the angle change in the IMU solution between the current and the two latest particle

filter updates. Direction of travel of each particle is changed according to the angle change in the inertial solution. Distance travelled is likewise taken using the previous update and the current IMU position solution.

A perturbation model adds small propagation noise for each particle. This is also called regularisation and prevents impoverishment. An experimental model was used according to the tests conducted on two different inertial setups. We discuss these next.

7.3.1 Defining the Perturbation Characteristics

Two different inertial sensor setups were tested. Microstrain 3DM GX3-25 and 3DM GX4-45 sensors were strapped to a left boot with tape. A laptop was used to sample inertial data at 100 Hz rate. Another setup used a cheaper Arduino inertial sensor breakout board and Bluetooth module with a connection to a laptop. This setup is shown in the Figure 7.12. A 50Hz recording rate was possible using this setup.

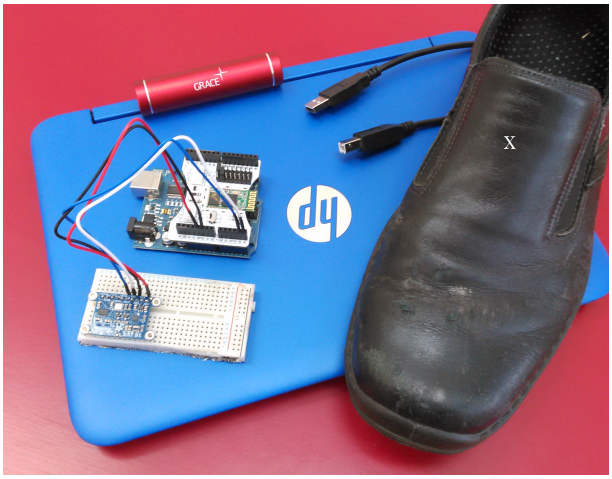


Figure 7.12: An arduino Kit used in derivation of the perturbation characteristics.

10 walks were conducted in the office with all three sensors. Figures 7.13 and 7.14 show results of the ten walks for two of the three sensors with track trails. 5 walks followed the office outline and 5 walks passed through work desk corridors.

Of these measured tracks two perturbation model attributes were derived for the sensors. The same zero velocity update detection method was used in all measurements. The ZUPT method was described previously.

Yaw angle and step length standard deviations were derived from the results. The difference of the yaw angle from the true yaw angle in the end of the track was calculated. Likewise, the position error was noted at the end of the walking track. In addition, approximate stepwise analysis was conducted. The true track was taped on the office floor and steps were carefully taken to follow the set yaw angle changes of 90 degrees at the office corners. For microstrain the angle deviation was approximated as $\pm$ 2 degrees/m. For the Arduino setup, the value was $\pm$ 6 degrees/m. Similarly, length deviation characteristics were $\pm$ 2cm/m and $\pm$ 5cm/m.

These values are used to perturb the particles' positions and directions of movement in the propagation stage of the particles. This is the regularisation as explained in chapter 4. Figure 7.15 shows the moments of time when the particle filter loop is triggered.

When a step has been detected, only the IMU measurement is used in propagating the particles. After propagation, the perturbation spreads the particles according to the perturbation model. No importance sampling is processed.

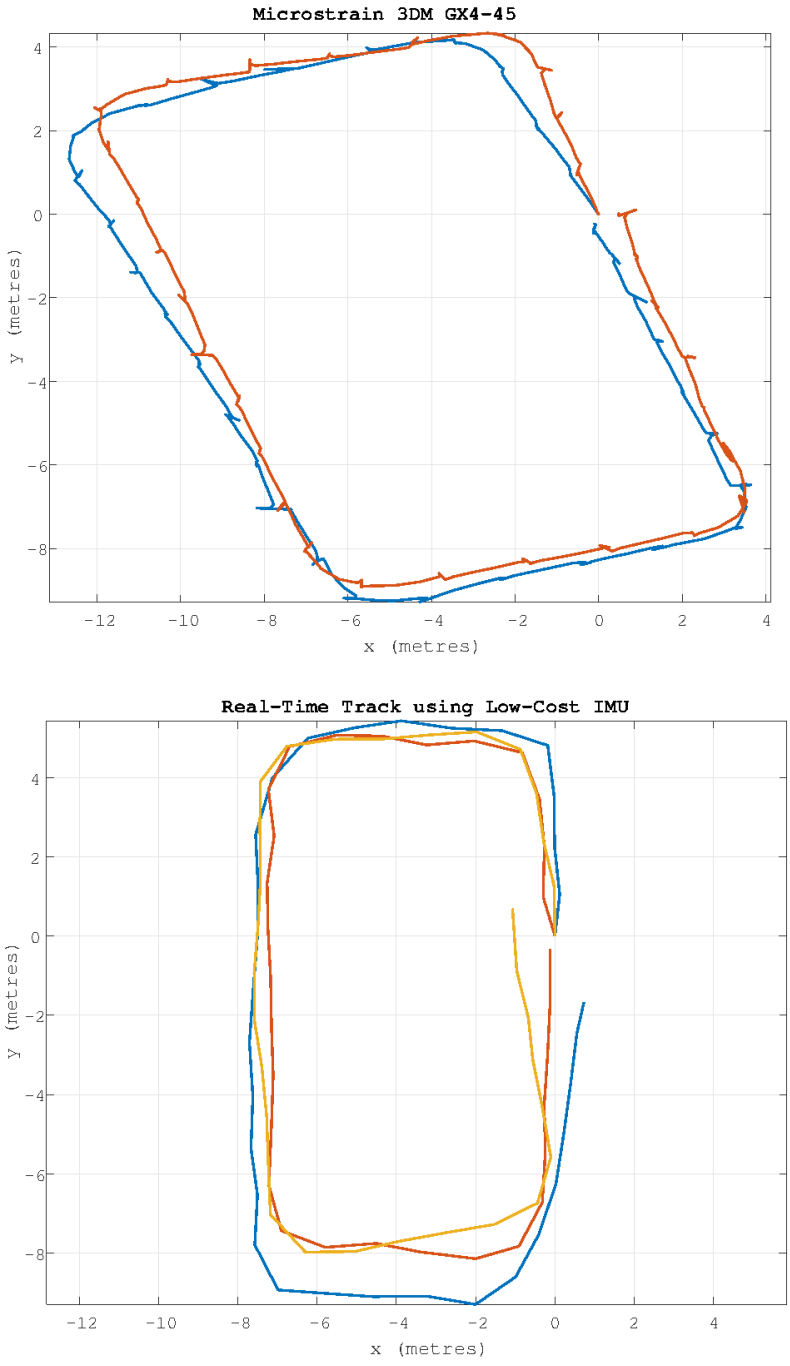


Figure 7.13: Example result tracks on experimenting for the perturbation characteristics.

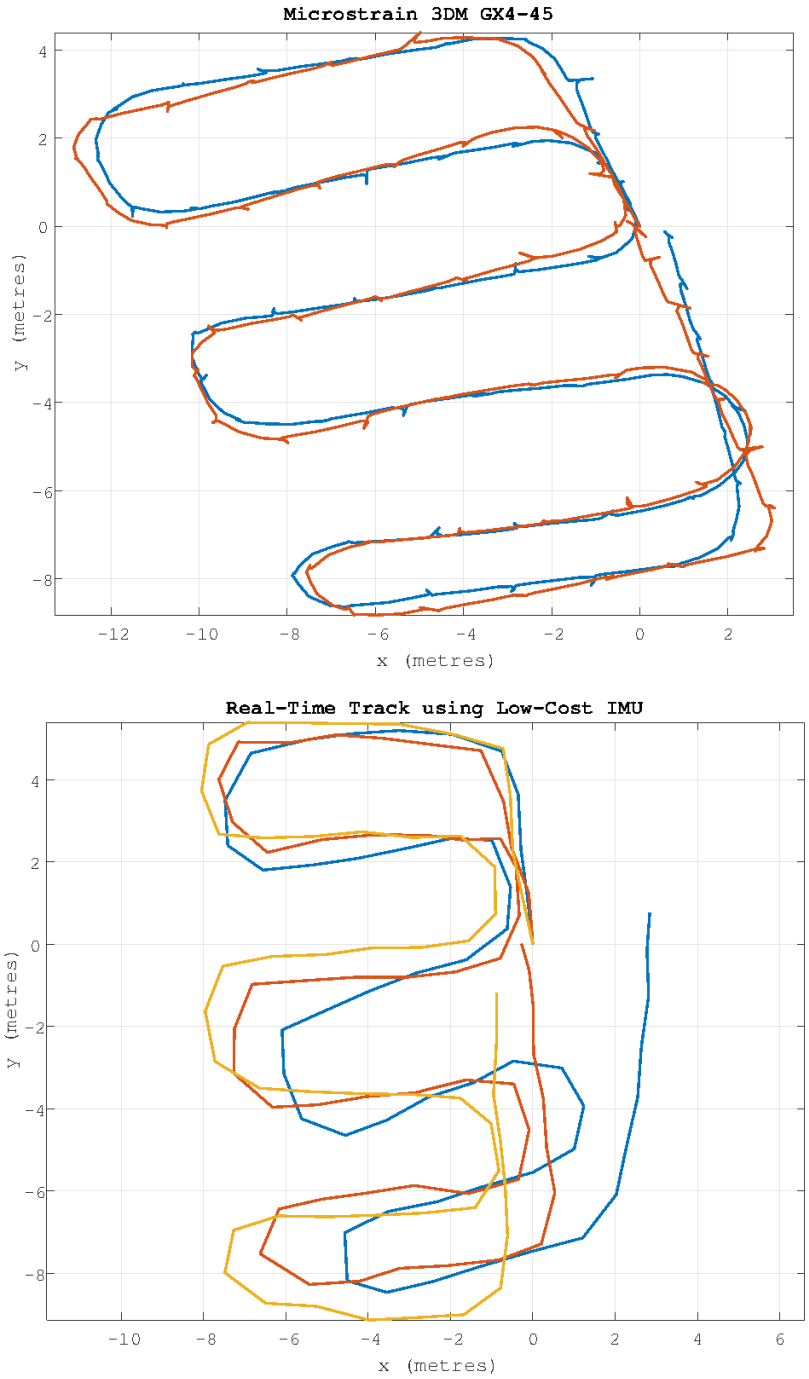


Figure 7.14: Example result tracks on experimenting for the perturbation characteristics.

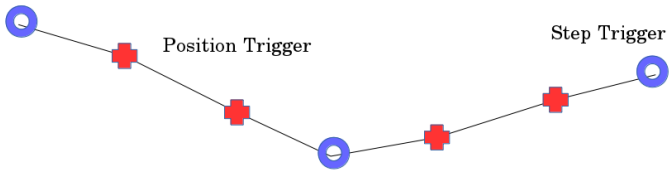


Figure 7.15: Particle filter is updated every time a step is encountered or a new measurement comes from the position fixing systems.

When a position update arrives, the importance sampling step is processed in addition to the particle propagation. A prior probability distribution function was used as the importance function. Then the importance update is simply according to the Equation 4.28 and the old importance is combined with the measurement probability distribution. New weight for particles is derived by multiplying and normalising the sample importance of the particle and the measurement importance density at the particle position.

7.3.2 Particle Filter Update using Tukey-style Distribution

The Decawave UWB anchors output only one range estimate for each anchor. The error estimate using least squares thus does not take into account the errors in the ranges that can be larger. The kit doesn't provide range measurement error estimates. These errors in range then affect the least squares result together with the anchor geometry. Sometimes the wrong ranges match and the position error estimate is very small, although in reality, the position is clearly wrong when compared to the reference system positions. This was noticed when testing the characteristics

of the UWB sensor subsystem as depicted in the Figure 7.8. In order to take into account the effect of the range errors, not dealt within the sensor subsystem, a novel approach is presented here.

The first alternative would be to measure a point and take the characteristic deviation of the measurements and use that including body shadowing effect. The second alternative would be to use a value from the datasheet, which was applied at first. When moving, the accuracy will get worse because of body shadowing, walls and poor placement of the anchors resulting in poor GDOP. Thus, the UWB position measurement error was increased manually to take into account these effects in the measurements shown in the Figure 7.8. This means simply increasing the estimated error value for each UWB system position measurement.

When using particle filter position updates and the Gaussian measurement update approach the particles closer to the measurements will get weighed more and are considered better. When we examine Figure 7.8 we notice that often the UWB position result is inside or very close to a wall. For a particle filter, this means that "wrong" particles near the wall will get too much weight in the importance sampling update. Similar phenomenon happens with GNSS measurements. Very often the result is 4 meters too much left and then jumps to be 4m too much on the right side of the correct position.

A novel update approach was created. This is called "the Tukey-style" measurement update. Depending on the technology (GNSS, UWB, BLE) and the measurement quality if provided, instead of using Gaussian probability density for mea-

surement update we use a "Tukey" -style probability density update. Figure 7.16 depicts this update probability distribution.

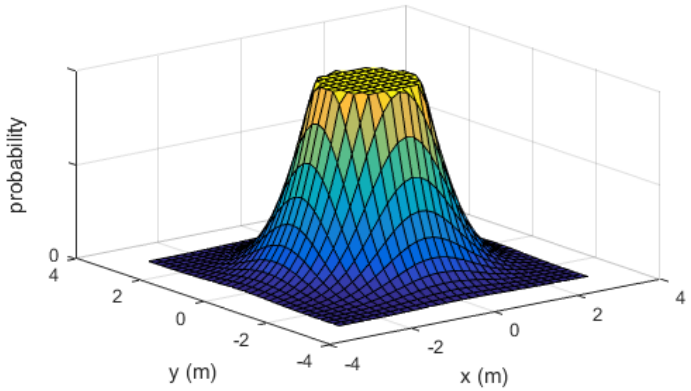


Figure 7.16: The Tukey-style window for the particle filter measurement update.

In a Tukey-style filter there is a flat area in the middle as shown in the Figure 7.16. Here the Tukey range can be called "the radius of immunity". The weights remain unchanged within this region. The surrounding area, around the Tukey circle, follows the Gaussian distribution defined by the error deviations from the least squares process in both x and y directions. Using this approach instead of the normal Gaussian with the least squares error estimate results, the particle filter survival and success rate was higher. More of the correct particles survived the importance sampling and the resampling elimination process because of this method, when previously due to weighting the wrong particles, the correct particles would be killed in the resampling stage.

Figure 7.17 depicts the Gaussian and Tukey-style measurement update differences. The probability mass inside the Gaussian distribution area equals that of the doughnut area around the Tukey-style area of inefficiency. Inside the Tukey-style

area the weights of the particles are unaffected.

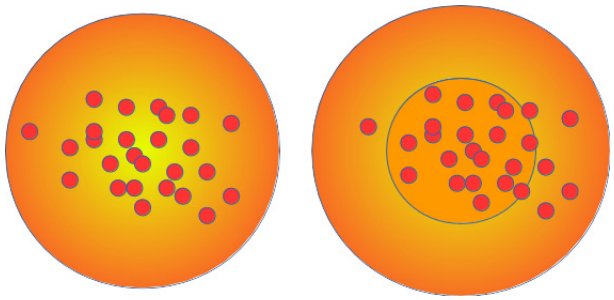


Figure 7.17: In the left is the Gaussian distribution for the measurement update. The Tukey-style window for the particle filter measurement update is on the right. The particles in the area in the middle retain their relative weight values.

In addition, when the UWB measurement solution was showing a result inside or behind a wall, the error deviation estimates of these measurement were increased even further. This is a simple sensor subsystem measurement integrity checking. This was placed in the availability system block within the navigation filter. To limit the amount of computation, no extensive data association methods were used. The addition of an integrity monitoring block could be an option parallel to the availability block. Next, we examine particle filtering results including the use of Tukey with UWB for the indoor scenario.

7.3.3 Resampling Strategy and Results Evaluation

The common approach for resampling is to use a measure for the particle cloud weights. This was mentioned previously in chapter 4 and is the effective sample size N_{eff}. When this measure reaches a threshold, resampling is initiated.

Resampling was used at every update to prevent the sample impoverishment

altogether. This makes the particle filter slower when particles are examined and modified twice instead of just one round. 4% of particles were resampled at each update. The particles with lowest weights were transformed to be copies of the better particles. This copy was a weighted random selection of a better particle, according to the weights of the better, the "survived", particles. In addition, small regularisation was applied for these transformed particles.

It was noticed that when the inertial mechanisation fails, like can be seen in Figure 7.10, resampling a higher percentage of particles will compensate for the errors uncompensated by the sensor subsystem. The inertial subsystem sampling frequency was low. This, together with the strapdown taping quality made the inertial sensor subsystem unable to track and compensate for the errors. But with higher resampling and/or more loose propagation model this error can be compensated for. The sort of errors, as can be seen in the 7.10, when the inertial track turn is erroneous, are harder to detect. We will present a novel method (track form matching) for detecting this in the chapter that follows, but for now we will use slightly more loose propagation model and resampling percentage. The resampling percentage increase can be thought of as a more frequent resampling when using the effective sample size measure.

Figure 7.18 shows the result track when using a particle filter for sensor fusion.



Figure 7.18: The resulting particle filter track.

Inertial track was used in the particle propagation step at each update (step and position). The BLE and UWB sensor subsystems provided positions with error estimates. The UWB sensor subsystem provided an error estimate that was noticed to be too small and this was partly compensated for by manually adjusting the particle filter update settings and the Tukey-style updating. The resulting track still shows the bad results from the UWB affecting the fusion track in the staircase. The inertial track quality is compensated for by the particle filter propagation characteristics and the walls constraining/killing wrong particles. The UWB results that were behind the walls were downweighed by adjusting the error estimate when

integrating with the particle filter.

Particle amount: Mapmatching compared the new particle location with the map walls similar to Perttula et al. (2014). If the new location had gone through a wall or was inside a wall it was resampled in a similar manner to what was described previously.

The initial heading was unknown. The initial particles' direction of movements were set according to a uniform distribution over the whole circle. The particle cloud was initialised at point (0,0) using 100 particles. Figure 7.19 depicts the position initialisation accuracy that is dependent on the particle amount.

The GNSS receiver output error is stated to be 2.5m. This is true on an open field. Within urban canyons and areas with less open sky, the error is in reality much larger.

A simulation was conducted on how many particles are needed to represent a position solution with less than a meter error. Using three values for 50% error deviation limits, 10000 initialisations were made for different number of particles. We can see here the error that the clouds usually have when trying to represent a position. According to this, we chose 100 particles for our implementation. The position solution is the weighed average of all particles' positions. GNSS sensor subsystem is the least accurate when compared with the constrained BLE and UWB systems and thus defines the number of required particles.

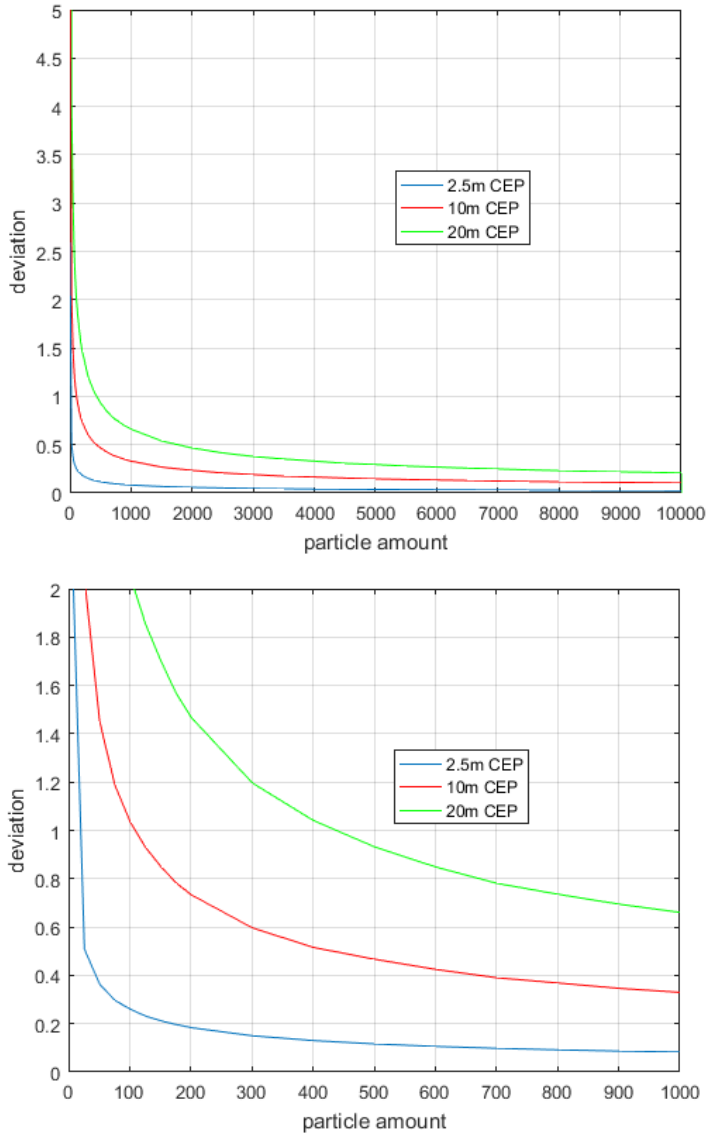


Figure 7.19: The particle amount to the position error relation when using different standard deviations and particle numbers. The right side figure shows a zoom of the left side.

Results: Figure 7.20 depicts the position error of the sensor fusion particle filter track.

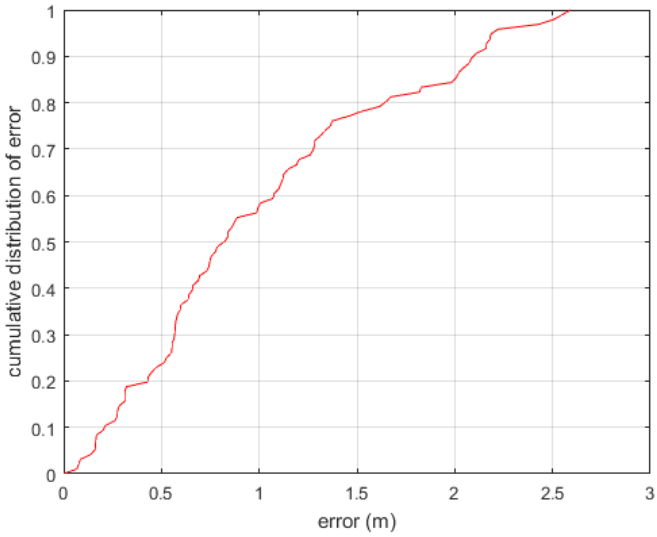


Figure 7.20: The particle filter fusion track accuracy when using 100 particles.

The track is more accurate than the individual systems by themselves. The initial lack of knowledge of a position and heading makes the system to first evolve towards the correct heading. When the first position fix comes from the BLE subsystem the particle filter is reinitialised according to the BLE position and errors. The headings are kept in initial values as they already represent an uniform distribution. After few steps, the particles moving towards wrong directions die and are transformed into the higher weight particles.

The wrong turns and misfit step length provided by the inertial sensor subsystem are compensated by the wall constraints and the particle propagation and resampling properties. Moreover, the position fixes kill particles propagating in the wrong direction.

7.4 Map Matching

Map matching is located in the availability block. It is processed after the pentagon buffer or particle filter updates. For the particle filter each particle position is compared while for pentagon buffer the collision logic compares the buffer points and the solution point. The pentagon buffer is presented in the next chapter.

Two checks are made. First is whether the point is inside a wall polygon. Second is whether the line formed by the current and previous point locations crosses any of the wall polygons. If either of the two checks are true then a collision has happened. In a particle filter this initiates resampling and with the pentagon buffer the buffer reduction is conducted.

7.5 Conclusions

The top-level particle filter system design and the implementation is proven to function well. Even with inaccurate sensor subsystems, with the help of the context detection and data association, the result is more accurate. Thus far only zero-update flag and stairs flag have been used in the context engine. Zero update flag controls and enhances the inertial mechanisation and triggers the particle filter step update. Stairs flag adjusts the particle filter and the zero-update detection method. Process and Measurement noise are increased in the inertial only Kalman filter and propagation characteristics are made looser in the particle filter. Zero update detection thresholds are adjusted for the stairs. The integrity check feature with UWB

sensor subsystem was placed in the Availability block seen in Figure 7.1.

With Bluetooth low energy fingerprinting it was noted that the deployment of the anchors was not successful in the lobby area. The same was true with the placement of the UWB anchors. When the user was walking in the lobby area, in the southwest corner, most of the measurement results pointed to a wrong location. It is clear that an extra BLE anchor would be needed here to enhance the system in resolving the solution in the open lobby area.

The UWB anchor placement geometry was poor. This with noisy range measurements made the UWB solution worse than it could be with wiser anchor placement. The BLE system accuracy with the constrained area reached 3 to 4 m level. UWB accuracy in this test indicated 2m accuracy level. The inertial system with update rate of 62.5Hz and 50Hz produced especially inaccurate turns. The 100Hz tests did not have as bad a turn tracking.

The particle filter is very robust and fails rarely. This is true if the propagation characteristics are kept loose. If the sensor subsystems are much more accurate than the ones used in this chapter the particle filter propagation and update properties can be kept tighter. The Tukey update performs better than the Gaussian update when the position measurements are not precise. The Tukey update lets the particles to follow the inertial mechanisation more inside the Tukey range.

Next, we look into the Kalman filter fusion.

Chapter 8

Kalman Filter and Pentagon Buffer

A Kalman filter is presented here with a pentagon buffer for faster computation, especially for outdoors and areas with less wall constraints. A 15-Error-State Kalman Filter tracks the attitude, velocity and position errors. In addition, biases for the gyro and accelerometer are taken into account. In the previous chapter we mentioned the inertial-only error state Kalman filter that uses zero velocity updates. In this chapter, a copy of this Kalman filter is introduced. The copy runs in parallel, the zero updates are applied and in addition updates from the position fixing sensor subsystems. Moreover, a parallel novel approach for position error tracking is presented together with a track form matching method for GNSS track quality determination. These are called the Pentagon buffer and the Track Form matching methods respectively.

8.1 Design Basis

Figure 8.1 depicts the block diagram of the Kalman filter design.

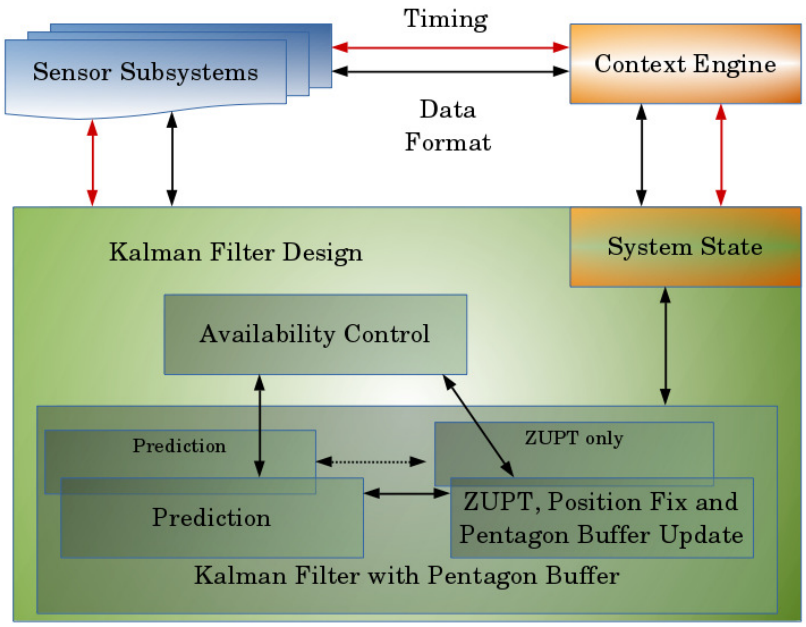


Figure 8.1: The Kalman Filter with Pentagon Buffer design block diagram.

The sensor subsystems included in this chapter study are the IMU and GNSS subsystems. The data and timing agreements between the blocks are the same as described previously. The context engine uses ZUPT and Collision logic correction parts.

The initial value is given for the heading. The first position fix is taken from the GNSS subsystem. As mentioned, the two Kalman filters run in parallel. The first one shown, in the back, is the same that was running with the Particle filter. This Kalman filter updates the inertial solution only with ZUPTs. Another Kalman

filter uses not only ZUPTs, but also position fixes outdoors from the GNSS sensor subsystem. The novel approach for error tracking and interaction with the map is called the Pentagon Buffer approach. We discuss this next.

8.2 Pentagon Buffer enhanced Kalman Filter

The Pentagon Buffer (PB) can be described to be similar to a Rao-Blackwellised Kalman Filter and an Unscented Kalman Filter. When the UKF propagates the sigma points for position, velocity and attitude, the pentagon buffer points only follow the position. This saves computational power and the filter doesn't suffer from dimensionality, which is the case with UKF as mentioned by Briers et al. (2003).

Figure 8.2 shows the Pentagon Buffer approach.

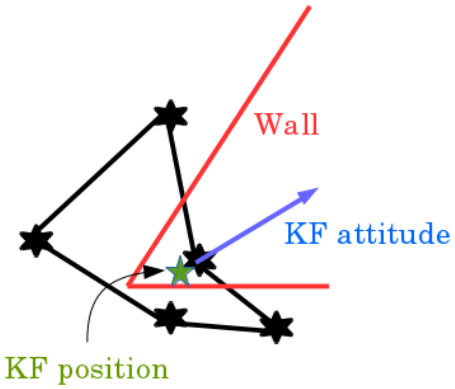


Figure 8.2: The Kalman Filter solution with Pentagon Buffer interacting with the map. The Kalman filter solution is shown in the middle. Five buffer points surround the solution at fixed angles. These five points are the 3-sigma points in their directions. The distance to these corners points change according to the inertial propagation characteristics and the positioning fixes.

The position solution is propagated each epoch, along with the Kalman filtered error estimates. This defines the centre of the Pentagon Buffer. When the first position fix from either the GNSS receiver or other sensor subsystems arrive, the position solution is initialised. Together with the position, the error estimates from the position fixing system are used to initalise the Pentagon buffer. The pentagon buffer has five corners. The distance from the solution to the five corners changes according to the pedestrian dead reckoning error propagation and the positioning fixes. The corners are independent of each other and thus the design has more degrees of freedom compared to for example an ellipse estimation of the error. From the centre, each corner is defined by a distance value. The angles for the corners are derived as explained in section 8.5. One corner is the direction of movement when the Kalman filter and the pentagon buffer are initialised. The distance values are derived initially from the standard deviation of the x and y components of the positioning fixing system error estimate. The points indicate the 3-sigma points in each direction. These are derived using the standard deviation of the first position fix. One corner is initialised pointing to the same direction as the foot yaw angle.

When a position solution is propagated by estimating the error using the Kalman filter, the Pentagon buffer points are propagated as well. This Pentagon Buffer points position propagation happens at the same triggering events as the particle filter propagation. These were either when step is detected or when a new position update arrives from the position fixing sensor subsystems.

The Buffer thus travels around the main solution. Although the foot yaw angle

changes and the direction of movement, the Pentagon buffer angles stay at fixed angles around the position solution. The angles do not follow the changes in the direction of motion. This is because the corners represents the position error. Similarly to the particle filter particle cloud the five corners are propagated independently. This means that the Pentagon Buffer points do not rotate although the main yaw solution in the centre turns. This imitates the behaviour of the particle cloud edges as can be seen in the image 5.3.

8.2.1 Pentagon Buffer Propagation

The Pentagon Buffer points' distance values from the centre are adjusted at each Buffer update. The distances are increased according to the perturbation model derived for the particle filter. These were defined to increase according to the travelled distance. Thus, since the last update the distance indicated by the mechanisation of the IMU measurements define how much the points' distances are increased. These are maximum estimates or the 3-sigma estimates from the perturbation study. Each direction grows at every Pentagon Buffer update. This buffer approach is acquired from the empirical perturbation model study, while the Kalman filter variance can be very optimistic on the error growth. Thus we have two error estimates for the position error behaviour.

Moreover, the Pentagon Buffer points positions are compared with the 3-sigma points of the positioning fixing systems' measurements. If the Buffer points are too far away they are returned to the 3-sigma range of the position measurement up-

dates. This is applied if the position fixing measurement quality value is trustworthy (threshold value).

8.2.2 Pentagon Buffer Collision Logic

Simple collision logic is introduced. When a wall is encountered as in Figure 8.2, the Pentagon Buffer point distances are reduced until either they are out of the wall or they become zero. If any of the corners become zero we have encountered a collision. Now, the yaw of the Kalman solution (Blue arrow in the Figure 8.2) is compared with each of the angles of the Pentagon Buffer corners.

The position solution is moved out of the wall, perpendicular to the blue arrow indicating yaw angle, towards the Pentagon Buffer point that had an angle closest to the direction of movement. This is, if the distance of that corner was greater than zero. If all the three closest corner distances are zero we move the position solution towards the closer corner that was behind. In the Figure 8.2 this is either of the two corners behind the direction the user was walking to. The corners are just outside of the sharp corner of the wall. The corner that has zero length is not included in the correction selection because the wall is likely to be on that side of the buffer.

The position solution is moved out of the wall, similarly as in the work presented by Mansour & Waters (2013) by applying a position update, to either side which had closer angle to that of the yaw angle. The distances are corrected after the corrective move so that no corners remain inside the wall or obstruction.

Figure 8.2 represents a problematic case. Either side of the obstruction could

be a possible correct solution. For now, we make a decision to choose the corner with the closest angle to the direction of movement. We will develop a novel filter structure and a more extensive collision logic in the chapter that follows, to take into account the more complex wall structures.

8.3 Evaluation

Three approximately 700-meter walks were conducted. One track result using the Pentagon Buffer Kalman filter with GNSS position updates is shown in Figure 8.3.

The track ends in front of the main door of the Nottingham Geospatial Building. This is seen next to the last epoch of the Error State Kalman filter, with the pentagon buffer drawn on the map. The track begins from the bike stands.

The pentagon buffer corrects the Kalman position solution according to the buffer collision logic. This is executed by applying a position correction measurement update defined by the Buffer collision logic to the Kalman filter. This is a position update for the Kalman filter, where the measurement z is the difference between the current position inside the wall and the corrected position outside the wall that was derived by the collision logic block. H is the measurement update matrix and involves only the position term. R indicates a very accurate measurement error, in order to move the solution out of the wall.

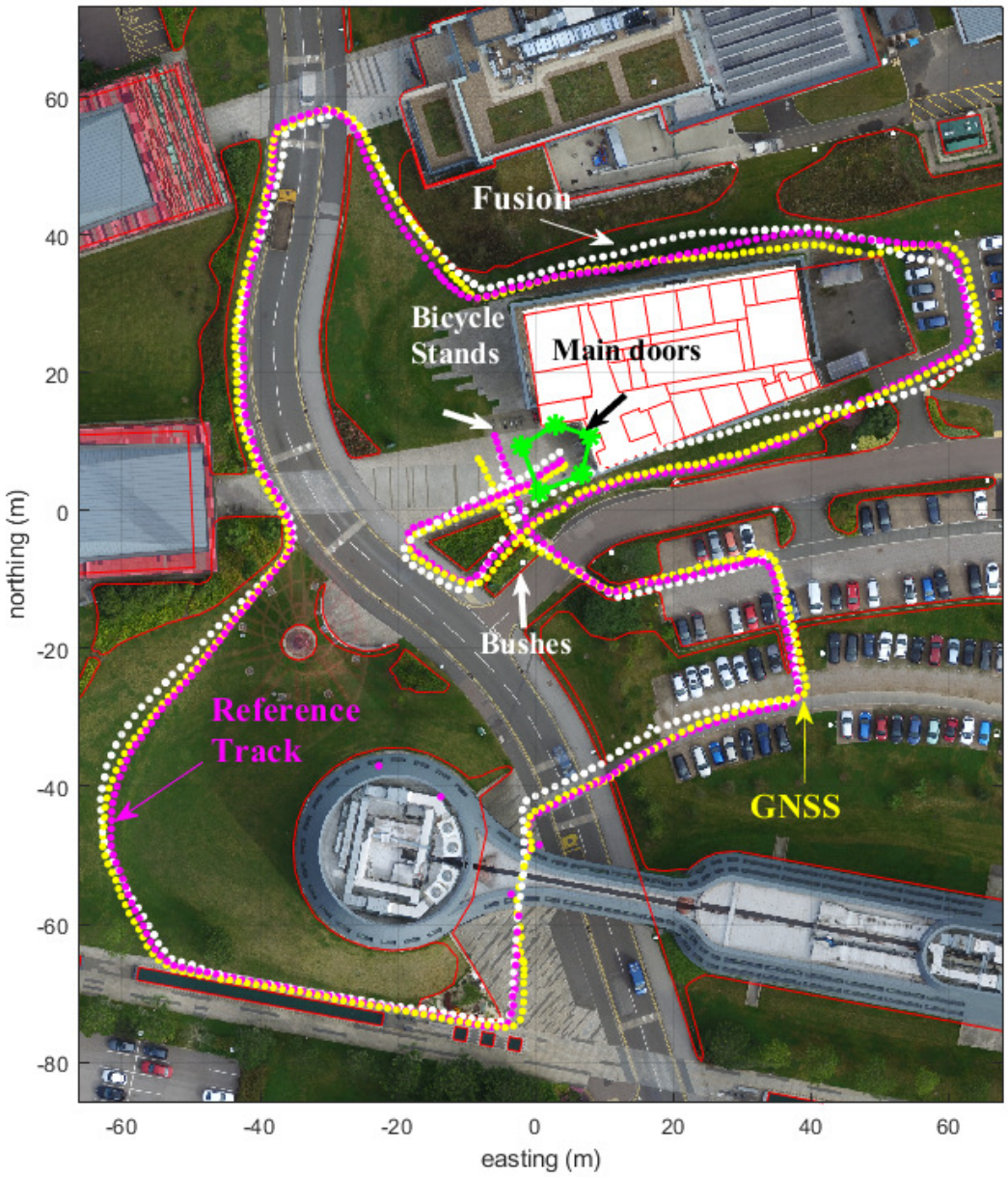


Figure 8.3: Error State Kalman Filter solution with Pentagon Buffer (white). GNSS track (yellow). Reference track using Leica RTK kit (magenta).

The collision with the wall, the constraint, and the weighting supresses the position error variance. This is accepted although the position error variance may become too small. The pentagon buffer is not affected by the Kalman filter position

error update and maintains the error estimation limits before and after the collision. The buffer is changed by the wall collisions, the position update 3-sigma limits and the error propagation of the pedestrian dead reckoning.

It can be seen that the cold start position solutions of the GNSS receiver are off the true track and make the Kalman solution to go first towards the bushes. The Pentagon Buffer helps by applying position updates that transfer the solution back to the pathway. Afterwards the GNSS offers very good quality position solutions. We can see the accuracy of the GNSS receiver and the accuracy of the Kalman fusion between inertial and GNSS solutions in Figure 8.4.

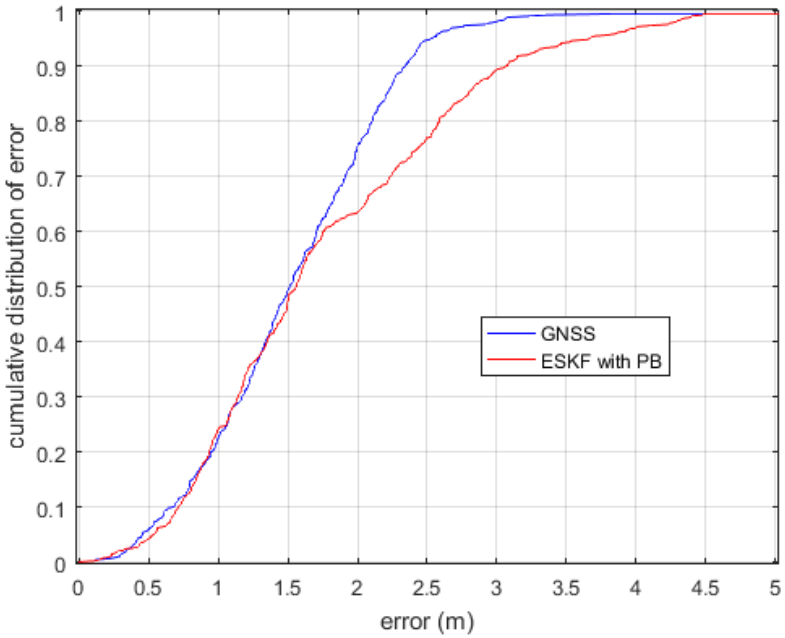


Figure 8.4: Single GNSS receiver and GNSS and IMU Kalman fusion accuracies.

We notice that adding the inertial sensor makes the accuracy worse. This is clear when we examine the inertial solution. This is shown in Figure 8.5.

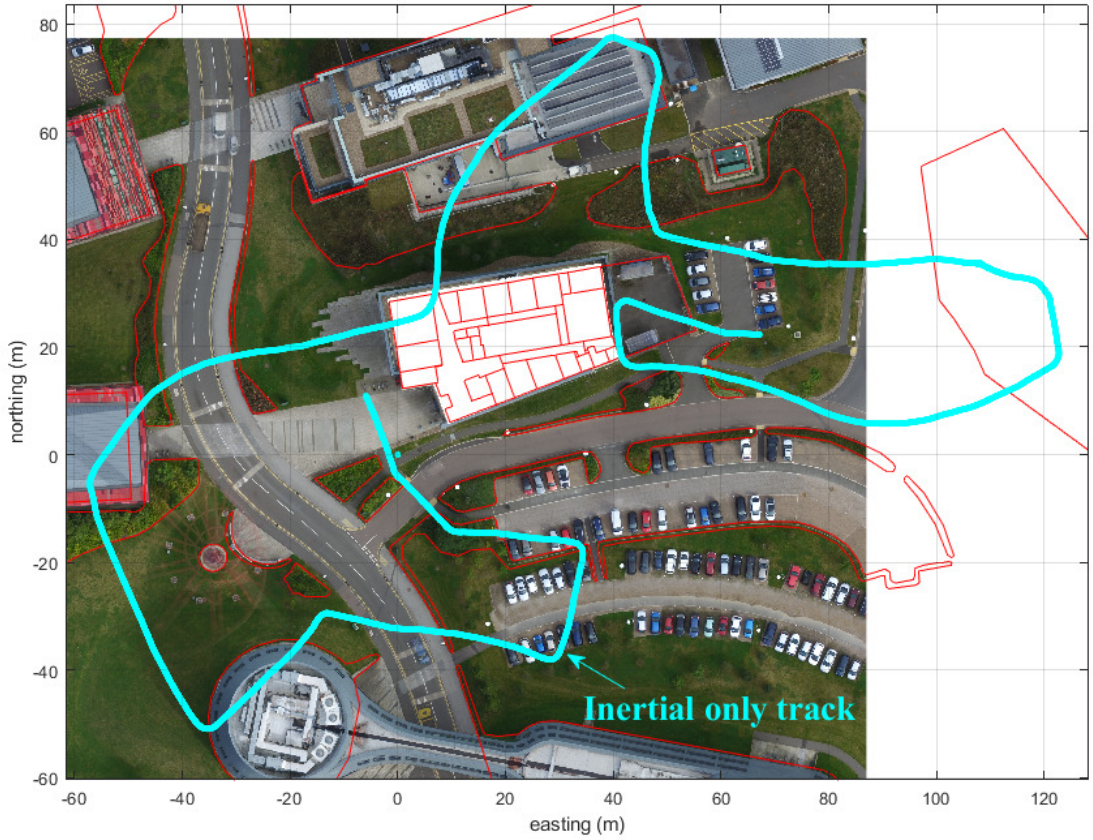


Figure 8.5: The mechanisation of the IMU output swerves slowly towards right.

This was the case for all three walks. The main cause is assumed to be the tape strapdown method onto the foot top or the used Zero Update detection method. The Microstrain cable was hanging on the outside of the user's left foot which produces unwanted motion on the attached IMU as well. This happens especially when jumping off the ground and when the foot lands. In addition, the used sensor saturates already at $\pm$ 5g. The acceleration at the hit with the ground can sometimes be more than 5g, and this is another possible error source for missing mechanisation information.

The used GNSS receiver produced a questionable result only once that seemed to

indicate coming from a multipath condition in between multiple artistically shaped buildings. This can be seen in Figure 8.6.

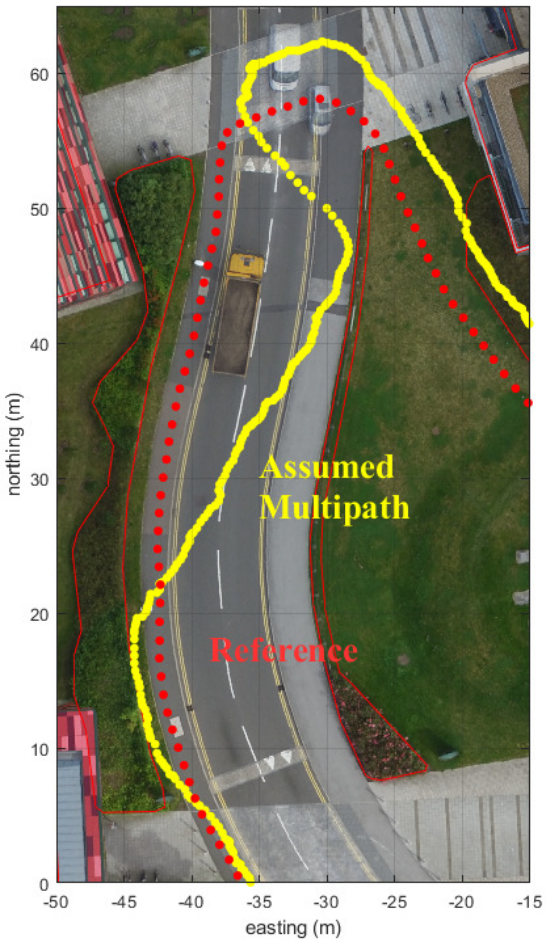


Figure 8.6: Assumed multipath condition noticed with the used GNSS receiver.

The correct track continued along on the pathway west side of the road. The receiver solution on the other hand continued across the road. A sudden drop in the horizontal accuracy was noticed at the top north side of the track and the position solution moves rapidly towards the correct track position. This happens after approximately 50 meters from the beginning of the "multipath" error. Possibly

one satellite with heavy multipath range results got completely blocked and was switched to satellite with better quality range results.

The fused solution is thus very useful, if and when the GNSS solution is lost or is degraded. Although the inertial solution swerves a bit to the right, it is still very useful if GNSS is lost. We simulated two cases for Figure 8.3 walk, a gap of 50 seconds and a gap of 100 seconds. Figure 8.7 shows a gap of 50s. Figure 8.8 shows a gap of 100s.

We can see that if the IMU track is of a good quality not all GNSS updates need to be used. The accuracy degradation is shown in Figure 8.9.

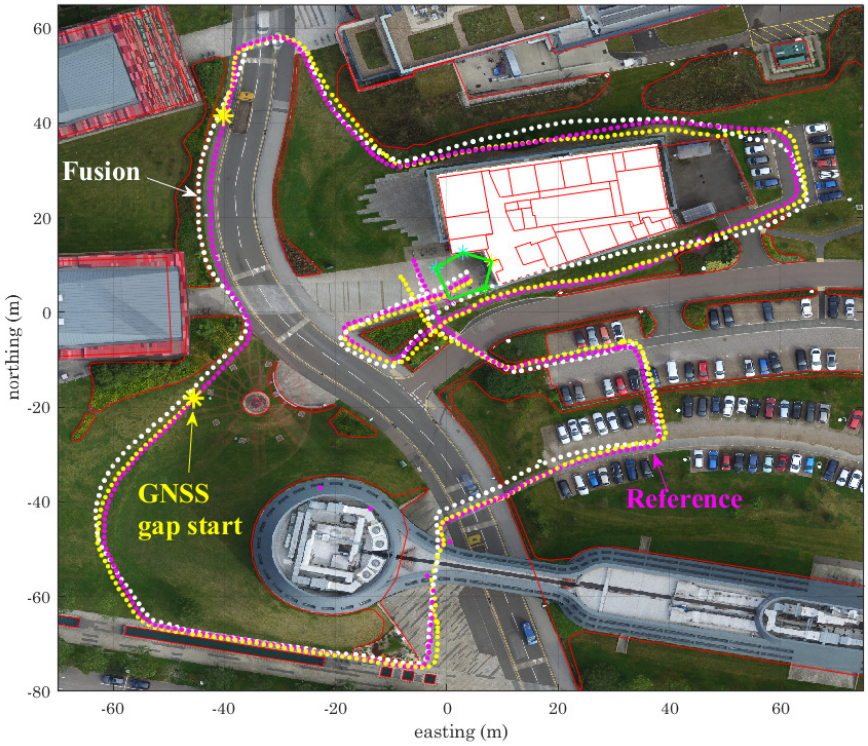


Figure 8.7: Simulated GNSS gap of 50 seconds. Yellow stars indicate the start and the stop of the gap.

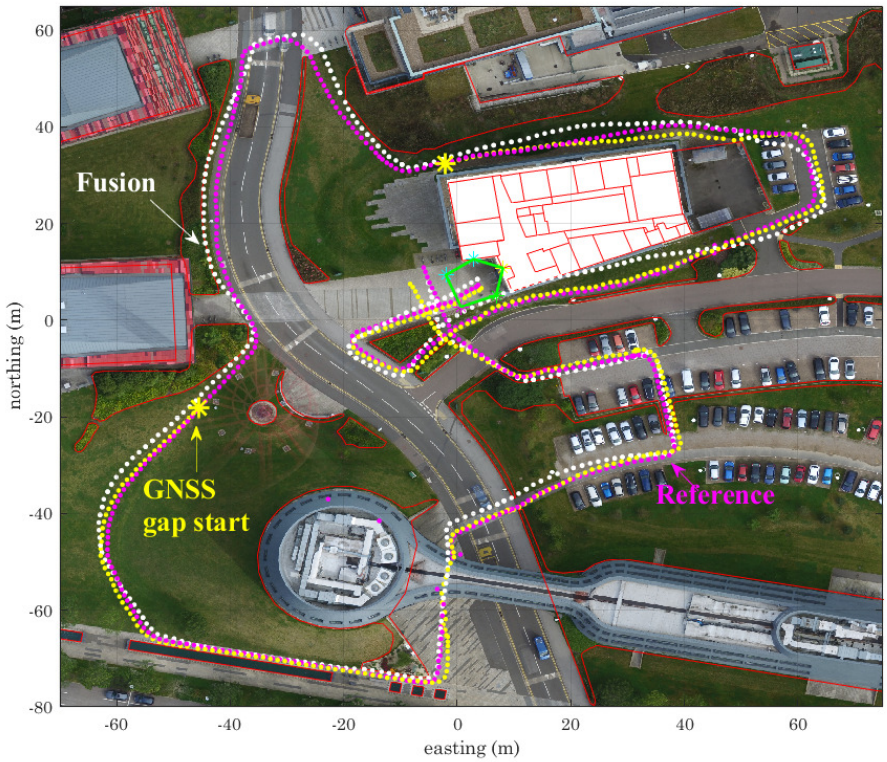


Figure 8.8: Simulated GNSS gap of 100 seconds. Yellow stars indicate the start and the stop of the gap.

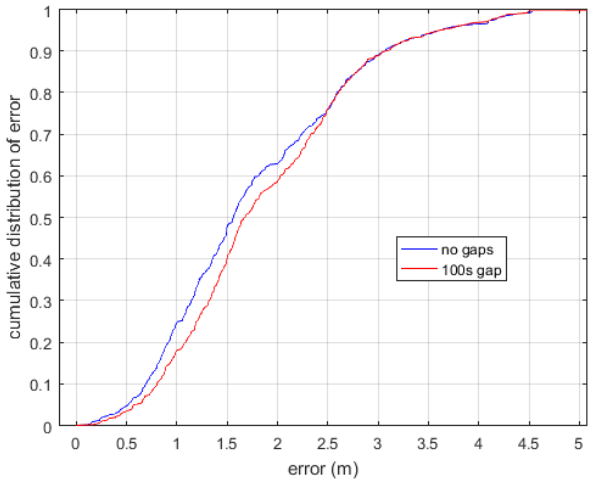


Figure 8.9: 100s gap in GNSS does not degrade the fusion accuracy too much.

One question remains; how to detect the GNSS error, when the GNSS receiver

cannot resolve its state or true quality of the positioning measurement.

We now propose a method for checking the integrity of the GNSS track by presenting a method on how to detect this GNSS receiver condition described by the Figure 8.6.

8.4 Track Form Matching

Inspired by the work of Lan & Shih (2014) a track form matching method was developed. The inertial track has a tendency to drift after few tens of seconds. This drift or error depends on Zero update detection quality, sampling rate, the sensor saturation, amount of turns taken and the speed of movement. The good side of the inertial mechanisation tracking is that the solution can be trusted within this time window of tens of seconds.

On the other hand, the GNSS receiver position accuracy is not very good (deviation of approximately 10m) within this time window. But on a longer term, within a time window of few hundreds of seconds, the GNSS receiver provides the constant positioning accuracy while dead reckoning will get lost. Both long-term and short-term properties can be used by examining the track forms. We present here two approaches in detecting the GNSS track form perturbations and inertial track inaccuracies within short and long-term time windows.

8.4.1 Short Term GNSS Track Inaccuracies

A typical error in the walks was noticed with the used GNSS receiver. This was shown in Figure 8.6. This deviation should first be detected and then acted upon.

Two trailing points technique was used to examine both the inertial and GNSS tracks. This is depicted in the Figure 8.10.

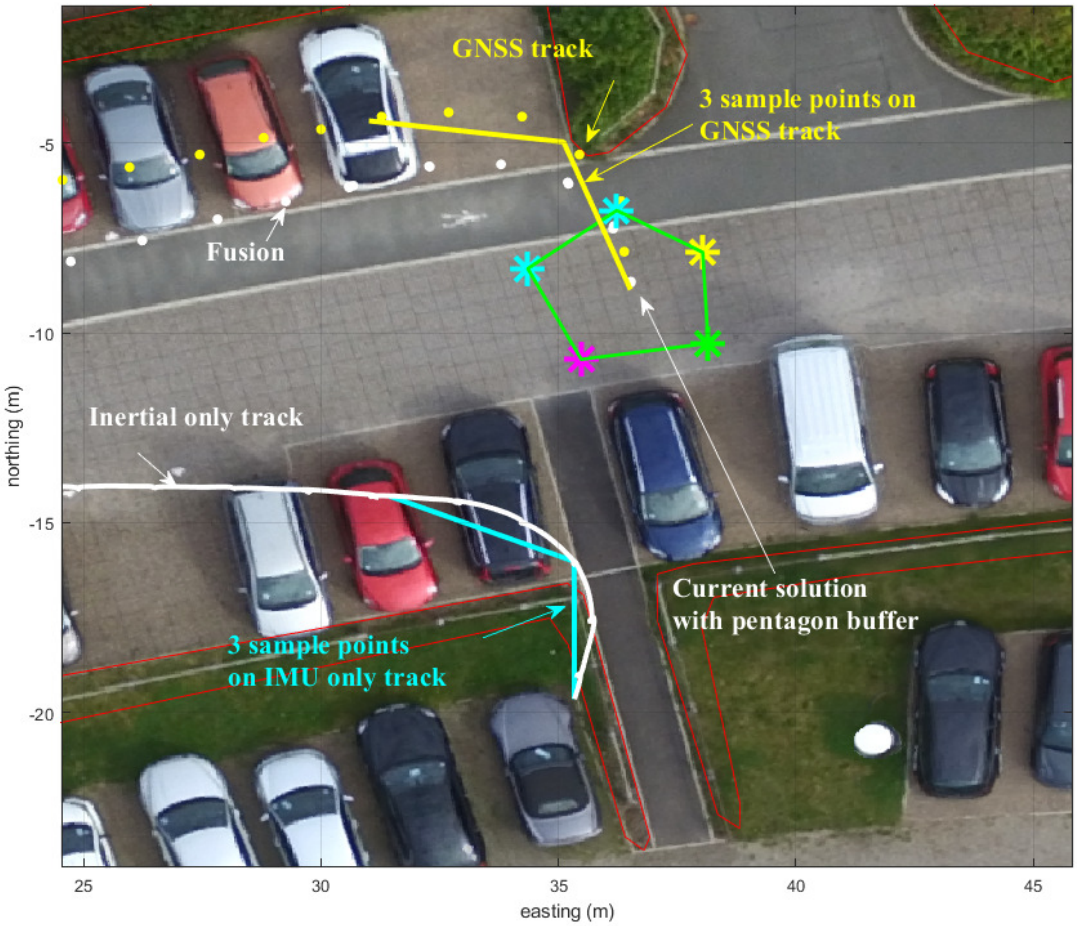


Figure 8.10: Three point samples were taken on both the inertial only track and the GNSS receiver tracks. Matching the two track forms gives an indication whether short term GNSS track quality is valid.

The difference of angles between the two track forms is derived. The angle for

one track is the angle between two lines formed by the three points. The Figure 8.11 shows the comparison between the angles between the GNSS and the inertial trails.

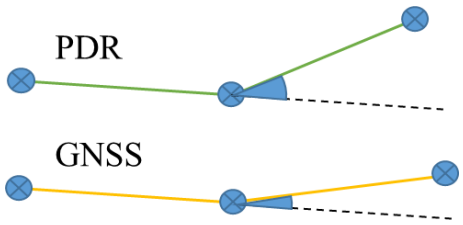


Figure 8.11: The angle between the GNSS and the inertial (PDR, pedestrian dead reckoning) tracks are compared.

Figure 8.12 shows the result of the angle differences between GNSS and inertial only tracks throughout the walk.

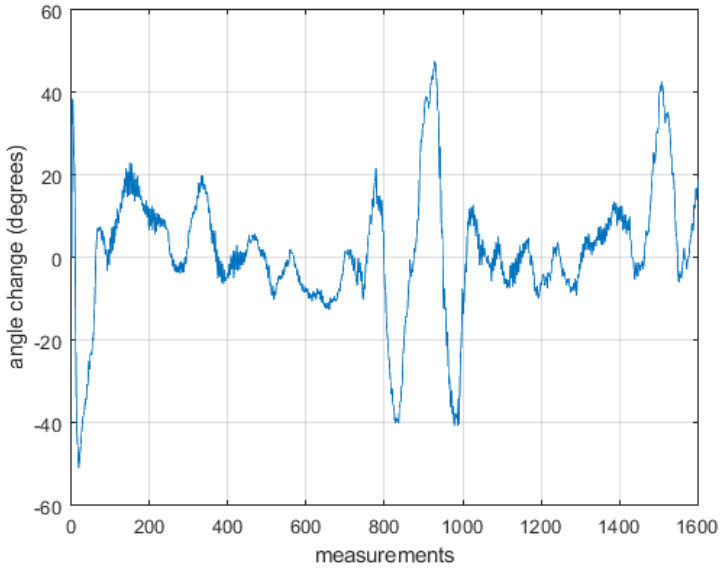


Figure 8.12: Angle difference between the three point samples on GNSS and inertial tracks.

A threshold defined a condition when the GNSS measurements were no longer trusted. This was set to 30 degrees. The 15-error state Kalman filter is used. The

GNSS measurements were weighed down while this condition occurred, which is why during the condition the GNSS measurement do not influence the fusion. The fusion trusts the inertial dead reckoning more during the condition. A delay of 30s was applied before turning the angle detection ON again and before returning to normal GNSS weighing. Figures 8.13 and 8.14 show the resulting tracks before and after the downweighing of GNSS measurements was turned on. The yellow stars indicate when the track form detection method downweighed the GNSS measurements.

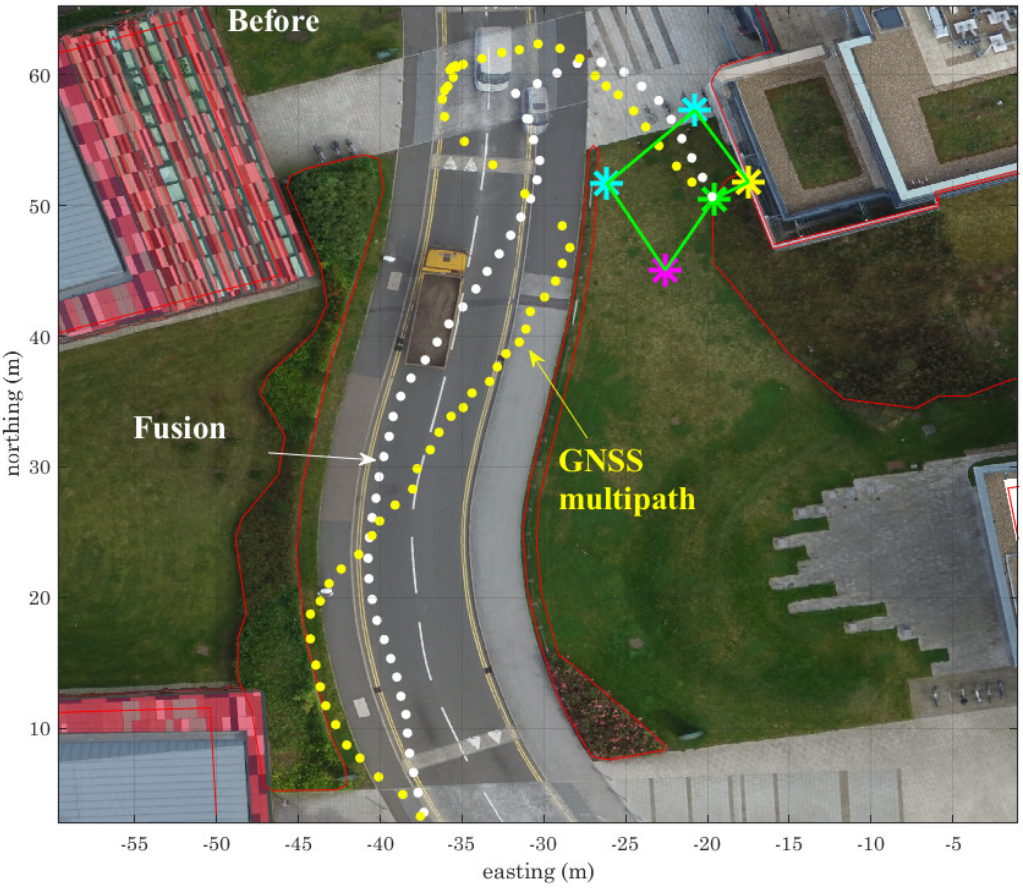


Figure 8.13: The perturbation in the GNSS trail is detected using the method.

It is clear that for this single case the switch works well. The resulting track

is closer to the true track when the GNSS updates are downweighed. The method is based on the more accurate performance of the inertial tracking over GNSS on shorter routes. It has to be remembered and also added into the code that this has a prerequisite that the user hasn't shuffled his feet or otherwise taken too many dynamic turns.

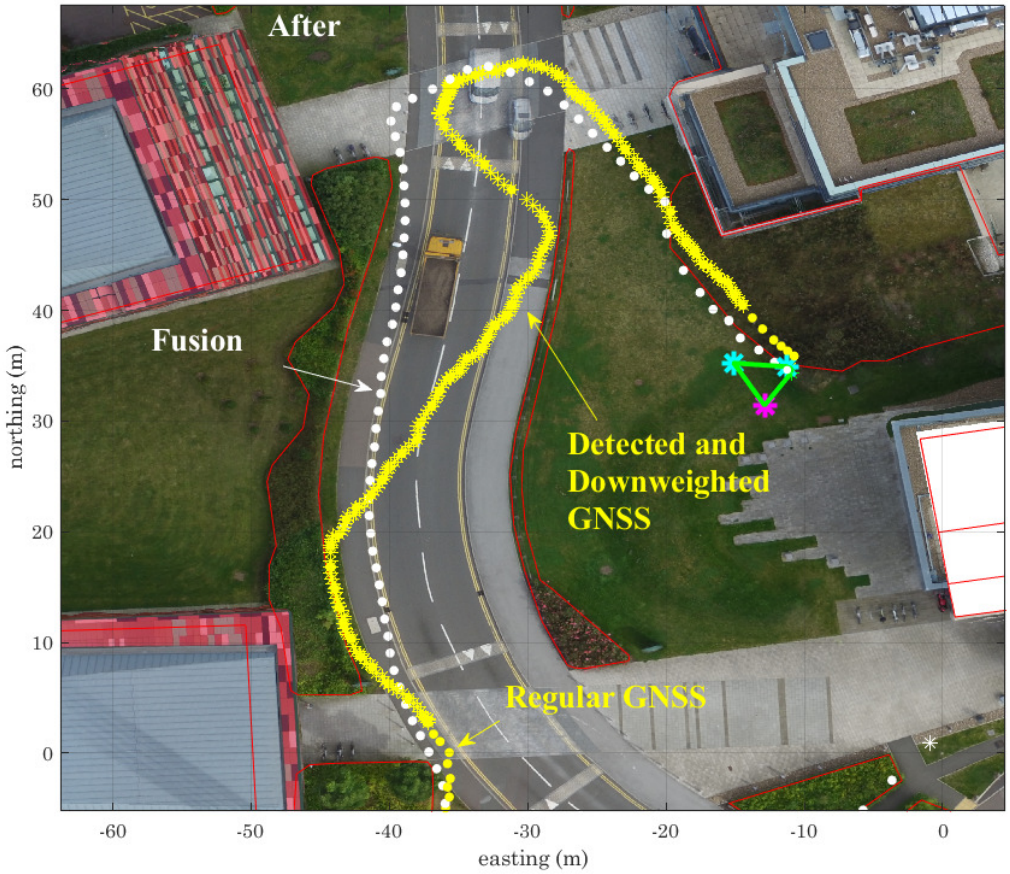


Figure 8.14: The resulting track after downweighting the GNSS measurements.

8.4.2 Longer Term inaccuracies within the IMU Track

A longer seven-point track form matching was applied in order to examine the effect of the IMU track drift. This is shown in the Figure 8.15.

Each of the six sections of the selected track were examined. The inertial track was fitted onto the GNSS track fixing the first two points as shown in the Figure 8.15. The cumulative error on either left or right side of the GNSS track was gathered. This error then indicated how much the inertial track had turned to the wrong way during the selected time window.

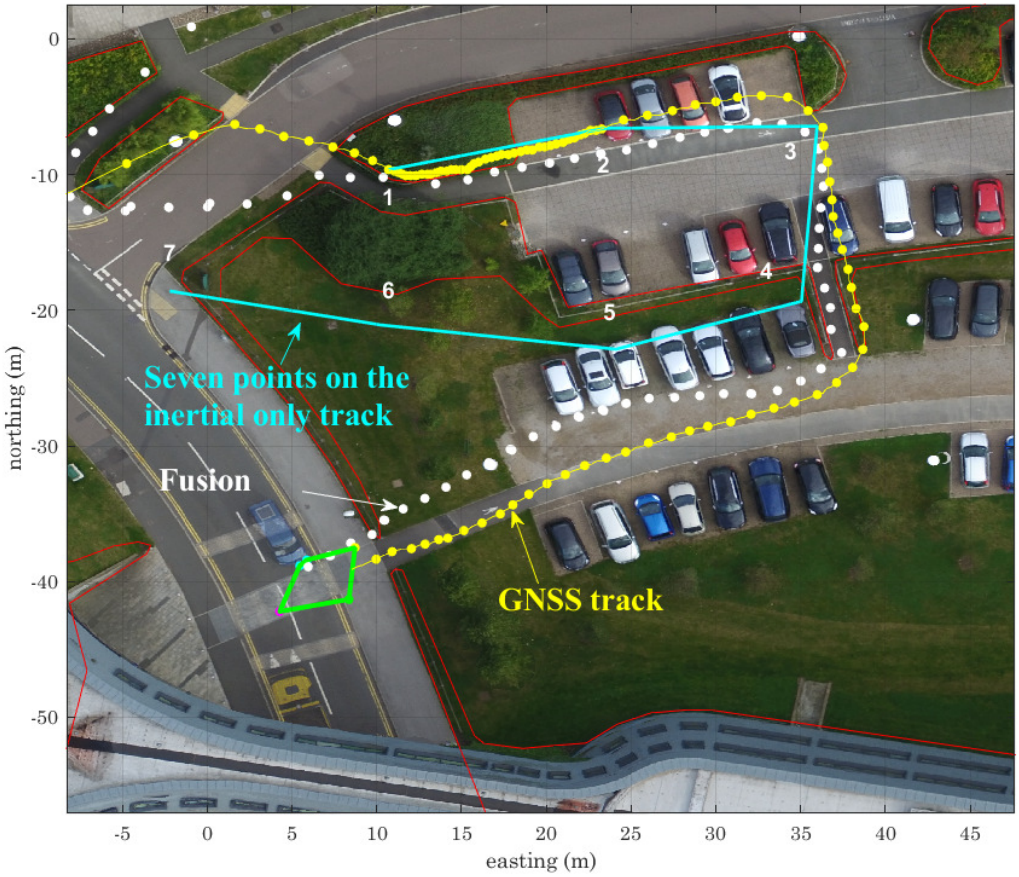


Figure 8.15: Seven points on the Inertial and GNSS track examine the long-term effect and quality of the inertial track. In this case the inertial track was all the way on the right side of the GNSS trail. Thus the correction is a turn adjustment to the left.

A corrective small yaw turn was applied as a measurement in the Kalman filter each step after this detection, depending on how far away the last points on the

two tracks were from each other. The Kalman filter 15-error state vector has the accelerometer and the gyro biases. These are constant throughout the walking tests. In future research the corrective yaw turn could be applied via changing the bias values instead of measurement updates. The track form matching correction turn update was executed at each 10s time interval window.

After walking forward, the inertial and GNSS track forms match better. This is seen in the Figure 8.16 and 8.17.

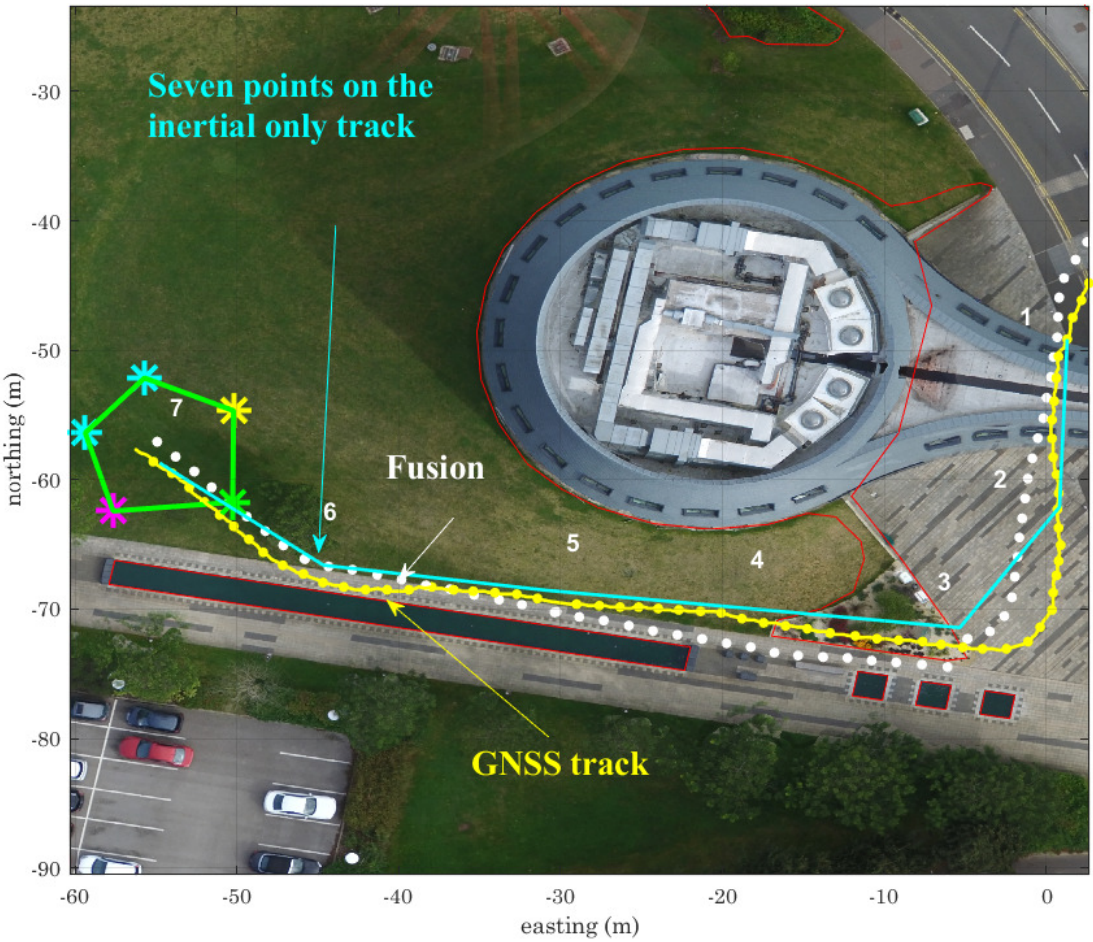


Figure 8.16: The applied correction suppresses some of the turn bias and brings the inertial track form closer to the GNSS track.

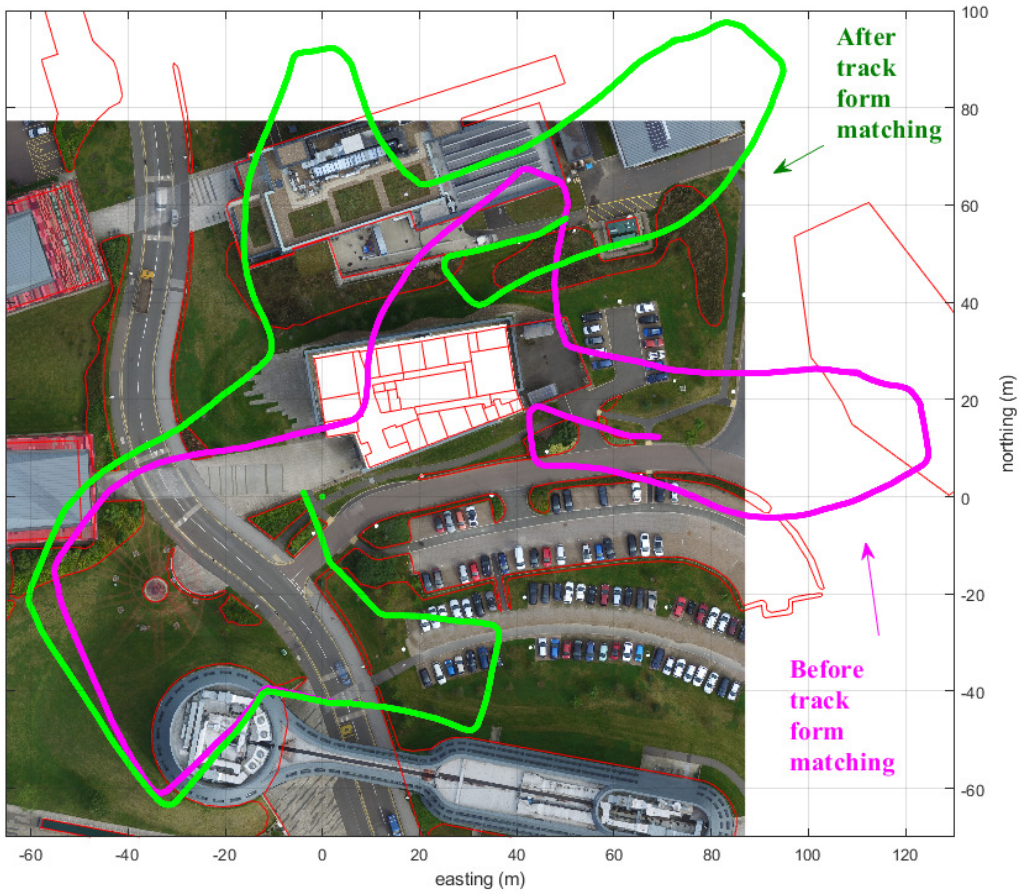


Figure 8.17: The magenta track shows the inertial only solution and the green track shows the inertial solution with the track form matching stepwise yaw correction.

The longer-term sectioning track form matching corrects the inertial solution if it has a tendency to turn repetitively onto either side. In some of the walks though, the inertial track turn bias does not remain constant. For example, a slip while walking and turning on a gravel layered ground could make the mechanisation misread the turn. Rather the total turn error is more dependent on individual turns. Some turns have errors that make the mechanisation result to swerve more to the other yaw direction. An improved track form matching approach is introduced in the next

chapter.

This method demands a good quality GNSS track. Thus, it can't be used when the short-term track form matcher indicates deformation in the GNSS track.

The resulting track is shown in Figure 8.18 when both the short-term and the long-term integrity measures were combined. the resulting track is shown across the zone where the GNSS solution is affected by the multipath. The long-term fixing was not used while the GNSS perturbation was detected.

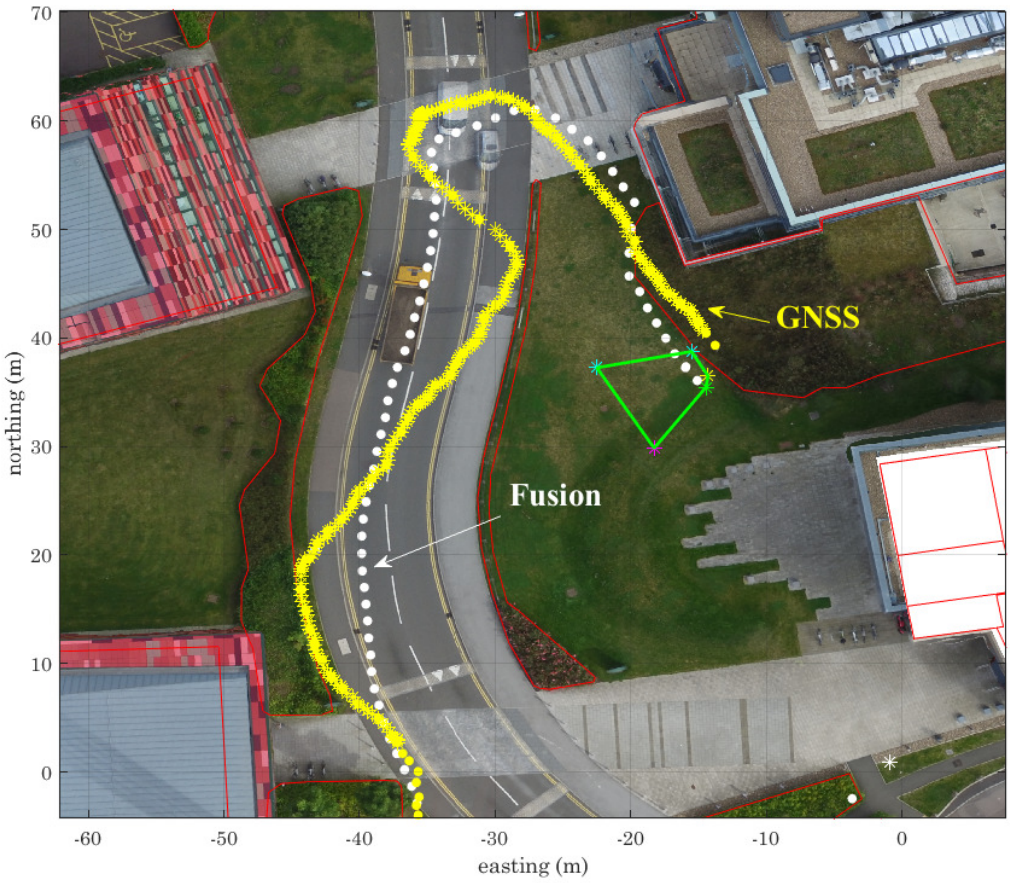


Figure 8.18: Inertial track is trusted when GNSS track fails. The corrective turn, defined by the long-term track form matching, on the inertial track is applied at every step.

8.5 Using Pentagon Buffer properties in Initialisation

We can exploit the dynamic features of the buffer when converting from the particle filter into the pentagon buffer enhanced Kalman filter. Figure 8.19 represents a switch scenario.

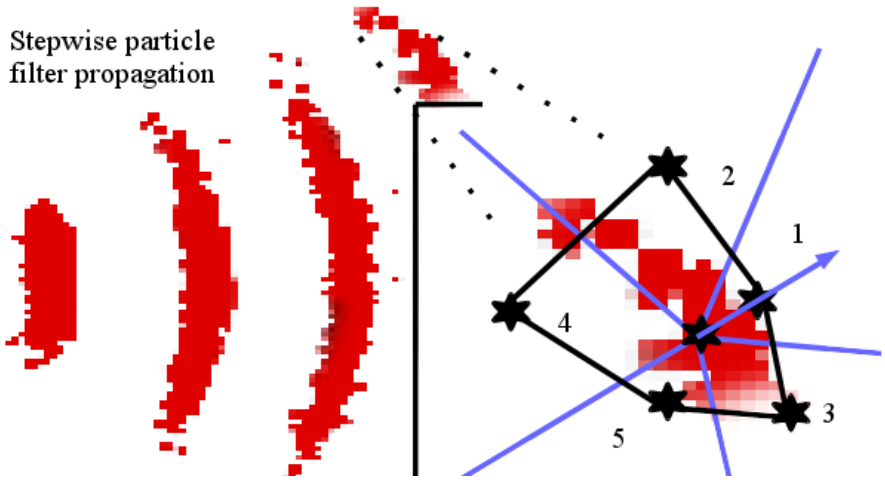


Figure 8.19: Pentagon buffer properties will retain more information of the particle cloud than the Kalman filter covariance alone.

When the switch trigger evaluates as true, the Kalman filter is initialised first. Then a buffer is constructed. The position solution for the Kalman filter is the particle filter weighted average of particles' positions. We divide the particles into five sections according to the direction of motion information from the particle filter. The direction of movement is shown in blue in the Figure. For each sector, the standard deviation of the particles in that sector is calculated. Now we can derive five 3-sigma values for each sector. These are then used to initialise the pentagon buffer point distances.

Also, when converting into particle filter, particles can first be spawn and then the pentagon buffer can be used as a polygon for map matching. The particles outside of the pentagon buffer are then resampled into the pentagon buffer. This creates a particle cloud that again retains more of the error information than just simply using the Kalman covariance. Sectorwise particle seeding is also an alternative method for conversion.

8.6 Conclusions

We presented a pentagon buffer to fix the position when the main solution collides with the map walls. This approach, using collision logic works by moving the solution out of the wall. In some cases, the filter updates the position into one of two possible locations like that in Figure 8.2. This is a problem, since we cannot be sure which is the correct location. In the next chapter, a solution to this is presented by a combined usage of both the particle and the Kalman filters and an improved Pentagon Buffer collision logic.

Two track form evaluation methods were presented. The short term inertial and the GNSS tracks are compared. The short-term method evaluates the quality of the GNSS track according to the short inertial track. The detection method shows whether it is feasible to use the GNSS measurement update. This method was implemented in the availability check box shown in Figure 8.1.

Similar to the short track form matching the longer-term track form matching

checks the yaw swerve tendency in the inertial track. In this chapter, if the inertial track seems to be turning too much on either to the left or right, we applied a corrective yaw correction each step. A new developed method is introduced in the following chapter instead of the stepwise yaw correction.

We have presented the particle and Kalman filter methods for indoors and outdoors respectively. Moreover, the Tukey style update and track form matching methods were introduced with preliminary tests. Now in the chapter that follows, all these used filters and methods and their good sides are now next designed into a system for seamless indoor-outdoor navigation.

Chapter 9

Seamless Navigation Filter

The third and final design iteration of the system combines the three previous chapters. The context engine in its entirety is brought to control the usage of the pentagon buffer enhanced error state Kalman filter and the particle filter. The design is named either as the tripartite filter, referring to the inertial only, the Kalman and the particle filters, or as the dual-mode filter, referring to the particle or the Kalman filter that produces the final fused solution. The big picture of the final design is presented first, showing the function and interconnections between all three combined: the context inference, the navigation filters and the sensor subsystems.

Following this, the outdoor performance of the dual-mode filter with the track form matching is examined. The particle filter is further examined and an enhanced version of the BLE sensor subsystem tackles the issues encountered in the previous chapters. UWB anchor geometry is reconsidered; and finally, the context engine operation is evaluated.

Parallel execution is considered, and whether the mobile devices have capabilities for this. How to use the available computational power most effectively is discussed. Track form matching is the technique that was developed to aid correcting errors caused by multipath. Multipath detection using track form matching deals with how to detect the multipath condition and when to command the context engine to tackle it.

9.1 Design Basis

Figure 9.1 depicts the block diagram of the tripartite filter design.

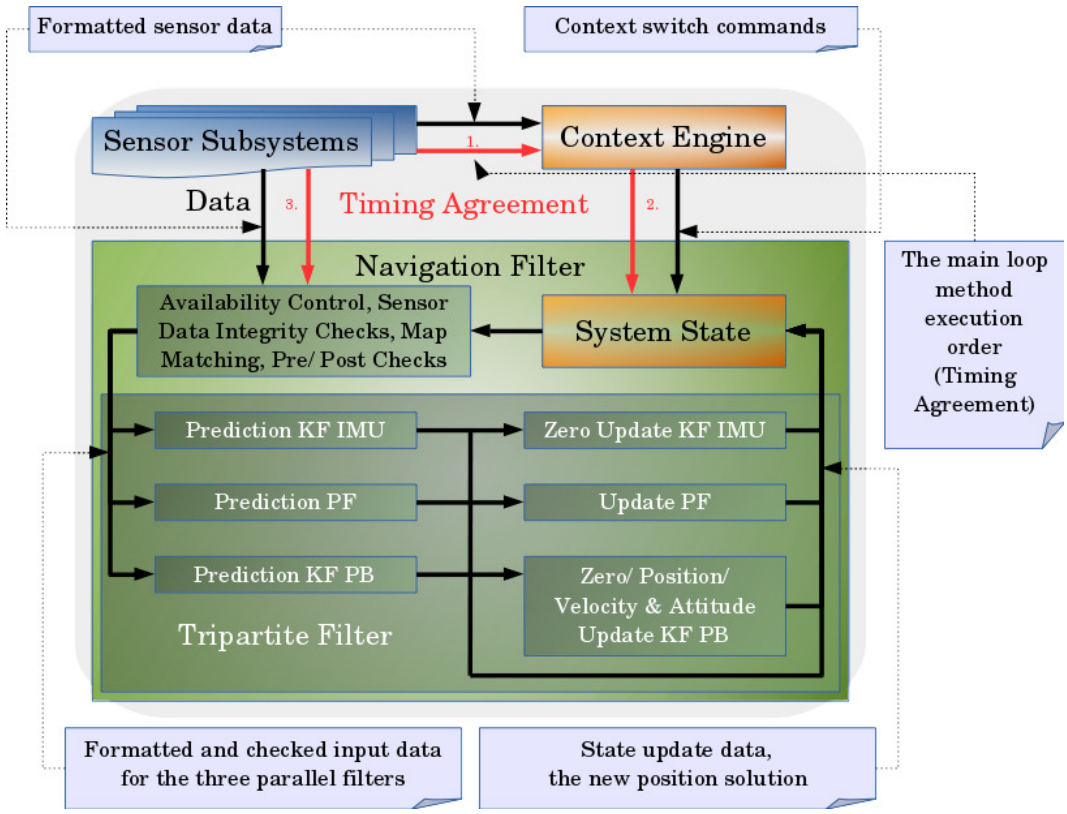


Figure 9.1: The tripartite filter shows the final design of the seamless context-sensitive navigation filter for indoors and outdoors.

The modified sensor subsystem versions are explained later. The data interface agreements remain the same and data flow is away from the sensor subsystems. The context engine asks for the position information from the navigation filter, but does not modify the position. The context engine changes the state of the navigation filter according to the current measurements and the history of position solutions. The timing agreement is shown in Figure 9.1. First, the context engine derives the necessary inference using the sensor subsystem data. The inferred output from the context engine then sets the status of the navigation filter. Finally, the navigation filter processes the epoch according to the modified status.

The availability block is responsible for data association for fusion. This is both before and after the epoch is processed. In a way, it is considered as a data integrity check block. Before the epoch from the sensor subsystems is processed, the availability block checks the subsystem measurements against the history and location information of the track. In the implementation for this work, this was used to identify whether the sensor subsystem position measurement was inside a wall or on the other side of the wall, in which case the measurement was given lower weight and considered a corrupted measurement.

The context engine itself is independent of the memory aspects and infers the current epoch context only. Availability is more closely connected to the navigation filter and the solution, and the data association rules are applied here within the same block with the tripartite filter. Previous positions and track history are recorded within the availability block. The availability block provides history infor-

mation for the context engine when queried, and is thus responsible for the memory aspects of the filter.

After the epoch is processed, final checks are made in the availability block for the solution. This is the mapmatching stage, where the new solution is checked against the map. If the logic finds that something went wrong, adjustments are made and the epoch is retaken using the new settings. This logic was explained previously as being the pentagon buffer collision logic and particle filter particle cloud property check and map matching. This block holds the novelty that is found in the structuring of the responsibilities and the more robust filter selection and solution correction processes.

9.2 Outdoor Performance

Figure 9.2 depicts the accuracy of the final filter design outdoors for the 3 combined walk recordings. The outdoor accuracy is dependent on the GNSS and the inertial sensor subsystems qualities. Moreover, the filter settings define the accuracy that can be achieved. In more detail, this means which inertial sensor propagation model settings are used and how the positioning fix qualities are adjusted in the context engine and in the availability blocks. For the conducted test cases the mean error outdoors was 1.81 metres.

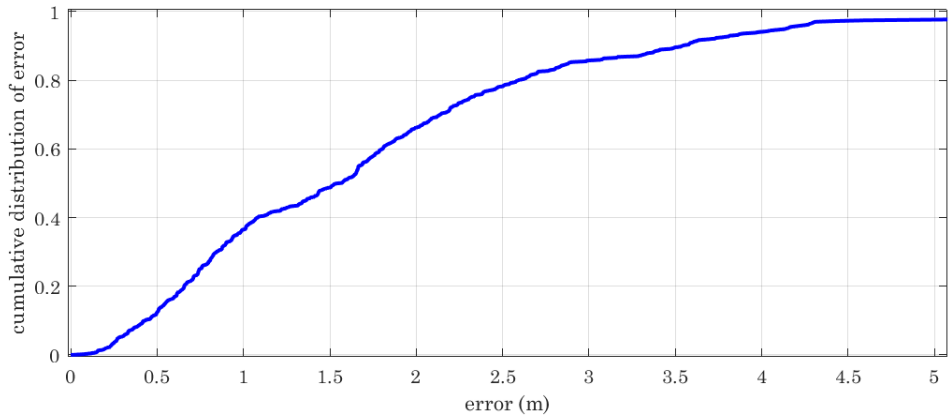


Figure 9.2: The tripartite filter accuracy outdoors. In this example run, the particle filter was on in the beginning for approximately 10 metres and near the end of the track for approximately 20 metres.

The above figure shows a typical result for our measurements using the typical settings for both particle and Kalman filters outdoors. The inertial solution had a tendency to drift and swerve to the right. The GNSS position was mostly, within 2 to 4 metres, close to the truth, although under the southern building bridge and next to the northern buildings the horizontal accuracy did get worse.

Figure 9.3 depicts the track of the dual-mode filter outdoors.

The filter starts with the particle filter on. The bush wall kills particles so that the particle scatter falls below the threshold and the pentagon buffer switch happens. The pentagon buffer enhanced Kalman filter runs until a collision is encountered on the north-east side of the track (Northing 15m, Easting 50m). This collision can't be resolved by the collision logic as described in the previous chapter and the particle filter is initialised.

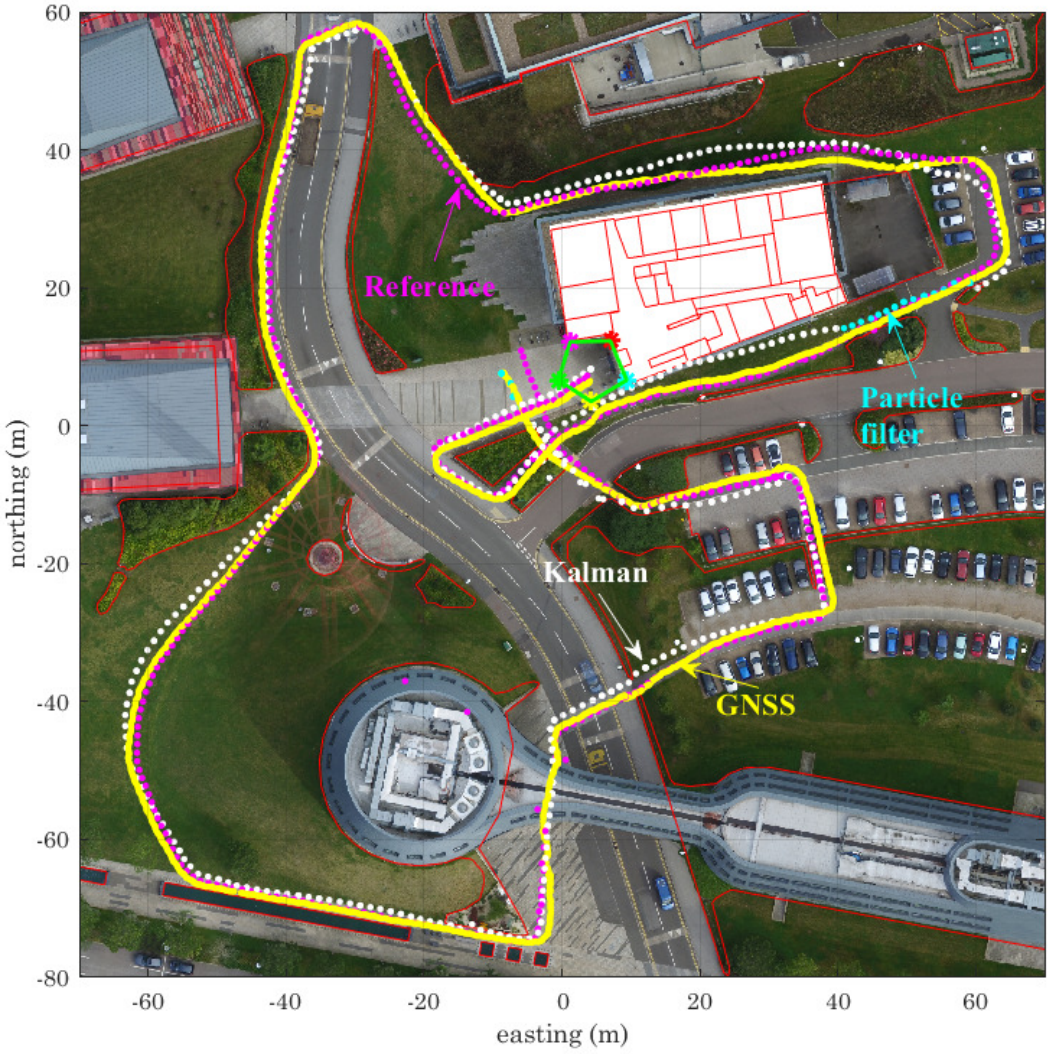


Figure 9.3: The tripartite filter track result is shown in white when the pentagon buffer enhanced Kalman filter solution was used. Cyan shows the parts where particle filter was used.

This collision is shown in more detail in Figure 9.4.

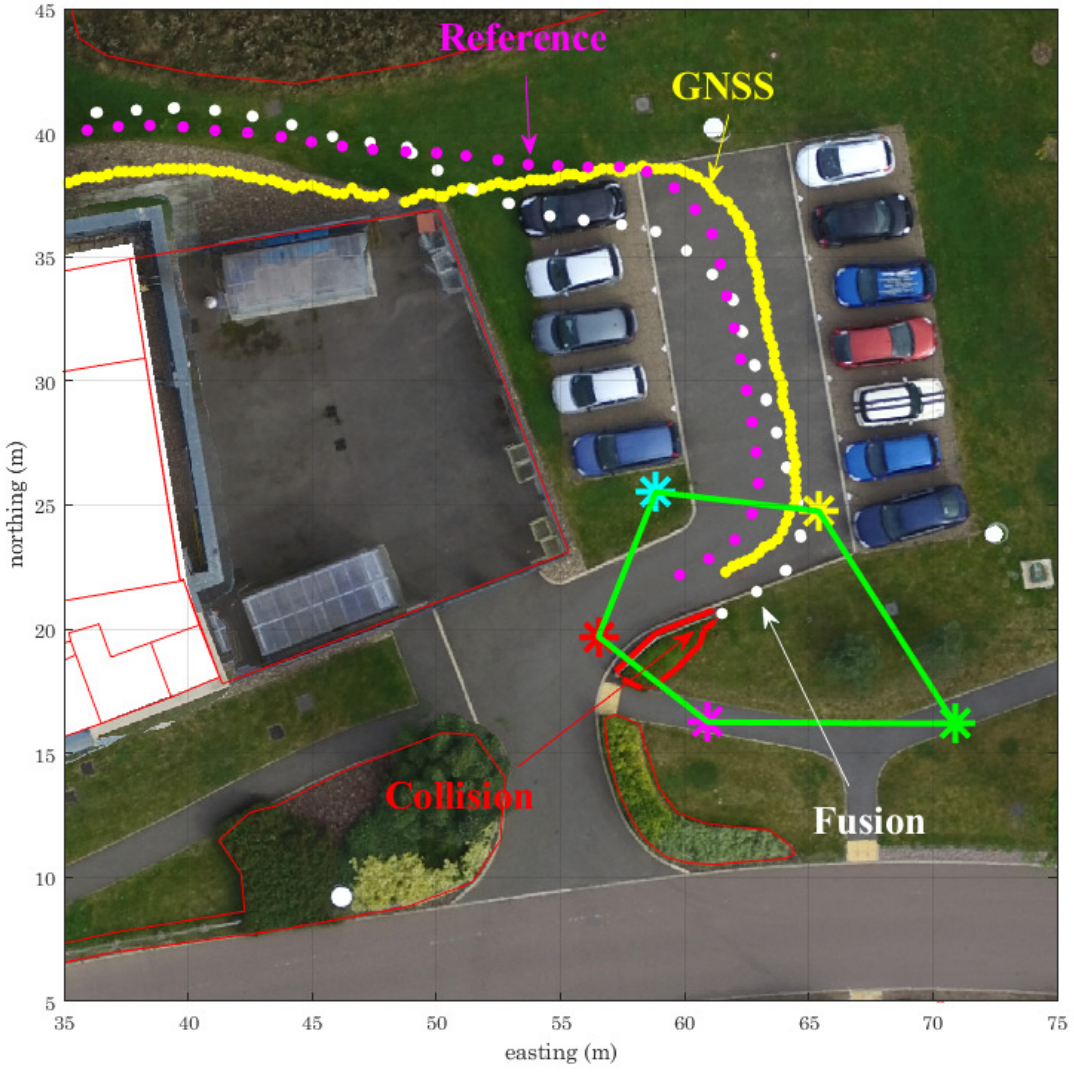


Figure 9.4: The pentagon buffer collision logic is set to switch to particle filter when this kind of collision happens, where the correct solution has two possible sides to go around the object.

The particle filter resolves the situation when the particles on the left side of the object die. This resolution is shown in Figure 9.5.

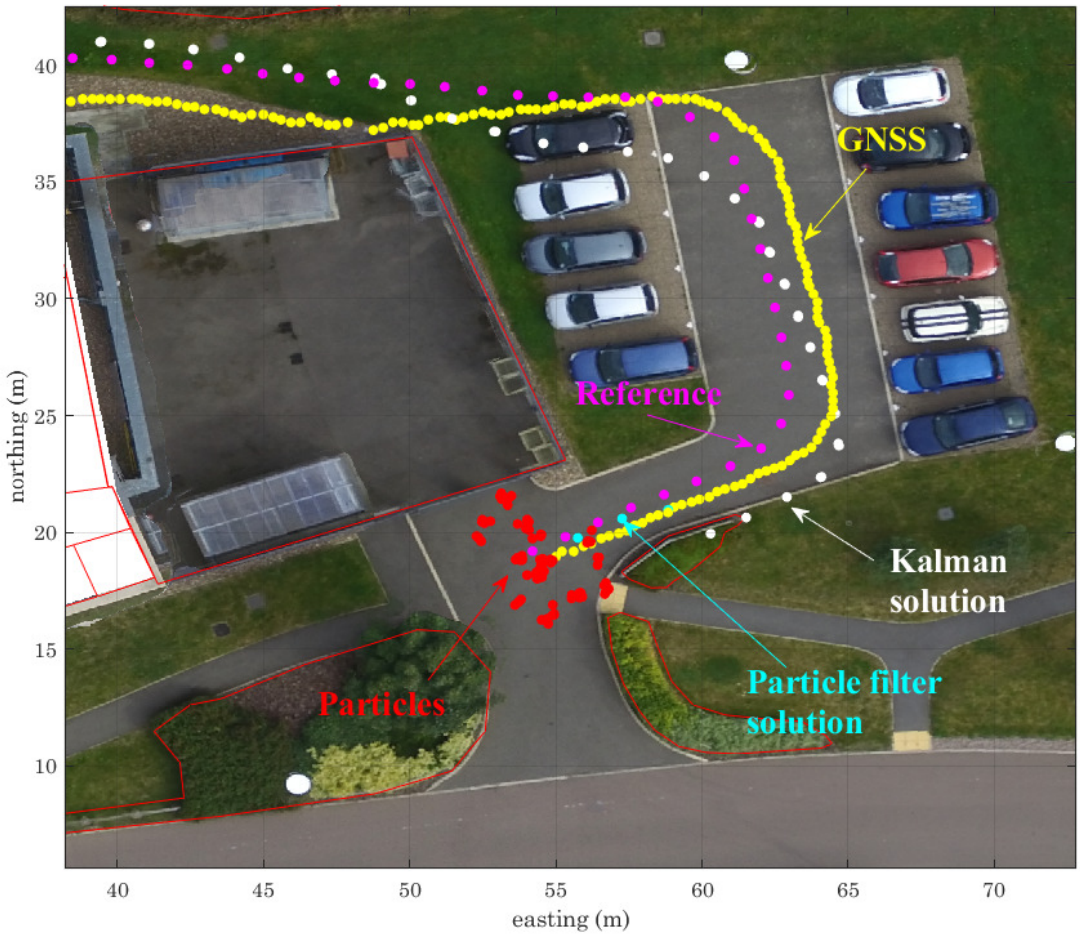


Figure 9.5: The particle filter quickly resolves the collision event.

After few steps, the particle scatter again diminishes and pentagon buffer Kalman filter is initialised. We can see one troublesome characteristic with the particle filter when using the weighted average position solution concerning map matching. This is shown in Figure 9.6.

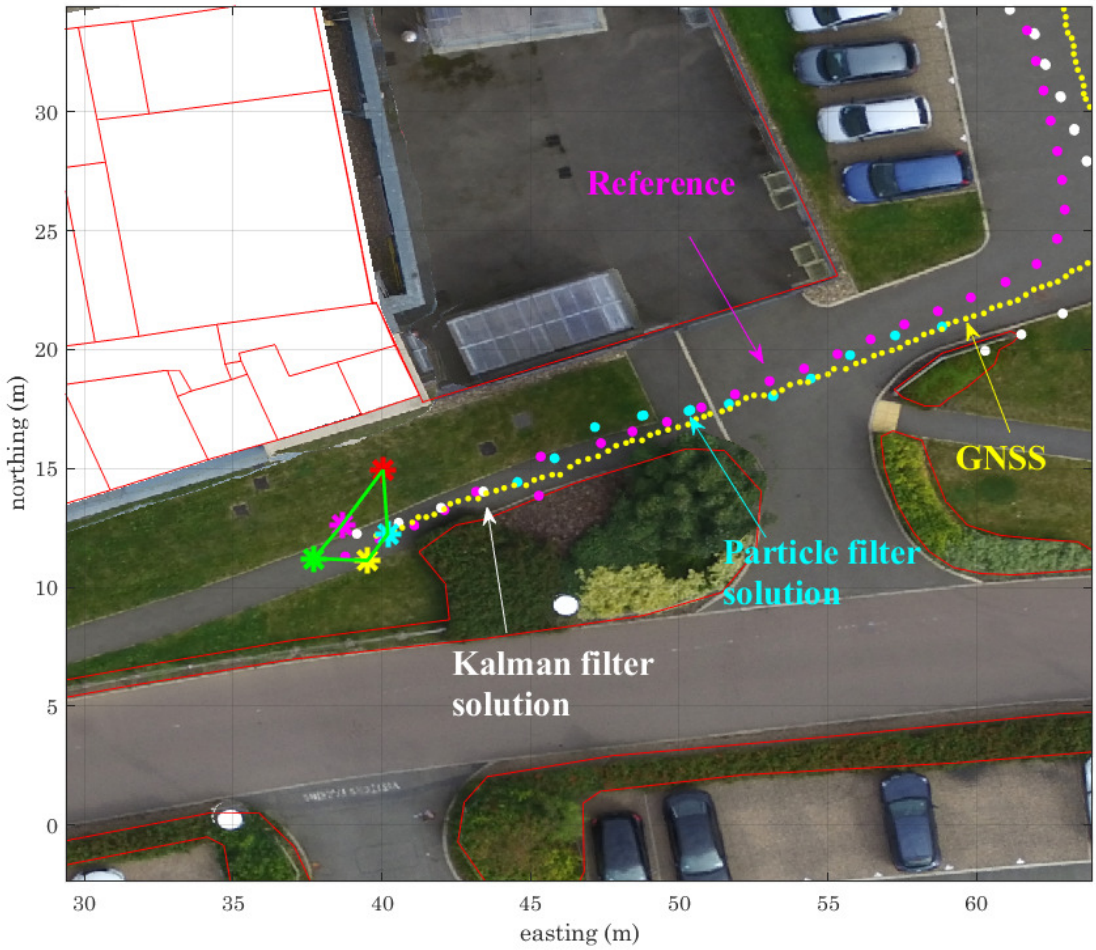


Figure 9.6: The particle filter switches again to the pentagon buffer enhanced Kalman filter.

The trouble with using particle filter, weighted mean solution, and map matching - comes when the solution is near the wall. This is seen in Figure 9.6. The particles on the left die and the particles on right continue spreading to the right. This means if the correct solution is just next to the wall, the weighted mean of the particles tends to still point it further away to the right off the wall. This is fixed partly by limiting the Tukey range and putting more weight on the GNSS measurement, but it is still insufficient. A possible fix would be to use the pentagon buffer also around

particle filter solution - and not use map matching for the particles, so that the statistical balance would remain. A pentagon buffer would again be the correcting method like it is in around the Kalman solution.

9.2.1 Particle Filter Characteristics

The GNSS system in this experiment is the most inaccurate of the used sensor subsystems. It thus defines the minimum limits for the particle filter performance. The outdoor track was processed using different number of particles and with only the particle filter on. Figure 9.7 depicts the accuracy of the runs.

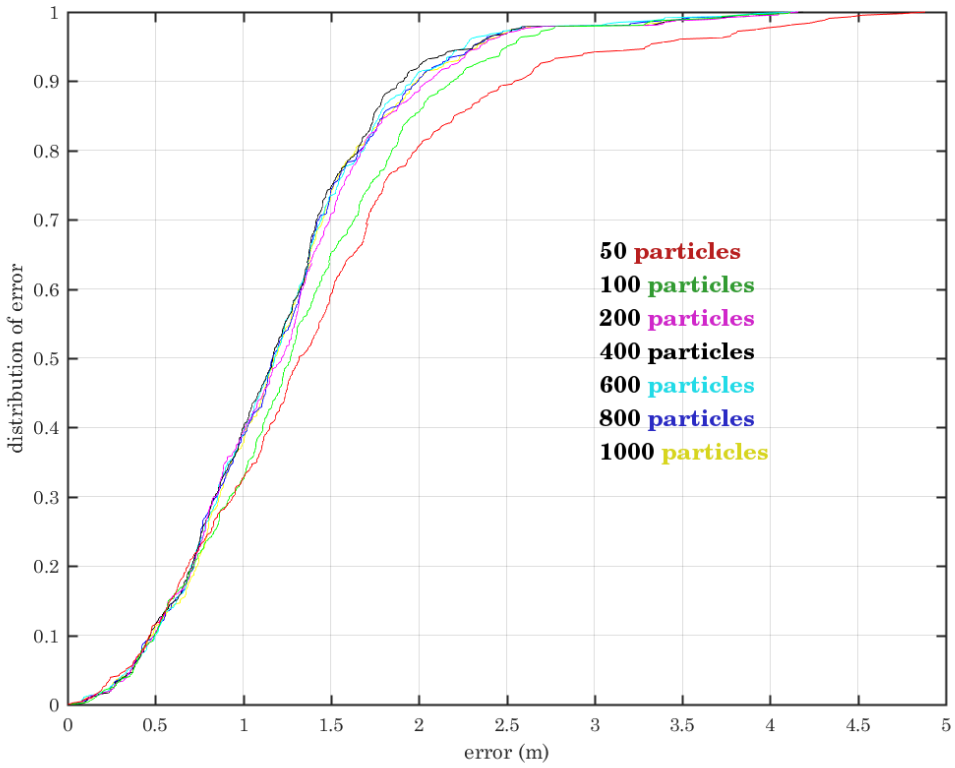


Figure 9.7: Using only the particle filter for different numbers of particles, we can conclude that 200 or more particles are necessary to represent a solution in our tests.

It is clear from the figure that 200 particles is the minimum number for representing a position in our tracking scenario outdoors. If fewer particles are used the accuracy will deteriorate. This agrees with the study made in the previous chapter concerning the particle filter (Figure 7.19).

When we turn on the Kalman filter and allow the switch back and forth with the particle filter, we can study the computational load and gain in speed when using the novel dual-mode filter design. Figure 9.8 depicts this enhancement.

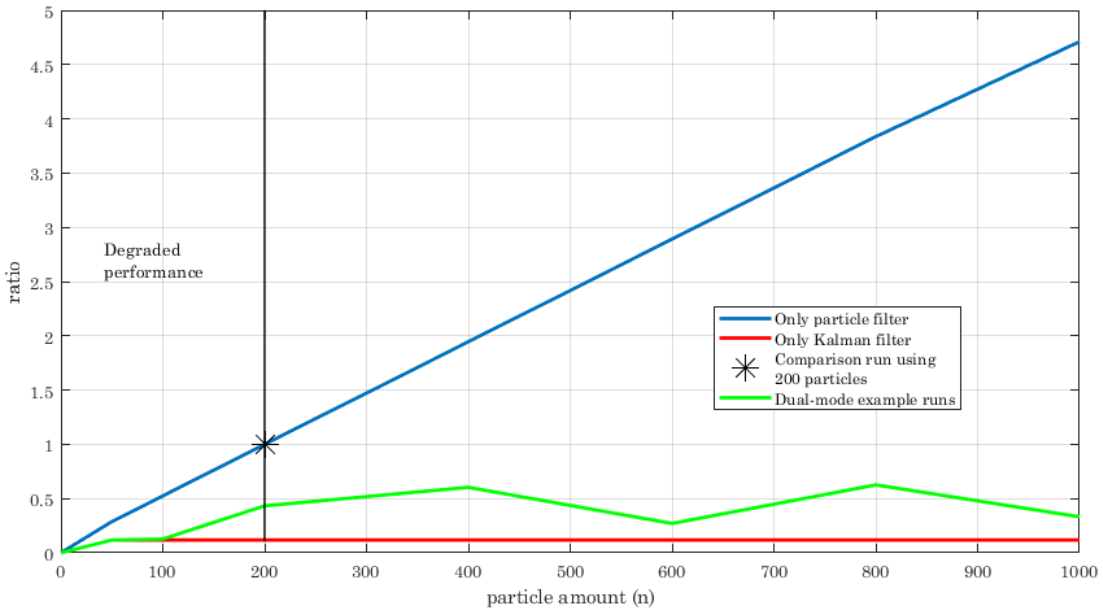


Figure 9.8: Using only particle filter, the computational load gets linearly greater. By applying the Kalman filter when necessary, computational power is saved. The 200-particle filter load is taken as the reference. The blue line represents the particle filter-only runs, the red line describes the pentagon buffer enhanced Kalman filter runs, and the green line describes example runs using the novel dual-mode design.

The code is written in MATLAB and could easily be further optimised for more effective programming languages. The most expensive section, in terms of computation, is the map matching of the particles. The saving of computational power is

thus evident according to the test runs. The pentagon buffer Kalman filter runs ten times faster than the 200-particle filter. The number of particles could be adjusted according to the situation; however, in the work conducted for this thesis, this was kept constant for simplicity. If the quality of the sensors is low, then more particles would be needed to represent the solution. And in the case of a filter failing, more particles when initialising would also help in acquiring the correct solution more robustly and with greater accuracy.

9.2.2 Track Form Matching Characteristics

Track form matching was introduced in the previous chapter. Inertial track quality stays consistent over a few tens of metres, after which the error tends to be closer to few metres which is close to the accuracy of the GNSS receivers. Conversely, the GNSS receiver short-term accuracy is around few to ten metres. These are complementary characteristics which can be exploited.

Our approach for short term was to examine the track forms of the inertial-only solution and the GNSS solution. For the final implementation, three short-term track form matchers were created as introduced in the previous chapter. The three track forms had distance resolution of 3, 10 and 20 metres respectively. This means that three points on both the inertial and GNSS tracks were selected. The points were separated by the distance resolutions, making altogether nine points.

An angle difference between the two tracks could then be detected. An angle difference of 25 degrees was set for all short track form matchers. A delay value was

used to upkeep the angle detection state for 15 seconds after the condition ceased. A distance based switching back ON would allow the condition to automatically reset when the track forms match again. But for the conducted test walks the time based approach is applicable since the movement is continuous, without stops. This triple matcher was then processed alongside the dual-mode filtering solution on a corrupted track. Figure 9.9 depicts the detection result.

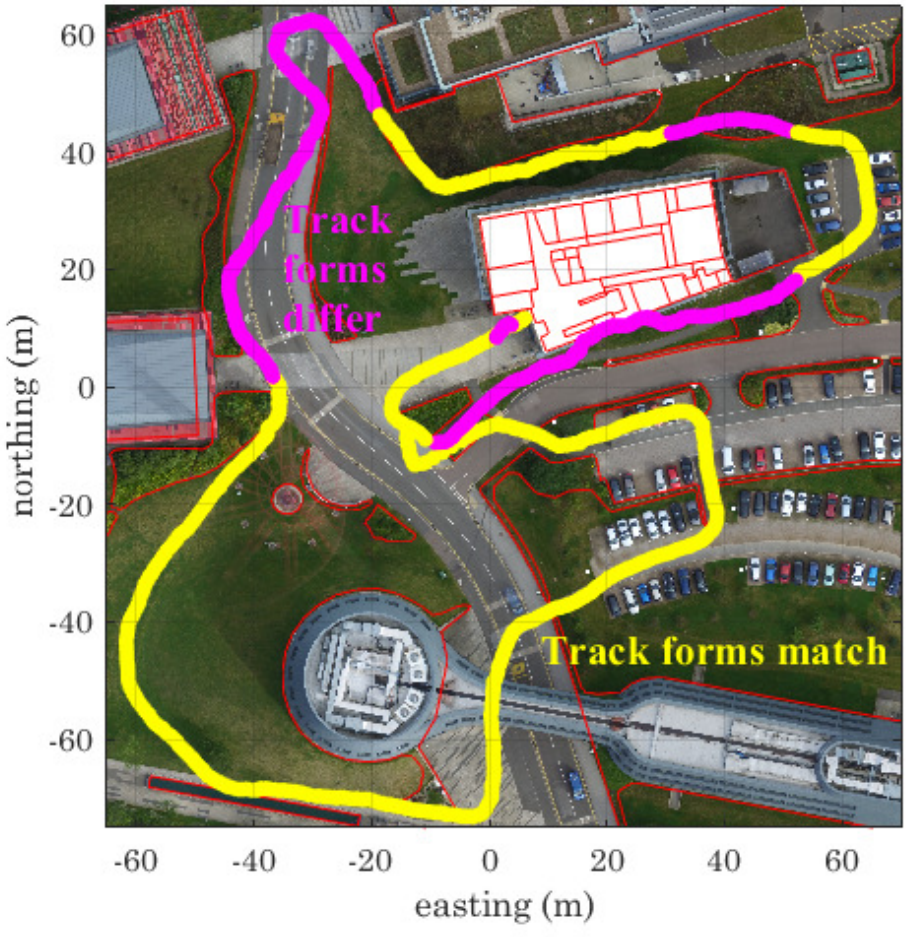


Figure 9.9: Short track form matching result using three parallel short track form matchers with distance resolutions of 3, 10 and 20 metres. The magenta track shows when one or more track form matchers indicated an angle difference greater than the threshold of 25 degrees.

For comparison, the right swerving inertial track is shown in the Figure 9.10. This method assumes that the inertial track quality stays good for 40 metres. The last detected angle difference south of the Nottingham Geospatial Building (Northing 10m, Easting 30m) in the previous example, was due to the inertial track swerving towards right, rather than the GNSS track quality - as can be inferred by looking at the inertial track form at that point in Figure 9.10.

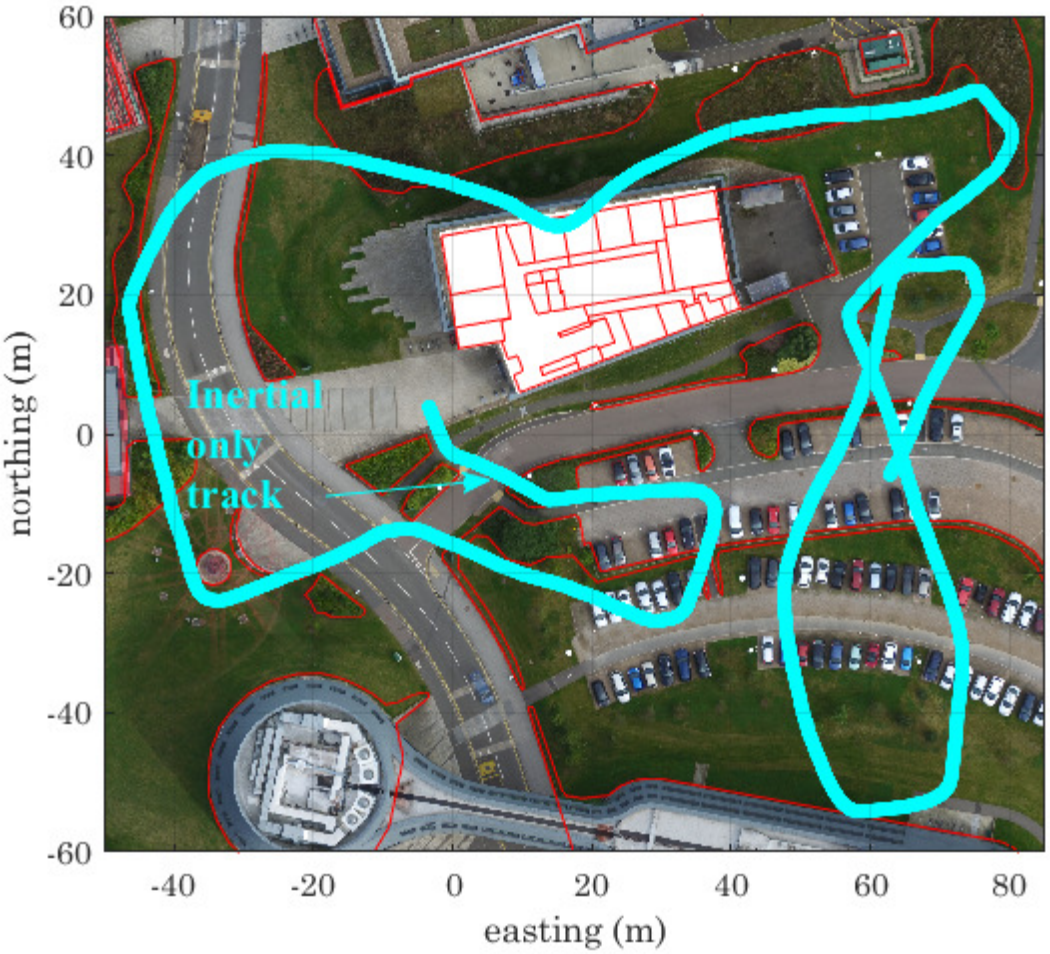


Figure 9.10: The inertial-only track swerves to the right.

This is where the advantage of using long-term track form matching comes into

play. When we correct the inertial track using a modified long-term track form matching method, we get the following detection result shown in Figure 9.11.

The long-term correction was used only if the short-term track forms were good. The corrected inertial-only track is shown in Figure 9.12.

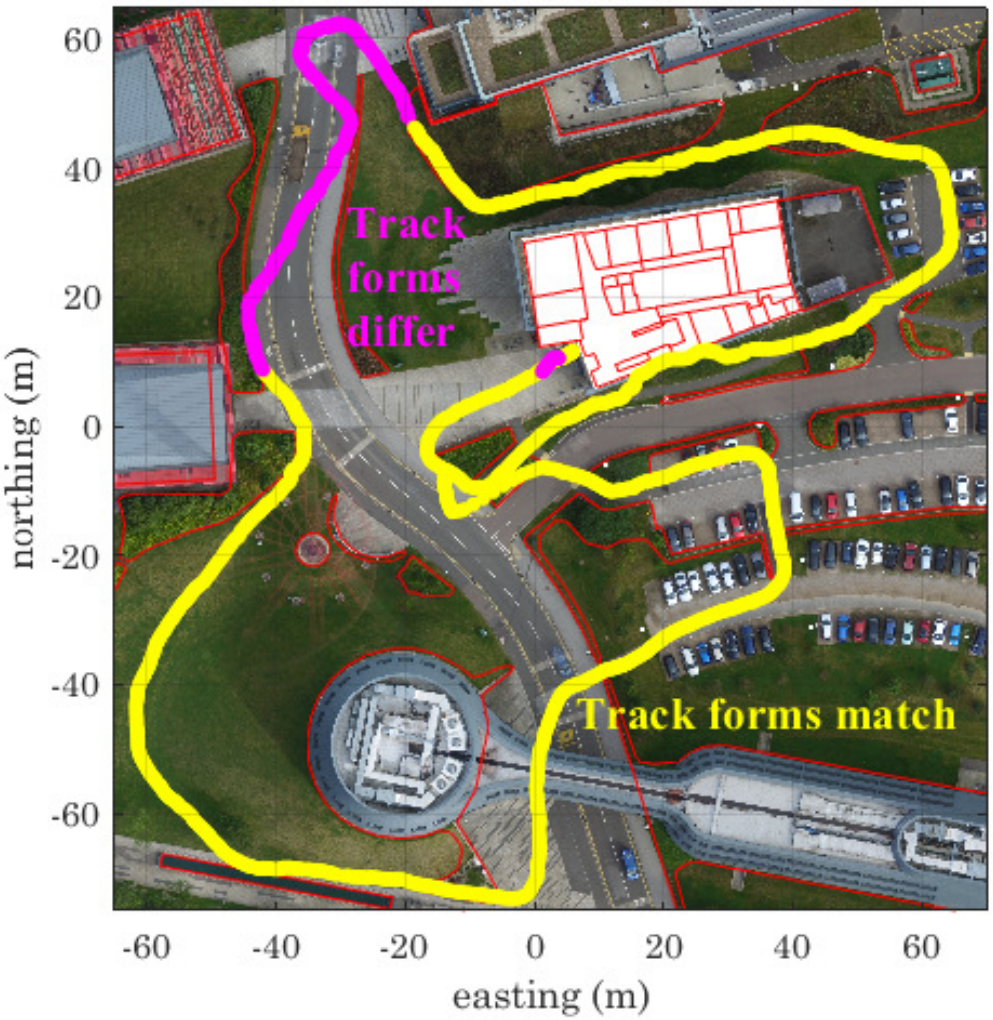


Figure 9.11: Using long track form matching the inertial track can be corrected and the track combined form matching results in the detection of the GNSS track anomaly.

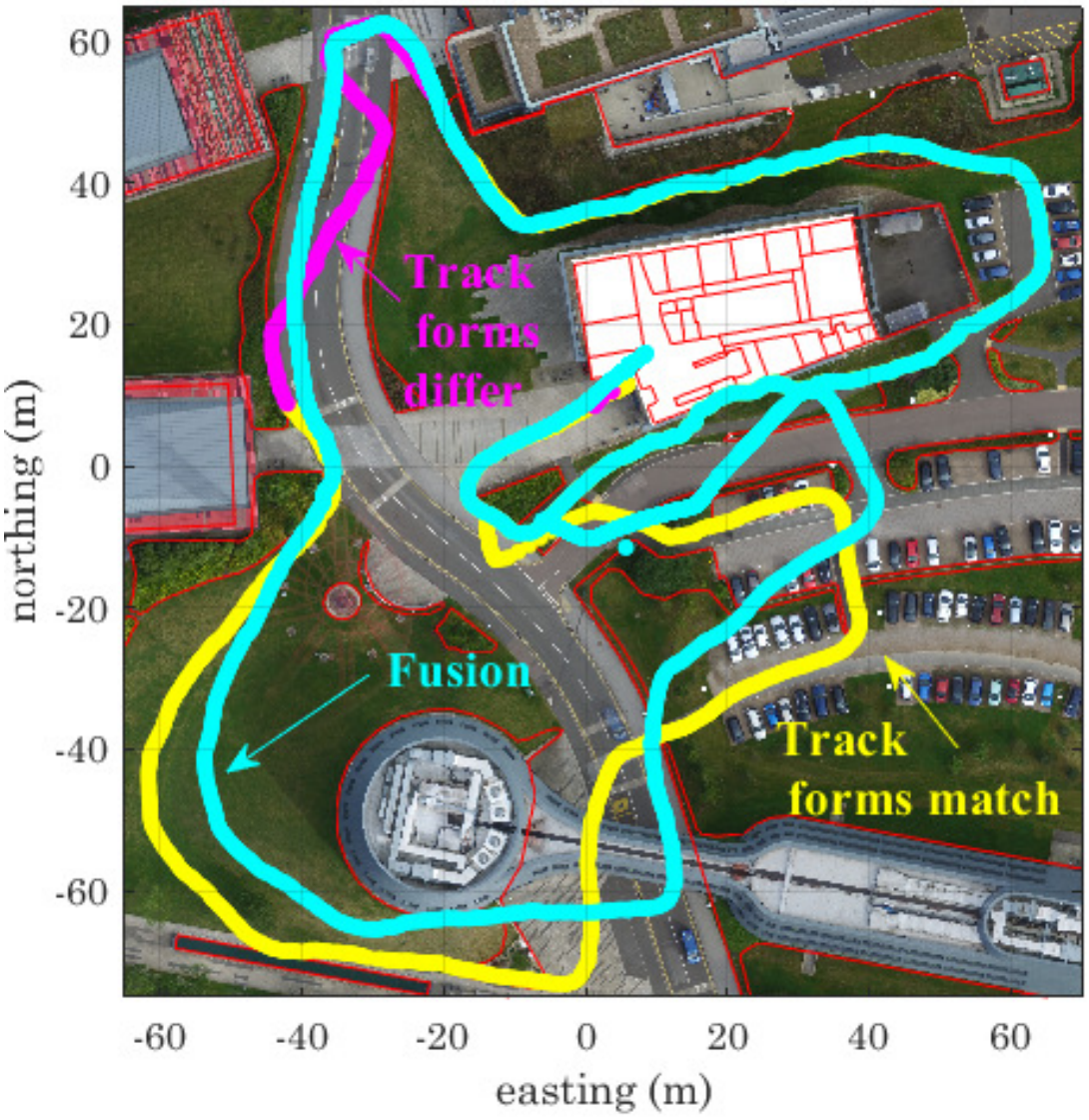


Figure 9.12: Using long track form matching the inertial track can be corrected. The corrected track is shown here in cyan.

The long-term track form correction method used was modified from the one described in the previous chapter. Similarly, seven points were selected on the inertial track 10m apart. The procedure is explained here.

1. Instead of fitting the two earliest points of the inertial track to the GNSS

track, the centre of mass point was calculated for the 7 GNSS track points.

2. The same mass centre point was also calculated for the inertial track.

3. The inertial track was then moved so that the two centres of mass were at the same location.

4. The inertial track was then rotated so that the angle was found where the minimum least squares distance between all corresponding points on the inertial and GNSS tracks were the shortest.

5. After this, the start of the inertial track was moved to the corresponding GNSS position.

6. A weighted attitude update was now applied so that the inertial track attitude was turned towards the direction indicated by the GNSS track.

7. A weighted position correction update was applied to the inertial-only track at the end of the track. The new position was in between the inertial track end point and the corresponding point on the GNSS track according to a preselected **correction percentage**.

8. The remainder of the track was smoothed backwards until the track-form matching start point. The shift, how much the update moved the end of the track was noted down. The epochs, backwards until the beginning of the track form matched section, were shifted by this shift, multiplied by a declining percentage. The percentage for the end of the track was 100% of the

shift declining to 0% of the shift for the epoch at the beginning of the track form matched section.

This method worked better than the first version explained in the previous chapter. The correction percentage was 48% which is relatively high since it is applied six times on each section that goes through the long track form matching. This means that the inertial track will be shifted six times according to the correction percentage. The lower the correction percentage, the less the inertial-only track is transformed towards the track form matching result.

9.3 Indoor Performance

Figure 9.13 depicts the tripartite filter accuracy indoors. Figure 9.14 depicts the tracks on the map. The mean error indoors was 1.14 metres.

The tripartite or the dual-mode solution reaches sub-metre accuracy when UWB positioning is used in the lobby area and has line of sight condition to the user. Going to the northern corridor, only inertial solution is available and the error grows although it is constrained by the walls. When returning to the lobby from the eastern side of the building, the UWB system again aids the fused solution to be very accurate. When entering the upper part of the staircase (lower right corner), the UWB system is somewhat less helpful in the fusion process, because of the non-line of sight condition with one anchor.

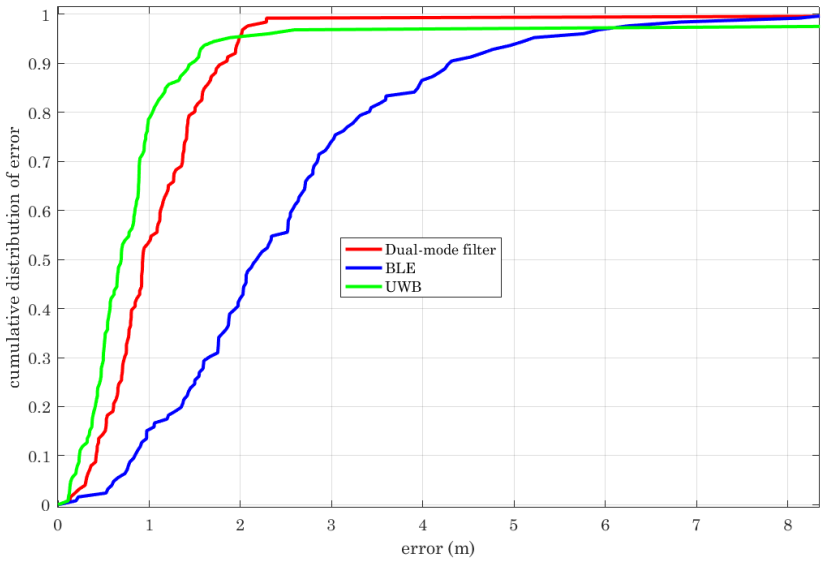


Figure 9.13: Dual-Mode particle/Kalman filter accuracy in red; UWB subsystem accuracy in green; BLE subsystem accuracy in blue. These are calculated only when the reference system was available.

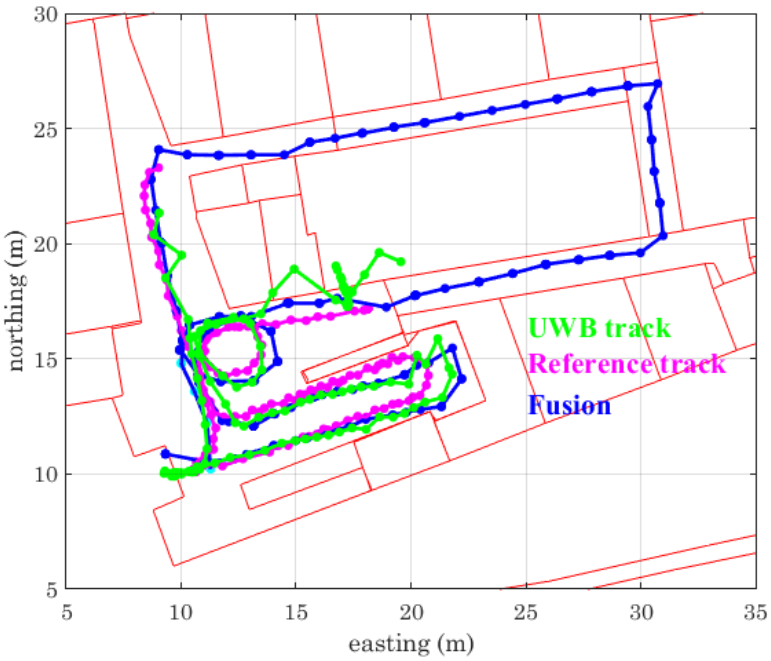


Figure 9.14: Blue track shows the tripartite filter trail; green shows the UWB sensor track; and magenta is the reference track measured using a Leica total station.

Figure 9.15 presents the anchor locations on the map.

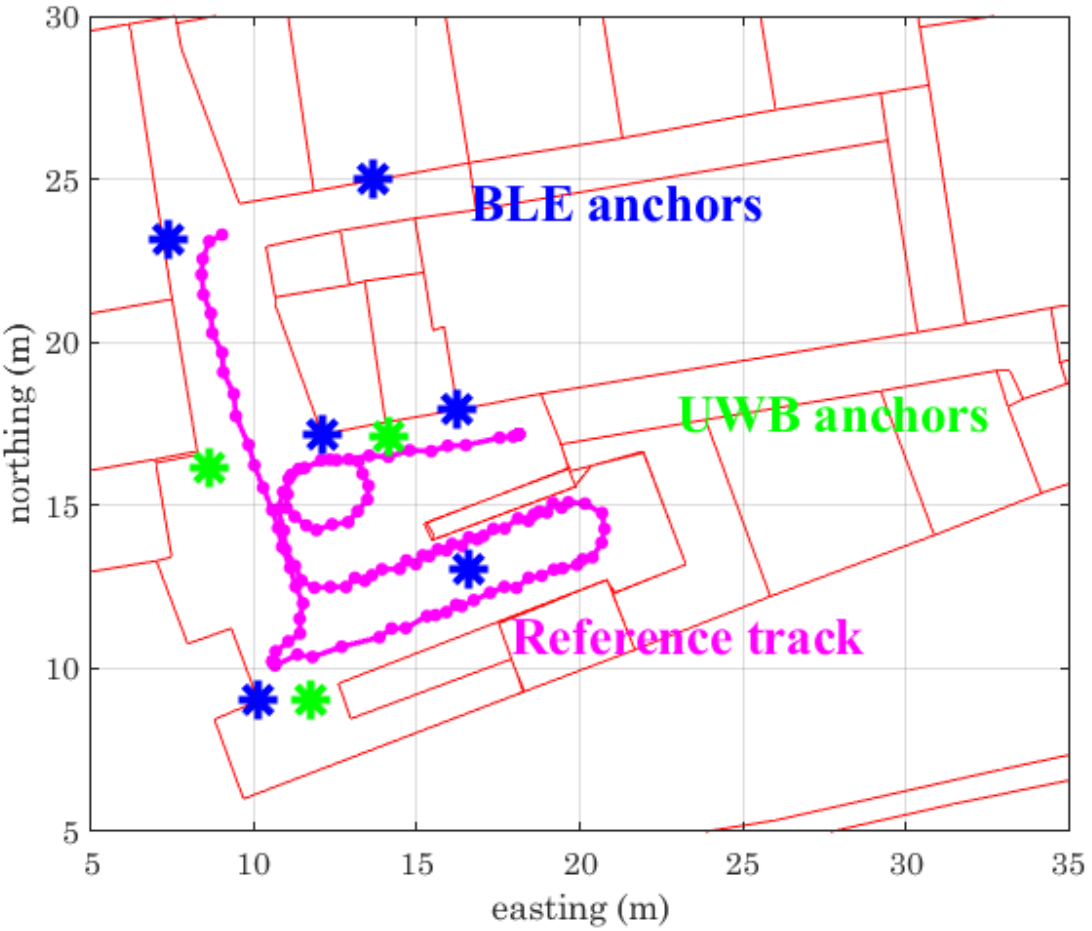


Figure 9.15: Blue stars show the locations of the BLE anchors. Green stars mark the UWB anchor locations.

The new implementation for the UWB system functions better than the first setup in the previous chapters. This is mainly due to the better GDOP by placing the anchors smartly as devised by Kirchhof (2013). The positioning stays acceptable, even behind one wall structure in the corridor that runs from south to north (northing coordinates 16m to 23m). This is also because a simple range compensation was used at the top of the stairs and in the northern corridor for the top right

anchor. This is to aid the non-line of sight condition that occurs at these locations for this anchor. The resulting track behind one-wall structure corresponds to the performance discussed by Meng et al. (2012). The UWB system now covers the whole of the open lobby area which was challenging for the BLE system.

The novel method used in BLE positioning functions as well as the fingerprinting method used previously. The advantage is that the laborious fingerprint recording is not needed. We now discuss this new version of the BLE positioning fixing systems in more detail.

9.3.1 Novel BLE positioning

The accuracy of the novel approach for BLE fingerprinting reached similar accuracy (Figure 9.13) to the fingerprinting method tested in the previous chapters. In this approach, the fingerprinting recording was replaced with less laborious recording of the anchor RSS-to-distance relationship curves. For each anchor, an individual conversion curve was measured by moving the receiver meter by meter until 20m away in an open area environment. The radiomaps were then created in a similar manner to walless pathloss modelling. For each anchor, the gridmap contained RSS value at each point according to the individual distance RSS conversion curve.

This approach removes the need for the laborious fingerprinting recording offline session. It is also very simple to set the anchors to send their positions. This removes the need to have anchor location information in the map on the mobile device. The anchor placement on the mobile device map could be placed when the user arrives

to the proximity of this BLE anchor.

The same position acquisition algorithm was used for these radiomaps that was used in the fingerprinting approach in the previous chapters. Figure 9.16 presents a positioning solution using this method.

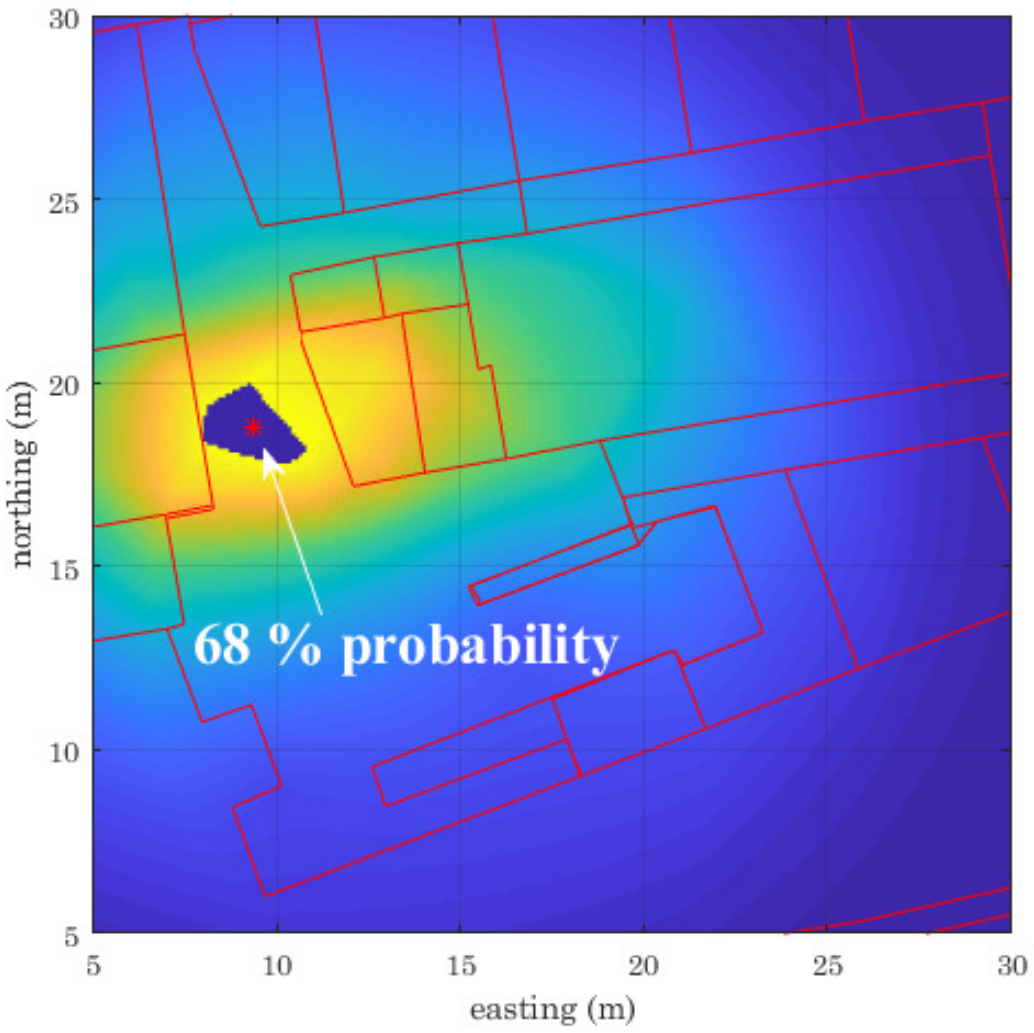


Figure 9.16: Example of a BLE positioning result using the pathloss model-based grid radiomap creation method.

9.4 Seamless Performance

Four additional walks were conducted. This time the walks were continuous and included going indoors and coming outdoors. No reference track was collected during these continuous seamless trials. The walked tracks outdoors and indoors go the same way as the tracks explained previously in this chapter. The outdoor track was also walked in the reverse direction when leaving the building. We first examine the improved parts of the navigation system before investigating performance.

9.4.1 State Machine

Ruppelt et al. (2016) used a state machine in their design. The state machine of the navigation filter bank describes the implemented context inference in action. Two state flags - indoor and particle filter flags - define the main states. The stairs, entrance and ZUPT flags are secondary flags contributing to the filter setting changes. Figure 9.17 describes the four main states using the large squares.

The secondary flag blocks are smaller. The ZUPT and Stairs flags adjust the Mechanisation settings while Entrance flag enables the vertical indoor-outdoor switch checks. A horizontal change describes the transfer between the two filters: the particle filter (PF) and the Pentagon Buffer Kalman filter (PB).

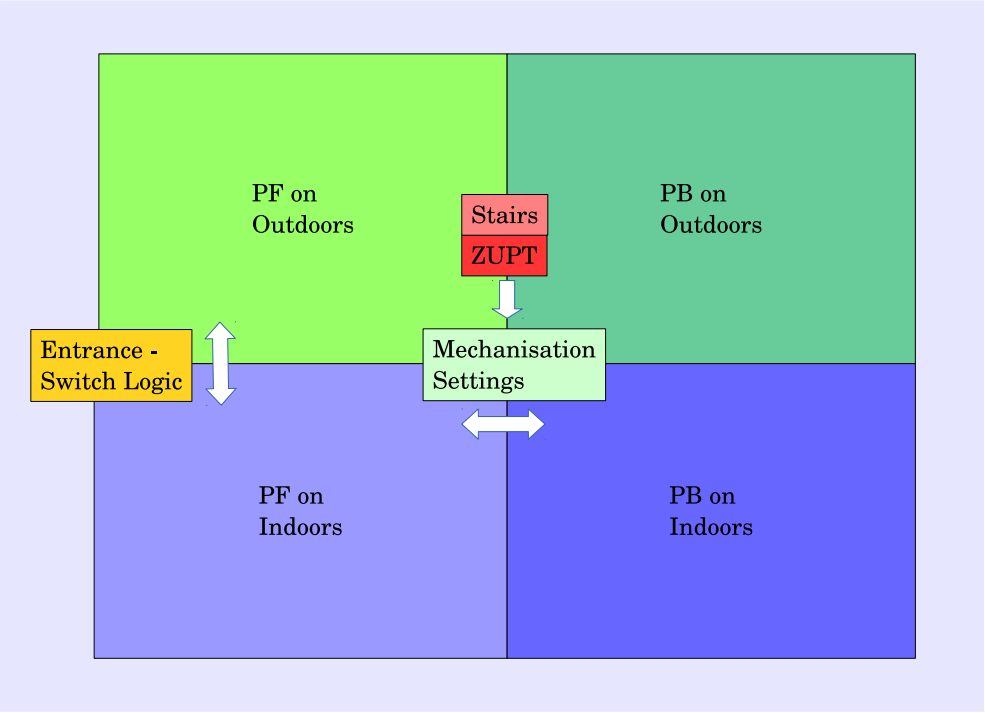


Figure 9.17: The state machine describes the context engine operation and the control transfer between the particle filter (PF) and pentagon buffer Kalman filter (PB).

9.4.2 Novel Collision Logic for Pentagon Buffer

Figure 9.18 presents the enhanced collision logic.

In the new collision logic that was developed, the black trail shows the colliding solution. When a collision occurs, the two green points represent the solution options for a new correct position that was described in the first version. The green positions start from the position where the collision was detected. Aiding blue points are used starting from the two previous step locations. Collision detection uses two methods: a check if the point is inside a polygon is the first method, and the second checks

whether two lines are crossing each other. The green and the blue points are moved perpendicularly, until a trace is found where no points are inside a wall polygon or no lines are crossing the wall structures. This is for both green circles.

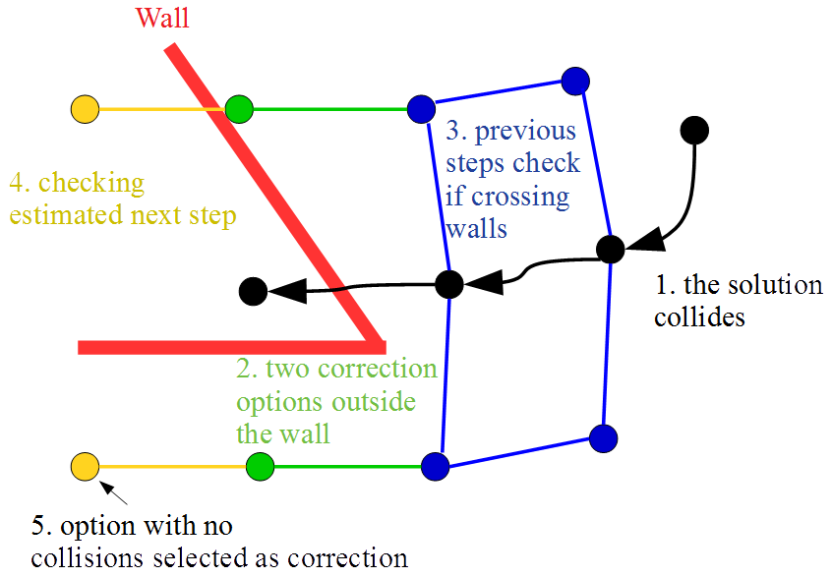


Figure 9.18: Enhanced collision logic using aid points.

An additional check is made for the next predicted step. In our example, the bottom trail is selected as the correct position. If none of the traces work, that is the green solutions are far away (threshold) or no free path was found for the lines, then the particle filter is initialised.

In case the particle filter fails, which is when every particle ends up inside a wall, the latest position update or track location is used to reinitialise the particles.

9.4.3 Particle filter with Pentagon Buffer

As was noticed when discussing outdoor performance, map matching with the particle filter causes the weighted average solution to move away from the walls - even though the correct solution could be just next to a wall. This was because walls kill particles and naturally the particle cloud scatters away from the wall. Moreover, map matching with each particle is the most computationally expensive operation in our design. To tackle these two issues, we place the pentagon buffer around the particle cloud. The buffer is now let to interact with the map and map matching is not used on particles. This means that the statistical information kept within the particle filter will only be affected by the position fixing updates.

The option to map match particles is retained to be used when necessary. This is necessary especially when no position updates are available. The centre of the buffer is at the same location as the weighted average of the particle cloud positions. If this centre enters a wall, a collision logic is initiated and an offset is added to the pentagon buffer centre according to the new corrected position. This pentagon buffer centre, including the offset from the particle filter weighted average solution, is now the solution for the whole filter. The offset is programmed to diminish over time. This means that if the walls have guided the pentagon buffer solution away from the particle filter, it will automatically return closer when entering a more open or wall-free area. If the correction for the pentagon buffer fails, map matching is turned on for the particles and pentagon buffer is shut down. Figure 9.19 presents

the results of the particle filter with pentagon buffer. We can see that the buffer bumped into a wall but the solution still stays near when using the pentagon buffer. If a particle filter with map matching would have been used, the particles on the top right would have been killed by the sharp wall on the top of the figure - and the solution would be more westerly.

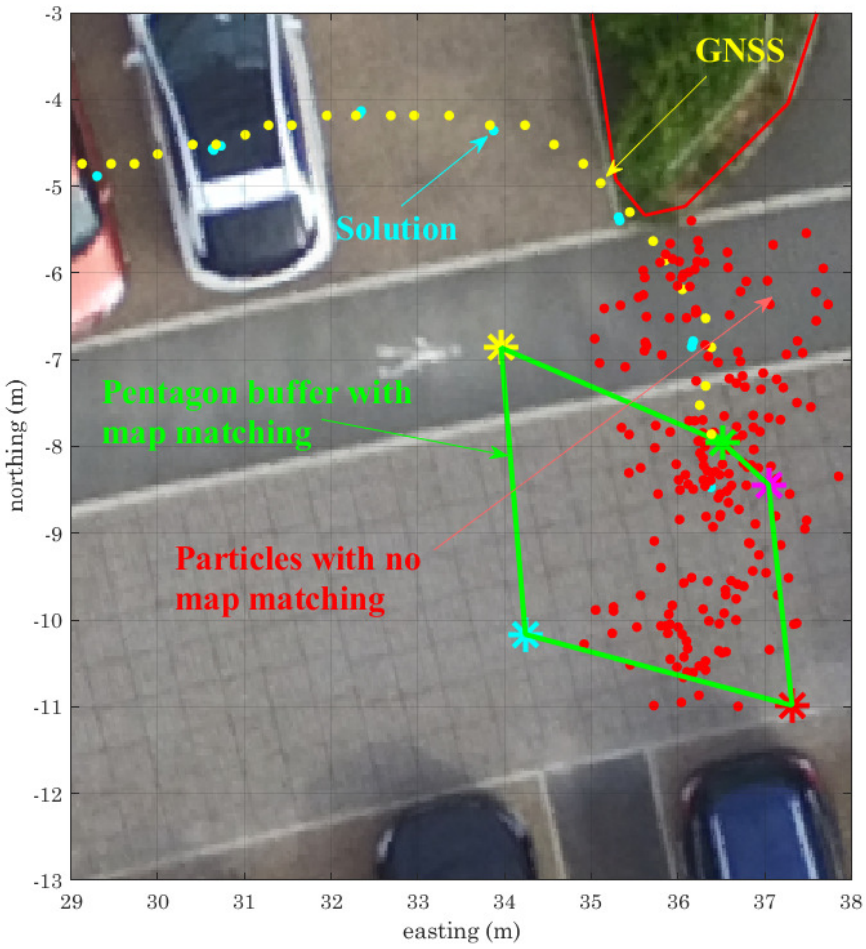


Figure 9.19: Particle filter with pentagon buffer.

This adds a new filter state in between the particle filter and the pentagon buffer Kalman filter. For simplicity, we initialised the pentagon buffer particle filter only

when the particle filter was on. The initialisation of the pentagon buffer particle filter from the pentagon buffer Kalman filter was left for future study. This is shown in Figure 9.20 using the white arrows on the top.

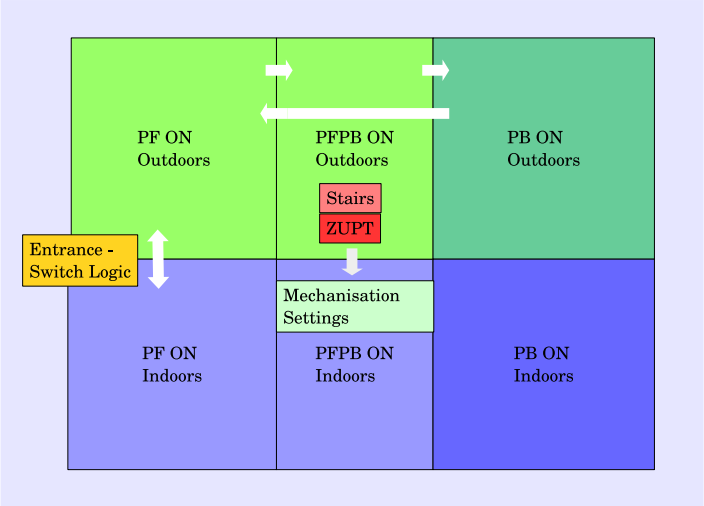


Figure 9.20: The state machine describes the context engine operation and the control transfer between particle filter (PF), particle filter pentagon buffer (PFPB) and pentagon buffer Kalman filter (PB).

9.4.4 Seamless System Characteristics

The accuracies for the combined seamless walks outdoor and indoor were comparable to the accuracies of the walks conducted in the chapters 7 and 8. In addition, the Kalman filter pentagon buffer stayed on the whole time for all the measured tracks after the initial conversion from particle to Kalman filter, because of the novel collision logic. Thus, the interesting factor to examine were the outdoor to indoor and the indoor to outdoor transitions.

Figure 9.21 shows a challenging transition of the four walks from outdoors to

indoors. The novel pentagon buffer logic is able to correct the position into the entrance. The switch condition is seen completed when the stepwise marking colour of the track turns from white to blue. The switch condition was explained in the chapter 6.

When going outdoors the solution depends on an inertial sensor only for a short while before the GNSS receiver horizontal accuracy improves. This can be seen in Figure 9.22. The yellow GNSS track position is off first by few metres before settling near to the true walk trail. The GNSS position weighting increases in the Kalman filter as horizontal accuracy improves.

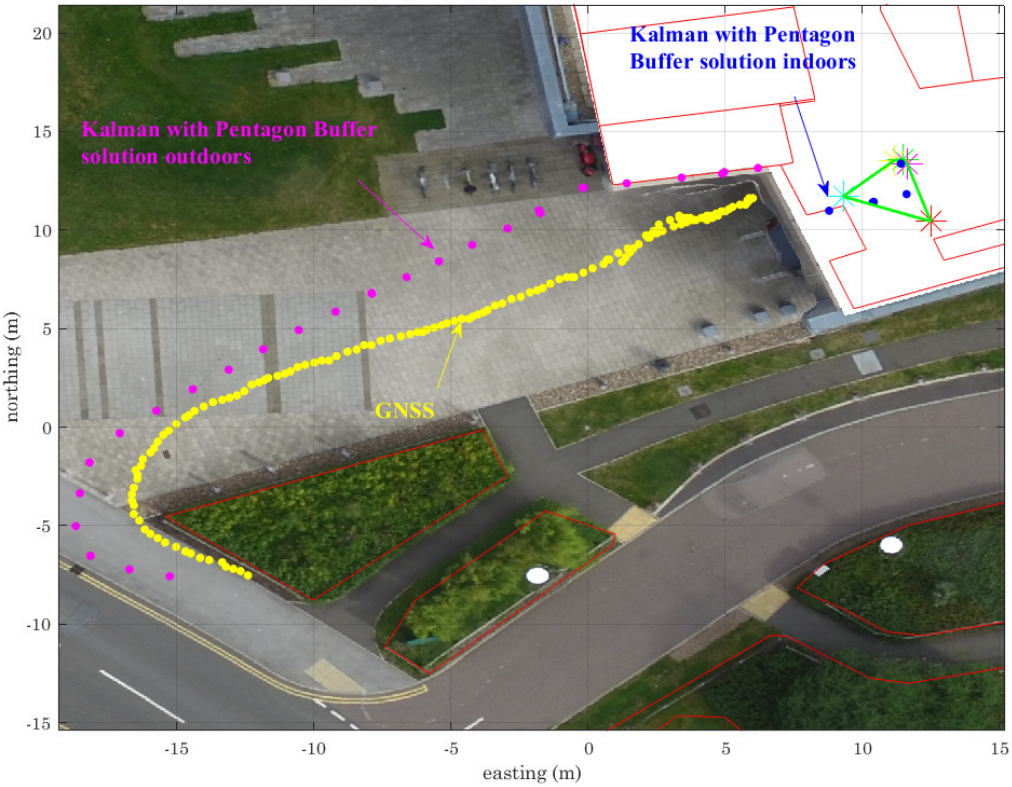


Figure 9.21: The pentagon buffer Kalman solution enters indoors successfully.

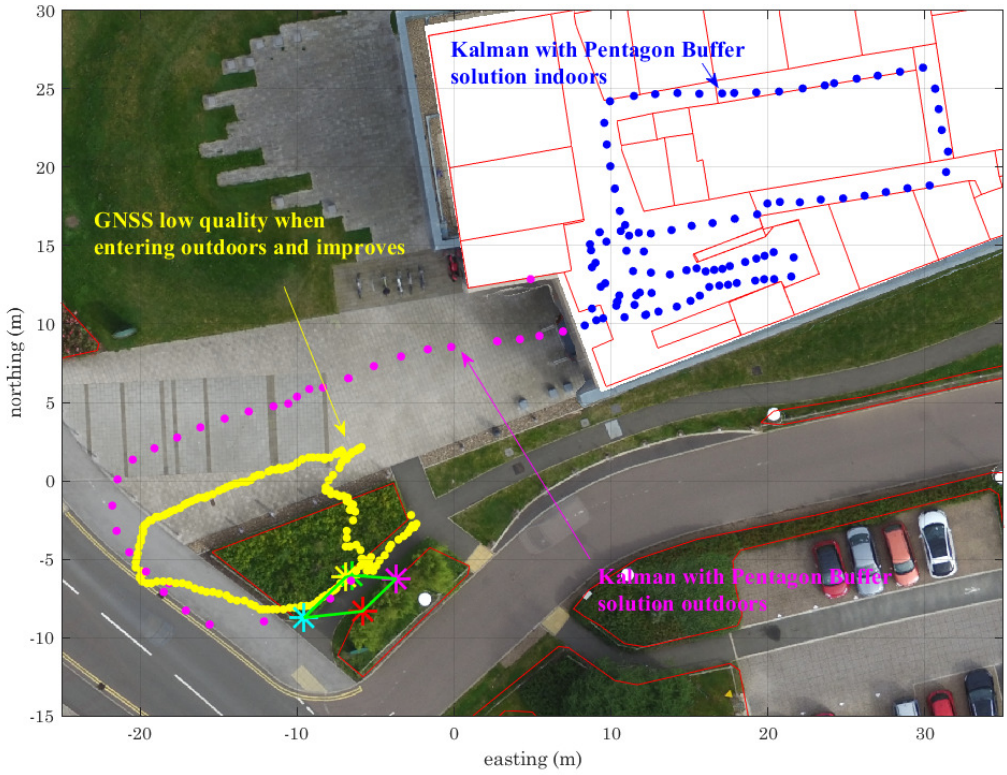


Figure 9.22: The pentagon buffer Kalman solution enters outdoors successfully.

Indoor to outdoor performance was worse with GNSS because of the GNSS receiver cold start as mentioned also by Hejc et al. (2014).

A further eight walks were conducted to examine the transition. Four from outdoor to indoors and four from indoors to outdoors. BLE and UWB anchors were placed at the entrance and the values were recorded during the walks. The GNSS horizontal accuracy change was examined.

Figure 9.23 shows the inertial track in white and the GNSS track in yellow.

A very steep drop in horizontal accuracy occurs when entering indoors. This can be seen in Figure 9.24.

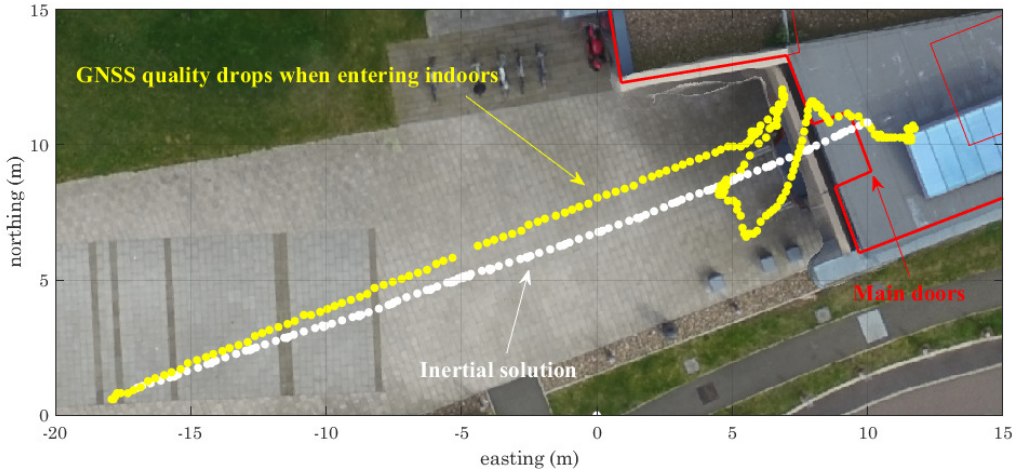


Figure 9.23: Outdoor to indoor transition trails for inertial (white) and GNSS (yellow) solutions.

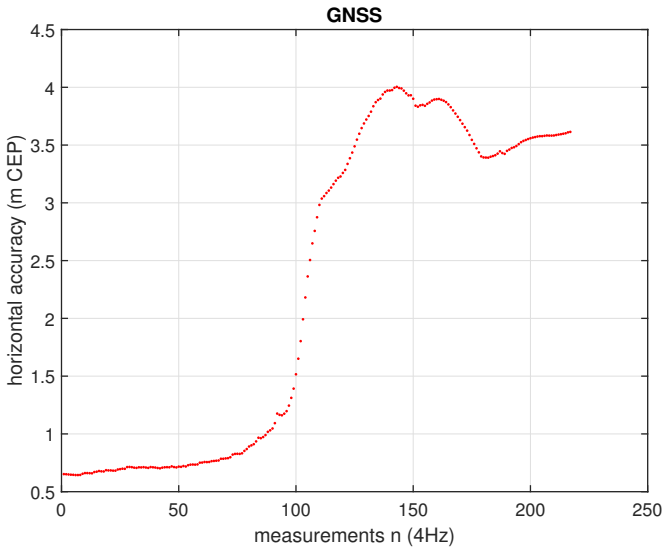


Figure 9.24: GNSS receiver horizontal accuracy when entering indoors.

Figures 9.25 and 9.26 show the equivalent diagrams when going outdoors from indoors.

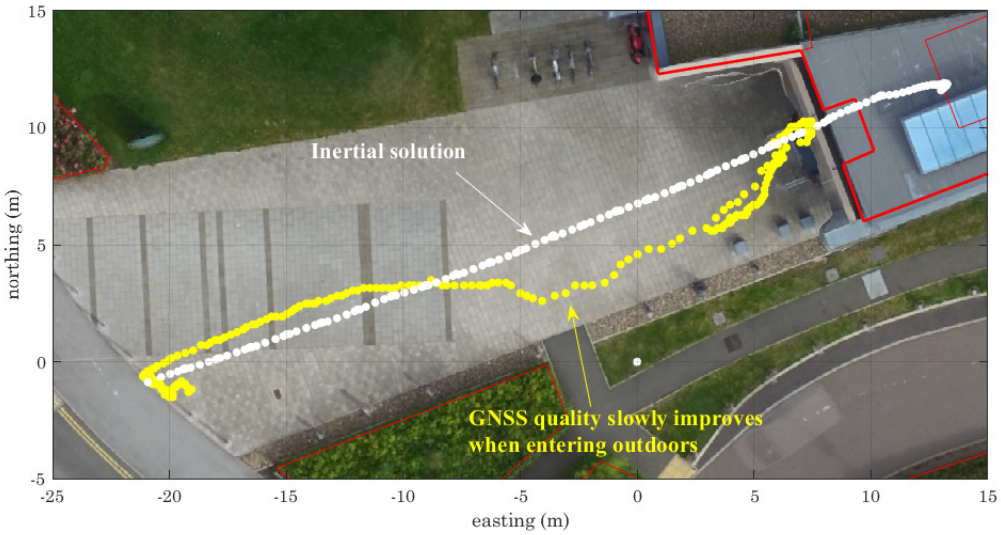


Figure 9.25: Indoor to outdoor transition trails for inertial (white) and GNSS (yellow) solutions.

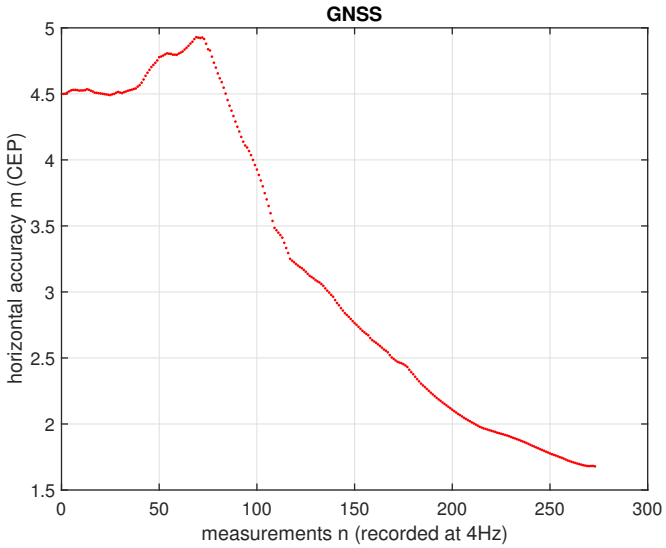


Figure 9.26: GNSS receiver horizontal accuracy when going outdoors.

It is clearly seen that when going outdoors it takes few to tens of seconds before the horizontal accuracy improves.

To aid in the transition, approach sensors could help in fixing the position when

approaching or leaving an entrance. BLE and UWB anchors were tested at the entrance while the user passed. Figures 9.27 and 9.28 show the range measurement for the UWB anchor and the RSS measurement for the BLE anchor. The UWB is much more accurate and with the UWB anchor it is easier to detect the outdoor-indoor context switches more reliably.

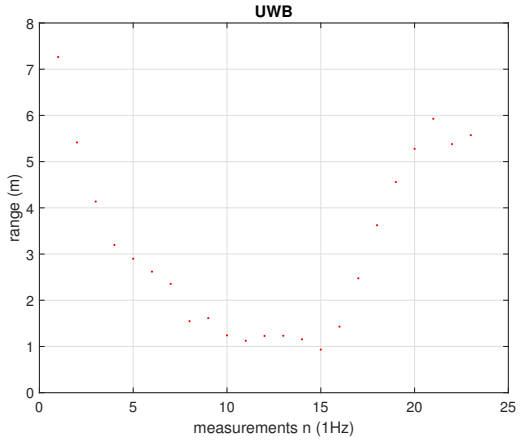


Figure 9.27: UWB ranging results while passing the anchor.

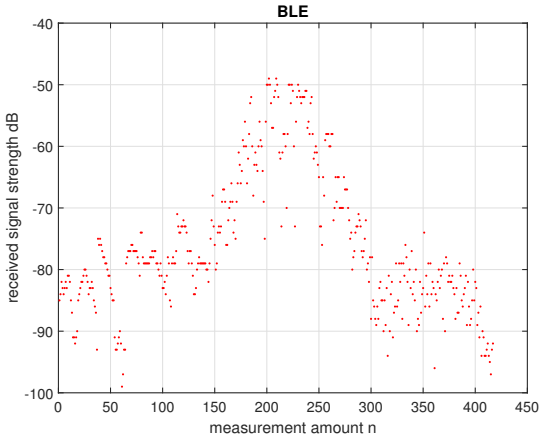


Figure 9.28: BLE signal strength results while passing the anchor.

9.5 Concluding Remarks

The novel seamless navigation filter design was tested and operates correctly. The following chapter contains a detailed summary, lessons learnt and future research suggestions.

Chapter 10

Communication and Conclusions

The work has been assessed according to the publications in conferences: Basiri et al. (2014, 2015), Peltola et al. (2016), Daniel et al. (2016), Peltola et al. (2017). and a journal: Basiri et al. (2017). A novel seamless navigation system was successfully designed and tested. This chapter summarises the experiences encountered and the likely future research opportunities. A light metric for seamlessness is considered.

10.1 On Context Inference

One research question included the examination of context inference for improving system performance. This goal was clearly achieved. The designed system is capable of switching between the particle and the Kalman filters depending on the environment.

The developed general system design takes into account the configurational flex-

ibility. This means that the sensor subsystems can be replaced by improved versions in the future. This is enabled by the plug-and-play design. Plug-and-play design includes the separation of the implementation choices and the top-level design. The interface agreements between the blocks play a crucial role. Future implementations could consist of multiple similar sensor subsystems. The blackboard model could then be used to select the best of the subsystems. This could be essential if, for example, a Bluetooth system loses an anchor and thus also accuracy. More advanced integrity checking would be necessary.

The system is also environmentally adaptable. The system can detect the environment and act upon it. The settings for stairs and settings for indoors, where different sensor subsystems are used, were switched ON successfully when the indoor flag was raised by the context engine. When getting closer to the entrance, the context transfer inference code blocks were activated. This saves computational power by not running the checks unnecessarily.

10.1.1 Filter transfer

The filter transfers were successful. The computational power management was improved by the filter switching logic. The particle filter is the most robust filter, which is more suited to and capable of deriving the initial heading at the start when the system switches on. This is achieved through positioning fixing systems killing particles that are heading in the wrong direction. Also, when the filter fails, a more sparse particle cloud can be re-initialised in order to recover efficiently. The initial

movement heading distribution was uniform. The first positioning fix initialised particle positions according to the error estimate from the positioning fixing system.

When the particle filter indicated a clear unimodal distribution, the control was transferred to the Kalman filter. This saves computational power. In between - when the particle filter scatter was greater, yet still representing an unimodal solution - the particle cloud was surrounded by the pentagon buffer. When using only the particle filter, the map matching method for each particle used a massive amount of processor time. When turning the pentagon buffer on, map matching of the particles was stopped. The pentagon buffer interacted with the map instead of each particle; this saved computational power.

10.1.2 On Map Matching

A high definition UAV survey image map was georeferenced and transferred into MATLAB using a local tangent plane coordinate system. In addition, the KML wall files were used to represent constraints for user movements in the map area. The walls contained information on their passability. The doors were considered as walls that are passable. Future maps could be enhanced by adding temporal attributes to the wall structures. The inference for displaying only the necessary information could also enhance the overall system performance by showing just the necessary related information on the map.

The pentagon buffer interacted with the walls alongside the particle and the Kalman solutions. This novel approach eased the workload of the processor and

guided the solution according to the characteristics of the changing environment. The particle cloud can disperse into two if positioning fixes are not present. A future study could involve particle cluster detection methods. The separate clusters could be surrounded by multiple pentagon buffers. In this way, the multiple trail guesses could be upheld until a new position fix is available.

The collision logic handled the pentagon buffer interaction with the walls. Yet further development of the collision logic could initiate multiple Kalman solutions when needed. This is especially the case when an obstruction interacts with the buffer and could split the solution into two. These Kalman solutions could then be surrounded by their own buffers. Multiple target tracking could benefit from this approach. To save power, the buffer approach could be used instead of the particle filter to track the targets whose position is presentable by the unimodal Kalman filter.

10.1.3 On Abstraction

Our implementation did not use any virtual sensors other than the map. For example, if calendar information was added into the system, the context inference could enhance the navigation filter status derivation according to this extra information, or information from other similar future virtual sensors. This would be executed by inputting the information from the calendar into the context engine inference process, which would then adjust the navigation filter state to be more optimal with this extra information from the calendar.

The availability block stored the defined amount of history data. There are reprocessing options available in the literature that could be added, for example by Lan & Shih (2014), to enhance the positioning solution using this history data. A good example would be smoothers to reprocess parts of the tracks. The availability block was responsible for map matching, integrity checks, history records, status checks and filter iteration initialisation. Optimisation, especially of this block, could further enhance the flow of the system.

The state machine design is closely related to the availability block design. This could be the focus of more research. The efficient use of the filter bank is essential, and more filters could be added to the bank and run in parallel or individually.

The novel track form matching exploited short-term features of the inertial solution and the long-term features of the GNSS solution. If the GNSS track was very bad, like is the case with cheaper receivers on mobile devices, the usage of only the foot mounted inertial solution with track form updates could work even better than continuous updates with poor GNSS. This also could be a future research topic. More different sensor types should be tested with the novel design and features added while testing these sensors.

10.2 On Sensor Fusion

The fusion and technology features were examined according to the research questions. First, the literature was reviewed for the current technologies used in indoor

positioning. The indoor positioning still poses the greater challenge for achieving seamlessness. The filter accuracy and the computational load are the two main priorities to be tackled in the navigation system design process. The trianlge between the usability, computing power and accuracy should be evaluated.

10.2.1 Technologies and Methods

An analytic hierarchy process revealed the best fit sensors for our needs. The camera and the light-based positioning is very accurate, but was excluded because the sensors are not the most userfriendly. Instead, UWB, BLE, Inertial and GNSS technologies survived. These were the technologies our novel seamless system is based on.

For these technologies, a revision and testing of navigation solution methods was conducted. The best approaches were then implemented into the novel tripartite filtering structure. For Bluetooth low energy technology we ended up with the second approach. This was the use of anchorwise individual RSS to distance conversion tables. These tend to change according to the battery level. This should be taken into account in future designs. For example, if the anchor sends the current battery level with its position this would enable independency of the map and compensate for the conversion table errors. The fingerprinting system is possibly even more accurate, but the laborious setup of the system makes it not likely reach any big data markets.

The UWB system was the most accurate positioning fixing system that we in-

cluded into the design implementation. The UWB anchor geometry has to be taken into account and anchor placements designed accordingly. This system can penetrate one wall structure. We used the least squares to derive position from the range results. A wall pathloss compensation can be used for anchors that are likely to have non-line of sight conditions. This is helpful in the least squares process and improves the UWB subsystem accuracy. In our tests only one anchor was corrected. Otherwise, the anchors were behind multiple walls and the signal could not reach the user receiver.

10.2.2 Characteristics Adjustments

When approaching an entrance, only the IMU was trustworthy. Adding an approach sensor to the entrance would help, especially in the case when the user leaves the building. This is the most critical region requiring additional help. An UWB anchor would be best, though BLE anchors could work as well.

The computational load particularly depends on the particle amount. This relationship between particle amount and filter accuracy was examined. A further study could be relevant for how to change the amount of particles necessary in different contexts. More particles are clearly necessary if the area has many walls and no position updates are available. For example, if the user goes into a room but then walks parallel to the wall alongside corridor on the other side of the wall. The system in this thesis needed at least 200 particles.

A Tukey-style position update for the particle filter was developed to compensate

for the too small or untrustworthy sensor quality estimates from the GNSS, UWB and BLE systems. The Tukey update includes a safe zone around the position fix where the particle weight is not affected. This improved the performance, for example, in cases where the sensor subsystem continuously indicated positions behind a wall instead of a corridor. The "correct particles" in the corridor survived. The pentagon buffer particle filter also tackled this issue.

The inertial system propagation properties were tested by a measurement campaign. Our model used a similar increase in uncertainty for all directions. A more sophisticated model could help in making the boot-mounted strapdown inertial error estimation more efficient.

10.2.3 Performance and Achievements

The different filters are suited for different tasks. The approach presented in this thesis enables the future navigation applications to exploit the results for designing more adaptable systems. The novel integrated approach is more intelligent, robust and autonomous compared to using only a single type of navigation filter. The best features of the particle filter can be exploited in situations when the area has many walls and an unique solution is difficult to track using a Kalman filter. Vice versa, when the environment is less wall-rich, computational power can be saved by using the lighter Kalman filter.

The developed design enables plug-and-play approach for sensor subsystems offering separate positioning solutions. The parallel context engine can then either

pick the best or command the navigation filter to fuse the best quality solutions into the most optimal final navigation solution.

10.3 Seamlessness

The designed seamless system is implementable on mobile devices. A metric could be useful for assessing the level of seamlessness. We need to first define the affecting attributes. These are: the sensor availability on both the device and the environment, the sensor type and the sensor accuracy. We also need a restricted area inside which we can assess the seamlessness. The further derivation is left for future research.

Bibliography

Adusumilli, S., Bhatt, D., Wang, H. & Devabhaktuni, V. (2014), Improving low-cost ins/gps positioning accuracy using quantile regression forests **LR-73**, *in* 'ION/ITM', pp. 175–180.

Afzal, M. H., Renaudin, V. & Lachapelle, G. (2011), *Magnetic Field based Heading Estimation for Pedestrian Navigation Environments* ***LR-96***, Indoor Positioning and Indoor Navigation (IPIN).

Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M. A. & Al-Khalifa, H. S. (2016), 'Ultra wideband indoor positioning technologies: Analysis and recent advances', *Sensors* **16**(5).

Alpaydin, E. (2014), Introduction to machine learning, *in* 'MIT Press, Cambridge'.

Arulampalam, M., Maskell, S., Gordon, N. & Clapp, T. (2002), 'A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking', *Trans. Sig. Proc.* **50**(2), 174–188.

URL: *http://dx.doi.org/10.1109/78.978374*

Aufderheide, D. & Krybus, W. (2010), *Towards Real-Time Camera Egomotion Estimation and Three-Dimensional Scene Acquisition from Monocular Image Streams* ***LR-43***, Indoor Positioning and Navigation(IPIN).

Ayaz, A. S., Pany, T. & Eissfeller, B. (2010), Assessment of gnss signal acquisition sensitivities for indoor and urban scenarios, *in* '2010 IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops', pp. 223–227.

Bai, Y., Jia, W., Zhang, H., Mao, Z. H. & Sun, M. (2013), Helping the blind to find the floor of destination in multistory buildings using a barometer **LR-89**, *in* '2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)', pp. 4738–4741.

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

Basiri, A., Lohan, E. S., e Silva, P. F., Peltola, P., Hill, C. & Moore, T. (2014), Overview of positioning technologies from fitness-to-purpose point of view, *in* 'International Conference on Localization and GNSS 2014 (ICL-GNSS 2014)', pp. 1–7.

Basiri, A., Lohan, E.-S., Moore, T., Winstanley, A., Peltola, P., Hill, C., Amirian,

P. & Figueiredo E Silva, P. (2017), 'Indoor location based services challenges, requirements and usability of current solutions', *Computer Science Review* .

Basiri, A., Peltola, P., e Silva, P. F., Lohan, E. S., Moore, T. & Hill, C. (2015), Indoor positioning technology assessment using analytic hierarchy process for pedestrian navigation services, *in* '2015 International Conference on Location and GNSS (ICL-GNSS)', pp. 1–6.

Bellusci, G. (2010), Ultra-Wideband Ranging for Low-Complexity Indoor Positioning Applications, PhD thesis, The University of Pisa.

Bisio, I., Delfino, A. & Lavagetto, F. (2015), A simple ultrasonic indoor/outdoor detector for mobile devices, *in* '2015 International Wireless Communications and Mobile Computing Conference (IWCMC)', pp. 137–141.

Boll, S., Asif, A. & Heuten, W. (2011), 'Feel your route: a tactile display for car navigation **LR-82**', *IEEE Pervasive Computing* **10**(3), 35–42.

Bonenberg, L. K., Hancock, C. & Roberts, G. W. (2013), 'Locata performance in a long term monitoring **LR-46**', *Journal of Applied Geodesy* **7**(4), 271–280.

Bonenberg, L. K., Roberts, G. W. & Hancock, C. M. (2011), 'Using Locata to augment GNSS in a kinematic urban environment **LR-47**', *Archives of Photogrammetry, Cartography and Remote Sensing, Vol. 22, 2011, p. 63-74* **22**, 63–74.

Brajdic, A. & Harle, R. (2013), Walk detection and step counting on unconstrained smartphones, *in* 'Proceedings of the 2013 ACM International Joint Conference

on Pervasive and Ubiquitous Computing', UbiComp '13, ACM, New York, NY, USA, pp. 225–234.

URL: *http://doi.acm.org/10.1145/2493432.2493449*

Bras, L., Nuno, B. C., Pinho, P., Kulas, L. & Nyka, K. (2012), 'A review of antennas for indoor positioning systems', *International Journal of Antennas and Propagation* . Copyright - Copyright 2012 Luis Brs et al. Luis Brs et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited; Last updated - 2015-04-05.

URL: *https://search.proquest.com/docview/1282316114?accountid=27303*

Bray, D. E. & Stanley, R. K. (1997), *Nondestructive Evaluation, A Tool in Design, Manufacturing and Service*, CRC Press.

Brena, R. F., Juan, P. G., Galvn-Tejada, Carlos, E., Muoz-Rodriguez, D., Vargas-Rosales, C. & Fangmeyer, James, J. (2017), 'Evolution of indoor positioning technologies: A survey', *Journal of Sensors* . Copyright - Copyright 2017 Ramon F. Brena et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited; Last updated - 2017-04-15.

URL: *https://search.proquest.com/docview/1885959392?accountid=27303*

Briers, M., Maskell, S. R. & Wright, R. (2003), A rao-blackwellised unscented kalman

filter, *in* 'Sixth International Conference of Information Fusion, 2003. Proceedings of the', Vol. 1, pp. 55–61.

Candy, J. V. (2008), *Bayesian Signal Processing, Classical, Modern, and Particle Filtering Methods*, Wiley.

Cerruela Garca, G., Luque Ruiz, I. & Gmez-Nieto, M. . (2016), 'State of the art, trends and future of bluetooth low energy, near field communication and visible light communication in the development of smart cities', *Sensors* **16**(11).

Chui, C. K. & Chen, G. (2009), *Kalman FIltering with Real-Time Applications*, Springer.

Ciurana, M., Barcel-Arroyo, F. & Martn-Escalona, I. (2011), 'Comparative performance evaluation of {IEEE} 802.11v for positioning with time of arrival **LR-90**', *Computer Standards & Interfaces* **33**(3), 344 – 349.
URL: *http://www.sciencedirect.com/science/article/pii/S0920548910001005*

Clark, M., Linnane, F., Sharples, S. D. & Somekh, M. G. (1998), 'Frequency control in laser ultrasound with computer generated holography', *Applied Physics Letters* **72**(16), 1963–1965.

Clark, M., Sharples, S. D. & Somekh, M. G. (1998), Fast laser generated Rayleigh wave scanning microscope, *in* '1998 IEEE Ultrasonics Symposium', IEEE and UFFC, IEEE, Piscataway, NJ, USA.

Clark, M. & Smith, R. (1996), 'A direct-search method for the computer design of holograms', *Optics Communications* **124**(1-2), 150–164.

Contreras, D., Castro, M. & de la Torre, D. S. (2017), 'Performance evaluation of bluetooth low energy in indoor positioning systems **LR-100**', *Transactions on Emerging Telecommunications Technologies* **28**(1), e2864–n/a. e2864.
URL: *http://dx.doi.org/10.1002/ett.2864*

Damian, D. D., Newton, T. H., Pfeifer, R. & Okamura, A. M. (2015), 'Artificial tactile sensing of position and slip speed by exploiting geometrical features', *IEEE/ASME Transactions on Mechatronics* **20**(1), 263–274.

Daniel, O., Raasakka, J., Peltola, P., Frhle, M., Rivero-Rodriguez, A., Wymeersch, H. & Nurmi, J. (2016), *Blind sub-Nyquist GNSS signal detection*, IEEE, pp. 6575–6579.

Davidson, P. & Piche, R. (2016), 'A survey of selected indoor positioning methods for smartphones', *IEEE Communications Surveys Tutorials* **PP**(99), 1–1.

Dean, A. J., Langelaan, J. W. & Brennan, S. N. (2010), Improvements in terrain-based road vehicle localization by initializing an unscented kalman filter using particle filters, *in* 'Proceedings of the 2010 American Control Conference', pp. 700–707.

Decawave.com (12/ 2016), Decawave application notes, Technical report.

DeCuir, J. (2014*a*), 'Introducing bluetooth smart: Part 1: A look at both classic and new technologies. **LR-3**', *IEEE Consumer Electronics Magazine* .

DeCuir, J. (2014*b*), 'Introducing bluetooth smart: Part ii: Applications and updates. **LR-4**', *IEEE Consumer Electronics Magazine* .

Deng, Z., Yu, Y., Yuan, X., Wan, N. & Yang, L. (2013), 'Situation and development tendency of indoor positioning **LR-30**', *China Communications* **10**(3), 42–55.

Dogruer, C. U. (2014), Estimation of odometer parameters with mmae and lse, *in* '2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics', pp. 1728–1733.

Doucet, A., de Freitas, N. & Gordon, N. (2001), *Sequential Monte Carlo Methods in Practice*, Springer.

Doucet, A. & Johansen, A. M. (2011), 'A tutorial on particle filtering and smoothing: fifteen years later'.

Eid, B. F., Durkee, S. R. & Darwin, K. P. (n.d.), 'Universal sensor interface system and method **LR-102**'.

Esteban, J., Starr, A., Willetts, R., Hannah, P. & Bryanston-Cross, P. (2005), *A review of data fusion models and architectures: Towards engineering guidelines.* ***LR-48***, Neural Computing & Applications.

Faragher, R. & Harle, R. (2015), 'Location fingerprinting with bluetooth low energy beacons', *IEEE Journal on Selected Areas in Communications* **33**(11), 2418–2428.

Faragher, R. M., Sarno, C. & Newman, M. (2012*a*), Opportunistic radio slam for indoor navigation using smartphone sensors **LR-57**, *in* 'Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION', pp. 120–128.

Faragher, R. M., Sarno, C. & Newman, M. (2012*b*), Opportunistic radio slam for indoor navigation using smartphone sensors **LR-57**, *in* 'Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium', pp. 120–128.

Feng, G., Shen, C., Long, C. & Dong, F. (2015), Gdop index in uwb indoor location system experiment, *in* 'SENSORS, 2015 IEEE', pp. 1–4.

Fernandez, D., Barcelo-Arroyo, F., Martin-Escalona, I., Ciurana, M., Jofre, M. & Gutierrez, E. (2011), Fusion of wlan and gnss observables for positioning in urban areas: The position ambiguity, *in* '2011 IEEE Symposium on Computers and Communications (ISCC)', pp. 748–751.

Ferreira, K. R., Vinhas, L., Monteiro, A. M. V. & Camara, G. (2012), Moving objects and kml files, *in* 'Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference on', pp. 355–359.

Figueiras, J. & Frattasi, S. (2010), *Mobile Positioning and Tracking: From Conventional to Cooperative Techniques*, Wiley.

Figueiredo e Silva, P., Basiri, A., Lohan, E. S., Pinchin, J., Hill, C. & Moore, T. (2015), On the impact of intra-system interference for ranging and positioning with bluetooth low energy, *in* 'Proceedings of the 5th International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments', MELT '15, ACM, New York, NY, USA, pp. 1:1–1:4.

URL: *http://doi.acm.org/10.1145/2830571.2830573*

Filgueiras, T. S., Lima, A. C. O., Baima, R. L., Oka, G. T. R., Cordovil, L. A. Q. & Bastos, M. P. (2016), Vibrotactile sensory substitution on personal navigation: Remotely controlled vibrotactile feedback wearable system to aid visually impaired, *in* '2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)', pp. 1–5.

Frontoni, E. & Zingaretti, P. (2006), *An Efficient Similarity Metric for Omnidirectional Vision Sensors*, 2nd European Conference on Mobile Robots (ECMR 05).

Fujimoto, M. & Nakamori, E. (2011), 'A broad-typed multi-sensing-range method for indoor position estimation of passive rfid tags. **LR-13**', *Indoor Positioning, IPIN* .

Gezici, S., Tian, Z., Giannakis, G., Kobayashi, H., Molisch, A., Poor, H. & Sahinoglu, Z. (2005), 'Localization via ultra-wideband radios: A look at positioning aspects of future sensor networks. **LR-12**', *IEEE Signal Processing Magazine, 22(4), 7084.* .

Groves, P. (2013*a*), *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition:*, GNSS/GPS, Artech House.
URL: *https://books.google.co.uk/books?id=t94fAgAAQBAJ*

Groves, P. D. (2013*b*), 'The pnt boom: Future trends in integrated navigation', *Inside GNSS* **8(2)**, 44–49.

Groves, P. D., Jiang, Z., Wang, L. & Ziebart, M. (2012), Intelligent urban positioning, shadow matching and non-line-of-sight signal detection, *in* 'Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing, (NAVITEC), 2012 6th ESA Workshop on', pp. 1–8.

Groves, P. D., Wang, L., Walter, D., Martin, H., Voutsis, K. & Jiang, Z. (2014), The four key challenges of advanced multisensor navigation and positioning **LR-78**, *in* '2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014', pp. 773–792.

Hao, J. (2015), Indoor Collaborative Positioning based on a Multi-sensor and Multi-user System, PhD thesis, The University of Nottingham.

Hao, Y., Zhang, Z. & Xia, Q. (2010), Research on data fusion for sins/gps/ magnetometer integrated navigation based on modified cdkf **LR-49**, *in* 'Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on', Vol. 2, pp. 1215–1219.

Hardegger, M., Roggen, D. & Tröster, G. (2015), '3d actionslam: wearable per-

son tracking in multi-floor environments', *Personal and Ubiquitous Computing* **19**(1), 123–141.

URL: *http://dx.doi.org/10.1007/s00779-014-0815-y*

Harle, R. (2013), 'A survey of indoor inertial positioning systems for pedestrians **LR-36**', *IEEE Communications Surveys Tutorials* **15**(3), 1281–1293.

Hartmann, F., Rifat, D. & Stork, W. (2016), Hybrid indoor pedestrian navigation combining an ins and a spatial non-uniform uwb-network, *in* '2016 19th International Conference on Information Fusion (FUSION)', pp. 549–556.

Hasani, H., Peixeiro, C., Skrivervik, A. K. & Perruisseau-Carrier, J. (2015), 'Single-layer quad-band printed reflectarray antenna with dual linear polarization', *IEEE Transactions on Antennas and Propagation* **63**(12), 5522–5528.

Hasani, M., Talvitie, J., Sydanheimo, L., Lohan, E. & Ukkonen, L. (2015), 'Hybrid wlan-rfid indoor localization solution utilizing textile tag. **LR-15**', *IEEE Antennas and Wireless Propagation Letters, PP(99)* .

Haug, A. (2012), *Bayesian Estimation and Tracking*, Wiley, Hoboken.

He, Z., Hu, Y., Wu, J., Wang, J., Hou, J. & Wang, K. (2011), A comprehensive method for multipath performance analysis of gnss navigation signals, *in* 'Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on', pp. 1–6.

Hejc, G., Seitz, J., Boronat, J. G. & Vaupel, T. (2014), *Seamless Indoor Outdoor Positioning Using Bayesian Sensor Data Fusion on Mobile and Embedded Devices* ***LR-55***, IEEE/ION GNSS+.

Hide, C. (2003), Integration of GPS and Low Cost INS measurements, PhD thesis, The University of Nottingham.

Hide, C., Botterill, T. & Andreotti, M. (2010), *Low cost vision-aided IMU for pedestrian navigation* ***LR-51***, Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS).

Hide, C., Moore, T. & Hill, C. (2007), 'A multi-sensor navigation filter for high accuracy positioning in all environments **LR-109**', *The Journal of Navigation* **60**(3), 409–425. Copyright - The Royal Institute of Navigation; Last updated - 2015-08-15.

URL: *https://search.proquest.com/docview/229600734?accountid=27303*

Höflinger, F., Hoppe, J., Zhang, R., Ens, A., Reindl, L., Wendeberg, J. & Schindelhauer, C. (2014), Acoustic indoor-localization system for smart phones **LR-87**, *in* 'Systems, Signals Devices (SSD), 2014 11th International Multi-Conference on', pp. 1–4.

Hol, J. D., Schon, T. B. & Gustafsson, F. (2006), On resampling algorithms for particle filters, *in* '2006 IEEE Nonlinear Statistical Signal Processing Workshop', pp. 79–82.

Honorato, J. L., Spiniak, I. & Miguel, T.-T. (2008), *Human Detection Using Thermopiles*, IEEE Latin American Robotic Symposium, 2008. LARS '08.

http://en.wikipedia.org/wiki/IEEE_802.11 (n.d.), *IEEE 802.11* ***LR-21***.

http://indoo.rs/bluetooth-low-energy-solution released/ (n.d.), *indoo.rs* ***LR-28***.

https://www.bluetooth.org/en-us/specification/adopted—specifications (2017), *Bluetooth specification* ***LR-2***.

http://www.researchgate.net/profile/Lukasz_Bonenberg/publications (n.d.), *Bonenberg* ***LR-31***.

http://www.ubisense.net/en/products/smart factory (n.d.), *UbiSense* ***LR-29***.

Hu, C., Chen, W., Chen, Y. & Liu, D. (2003), Adaptive kalman filtering for vehicle navigation **LR-106**, *in* 'Positioning, Vol. 1 No. 4'.

Humphreys, T. E., Murrian, M., van Diggelen, F., Podshivalov, S. & Pesyna, K. M. (2016), On the feasibility of cm-accurate positioning via a smartphone's antenna and gnss chip, *in* '2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)', pp. 232–242.

Jee, G. & Song, J.-H. (2010), A study on gps/dr car navigation system using vehicle movement information **LR-84**, *in* 'Coordinates'.

Jimenez, A. R., Seco, F., Prieto, J. C. & Guevara, J. (2010), 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', pp. 135–143.

Jing, H. (2013), *Collaborative Positioning for Indoor Mobile Users* ***LR-64***, NGI, 2nd year report.

Jones, L. A., Held, D. & Hunter, I. (2010), Surface waves and spatial localization in vibrotactile displays **LR-81**, *in* '2010 IEEE Haptics Symposium', pp. 91–94.

Julier, S. J. & Uhlmann, J. K. (2002), Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations, *in* 'Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)', Vol. 2, pp. 887–892 vol.2.

Julier, S. & Uhlman, J. (2009), 'General decentralized data fusion with covariance intersection', *Handbook of Multisensor Data Fusion, Theory and Practice* .

Kajioka, S., Mori, T., Uchiya, T. & Matsuo, H. (2014), 'Experiment of indoor position presumption based on rssi of bluetooth le beacon **LR-5**'.

Khalifa, S., Hassan, M. & Seneviratne, A. (2013), Adaptive pedestrian activity classification for indoor dead reckoning systems, *in* 'International Conference on Indoor Positioning and Indoor Navigation', pp. 1–7.

Khan, M. I. & Syrjärinne, J. (2013), Investigating effective methods for integration of building's map with low cost inertial sensors and wifi-based positioning **LR-92**, *in*

'Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on', pp. 1–8.

Kirchhof, N. (2013), Optimal placement of multiple sensors for localization applications, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on', pp. 1–10.

Kivimäki, T., Vuorela, T., Peltola, P. & Vanhala, J. (2014), 'A review on device-free passive indoor positioning methods **LR-39**', *International Journal of Smart Home* **8(1)**, 71–94.

Koch, W. (2014), *Tracking and Sensor Data Fusion*, Springer.

Koivisto, M., Nurminen, H., Ali-Lytty, S. & Piche, R. (2015), *Graph-Based Map Matching for Indoor Positioning*, IEEE. ISBN of the USB proceedings. Will appear in IEEE Xplore. ORG=ase,0.8 ORG=mat,0.2.

Kok, M., Hol, J. D. & Schn, T. B. (2015), 'Indoor positioning using ultrawideband and inertial measurements', *IEEE Transactions on Vehicular Technology* **64**(4), 1293–1303.

Kolakowski, M. & Djaja-Josko, V. (2016), Tdoa-twr based positioning algorithm for uwb localization system, *in* '2016 21st International Conference on Microwave, Radar and Wireless Communications (MIKON)', pp. 1–4.

Kray, C. & Kortuem, G. (n.d.), 'Adaptive positioning for ambient systems **LR-107**'.

Kuusniemi, H., Bhuiyan, M. Z. H., Strm, M., Sderholm, S., Jokitalo, T., Chen, L. & Chen, R. (2012), Utilizing pulsed pseudolites and high-sensitivity gnss for ubiquitous outdoor/indoor satellite navigation **LR-40**, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on', pp. 1–7.

Kuusniemi, H., Chen, L., Pei, L., Liu, J., Chen, Y., Ruotsalainen, R. & Chen, R. (2011), *Evaluation of Bayesian Approaches for Multi-sensor Multi-network Seamless Positioning* ***LR-65***, ION GNSS.

Kwak, M. & Chong, J. (2010), A new double two-way ranging algorithm for ranging system, *in* '2010 2nd IEEE InternationalConference on Network Infrastructure and Digital Content', pp. 470–473.

Laaraiedh, M., Amiot, N. & Uguen, B. (2013), Refined characterization of rssi with practical implications for indoor positioning **LR-74**, *in* 'Positioning Navigation and Communication (WPNC), 2013 10th Workshop on', pp. 1–5.

Ladetto, Q. (2000), On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive kalman filtering **LR-108**, *in* 'Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000), Salt Lake City, UT', pp. 1735–1740.

Lamport, L. (2.2.1993), *Latex: A Document Preparation System*, second edn, Addison-Wesley, Reading, Massachusetts.

Lan, K. C. & Shih, W. Y. (2014), 'Using smart-phones and floor plans for indoor location tracking - withdrawn', *IEEE Transactions on Human-Machine Systems* **44**(2), 211–221.

Laskar, Z., Huttunen, S., C., D. H., Rahtu, E. & Kannala, J. (2016), Robust loop closures for scene reconstruction by combining odometry and visual correspondences, *in* '2016 IEEE International Conference on Image Processing (ICIP)', pp. 2603–2607.

Laverne, M., George, M., Lord, D., Kelly, A. & Mukherjee, T. (2011), Experimental validation of foot to foot range measurements in pedestrian tracking **LR-66**, *in* 'ION GNSS'.

Le, L. & Gebre-Egziabher, D. (2016), Constrained, networked inertial navigation for human and humanoid robot feet pose estimation, *in* '2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)', pp. 76–84.

Li, H. (2014), 'Low-cost 3d bluetooth indoor positioning with least square. **LR-6**', *Wireless Personal Communications, 13311344.* .

Lin, T., Ma, M., Broumandan, A. & Lachapelle, G. (2013), 'Demonstration of a high sensitivity {GNSS} software receiver for indoor positioning', *Advances in Space Research* **51**(6), 1035 – 1045. Recent progresses on Beidou/COMPASS and other Global Navigation Satellite Systems (GNSS) - I.

URL: *http://www.sciencedirect.com/science/article/pii/S0273117712003845*

Linnane, F. (1998), Investigation of an all-optical laser ultrasound SAW system with respect to velocity measurement and imaging, PhD thesis, The University of Nottingham.

Liu, H., Darabi, H., Banerjee, P. & Liu, J. (2007), 'Survey of wireless indoor positioning techniques and systems', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **37**(6), 1067–1080.

Liu, M., Ho, H. P., Somekh, M. G. & Weaver, J. M. R. (1995), 'Noncontacting optical-generation of focused surface acoustic waves using a customized zoneplate', *Electronics Letters* **31**(4), 264–265.

Liu, X., Leung, H., Valin, P. & Boss, . (2012), Multisensor joint tracking and identification using particle filter and dempster-shafer fusion **LR-59**, *in* '2012 15th International Conference on Information Fusion', pp. 902–909.

Lohan, E.-S., Talvitie, J., Figueiredo e Silva, P., Nurminen, H., Ali-Lytty, S. & Piche, R. (2015), *Received Signal Strength models for WLAN and BLE-based indoor positioning in multi-floor buildings*, IEEE. ORG=elt,0.7 ORG=ase,0.2 ORG=mat,0.1.

Lohan, E. S., Talvitie, J. & Granados, G. S. (2016), Data fusion approaches for wifi fingerprinting, *in* '2016 International Conference on Localization and GNSS (ICL-GNSS)', pp. 1–6.

Lucas-Sabola, V., Seco-Granados, G., Lpez-Salcedo, J. A., Garca-Molina, J. A. & Crisci, M. (2016), Cloud gnss receivers: New advanced applications made possible,

in '2016 International Conference on Localization and GNSS (ICL-GNSS)', pp. 1–6.

Luo, R. C., Chang, C. C. & Lai, C. C. (2011), *Multisensor Fusion and Integration: Theories, Applications, and its Perspectives* ***LR-50***, IEEE/SENSORS.

Madgwick, S. (2013), Gait tracking with x-imu, *in* 'x-io.co.uk/gait-tracking-with-x-imu/ referred 7.7.2017'.

Mansour, M. F. & Waters, D. W. (2013), Map-assisted kalman filtering, *in* '2013 IEEE International Conference on Acoustics, Speech and Signal Processing', pp. 3208–3212.

Mautz, R. (2012), Indoor positioning technologies **LR-1**, PhD thesis, ETH Zurich. **URL:** *http://e-collection.library.ethz.ch/view/eth:5659*

Meng, X., Gao, Y., Kwok, K. H. & Zhao, H. (2012), Assessment of uwb for ubiquitous positioning and navigation **LR-69**, *in* 'Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), 2012', pp. 1–6.

Mikhaylov, K., Tikanmki, A., Petjjrvi, J., Hmlinen, M. & Kohno, R. (2016), On the selection of protocol and parameters for uwb-based wireless indoors localization, *in* '2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)', pp. 1–5.

Mitchell, H. B. (2007), *Multi-Sensor Data Fusion : An Introduction*, Springer, Berlin, Heidelberg.

Moghtadaiee, V., Dempster, A. & Lim, S. (2011), 'Indoor localization using fm radio signals: A fingerprinting approach. **LR-8**', *International Conference on Indoor Positioning and Indoor Navigation, IPIN 2011, 39.* .

Mulloni, A., Wagner, D., Barakonyi, I. & Schmalstieg, D. (2009), 'Indoor positioning and navigation with camera phones', *IEEE Pervasive Computing* **8**(2), 22–31.

Navratil, V., Karasek, R. & Vejrazka, F. (2016), Position estimate using radio signals from terrestrial sources, *in* '2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)', pp. 799–806.

Nguyen, T. L., Zhang, Y. & Griss, M. (2010), Probin: Probabilistic inertial navigation **LR-72**, *in* 'The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)', pp. 650–657.

Nurminen, H., Talvitie, J., Ali-Loytty, S., Muller, P., Lohan, E., Piche, R. & Renfors, M. (2012), 'Statistical path loss parameter estimation and positioning using rss measurements in indoor wireless networks. **LR-10**', *International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012.* .

Palumbo, F., Barsocchi, P., Chessa, S. & Augusto, J. C. (2015), A stigmergic approach to indoor localization using bluetooth low energy beacons, *in* 'Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on', pp. 1–6.

Pateriya, R. K. & Sharma, S. (2011), The evolution of rfid security and privacy: A

research survey **LR-14**, *in* 'Communication Systems and Network Technologies (CSNT), 2011 International Conference on', pp. 115–119.

Paull, L., Saeedi, S., Seto, M. & Li, H. (2014), 'Auv navigation and localization: A review', *IEEE Journal of Oceanic Engineering* **39**(1), 131–149.

Pelosi, M., Pedersen, G., Rosa, F. D. & Nurmi, J. (2012), 'The role of the propagation environment in rss-based indoor positioning using mass market devices. **LR-9**', *LAPC 2012 - 2012 Loughborough Antennas and Propagation Conference.* .

Peltola, P., Hill, C. & Moore, T. (2016), Particle filter for context sensitive indoor pedestrian navigation, *in* '2016 International Conference on Localization and GNSS (ICL-GNSS)', pp. 1–6.

Peltola, P., Hill, C. & Moore, T. (2017), Adaptive real-time dual-mode filter design for seamless pedestrian navigation, *in* '2017 International Conference on Localization and GNSS (ICL-GNSS)'.

Perlmutter, M. & Robin, L. (2012), High-performance, low cost inertial mems: A market in motion! **LR-62**, *in* 'Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION', pp. 225–229.

Perttula, A., Leppkoski, H., Kirkko-Jaakkola, M., Davidson, P., Collin, J. & Takala, J. (2014), 'Distributed indoor positioning system with inertial measurements

and map matching', *IEEE Transactions on Instrumentation and Measurement* **63**(11), 2682–2695.

Pesyna, K. M., Heath, R. W. & Humphreys, T. E. (2014), Centimeter positioning with a smartphone-quality gnss antenna, *in* 'ION GNSS'.

Pinchin, J., Hide, C. & Moore, T. (2012*a*), A particle filter approach to indoor navigation using a foot mounted inertial navigation system and heuristic heading information **LR-41**, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on', pp. 1–10.

Pinchin, J., Hide, C. & Moore, T. (2012*b*), The use of high sensitivity gps for initialisation of a foot mounted inertial navigation system **LR-42**, *in* 'Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION', pp. 998–1007.

Popleteev, A., Osmani, V. & Mayora, O. (2013), 'Investigation of indoor localization with ambient FM radio stations', *CoRR* **abs/1308.1606**.
URL: *http://arxiv.org/abs/1308.1606*

Prete, A. D., Nori, F., Metta, G. & Natale, L. (2012), Control of contact forces: The role of tactile feedback for contact localization **LR-83**, *in* '2012 IEEE/RSJ International Conference on Intelligent Robots and Systems', pp. 4048–4053.

Quddus, M. A., Ochieng, W. Y. & Noland, R. B. (2007), 'Current map-matching algorithms for transport applications: State-of-the art and future research direc-

tions', *Transportation Research Part C: Emerging Technologies* **15**(5), 312 – 328.
URL: *http://www.sciencedirect.com/science/article/pii/S0968090X07000265*

Quinchia, A. G., Ferrer, C., Falco, G., Falletti, E. & Dovis, F. (2012), Analysis and modelling of mems inertial measurement unit **LR-71**, *in* '2012 International Conference on Localization and GNSS', pp. 1–7.

Rahim, K. A. (2012), Heading drift mitigation for low-cost inertial pedestrian navigation., PhD thesis, The University of Nottingham.

Raitoharju, M., Nurminen, H. & Piché, R. (2015), 'Kalman filter with a linear state model for pdr+wlan positioning and its application to assisting a particle filter', *EURASIP Journal on Advances in Signal Processing* **2015**(1), 33.
URL: *http://dx.doi.org/10.1186/s13634-015-0216-z*

Rasmussen, C. E. & Nickisch, H. (2010), 'Gaussian processes for machine learning (gpml) toolbox', *J. Mach. Learn. Res.* **11**, 3011–3015.
URL: *http://dl.acm.org/citation.cfm?id=1756006.1953029*

Rein, K. & Biermann, J. (2013), *Your high-level information is my low-level data - A new look at terminology for multi-level fusion* ***LR-54***, IEEE/FUSION Information Fusion.

Reinstein, M. & Hoffmann, M. (2011), Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer **LR-85**, *in* 'Robotics and Automation (ICRA), 2011 IEEE International Conference on', pp. 617–624.

Rishabh, I., Kimber, D. & Adcock, J. (2012), Indoor localization using controlled ambient sounds **LR-88**, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on', pp. 1–10.

Rivero-Rodriguez, A., Leppkoski, H. & Pich, R. (2014), Semantic labeling of places based on phone usage features using supervised learning, *in* '2014 Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)', pp. 97–102.

Ruotsalainen, L., Kuusniemi, H. & Chen, R. (2010), *Overview of methods for visual-aided pedestrian navigation* ***LR-70***, Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS).

Ruotsalainen, L., Kuusniemi, H. & Chen, R. (2011), *Heading Change Detection for Indoor Navigation with a Smartphone Camera*, 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN).

Ruppelt, J., Kronenwett, N., Scholz, G. & Trommer, G. F. (2016), High-precision and robust indoor localization based on foot-mounted inertial sensors, *in* '2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)', pp. 67–75.

Santoso, F., Garratt, M. A. & Anavatti, S. G. (2017), 'Visual x2013;inertial navigation systems for aerial robotics: Sensor fusion and technology', *IEEE Transactions on Automation Science and Engineering* **14**(1), 260–275.

Sasiadek, J. Z., Wang, Q. & Zeremba, M. B. (2000), Fuzzy adaptive kalman filtering

for ins/gps data fusion **LR-105**, *in* 'Proceedings of the 2000 IEEE International Symposium on Intelligent Control. Held jointly with the 8th IEEE Mediterranean Conference on Control and Automation (Cat. No.00CH37147)', pp. 181–186.

Schatzberg, U., Banin, L. & Amizur, Y. (2014), Enhanced wifi tof indoor positioning system with mems-based ins and pedometric information **LR-77**, *in* '2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014', pp. 185–192.

Schmidt, G. T. & Phillips, R. E. (2010), Ins/gps integration architectures, Technical report, DTIC Document.

Scruby, C. B., Dewhurst, R. J., Hutchins, D. A. & Palmer, S. B. (1982), *Research Techniques in Nondestructive Testing*, Vol. 4, Academic Press, San Diego, CA, USA, chapter 5, pp. 281–327. (Editor: R. S. Sharpe).

Seco, F., Plagemann, C., Jimnez, A. R. & Burgard, W. (2010), Improving rfid-based indoor positioning accuracy using gaussian processes, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on', pp. 1–8.

Sendra, S., Fernandez, P., Turro, C. & Lloret, J. (2010), Ieee 802.11a/b/g/n indoor coverage and performance comparison, *in* 'Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on', pp. 185–190.

Serra, R., Knittel, D., Croce, P. D. & Peres, R. (2016), 'Activity recognition with

smart polymer floor sensor: Application to human footstep recognition', *IEEE Sensors Journal* **16**(14), 5757–5775.

Sharp, I., Yu, K. & Guo, Y. J. (2009), 'Gdop analysis for positioning system design', *IEEE Transactions on Vehicular Technology* **58**(7), 3371–3382.

Sharp, I., Yu, K. & Hedley, M. (2012), 'On the gdop and accuracy for indoor positioning', *IEEE Transactions on Aerospace and Electronic Systems* **48**(3), 2032–2051.

Siciliano, B. & Khatib, O. (2008), *Handbook of Robotics*, Springer.

Skog, I., Nilsson, J. O. & Hndel, P. (2010), Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on', pp. 1–6.

Skulich, M., Bonenberg, L. & Meng, X. (2013), 'Utilisation of uwb in the mining excavation collapse scenario. **LR-11**', *Proceedings of 2nd International Symposium on Deformation Monitoring (JISDM)* .

Snidaro, L., Visentini, I., Llinas, J. & Foresti, G. (2013), *Multi-Sensor Fusion for Navigation of Autnomous Vehicles*, IEEE/FUSION Information Fusion.

Sola, J. (2017), Quaternion kinematics for the error state kalman filter, *in* 'www.iri.upc.edu/people/jsola/JoanSola/objectes/notes/kinematics.pdf referred 7.7.2017'.

Soloviev, A. (2014), *Multi-Sensor Fusion for Navigation of Autnomous Vehicles*, IEEE/ION GNSS+.

Somekh, M. G., Liu, M., Ho, H. P. & See, C. W. (1995), 'An accurate non-contacting laser based system for surface wave velocity measurement', *Measurement Science and Technology* **6**, 1329–1337.

Talvitie, J., Lohan, E. & Renfors, M. (2014), 'The effect of coverage gaps and measurement inaccuracies in fingerprinting based indoor localization. **LR-7**'.

Tamazin, M. E. H. R. (2011), Benefits of combined gps/glonass processing for high sensitivity receivers, *in* 'Masters Abstracts International', Vol. 50.

Tejmlova, L., Sebesta, J. & Zelina, P. (2016), Artificial neural networks in an inertial measurement unit, *in* '2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA)', pp. 176–180.

Torres-Solis, J., Falk, T. H. & Chau, T. (2010), 'A review of indoor localization technologies: towards navigational assistance for topographical disorientation **LR-35**', *Ambient Intelligence, Felix Jesus Villanueva Molina (Ed.), InTech* .

Trawny, N. & Roumeliotis, S. (2005), Indirect kalman filter for 3d attitude estimation.

Tunca, C., Pehlivan, N., Ak, N., Arnrich, B., Salur, G. & Ersoy, C. (2017), 'Inertial sensor-based robust gait analysis in non-hospital settings for neurological disorders', *Sensors* **17**(4).

Viswanathan, S. & Srinivasan, S. (2015), Improved path loss prediction model for short range indoor positioning using bluetooth low energy, *in* '2015 IEEE SENSORS', pp. 1–4.

Walder, U. & Bernoulli, T. (2010), Context-adaptive algorithms to improve indoor positioning with inertial sensors **LR-80**, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on', pp. 1–6.

Walter, C. & Schweinzer, H. (2014), Locating of objects with discontinuities, boundaries and intersections using a compact ultrasonic 3d sensor, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2014 International Conference on', pp. 591–600.

Wan, C., Zhao, L., Ding, Y. & Xue, S. (2017), 'A two-phase ranging algorithm for sensor localization in structural health monitoring', *Advances in Mechanical Engineering* **9**(1), 1687814016685964.

URL: *http://dx.doi.org/10.1177/1687814016685964*

Wang, H., Bauer, G., Kirsch, F. & Vossiek, M. (2012), Optimization of fusion algorithm for hybrid pedestrian localization and navigation **LR-60**, *in* 'Positioning Navigation and Communication (WPNC), 2012 9th Workshop on', pp. 163–168.

Wang, J., Gao, Y., Li, Z., Meng, X. & Hancock, C. M. (2016), 'A tightly-coupled gps/ins/uwb cooperative positioning sensors system supported by v2i communication', *Sensors* **16**(7).

Wang, Y., Virrantaus, K., Pei, L., Chen, R. & Chen, Y. (2012), *3D Personal Navigation in Smart Phone using Geocoded Images* ***LR-61***, GPS World.

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

Wirola, L., Laine, T. A. & Syrjrinne, J. (2010), Mass-market requirements for indoor positioning and indoor navigation **LR-37**, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on', pp. 1–7.

Won, P., Biglarbegian, M. & Melek, W. (2015), 'Development of an effective docking system for modular mobile self-reconfigurable robots using extended kalman filter and particle filter', *Robotics* **4**(1), 25–49. Copyright - Copyright MDPI AG 2015; Last updated - 2015-07-13.

URL: *https://search.proquest.com/docview/1672869008?accountid=27303*

Woodman, O. & Harle, R. (2008), Pedestrian localisation for indoor environments, *in* 'Proceedings of the 10th International Conference on Ubiquitous Computing', UbiComp '08, ACM, New York, NY, USA, pp. 114–123.

URL: *http://doi.acm.org/10.1145/1409635.1409651*

www.navipedia.net (Accessed 20.9.2017), *Conventional Celestial Reference System*, ESA.

www.opengeospatial.org (Accessed 16.2.2017), *Open Geospatial Consortium web site, www.opengeospatial.org*, Open Geospatial Consortium, IndoorGML Indoor Location Standards Working Group.

Xi, J., Yang, C., Mason, A. & Zhong, P. (2006), Adaptive multi-sensor interface system-on-chip **LR-103**, *in* '2006 5th IEEE Conference on Sensors', pp. 50–53.

Yang, C. & Soloviev, A. (2014), Covariance analysis of spatial and temporal effects of collaborative navigation **LR-76**, *in* '2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014', pp. 989–998.

Ye, J., sheng Ma, X. & Yang, L. (2008), Design and realization of ahp toolbox in matlab, *in* '2008 IEEE International Conference on Granular Computing', pp. 740–745.

Yin, H., Guo, H., Deng, X., Yu, M. & Xiong, J. (2014), Pedestrian dead reckoning indoor positioning with step detection based on foot-mounted imu **LR-58**, *in* 'ION/ITM', pp. 186–192.

Yoshiike, N., Morinaka, K., Hashimoto, K., Kawaguri, M. & Tanaka, S. (1999), *360 direction type human information sensor* ***LR-101***, Sensors and Actuators A: Physical.

Yus, R., Mena, E., Ilarri, S. & Illarramendi, A. (2014), 'Sherlock: Semantic management of location-based services in wireless environments', *Pervasive and Mobile Computing* **15**, 87 – 99. Special Issue on Information Management in Mobile Ap-

plicationsSpecial Issue on Data Mining in Pervasive Environments.

URL: *http://www.sciencedirect.com/science/article/pii/S1574119213001090*

Zampella, F., Angelis, A. D., Skog, I., Zachariah, D. & Jimenez, A. (2012), A constraint approach for uwb and pdr fusion **LR-68**, *in* 'Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on', pp. 1–9.

Zampella, F., Khider, M., Robertson, P. & Jimenez, A. (2012), *Unscented Kalman filter and Magnetic Angular Rate Update (MARU) for an improved Pedestrian Dead-Reckoning* ***LR-63***, IEEE/ION Position Location and Navigation Symposium (PLANS).

Zaugg, D. A., Samuel, A. A., Waagen, D. E. & Schmitt, H. A. (2003), A combined particle/kalman filter for improved tracking of beam aspect targets, *in* 'IEEE Workshop on Statistical Signal Processing, 2003', pp. 535–538.

Zhang, R., Höflinger, F. & Reindl, L. (2013), 'Tdoa-based localization using interacting multiple model estimator and ultrasonic transmitter/receiver **LR-86**', *IEEE Transactions on Instrumentation and Measurement* **62**(8), 2205–2214.

Zhuang, Y., Yang, J., Li, Y., Qi, L. & El-Sheimy, N. (2016), 'Smartphone-based indoor localization with bluetooth low energy beacons', *Sensors* **16**(5), 596. Copyright - Copyright MDPI AG 2016; Last updated - 2016-05-14.

URL: *https://search.proquest.com/docview/1788754497?accountid=27303*

Zichar, M. (2013), Geovisualization-related issues with cognitive aspects, *in* 'Cogni-

tive Infocommunications (CogInfoCom), 2013 IEEE 4th International Conference on', pp. 503–508.

Appendix A

Levelling using Quaternions

The levelling was adapted from the works by Sola (2017), Trawny & Roumeliotis (2005), Madgwick (2013). The following pseudocode describes the process of levelling during Zero Velocity Updates while the person was walking.

Incremental correction during each epoch, while the ZUPT is on.

1. The down component is taken in the local tangent plane coordinate frame of the normalised quaternion that represents foot attitude. This is also the direction estimate of g.

$$v = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

2. Cross product is taken between this vector and the measured normalised acceleration vector in the stagnant IMU body frame. The normalised measurement indicates the correct down direction for g. The cross product is orthogonal

to both the estimated and the measured directions of g. Thus, it can also function as the rotation axis. The magnitude of the resulting vector is proportional to the angle between the two vectors. This vector represents the difference between the measured and the estimated attitude of the stagnant foot on the ground.

$error = v \times (\frac{f_b}{|f_b|})$

3. We calculate the angle between the two vectors.

   $angle = atan2(|error|, v \bullet \frac{f_b}{|f_b|})$

4. We define a constant epochwise proportional factor, K_p that indicates the amount of corrective turn for each epoch. The larger the K_p, the slower the error is corrected. This means the estimated attitude turns towards the correct measured attitude. If this is set too small, the correction might overshoot and even turn unstable. The turn is derived as a quaternion.

   $scalar = cos(angle/(2K_p))$

   $axis = \frac{error}{|error|}. * sin(angle/2K_p)$

   $correction_quaternion = [scalar\ axis]$;

5. This adjusting quaternion can now be converted into euler angles and inserted as an attitude error update, together with the gyro bias estimate, to the Kalman filter.

We didn't apply smoothing stepwise corrections as, for example, Tunca et al. (2017). They used a Rauch-Tung-Striebel smoother. Moreover, Madgwick (2013)

used stepwise smoothing of the previous step. He recorded the velocity error at the first zero velocity update after step had been taken and compensated this value linearly by reprocessing the previous step epochs. A similar approach could also be used for the position drift. These are left as possible future additions.