

The BUZZ:

**Narrowband Ultrasonic Positioning
for Wearable Computers**

Michael Ronan McCarthy



A thesis submitted to the University of Bristol in accordance with the requirements for
the degree of Doctor of Philosophy in the Faculty of Engineering, Department of
Computer Science.

January 2007

c. 51 000 words

Abstract

Indoor environments—characterised by obstacles such as walls, floors, ceilings, and furniture—provide a number of challenges for location sensing. In particular, these environments prohibit the use of common outdoor systems such as GPS. Many inexpensive indoor positioning systems use the properties of ultrasound to provide spatial information to their algorithms. Traditionally, this information is in the form of range measurements, obtained using a combination of ultrasonic signals and electromagnetic signals. The main disadvantage to this dual-medium approach is that it requires more hardware than an approach that only uses one medium.

In this thesis, we demonstrate that it is possible to position mobile computers in an indoor environment using only narrowband ultrasound. We describe the design, implementation and evaluation of two novel systems: the Synchronous BUZZ and the Asynchronous BUZZ positioning systems. In both systems, the only form of measurement is from narrowband ultrasound signals originating from beacons in the environment. Positioning is made possible through the use of transmission patterns, which communicate timing information from the infrastructure to receiving wearable devices. Compared with traditional systems, our approach saves on component costs and power consumption, while improving on form factor.

The evaluation of the systems is done through the use of a custom built simulator and two real-world experiments. The simulator uses a constrained random-walk to generate realistic paths of a wearable user. Sensor noise, reflections, occlusions, as well as beacon locations can all be controlled within the simulator’s environment. The proposed real-world experiments compare the paths generated by the positioning algorithms to two different ground-truths: a known static path followed by the wearable and one captured by a high-speed camera. Based on the results of the evaluations, we provide recommendations for the uses of our systems.

Declaration

I declare that the work in this dissertation was carried out in accordance with the Regulations of the University of Bristol. The work is original, except where indicated by special reference in the text. No part of the dissertation has been submitted for any other degree in the United Kingdom or overseas. Any views expressed in the dissertation are those of the author.

Signed:

Date:

Michael Ronan McCarthy

Acknowledgements

My first thank you goes to Henk Muller. Over the last five or six years he has been a team-mate, a supervisor, a colleague and most of all a friend. Who knows where I would be if we had not met on the volleyball court. I am grateful for his time, patience, ideas and insight. While writing this, I considered making a reference to his height (six-five) and the quote, “standing on the shoulders of giants,” but could not think of any way of phrasing it without visualising the ridicule I would receive. In any case, I hope that *thank you* is enough to express my gratitude and respect. Thank you.

I am happy that I could share in the PhD struggle with some great people. From the wearables lab, Paul, Cliff and Ian have been great sources of inspiration and companionship. An extra thanks to Paul for our philosophical chats about the long road to a PhD. And to Cliff for all of his help with hardware, experiments, installations, and dodgy flights to Japan. From the rest of the department: Amoss, Mark, Denis, Rob, Kate, Veronica, Sion, Oli, Rich. Thanks for your help and friendship.

The Computer Science Department at the University of Bristol has been a great place to work. The people within the department have made the last five years genuinely enjoyable. I have appreciated the drinks at the pub, chats over lunch and donuts, and the stimulating, relaxed and welcoming atmosphere. Many thanks to Angus the-camera-man, and Martijn the chef. Thanks to Dave G. for taking time out of his busy schedule to help with my experiments.

I am grateful to have been funded by such a grand initiative, the Equator Interdisciplinary Research Collaboration. It has introduced me to new fields of research and provided opportunities to meet a myriad of fascinating individuals.

One thousand thank-yous to my friends and family back home and around the world. I would not be who I am without the influence of the many actors in the play that is my life. There is not enough room in fifty *Acknowledgements* sections to express my appreciation to you all.

Lastly, to Fiona, my wife to be. Words cannot express my gratitude for your patience, support, encouragement and love. I could not have done any of this work without you in my life. I am the luckiest guy in the world.

This work is dedicated to Fiona, the love of my life.

Contents

List of Figures	xv
List of Tables	xvii
List of Abbreviations	xix
1 Introduction	1
1.1 Contributions	4
1.2 Applications	5
1.2.1 The Synchronous BUZZ	6
1.2.2 The Asynchronous BUZZ	7
1.3 Roadmap	10
2 Location Sensing	11
2.1 Location Concepts	11
2.1.1 Infrastructure and Mobile Devices	11
2.1.2 Positioning versus Tracking	13
2.1.3 Location as a Position	14
2.1.4 Relative versus Absolute	15
2.1.5 Performance	16
2.2 Position Recovery	17
2.2.1 Angular Model	17
2.2.2 Ranging Model	19
2.2.3 Relational Velocity Model	23
2.3 Observing Spatial Phenomena	26
2.3.1 Electromagnetic Phenomena	27
2.3.2 Acoustic Phenomena	32
2.4 Localisation Algorithms	35
2.4.1 Analytical Solutions	35
2.4.2 Simple Numerical Solutions	37
2.4.3 Error Minimisation	38
2.4.4 State Space Methods and the Kalman Filter	40
2.5 Conclusions	46

3 The BUZZ	47
3.1 People Positioning	47
3.1.1 Process Covariance for the Position Model	50
3.1.2 Process Covariance for the Position-Velocity Model	50
3.2 Ultrasound	52
3.2.1 Sensing Narrowband Ultrasound	53
3.2.2 Handling Anomalous Measurements	56
3.2.3 Beacon Density	57
3.3 Self Positioning Mobile Devices	59
3.3.1 Mobile Device as a Receiver	59
3.3.2 Light-weight Positioning Algorithms	62
3.4 Evaluation Metrics	66
3.4.1 Evaluation Concepts and Techniques	66
3.4.2 Simulations	68
3.4.3 Real World Evaluation	71
3.5 Conclusions	77
4 The Synchronous BUZZ	79
4.1 Connected Infrastructure	80
4.2 Positioning Algorithm	82
4.2.1 Kalman filter Parameters	82
4.2.2 Getting a Fix: Initialisation and Stability Monitoring	84
4.3 Evaluation through Simulation	87
4.3.1 Sensor Noise	88
4.3.2 Occlusions and Number of Beacons	89
4.3.3 Beacon Placement	91
4.4 Real World Evaluation	92
4.4.1 Stationary Performance	92
4.4.2 Path Comparison Analysis	95
4.4.3 Video Ground-truth	99
4.5 Discussion	101
4.5.1 Real-world Errors	101
4.5.2 Comparative Performance of the Algorithms	102
4.5.3 Errors in Calibration and Ground-truths	103
4.5.4 Resource Usage and Deployment	104
4.6 Conclusions	105
5 The Asynchronous BUZZ	107
5.1 Independent Beacons	108
5.1.1 Positioning Methods	108
5.1.2 Beacon Transmission Considerations	109
5.1.3 Beacon Hardware	115
5.2 Positioning Algorithm	117
5.2.1 Initialisation	117

5.2.2	Doppler Filter	119
5.2.3	Pseudorange Filter	121
5.2.4	Chirp-Beacon Association	123
5.3	Evaluation through Simulation	125
5.3.1	Sensor Noise	126
5.3.2	Monte Carlo Exploration	129
5.3.3	Occlusions	133
5.3.4	Reflections	135
5.4	Real World Evaluation	136
5.4.1	Initialisation Data	138
5.4.2	Three Trials	138
5.4.3	Position Seed	142
5.5	Movement Limitations	144
5.6	Discussion and Conclusions	146
6	Conclusion	149
6.1	Contributions	150
6.2	Limitations	151
6.3	Further Research	153
Bibliography		157

List of Figures

1.1	Corner of the Great Oak Room in the Red Lodge	9
1.2	Close-ups of three beacons from the Asynchronous BUZZ.	9
2.1	The Angular Model of position recovery	17
2.2	Position recovery using the Ranging Model	20
2.3	Visualisation of the Velocity Model for position and velocity recovery .	23
3.1	A generic control system	49
3.2	The effect of distance-related attenuation on chirp arrival time	54
3.3	Beacon transmission cone	58
3.4	The definition of a chirp as observed by the mobile device	60
3.5	The old mobile device: ADS Bitsy based wearable system	61
3.6	The latest mobile device: Gumstix based wearable system	62
3.7	Simulator position output for the XY axes	70
3.8	Circular path mobile device during dynamic evaluation experiments .	73
3.9	Error from the circular path in three dimensions	75
3.10	Frame of video from camera evaluation	76
3.11	Measurement buffering	77
4.1	Transmission and reception times of a six beacon synchronous system .	81
4.2	State machine framework	85
4.3	Simulation 3D error CDFs with ideal data	87
4.4	Sensor noise versus SEP for P- and PV-filters	88
4.5	Occlusions versus SEP 50% and number of beacons	90
4.6	Performance for different beacon configurations	91
4.7	3D error CDF (SEP) for 18 10-minute stationary trials	93
4.8	Stationary performance at two points	94
4.9	Spinning chair performance near the centre of the room	96
4.10	Dynamic positioning results for an outermost location	98
4.11	Total dynamic performance	99
4.12	Results from the camera experiment	100
4.13	Error CDF for the camera experiment	101
5.1	Channel occupancy	112
5.2	Occupancy versus beacon period	114

5.3	Asynchronous beacons	116
5.4	Initialisation Sequence	118
5.5	Dynamic simulation results with ideal data.	125
5.6	The latency in the Doppler Filter	126
5.7	Sensor noise versus SEP	127
5.8	X -axis position at $\sigma = 2.4$ cm	128
5.9	Monte Carlo simulation results using a cubic beacon configuration	131
5.10	Monte Carlo simulation results using the lab beacon configuration	132
5.11	δ measured along three lines in 6D space	133
5.12	Occlusion probability versus SEP	134
5.13	Chirp-beacon association statistics versus occlusion probability	135
5.14	Reflection probability versus SEP	136
5.15	Chirp-beacon association statistics versus reflection probability	137
5.16	Path estimated by the Asynchronous BUZZ over three real-world trials .	140
5.17	Receiver moving towards beacon	141
5.18	Vertical error CDFs	142
5.19	Results from position seed experiment	143
5.20	3D position error versus time	145

List of Tables

3.1	Recommended beacon densities	59
4.1	State variables	83
4.2	Position model equations	84
4.3	Position-velocity model equations	84
4.4	Simulation results for ideal data (values in centimetres)	87
4.5	Performance figures for the Synchronous BUZZ	98
5.1	Doppler method extended Kalman filter equations	121
5.2	Pseudorange method equations	121
5.3	Dynamic simulation results with ideal data	125
5.4	Output of the pfind algorithm	138

List of Abbreviations

ADS	Applied Data Systems
BUZZ	Collective name of the positioning systems described in this thesis (derived from the words “Bristol” and “Ultrasound”)
CDF	Cumulative Distribution Function
CEP	Circular Error Probable
COSPAS	Cosmicheskaya Sistema Poiska Avariynich Sudov (Russian for Space System for the Search of Vessels in Distress, see SARSAT)
CPU	Central Processing Unit
DME	Distance Measuring Equipment
DOP	Dilution of Precision
DRMS	Distance Root Mean Square
EPSRC	Engineering and Physical Sciences Research Council
GPS	Global Positioning System
IIR	Infinite Impulse Response
IRC	Interdisciplinary Research Collaboration
ISWC	International Symposium of Wearable Computers
LCM	Lowest Common Multiple
LEP	Linear Error Probable
MRSE	Mean Radial Spherical Error
NOP	No operation instruction
PCB	Printed Circuit Board
PIC	Microchip’s PICmicro microcontroller

PIT	Point in Triangle
P-model	Position Model
PV-model	Position-Velocity Model
RAM	Random Access Memory
RC	Resistor Capacitor
RF	Radio Frequency
RMS	Root Mean Square
SARSAT	Search and Rescue Satellite Aided Tracking
SCAAT	Single Constraint at a Time
SEP	Spherical Error Probable
SLAM	Simultaneous Localisation and Mapping
SPL	Sound Pressure Level
UbiComp	International Conference on Ubiquitous Computing
VHF	Very High Frequency
VOR	VHF Omnidirectional Ranging

Chapter 1

Introduction

A *context aware* computer is one that is able to automatically determine information about its operating environment. This ability is particularly desirable when the environment is dynamic or continually changing. In the realm of ubiquitous computing, where computers are embedded within the surroundings, a computer may be programmed to adapt to a flux of different users, tailoring its services based on their tastes and preferences. In contrast, a mobile or wearable computer may be programmed to provide services based on the different circumstances into which it is carried or worn. For either computing paradigm, contextual information can be used to provide enhanced applications and services.

There are many different types of sensors available to designers of context aware applications. These sensors are able to measure various forms of context which can be used to make assumptions about the environment in which the sensor is situated. To give an example, there have been a number of research projects that combine temperature, light, motion and sound sensors to create context aware wearable computers. Two specific projects are *SenSay* [115] and *eWatch* [117] both developed at Carnegie Mellon University. Applications running on these platforms form hypotheses about the operating surroundings and adapt accordingly; for instance, by adjusting the ring volume on an attached mobile phone.

One of the most useful forms of context is location. For a computer attempting to understand its surroundings, the location of users and/or objects is very important. Consider an example of a diary application on a handheld computer. Let us assume that its user wishes to be reminded of particular events in the diary for the sake of punctuality. For this task, the sensing of location can allow the diary to adjust its reminders based on the time it takes to travel between the user’s current location and the location of the event—a service that is far more valuable than, say, a five minute warning. If the user in our example is currently located in Bristol, and it takes two and a half hours to get to an interview scheduled in London for 15:00, the diary application can prompt her to begin her journey at 12:30, thus ensuring that she arrives on time.

Indeed, there are many examples in which location sensing has already been employed in a number of domains and industries. Search and rescue operations use equipment to locate lost sailors and downed aeroplanes [73, 74, 67, 95]. There are also systems that allow rescuers to find people trapped in avalanches [87, 59, 123]. The earliest location systems were invented for navigation, such as the ground beacon based air navigation system called VHF Omnidirectional Ranging (VOR) [63, 96]. Since the 1960’s, pilots have used on-board VOR equipment to navigate to their destinations. The first satellite based system was the US Navy’s TRANSIT system [98, 46]. Operational in 1968, it provided coarse and intermittent two-dimensional positioning for equipment on the ground. TRANSIT’s successor, the Global Positioning System (GPS) [98, 46], improved on TRANSIT by providing more accurate three-dimensional position estimates at a higher frequency. It has been in use since the early 1990’s in a myriad of military and civilian applications.

More recently, location systems have been employed by mobile, wearable or ubiquitous systems to support applications in areas such as gaming, tourism, and art. Examples of game applications include: *HeadRacer*, a racing game that uses head movements to steer a penguin down snowy slopes [106]; *Feeding Yoshi*, a long-term, wide area game based on the characteristics of WiFi networks [12]; *Can You See Me Now?*, a game that combines the experience of on-line and real-world GPS users [41]; *Real Tournament*, an augmented reality game using GPS and electronic compasses [88]; and the *Nintendo Wii*, a game console that uses the movement from players as its main form of input [84].

Some applications in the area of tourism are: *GUIDE*, a context-aware guide of the city of Lancaster [21]; a mixed-reality experience of the Mackintosh Interpretation Centre in

The Lighthouse in Glasgow [14]; and an application that allows city-goers to share their experiences with other visitors [13].

Examples of artistic applications include: *A Walk in the Wired Woods*, a location based audio experience involving images of natural woodland [65]; *Riot!*, an interactive audio play reenacting the 1831 Bristol riot [110]; *Come Closer*, an interactive audio and visual experience using the relative location of users [121]; and *Hosts*, a project erected in the Bath Abbey providing location based experiences for tag-wearing users [85].

To cope with the demands of the above types of applications, researchers have been adapting the techniques employed by earlier location systems. In general, this involves creating systems that are smaller, cheaper, and easier to use.

The process of sensing location is facilitated through spatial observations of the environment. These observations are provided through the use of different measurable phenomena such as electromagnetic or acoustic waves. GPS, for instance, uses radio-frequency (RF) signals, transmitted by satellites orbiting the earth, to provide observations of distance. This is possible since the transmission time of each signal is encoded within the signal itself. Devices receiving the signals use this information to derive distance measurements and thereby calculate their terrestrial latitude, longitude and elevation.

As a tool for application designers, GPS is widely available and its coverage is nearly global. However, its use is limited to spaces where its RF signals are unobstructed. As such, many spaces—including indoor environments—are not covered by GPS. Furthermore, the use of GPS is limited by its accuracy (resolution of metres) and estimation rates (approximately 1 Hz) since many applications require greater performance. In general, there is no one-size-fits-all location system. Different environments and different applications have different requirements that, currently, cannot be met by one system. As a result, researchers have created a number of different solutions tailored for specific uses, budgets and environments.

Many systems designed for indoor environments use narrowband ultrasound to observe the spatial properties of the targeted area. These properties are traditionally in the form of distance measurements, collected using a combination of RF and ultrasonic signals. The main principle behind these systems lies in the estimation of the time-of-flight of ultrasound. The RF signal is used as a synchronisation mechanism, supplying transmis-

sion times in a similar manner to the encoded information of a GPS signal. Put simply, distance is calculated by multiplying the speed of sound by the difference between two time measurements: the reception time and the transmission time of the ultrasonic signal. In these systems, the latter is provided by the RF signal.

In this thesis we present two indoor positioning systems—the Synchronous BUZZ and the Asynchronous BUZZ, collectively known as the BUZZ positioning systems—that do not require a separate synchronisation signal. They are able to *calculate the position of wearable computers using only narrowband ultrasound*. The research presented herein, attempts to explore the extent to which we can push narrowband ultrasound as a medium for location sensing. We show that, using stochastic modelling techniques, we are able to create a high performance system—the Synchronous BUZZ—that can be used for applications such as gaming, as well as a low-power system—the Asynchronous BUZZ—that can be retrofitted to aesthetically sensitive environments.

Our work is funded by the Engineering and Physical Sciences Research Council (EPSRC) through the *Equator* Interdisciplinary Research Collaboration (IRC) [38]. The design of our positioning systems is motivated by the cooperative nature of the Equator project. Specifically, we have aimed to design our positioning systems so that they can be used today, within the IRC’s multidisciplinary experience projects. As such, we have attempted to achieve a balance between novelty and practicality: we want our systems to push the envelope of positioning research while at the same time create systems that are accessible to our Equator colleagues.

The remaining sections in this chapter outline the contributions of our work, highlight a number of applications for our systems, and provide a roadmap for the structure of the thesis.

1.1 Contributions

Our research contributions are summarised as follows:

- In this thesis we demonstrate that it is possible to perform indoor positioning on wearable computers using only narrowband ultrasound. Our novel approach uses

transmission patterns to communicate timing information from the infrastructural systems to the receiving mobile devices.

- We have designed and implemented two low-cost positioning systems that each has a specific application target. The systems are functional and can be employed presently.
- We have developed a simulation environment that allows us to test the performance of our systems under different environmental conditions. The simulator has control over parameters such as the path of the mobile device, the number and placement of transmitting beacons, sensor noise, reflections and occlusions.
- We propose a novel framework for evaluating our systems. Our approach combines a number of different methods that explore the various behavioural properties of our algorithms and devices. Based on the results from our experiments, we provide recommendations for the use of the systems with respect to different applications and environments.

1.2 Applications

Location based technologies have already been around for a number of years. Recently, however, advances in the microelectronic and wireless communications industries have caused an increase in the use of mobile computers. The advances have made it possible to create “human centred” location technologies and have opened up a range of applications using them. To-date there have been a number of location systems developed to address the demands of applications running on personal mobile devices.

The BUZZ positioning systems have been designed to support indoor, wearable based applications that require medium-to-high performance at low cost. Each system provides a room-based positioning service that has been configured for capturing human motion. Taken individually, the systems have different properties that make them suitable for different applications. In the next sections we highlight the properties of each system and take a look at some possible applications of their use.

1.2.1 The Synchronous BUZZ

The design of the Synchronous BUZZ focuses on optimising performance. While it does not compare with the performance of more complex systems such as the *HiBall Tracker* [131] or *Constellation* [44], it is capable of providing position estimates within 5 cm accuracy at rates greater than 30 Hz. At this level of performance, not only is it possible to position a person as a whole, but it is also possible to independently position the head or limbs. And while full recovery of human movement requires a measurement frequency of greater than 40 Hz [129], we have found that our system is fast enough for many “high performance” applications.

To-date we have explored the use of the system with a number of gaming applications and art installations. We describe a selection of them here as examples of typical uses for the Synchronous BUZZ:

Bobsleigh: As part of an undergraduate gaming project, the Synchronous BUZZ was employed to provide head positioning for a multi-player bobsleigh game. Our self-positioning mobile devices were embedded into bicycle helmets worn by users facing a large projection screen. Each of the helmets used Bluetooth to transmit their position to a server running and displaying the game. The four players in the game had to work together—by moving their heads in unison—to guide their bobsleigh down the runs.

HeadRacer: To demonstrate our system at two conferences held in Japan in 2005 (the International Conference on Ubiquitous Computing (UbiComp) and the International Symposium of Wearable Computers (ISWC)), we modified Tux Racer, the single user, open-source game supplied with most Linux distributions, to respond to input from one of the bobsleigh helmets [106]. Similar to the bobsleigh game, the player uses head movements to guide Tux (the Linux penguin) down various snowy slopes, while sitting in front of a projection screen.

Jedi Game: Another undergraduate games project used the Synchronous BUZZ to track the position of a lightsaber. Two mobile devices were mounted on the shaft of a replica Star Wars lightsaber to capture its position and orientation. To play the game, the player holding the lightsaber stands in front of a projection screen and defends against various attacks from Stormtroopers and droids.

Walk in the Wireless Woods: A *Walk in the Wireless Woods* is an art installation that uses the position and orientation of a person’s head to spatialise streams of audio [89]. The user wearing the system wanders through a space that is populated with images of British woodland. A positioning device and a compass, mounted on the user’s headphones, allow the wearable computer to calculate the position of the user’s ears. Based on this, the audio streams are spatialised to sound like they are emanating from the images.

In general, the Synchronous BUZZ is a robust and accurate positioning system. Our main design aim is to get the most out of narrowband ultrasonic signals by tightly multiplexing them in the time domain. For this reason, single room applications, such as the ones we have mentioned, currently provide the most suitable environment for this system.

The synchronous, time multiplexed nature of the transmitted signals, means that there are constraints on the deployment of the system’s infrastructure. Specifically, the network of ultrasonic beacons, placed on the walls and ceiling of a room, are connected to a central control unit by a number of wires. The presence of these wires mean that the synchronous system is unsuitable for a number of environments. It was this drawback that motivated us to design the Asynchronous BUZZ, which we introduce in the next section.

1.2.2 The Asynchronous BUZZ

The Asynchronous BUZZ is based on a similar design as the Synchronous BUZZ. Specifically, it employs mobile devices that position themselves using signals transmitted by the infrastructure. The difference, however, is that the Asynchronous BUZZ does not control the transmission of the signals centrally. Each of the ultrasonic beacons, distributed throughout the environment, transmits ultrasound at periods determined by its own internal clock; oblivious to the behaviour of the other beacons. This design choice results in an *asynchronous* transmission pattern that each mobile device deciphers to calculate its position.

Aesthetically sensitive environments, such as museums and living rooms, provide motivation for the design. Without wires, the small beacons can be more easily retrofitted

to existing structures while reducing the visual impact on the environment in which they are deployed.

One of the applications we have targeted for our system is a wearable or handheld tour-guide for use within an Elizabethan house and museum in Bristol, UK. The house, the *Red Lodge*, is situated near Bristol's historic centre and is managed and maintained by the Bristol City Council. The two storey house was built in the late 16th century and has several rooms that are still in their original state with original wooden fittings and furniture. The rooms have no signs or markers in them, and the only guide currently available is a narrative printed on a laminate card. Visitors read the information on the card to learn about the history and significance of the different items and features in each of the rooms. We have noticed, however, that the descriptions can be confusing as it is not always easy to identify the location of unfamiliar features. Furthermore, if the curator wishes to change the location of one of the chairs or tables, for example, she must update the text on all of the cards.

We believe that an application based around our system can improve the service offered to visitors, as well as provide more flexibility for curators. We have identified one of the rooms, the *Great Oak Room*, as a candidate for a pilot experiment. It is a prime example of an environment where a wired system is prohibited and where aesthetics are of utmost importance. The Great Oak Room consists of oak-panelled walls, ledges, and a carved stone chimney, all of which are excellent locations to place our beacons out-of-reach and out-of-view. The beacons are small enough to be balanced on ledges as small as 7 mm wide. On wide ledges, the ultrasonic transducer can be mounted on the beacon's PCB so that the visible area is only 20×10 mm. On narrower ledges, where we are forced to mount the transducers perpendicular to the PCB, the visible area of the beacon is still only 26×16 mm. In both cases, all but the mesh face of the transducer (4 mm radius) can be camouflaged by painting the beacon with an appropriate colour.

An example beacon placement in the Great Oak Room can be seen in Figures 1.1 and 1.2. Figure 1.1 shows a snapshot of the corner of the fireplace and the door to the hallway. Three beacons are present in this scene, which are shown in close-up in Figure 1.2. In initial discussions with the curator, she observed that the reflective metal sides of the ultrasonic transmitter made the beacons visible and suggested painting them with a non-reflective paint. As can be seen in Figure 1.1, the bright metal makes the beacons stand



Figure 1.1: Corner of the Great Oak Room in the Red Lodge



Figure 1.2: Close-ups of three beacons from the Asynchronous BUZZ.

out in two locations as white dots. Such an effect is especially noticeable when flash photography is used.

1.3 Roadmap

This thesis is organised into six chapters (including this one). The remaining five chapters are structured as follows:

Chapter 2: Location Sensing. In this chapter we provide an introduction to location sensing. General location sensing concepts are explored and our terminology is defined. We provide explanation of the different methods for recovering position as well as the physical phenomena on which they are based. The chapter is concluded with an exploration of positioning algorithms used to fuse the recovery methods with measured phenomena.

Chapter 3: The BUZZ. Chapter 3 is intended to provide details on all of the properties common to the BUZZ positioning systems. The chapter starts with an exploration of the issues surrounding the task of positioning people. We then go into the general properties of ultrasound and considerations of position sensing with narrowband ultrasound. The hardware common to both systems is presented before we conclude with an introduction of our evaluation metrics.

Chapter 4: The Synchronous BUZZ. This chapter describes the design and implementation of the Synchronous BUZZ. We provide the operation details of two different positioning algorithms and contrast and compare their performance in a number of different simulations and real world experiments. We end the chapter with a discussion of the properties of the system, a reflection on our evaluation methods, and recommendations for the use of the different algorithms.

Chapter 5: The Asynchronous BUZZ. The Asynchronous BUZZ is presented in this chapter. We present the issues surrounding the task of positioning with asynchronous beacons and describe the hardware comprising them. The implementation details for two different algorithms based on two different position recovery methods (provided in Chapter 2) are described. These two methods are then evaluated using simulations and real world experiments. A discussion wraps up the chapter by comparing the results of the two algorithms.

Chapter 6: Conclusion. This chapter concludes the thesis by highlighting our research contributions, discussing the limitations of the BUZZ systems and outlining directions for future research.

Chapter 2

Location Sensing

Designing a positioning system requires the designer to understand location sensing concepts as well as the implications of certain design decisions. In this chapter we provide background to the main considerations of location system design. Our discussion starts with an explanation of the fundamental concepts, then moves on to explore different models of position recovery. Phenomena used for observing spatial relationships and algorithms for localisation are investigated in the remaining sections.

2.1 Location Concepts

In this section we outline some of the overarching concepts of localisation. The terms and conventions described here are important as they form the basis for our discussion in later sections and chapters.

2.1.1 Infrastructure and Mobile Devices

The equipment comprising a location sensing system can be separated into two groups. The first group, the *infrastructure*, consists of the components in the environment that are used to support the process of determining location. In the case of GPS, the infrastructure

includes all of the satellites orbiting the earth as well as the ground control stations that monitor and update the satellites. Similarly, the BUZZ uses an infrastructure of ultrasound transmitting beacons placed on the walls and ceiling of a room.

The second group of components comprising a location system is the *mobile device*. Mobile devices are attached to the objects being located and act as a reference for spatial measurements. In the case of GPS, GPS receivers form the “mobile device” part of the GPS system, and are used to measure the time-of-flight of radio signals broadcast by the satellites. For the BUZZ systems, a mobile device is composed of ultrasound receiving equipment and a processing unit. These units also measure time-of-flight, but the medium used is ultrasound rather than radio. In general, mobile devices are small so that they can be easily carried or worn by a user.

We note that the use of the two types of devices varies among location systems. In the examples that we have given so far, i.e. GPS and the BUZZ, infrastructural devices and mobile devices are used together to form the nodes in a unidirectional communication system. Some location systems, such as those that perform video tracking, use only an infrastructural component. This component consists of one or more stationary cameras placed in the environment, and is used to track the location of objects in the observed space. Examples of such systems include: W^4 , a system designed for monitoring people and their activities in outdoor environments [52], and a multi-camera, indoor system created as part of Microsoft Research’s EasyLiving project [77]. An advantage to these systems is that location sensing is unobtrusive. People are not required to carry anything and can be tracked as soon as they enter the scene. However, a main disadvantage for these kinds of tracking systems is the perceived intrusion of privacy.

In contrast, other location systems use only a mobile device component. For example, there are a number of systems that track the position of a mobile camera using only the stream of images in the video feed [22, 76, 94]. These systems fall under a domain of research called *Structure from Motion*, where the three dimensional structure of a scene is modelled from the input captured by a moving camera. Although both the location of the camera and the structure and location of objects in the scene are part of the estimated solution, location systems based on the concept focus more on the camera tracking aspects. In the robotics and mobile computing communities, the algorithms underpinning these location systems have been given the label *Simultaneous Localisation*

and Mapping (SLAM). A few researchers, such as Davison [28, 29] and Pupilli [102] have generalised SLAM to work with mobile and wearable applications.

2.1.2 Positioning versus Tracking

The difference between the ubiquitous and wearable models of computing lies in the computational architecture. For ubiquitous computing, processing is performed by the surroundings, where computers are meant to blend seamlessly into the background. Mobile or wearable computers, on the other hand, are designed to act as computerised extensions of the user and are present on the user's body.

A similar set of concepts exists in location sensing. *Tracking* is a form of localisation that uses centralised processing, similar to some forms of ubiquitous computing (not all ubiquitous systems are centralised; for example, some are designed as disaggregated nodes within a sensor network). Tracking systems, such as the Active Bat conceived at AT&T Cambridge, have a central processor that monitors or “tracks” the locations of objects or users within the covered area [128]. In contrast, *positioning*, also known as *localised location computation*, refers to a method where the task of calculation is bestowed upon the devices being located. A GPS receiver, for example, uses RF signals transmitted from satellites to position *itself*. In this system, the central infrastructure has no knowledge of the receivers.

Certain architectural decisions are important for distinguishing the design of a tracking system from that of a positioning system. Most systems that perform tracking, for example, employ active mobile devices that transmit signals to the infrastructure. Conversely, positioning systems tend to use passive mobile devices that only receive signals originating from the infrastructure. The main difference is in the direction that the signals travel. Since it is practical to group the measurements (received signals) together with the machinery that is processing them, tracked devices will, in general, contain a transmitting element while positioning devices will contain a receiving element.

Positioning and tracking also carry implications for issues such as scalability and privacy. Positioning systems, such as the BUZZ, generally scale better in terms of the number of mobile devices since the load of location processing is distributed. This implies that mobile devices within a positioning system need to be powerful enough to perform the

positioning calculations. With this in consideration, a tracking system may allow for smaller, lower power and cheaper mobile devices than a positioning system. However, the scalability of a tracking system is also affected by the complexity of managing communication in order to prevent collisions and bottlenecks. A positioning system does not need to consider this since its communication is unidirectional with signals travelling from the infrastructure to the listening devices.

Positioning systems are also inherently private—the location of a mobile device in a positioning system is known only to that device. In contrast, a tracking system holds the location of all mobile devices in a central system.

2.1.3 Location as a Position

The terms positioning and tracking described above refer to the *method* of determining location. This is different to the concept of location as *a description of the place where something exists*. As a description of place, location can be refined to take the form of *position*. Position is a description of location that is given in terms of geometrical coordinate systems. For example, the set of X , Y , and Z coordinates describing the location of an object in a three dimensional Cartesian space, such as (0.1 m, -2.3 m, 4.0 m), is a position. This type of location is precise and numerical; it cannot describe abstract or semantic concepts of location, such as “at the office”, “at home”, or “in the living room”.

However, position can be used to derive semantic notions of location by using additional information about the space and relationships within it. For instance, an application can assume that a person is near a building if the distance between the two, obtained from their positions, is within some threshold. In this case, position is used to facilitate the symbolic notion that “the person is near the building” where “near the building” is a semantic location. In this way, systems such as GPS, which provide estimates for position, can be used to support general location based applications.

We make the observation that it is important to consider the error and granularity of position estimates when they are used to infer semantic location. Position errors introduce ambiguity that make it difficult to map positions on to semantic locations, especially when the size of the semantic locations are small with respect to the errors [107]. For

example, a hand-held navigation application that uses a location system with an error of ± 10 m may wrongly inform a user that he is on a street adjacent to the one he is actually on. In cases like these, there is usually no way for the system to disambiguate likely locations, unless additional sensors or models are used.

The terms *symbolic location* and *physical location* have been used to further classify location as a description of place [60]. Physical location is equivalent to the concept of position while symbolic location refers exclusively to abstract human descriptions. For example, a set of GPS coordinates is a physical location while “near the building” represents a symbolic location.

This thesis is concerned with the process of determining location in the form of position or physical location. In the remaining text, references to the term *location*, unless stated otherwise, refer to this concept.

2.1.4 Relative versus Absolute

Hightower et al. classify location systems as either relative or absolute [60]. They state that *absolute* location systems are those which use the same coordinate system for every located device. GPS, which uses geographical coordinates to describe the position of all receivers (latitude, longitude and elevation) is an example of this type of system. In contrast, *relative* location systems are characterised by separate coordinate systems for different mobile devices. For these types of systems, two devices placed at the same location may provide different readings for location depending on which reference frame is used.

However, classifying a location system as absolute or relative depends on the point of view. Taking our room based BUZZ systems as an example, the coordinate space is shared by mobile devices as long as they stay within the same room and use the same infrastructure. In this sense, our systems can be considered *absolute*. However, there are circumstances where different coordinate systems can be used for devices at the same location. This situation arises in places where the coverage of two or more adjacent systems overlap, such as a door or hallway. In these areas, a device using a system in one room will provide a different set of readings than a device using a system in another. Taking this into consideration, our positioning systems can also be classified as *relative*.

On the whole, translating between relative location estimates and absolute estimates can be performed but requires knowledge of the relationship between the local coordinate system and the absolute coordinate system. Knowing the orientation and location of a room in terms of latitude, longitude and elevation, for example, would allow an application using a BUZZ system to output GPS coordinates, if desired.

2.1.5 Performance

The performance of a location system is dominated by its accuracy, precision, update rate, and latency. Each of these concepts are outlined below.

Accuracy is a measure of a system's expected error in terms of physical distance. For example, a location system can have an average error of 20 cm in either two or three dimensions. High accuracy refers to small distance values.

Precision is a measure of how often a user can expect to see location estimates within the stated accuracy. A performance value of 20 cm, 90% means that one can expect 90% of the estimates to be within 20 cm of their true locations.

The update rate of a location system specifies how often it produces estimates. In order to fully reproduce the motion of any object, the update rate must be high enough to prevent aliasing. High performance systems that track human movement, including that of the head and limbs, aim for rates greater than 40 Hz [129]. Tracking humans using a system with lower update rates (e.g. GPS which operates at around 1 Hz) can only produce snapshots of the true paths. Nonetheless, for many applications, low update rates can be tolerated.

Latency is a measure of how long it takes for a system to produce an estimate for a single location. For most interactive and real-time applications, it is desirable to make latency as low as possible. For example, it may not be useful to use a system that provides estimates several seconds after the object or person has moved from the reported location.

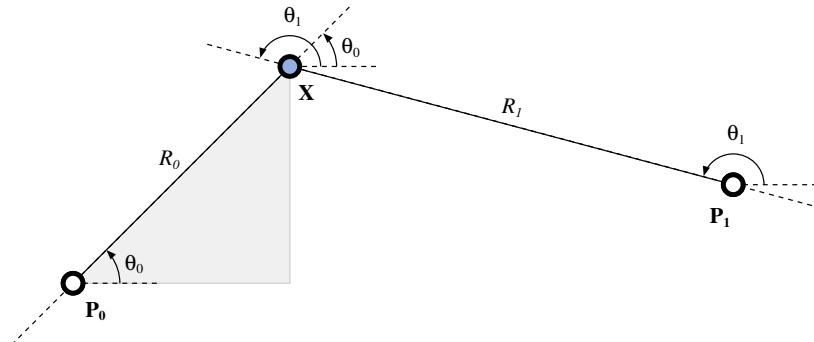


Figure 2.1: The angular relationship between the location of a mobile device, \mathbf{X} , and the fixed nodes \mathbf{P}_0 and \mathbf{P}_1 . The lines R_0 and R_1 drawn through \mathbf{P}_0 and \mathbf{P}_1 , at the angles θ_0 and θ_1 , intersect at \mathbf{X} . In a location system using this model, the known variables are \mathbf{P}_i and θ_i , while \mathbf{X} is unknown.

2.2 Position Recovery

All location systems are based on models of the physical world. Since location cannot be directly observed from our surroundings, we are forced to utilise relationships between position and the physical properties of the space. In this section, we explore some of the most common approaches to location sensing. Section 2.3 then moves on to discuss the observation and measurement of the spatial properties underpinning these approaches.

2.2.1 Angular Model

The angular method of location sensing uses measurements of angle. For this model, angles between the mobile device and a number of nodes fixed in the infrastructure are related to location using the geometric relationship therein. To make calculation possible, the location of the nodes must be known. Figure 2.1 shows the angular relationship between a mobile device at an unknown location, \mathbf{X} , and two fixed nodes at locations \mathbf{P}_0 and \mathbf{P}_1 . The locations of these three points define the angles θ_0 and θ_1 . The process of finding the mobile device's position involves taking measurements for θ_0 and θ_1 , and reversing the geometrical relationship to solve for \mathbf{X} . In terms of a visual representation of the solution, the position of the mobile device is given by the intersection of two lines drawn through each node \mathbf{P}_i at the measured angles.

The geometrical relationship can be expressed in terms of a system of equations. Examining the triangle created by the line joining \mathbf{P}_0 and \mathbf{X} (shown as R_0), a vertical line crossing through \mathbf{X} , and the horizontal line crossing through \mathbf{P}_0 , we can use simple trigonometry to derive the following equation for the position \mathbf{X} :

$$\mathbf{X} = \mathbf{P}_0 + R_0 \begin{bmatrix} \cos \theta_0 \\ \sin \theta_0 \end{bmatrix}$$

Similarly, we can perform the same analysis using \mathbf{P}_1 or any other reference node i . The general equation becomes:

$$\mathbf{X} = \mathbf{P}_i + R_i \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \end{bmatrix} \quad (2.1)$$

Using two reference nodes, this equation can be used to form a system of four equations involving four unknowns (X_x , X_y , R_0 , R_1). If the equations are independent—that is, the mobile receiver does not lie on a line joining the two reference nodes—the system is well constrained and solvable algebraically. In practice however, measurements for θ_0 and θ_1 contain errors that affect the final result. If these errors need to be accounted for, the system of equations can be solved using a form of error minimisation or state space filtering (Section 2.4). To extend the method to three dimensions, a third angle, measuring elevation, is required from at least one reference node.

There are a number of systems that use the angular method. The most common of these is a system called VOR [63, 96]. It is an aircraft navigation system that uses radio signals broadcast from land base stations. The base stations transmit two signals simultaneously. The first is broadcast in all directions while the second is transmitted in a directed beam that rotates around the station. The equipment on-board the aircraft extracts the phase difference between the two signals to calculate the angle to the base station. The ID of the station is encoded in the signal so that the station's position is known. Given two or more sets of signals, the VOR equipment uses the angular method to position the aircraft.

A number of indoor systems, inspired by VOR, have been created as well. One system uses wireless 802.11 signals [93]. Similar to traditional VOR, this system employs a rotating directional 802.11 base station aerial. As the aerial rotates, the station measures the signal strength to the mobile devices and calculates the angle at which they are located. These values are then sent back to the mobile devices, allowing them to calculate their positions.

Another angle-based system uses visible light as the signalling mechanism [92]. In this system, synchronised rotating light beacons are used to provide “listening” devices with relative angular measurements. The solution for position relies on a system of equations that assumes that each beam of light has the same angular velocity, that each beacon is identifiable through the use of an ID encoding, and that the phase differences between the rotating beams are known. The problem is more complex than the one shown in Figure 2.1 since the observed angles are taken relative to the beacons, rather than to a fixed direction, such as North for example.

Systems with tracking architectures can also employ the angular model. For example, mobile telephone network operators have explored its use in providing user location information to emergency services [137, 138]. Such systems differ from VOR style systems in that angles are measured at the base stations (infrastructure) rather than on the mobile devices. A centralised system collates the data from multiple base stations and calculates the position of phone users.

Angular models can also be combined with measurements of distance. There is a version of VOR, for example, which uses Distance Measuring Equipment (DME). The DME provides aircraft with distance estimates to transmitting base stations. The motivation for this is that, with the extra data, an aircraft can position itself using only one base station. In terms of the variables shown in Figure 2.1, the additional information makes R_0 and R_1 known quantities. VOR DME is, in a sense, a hybrid between an angular system and a ranging system.

2.2.2 Ranging Model

Ranging is a method of localisation that uses distance or range information. In the two dimensional case, the process of calculating position is a matter of resolving intersections of circles. In three dimensions, the process is equivalent to resolving the intersection of spheres. In general, the method of calculating position using distances is known as *trilateration* if three distances are used, or *multilateration* when more than three are used. Figure 2.2 shows how three distances are used to pin-point a mobile unit at location \mathbf{X} . Like the angular method, it is essential that the location of the reference nodes \mathbf{P}_i be known.

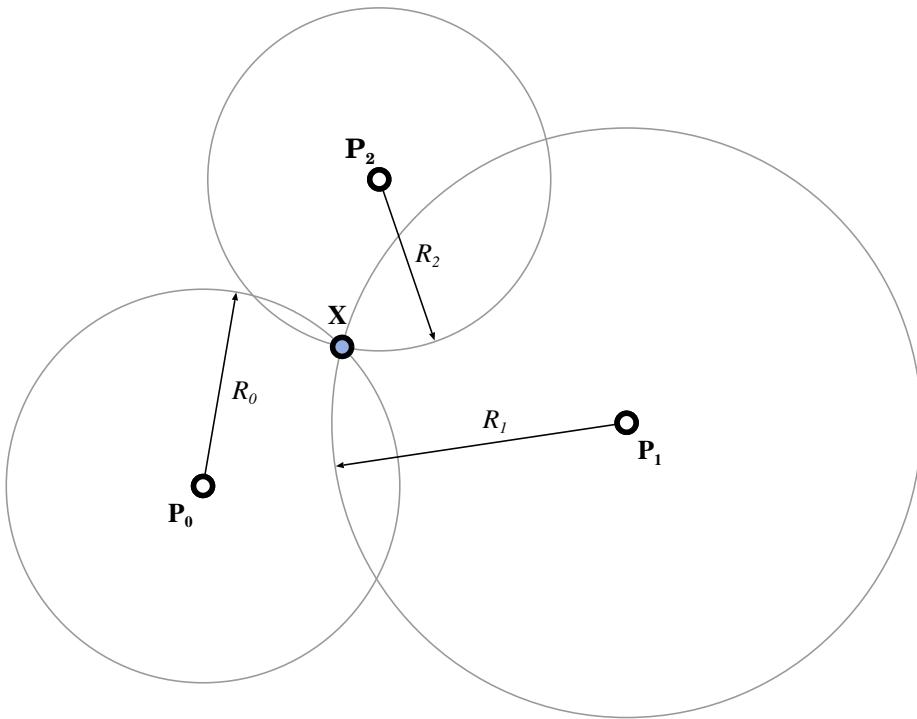


Figure 2.2: Range based location sensing is the process of resolving intersections of circles drawn around the known reference nodes \mathbf{P}_i . The range measurements, R_i , are the radius of the circles. Note that the location where they all intersect is \mathbf{X} , the unknown location of the mobile device.

In linear algebra, calculating the Euclidean norm of a vector provides a measure for its magnitude. In two or three dimensions, this magnitude is analogous to the physical length of the vector. The Euclidean norm is a generalisation of the Pythagorean theorem, which states that the length of the hypotenuse on a right-angled triangle is equal to the square root of the sum of its sides squared (i.e. the norm of $(X, Y, Z)^T$ is $\sqrt{x^2 + y^2 + z^2}$). We use the Euclidean norm to express the ranging model problem:

$$R_i = \|\mathbf{X} - \mathbf{P}_i\| \quad (2.2)$$

Here, the double bars denote the Euclidean norm. It operates on the vector pointing from \mathbf{P}_i to \mathbf{X} . By definition, this is the distance between the two points, shown in Figure 2.2 as R_i . Therefore, Equation 2.2 expresses the known distances R_i in terms of the unknown location \mathbf{X} and the known locations \mathbf{P}_i . Despite the non-linearity of the equation, closed form, algebraic solutions exist [39, 82]. In practice, however, many range base systems

employ iterative techniques to solve the equations [45, 70]. We discuss these further in Section 2.4.

Three reference nodes are required to solve for the coordinates of \mathbf{X} in two dimensions: two nodes give two possible points, and the third disambiguates them. In three dimensions, four nodes are required: the intersection of three spheres gives two possible points, a fourth disambiguates them. These are general requirements, however. In some instances—for example, when initial conditions are provided such that disambiguation is not necessary—it is possible to solve the problem using fewer nodes.

The locations of the reference nodes, themselves, is important. Their geometry determines whether the instances of Equation 2.2 provide enough constraints to solve the problem. For example, a collinear reference node geometry will not allow for the disambiguation of two solutions in two dimensions, regardless of the number of nodes. In three dimensions, a collinear geometry causes the solution to collapse into a set of overlapping circles. Without another reference node located off of the common line, it is impossible to find \mathbf{X} .

The quality of a reference node geometry can be described in terms of *Dilution of Precision* (DOP) [98, 48]. DOP is a measure of how the geometry of a system affects the propagation of measurement errors. Ideally, designers of location systems should aim to minimise DOP to prevent ranging errors from causing large position errors. In general, a system with a low DOP has a good spread of reference nodes along all axes. Based on this, collinear and coplanar reference node arrangements should be avoided.

An alternative form of ranging, known as *pseudoranging*, is used by some of the BUZZ algorithms. The technique is similar to ranging, however, an unknown constant offset in the range measurements is introduced to the equations. The most widespread use of pseudoranging is within the Global Positioning System [46, 98]. Specifically, the distances to the satellites, as calculated by a GPS receiver, are biased by a clock offset. The offset is a result of the receiver unit using a time reference different to that of the satellites. When the receiver calculates distance using the satellite encoded transmission time and the reception time recorded by its local clock, a “pseudo” distance is calculated instead. The GPS pseudoranging equation has the form:

$$\rho_i + cT = \|\mathbf{X} - \mathbf{P}_i\| \quad (2.3)$$

where ρ_i is the measured pseudorange, c is the speed of light, and T is the clock offset. In the process of solving for \mathbf{X} , the GPS receiver also solves for T . We note that T drifts over time due to the cut of the receiver clock's crystal (we return to this in Chapter 4).

Like multilateration, there are closed form, algebraic solutions to Equation 2.3 in the literature [9, 17, 80]. However, these works have shown that the presence of T means that the concept of intersecting spheres is no longer valid as a visualisation of the solution. Instead, the problem is more accurately visualised in terms of intersecting planes and revolving hyperbolae [80]. There are also implications for the geometry of the satellite reference nodes. As with the multilateration problem, collinear geometries are problematic. It has also been demonstrated that certain planar arrangements result in infinite solutions to the GPS pseudoranging problem [80]. As we have stated, planar geometries are generally undesirable because they result in a high dilution of precision perpendicular to the plane. Nonetheless, for non-planar geometries, a unique solution to the pseudoranging problem can be determined with four reference nodes—no disambiguating node is required despite the presence of an extra unknown variable.

The positioning systems designed as part of the research in this thesis use a form of pseudoranging. Unlike related work such as the Cricket Location Support System [101, 8, 118] and the Active Bat [128, 53], the BUZZ systems exploit the timing patterns of transmitted signals to avoid the use of a separate synchronisation medium. Our systems use only ultrasound, making the transmitting and receiving hardware smaller and lower power than the traditional RF / ultrasound approaches. These systems are described in detail in Chapters 4 and 5.

There is one form of distance based localisation that does not require the use of pre-located reference nodes in the environment. SLAM systems, as introduced in Section 2.1.1, are able to determine the location of a mobile unit *and* the location of reference nodes simultaneously. Early research on the SLAM problem was carried out in the robotics community using laser or ultrasonic range finders as the main observational input [119, 16, 30, 51, 36]. More recent research has focused on creating systems that attack the more difficult problem of determining 3D ego-motion without input from inertial sensors or robot control parameters [29, 102]. These systems model the location and orientation of a single lens camera using the rich information contained within the video feed. This involves identifying and tracking stable features within a stream of images while estimating the camera's angle and distance to these features. Like VOR

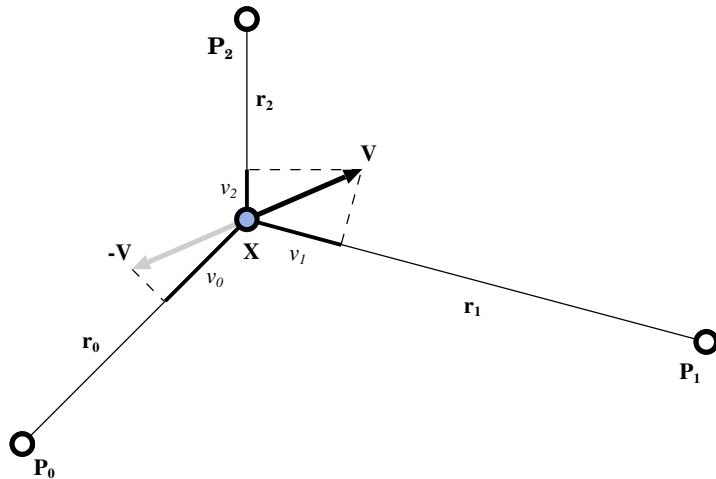


Figure 2.3: Visualisation of the Velocity Model for position and velocity recovery. The alignment of relative velocities v_i can only succeed if the mobile unit is located at \mathbf{X} . v_i are the signed scalar components of \mathbf{V} projected along the vector $\mathbf{r}_i = (\mathbf{X} - \mathbf{P}_i)$. The known values are \mathbf{P}_i and v_i . \mathbf{X} and \mathbf{V} are unknown

DME, these systems can be considered hybrid methods since angles and distances are used within the models.

2.2.3 Relational Velocity Model

Another location recovery method, the relational velocity method, uses the relative velocities of a number of reference nodes, which are observed with respect to the mobile unit. The reference nodes can be moving or stationary but their location and velocity must be known. Compared with the distance or angle based techniques, the relational velocity method is more difficult to portray visually. The presence of velocity in the model introduces another set of dimensions that make sketching a solution difficult. Despite this, we provide a depiction of the relational velocity method in Figure 2.3. It shows the position \mathbf{X} and velocity \mathbf{V} of the mobile device as well as the positions of the reference nodes \mathbf{P}_i . Although it is not necessary, we assume that the velocities of the reference nodes are zero. This makes our discussion simpler and foreshadows a method used by the Asynchronous BUZZ where the beacons in the infrastructure are stationary (see Chapter 5).

The relative velocity problem is formulated using the vector relationship:

$$\mathbf{P}_i = \mathbf{X} - \mathbf{r}_i$$

Differentiating with respect to time gives:

$$\frac{d}{dt} [\mathbf{P}_i] = \frac{d}{dt} [\mathbf{X}] - \frac{d}{dt} [\mathbf{r}_i]$$

which is equal to zero since we assume that \mathbf{P}_i is stationary. Taking the inner product with \mathbf{r}_i gives:

$$\mathbf{r}_i \cdot \frac{d}{dt} [\mathbf{X}] - \mathbf{r}_i \cdot \frac{d}{dt} [\mathbf{r}_i] = 0$$

For any vector \mathbf{a} with norm a it holds that $\mathbf{a} \cdot \frac{d}{dt} [\mathbf{a}] = a \frac{d}{dt} [a]$. Therefore the equation can be written:

$$\mathbf{r}_i \cdot \frac{d}{dt} [\mathbf{X}] - r_i \frac{d}{dt} [r_i] = 0$$

where $r_i = \|\mathbf{r}_i\|$. Assuming that r_i is never zero, we divide through by r_i :

$$\frac{\mathbf{r}_i}{r_i} \cdot \frac{d}{dt} [\mathbf{X}] - \frac{d}{dt} [r_i] = 0$$

From Figure 2.3 we can see that $\frac{d}{dt} [r_i] = v_i$, which is the velocity of the mobile device measured along the vector $\mathbf{r}_i = \mathbf{X} - \mathbf{P}_i$. Substituting $\mathbf{V} = \frac{d}{dt} [\mathbf{X}]$ for the velocity of the device and rearranging gives our main expression for the relational velocity method:

$$v_i = \frac{\mathbf{X} - \mathbf{P}_i}{\|\mathbf{X} - \mathbf{P}_i\|} \cdot \mathbf{V} \quad (2.4)$$

Using vector terminology, v_i is the projection of \mathbf{V} onto the vector $\mathbf{r}_i = (\mathbf{X} - \mathbf{P}_i)$, as depicted in Figure 2.3. To solve for the location of the device, the method uses the fact that there is a unique set of v_i for a particular location \mathbf{X} and velocity \mathbf{V} . Employing the method in practice involves taking measurements for the relational velocity scalars v_i and solving for the three-dimensional position and velocity of the mobile unit, \mathbf{X} and \mathbf{V} .

Several versions of the relational velocity method have been studied in the past [4, 18, 20, 19, 122, 81]. For the formulation of the problem expressed by Equation 2.4, no closed-form analytical solution exists. In practice, an iterative algorithm (error minimisation or state space) can be used to solve the set of non-linear equations. We assume that, since there are six unknown scalars in the equation, at least six equations (and, hence, reference nodes) are required to find the location of the mobile device.

We note that there are two conditions where the relative velocity method fails to provide enough constraints to calculate position. The first occurs when the reference nodes are arranged on a plane and the mobile device is located on, and is moving perpendicular to, this plane. In terms of Equation 2.4, the perpendicular arrangement will cause the inner product to equal zero, implying that the measured v_i 's will also be zero. In this instance, the equations no longer hold any information about the position of the mobile device since infinitely many values for \mathbf{X} and \mathbf{V} can satisfy the equation. The situation can be avoided by ensuring that the reference nodes are not located on a plane.

The other failure mode occurs when the mobile device is stationary. In this case, the relative velocities v_i are again measured as zero. If we assume that the reference nodes are not situated on a plane, and we know that the unit vectors $\frac{\mathbf{X} - \mathbf{P}_i}{\|\mathbf{X} - \mathbf{P}_i\|}$ cannot be zero, then the only conclusion we can make from the measurements is that \mathbf{V} is zero. We note that this holds for any value of \mathbf{X} , thus offering no information as to the location of the mobile device. Avoiding this failure mode is difficult in practice. We discuss it further in Chapter 5.

Another position recovery technique that uses a similar approach to the relative velocity model is known as the Point in Triangle (PIT) test [58]. It uses measurements for the relative direction of travel of a mobile unit (similar to the v_i measurements in the relative velocity model), to determine whether or not the unit is located inside a collection of triangles. The triangles are those formed by lines drawn between a subset of three reference nodes. Given more than three nodes, it is possible use the PIT tests to determine which polygon, formed by the intersecting triangles, contains the mobile unit. The strength of the technique is that it can be used when quantitative data is not available. For example, it is ideal for situations where it is possible to determine that a mobile unit has moved closer to or further from a reference node, but the amount that it has moved is either unreliable or unknown. The weakness of the approach is that it requires an exhaustive test of all the directions of travel. Unlike the relational velocity model which uses current relative velocities in a set of continuous equations, the PIT test requires that all directions of travel be tested against each reference node. This procedure is generally not feasible in practice.

There are a number of location systems that employ the relational velocity model. Although it is no longer in service, the first operational satellite navigation system, TRANSIT [46, 98], used Doppler shifts in radio signals to measure relational velocities and

subsequently solve for position. Each radio signal, transmitted by satellites in polar orbit, were encoded with the orbital parameters of the satellite transmitting it. This allowed a receiver on the earth to determine the location and velocity of the satellite reference node and, with the measured Doppler shift, calculate its own position.

The Search and Rescue Satellite-Aided Tracking (COSPAS-SARSAT) System [73, 74, 67, 95] also uses radio Doppler localisation. However, unlike TRANSIT, SARSAT uses a tracking architecture where the satellites receive signals from beacons on the ground. When activated in an emergency, the beacons transmit a signal that low orbit satellites intercept. The observed frequency shifts are then used to calculate the location of the beacon which is forwarded to relevant rescue authorities. The SARSAT system provides location information for emergency rescue operations involving people, marine vessels, or aircraft.

In the next section, we look into the process of observing physical phenomena and describe how location systems attain the measurements required by their computational models.

2.3 Observing Spatial Phenomena

The localisation methods described in the previous section rely on measurements of distance, angle or velocity. Attaining these measurements from the environment requires observation of physical phenomena. While some methods merely observe naturally occurring phenomena, such as video captured from a naturally or artificially lit scene, most methods actively stimulate the environment and measure the effects of the stimulus. GPS satellites, for example, actively create radio waves that receivers use to observe distance information. Various indoor systems also perform the same task but use acoustic signals instead.

In this section, we explore a number of different physical phenomena and examine their use in existing location systems. The systems described have been separated into two different categories based on the phenomena they use: *electromagnetic phenomena* and *acoustic phenomena*.

2.3.1 Electromagnetic Phenomena

Several location sensing systems employ electromagnetic media as a source of spatial observation. Electromagnetic waves travel at the speed of light and include frequencies in the radio and light bands.

Radio Waves

GPS uses the pseudoranging model described in Section 2.2.2. To obtain measurements for the model, GPS uses the relationship between satellite-receiver distance and the time it takes for a radio signal to travel between them. Given a model for this relationship (for example, radio waves travel at the constant speed of approximately $3 \times 10^8 \text{ ms}^{-1}$), the GPS receiver measures the time-of-flight of the signals and converts them into distances. By using a precise time reference, known as GPS time, the satellites are able to encode each signal with the time of transmission. A receiver receiving the signal measures the reception time on its local clock and calculates the difference between the two times. This produces the required distance measure for use with the pseudoranging model.

In order to provide the model with enough information to calculate position, the GPS signals are encoded with an ensemble of information known as the Navigation Message [48]. Among other pieces of data, the navigation message contains ephemeris information that gives the position of the transmitting satellite. Recall that this is required by the pseudoranging model to calculate the position of the receiver.

The pseudorange is extracted from the radio signal using *coarse acquisition* (C/A) codes. These codes uniquely identify the transmitting satellite and are repeated frequently within the signal so as to provide a steady stream of distance measurements. To calculate a pseudorange, the hardware in the GPS receiver time-shifts copies of these codes until they are aligned with the codes of the incoming signals. The magnitude of the shifts provide the time-of-flight of the signals, calculated with respect to the receiver's local clock.

The C/A codes and the navigation message are transmitted simultaneously through modulation with a carrier signal. There are currently two main GPS carrier signals: L1 which has a frequency of 1575.42 MHz, and L2 which has a frequency of 1227.60 MHz. The

use of multiple carriers allows for accurate measurement of ionospheric effects, provides resolution of the phase of the repeating C/A codes, and provides redundancy in the system [48].

Another location system that uses radio signals is SARSAT. The SARSAT system estimates the location of beacons using the Doppler shift of the radio signals that they emit. Doppler shift is the observed change in signal frequency that results from relative movement between the signal source and the observer. It occurs because the wavelength of a signal is compressed when the source and observer are moving towards one another and lengthened when they are moving apart. By using this relationship between observed frequency and movement, it is possible to determine the relative velocities of the source and observer and, hence, calculate location using the relative velocity model.

SARSAT employs a slightly different velocity model than the one given by Equation 2.4. In particular, the model assumes that the mobile device is stationary with a zero velocity and that the reference nodes (the satellites) are moving. It can be shown that the model equation takes the following form for satellites in circular orbits [122]:

$$v_i = -\frac{\mathbf{X} \cdot \mathbf{S}_i}{\|\mathbf{X} - \mathbf{P}_i\|} \quad (2.5)$$

In this equation, \mathbf{S}_i is the velocity of the satellite that measures the Doppler shift of the distress signal. The fact that the satellite infrastructure performs the measuring makes SARSAT a tracking system rather than a positioning system.

The SARSAT satellites process signals from emergency transponder beacons that transmit on frequencies of 121.5 MHz and 406 MHz. The signals carry information identifying the registered owner of the beacon. Modern beacons contain GPS receivers, and positions calculated by these receivers are also included in the transmitted distress signal.

There are a number of indoor location systems that also use radio signals. Many of these radio-based systems exploit the relationship between signal strength and transmitter-receiver distances. The relationship results from the attenuation of the radio signal as it propagates—the further away the receiver is, the weaker the signal. One such system, known as RADAR, uses signal strength fingerprints created by fixed 802.11 wireless networking base stations to estimate the positions of mobile users [7]. It requires a preprocessing phase to create a signal strength map of a building. Given this map, the

system uses on-line signal strength measurements to estimate the location of the users within the building. Another system called SpotON is based on a similar principle [61]. However, instead of using wireless networking hardware it employs custom built tags and base stations. In more recent research, signal strength signatures from mobile phone cell towers have also been used with signal strength maps [97]. In general, the technique of using signal strength maps to infer position is also known as *fingerprinting*.

The radio signal strength approach is attractive because off-the-shelf equipment such as 802.11 or Bluetooth network cards and mobile phones can be used. However, the main disadvantage to these methods is that the strength of radio signals is affected by transient objects in the environment. A number of researchers have therefore found that signal strength is too unreliable for use in a fine-grained indoor location system [112, 37]. They have discovered that achieving better than room-level accuracy often requires prohibitively complex models of the buildings being covered.

One system has attempted to tackle this problem by using power lines as the signal transmission medium [99]. By placing two tone generators on the power lines of a house or building, the system is able to bypass interaction with interfering walls, floors, etc. The tones are generated at opposite ends of the building and are adjusted to attenuate as they travel through the wires. A receiver within the building measures the signals radiated by the power lines and references a precomputed fingerprint map to determine its location. Since most of the distance travelled by the signals is through wire, the measured attenuation is more reliable than if the signals travelled through the air. The signal strength fingerprints are, therefore, more stable and provide a more consistent location reference.

In general, signal strength is used as an alternative to time-of-flight. Since it varies with the distance from the transmitting source, signal strength provides a way of estimating range. The complex effects of multipath fading and interference, which result from radio signals interacting with objects in the environment, have given rise to the fingerprinting methods as a way of overcoming them. The multipath effects are also the reason why time-of-flight is not used with conventional narrowband radio. In fact, using narrowband radio to provide time-of-flight measurements, as is done with GPS, is not feasible in indoor environments. Specifically, multipath artifacts—which are reflected and refracted versions of the direct signal—degrade the receiver’s ability to isolate the arrival time of the direct signal.

Recently, another form of radio location sensing has gained popularity for its ability to overcome this problem. Ultrawideband radio, which uses a wide range of frequencies, can be used in a mode that allows for the resolution of the direct signal [116, 56]. In this short pulse-width mode, direct nanosecond pulses are singled-out from their multipath reflections so that time-of-flight can be accurately measured [66, 132].

Ultrawideband also uses power efficiently. Because the power is spread across many frequencies, ultrawideband signals have a power spectral density that is magnitudes lower than equivalent narrowband systems. The result is that ultrawideband transmitters consume much less power. The amplitude of ultrawideband signals is also beneath the noise floor, meaning that they do not interfere with preexisting narrowband radio systems [133]. These benefits make the technology ideal for applications in the areas of sensor networks, mobile computing, ubiquitous computing, and many others.

A number of ultrawideband location systems have been proposed that use time-of-flight as well as angular measurements [71, 25]. One company, the Cambridge based Ubisense, markets ultrawideband location systems for indoor environments [124]. Their system, the Ubisense Platform, uses models that incorporate measurements of angle and distance, providing \sim 15 cm position accuracy.

Despite their benefits, the use of ultrawideband location systems is currently limited by their price. The expense of systems such as the Ubisense Platform has restricted their deployment outside of large, well funded projects.

Light Waves

Media that fall within or near visible light on the electromagnetic spectrum have been used in a number of location systems. Infrared was used in the first indoor system, the Active Badge System, to track people at the semantic level of rooms [127]. It uses networked infrared readers to identify and locate people carrying infrared transmitting tags. The Active Badge utilises the fact that infrared and visible light cannot penetrate surfaces such as walls and floors. Specifically, the sighting of a tag by a reader would indicate to the system that the person carrying the tag is in the same room as the reader.

The use of infrared motion sensors have been explored by Wren and Tapia for modelling and recognising the activities of building occupants [136]. These sensors only measure the presence of people in designated parts of the building. They cannot track individuals as they walk through a corridor, for example. By using time sequenced measurements from the infrared sensors, it is possible to make high level inferences about the intent and actions of people in the building. This information can then be used by intelligent in-building systems to improve factors such as productivity and safety.

The HiBall Tracker is a high performance tracking system that is able to achieve millimetre accuracy with very high update rates [131]. It uses a collection of multi-directional photo diode cameras (the HiBall) to observe a grid of infrared beacons on the ceiling. The view seen by the cameras is used to determine the position and orientation of the HiBall. The approach is similar to that taken by the camera based SLAM systems discussed in Section 2.2.2. However, the SLAM systems differ from the HiBall in that they do not require an infrastructure such as an array of beacons. Instead, they automatically determine and track features within the video feed of the camera.

A number of systems use optical signals to determine range. As mentioned in Section 2.2.2, the robotics community has used laser range finders to perform SLAM. Additionally, in the domain of sensor networks, the Lighthouse Location System performs positioning by using constantly rotating beams of light at fixed locations [111]. With this system, a Smart Dust device determines its distance from the light sources by observing the time that the beam is visible. Since the rotational velocity of the source is constant, this time is directly related to distance. With multiple light sources, the device is able to calculate its position using the ranging method.

Like radio signals, it is also possible to use the signal strength or the intensity of light to aid in determining position. The LuxTrace system, proposed by researchers at ETH in Zurich, uses shoulder mounted solar cells to measure the intensity of light coming from fluorescent office lights [104, 105]. By first creating an intensity map of the tracking areas (similar to the radio fingerprinting method), the system will use real-time measurements of light intensity to provide insight into the wearer's location and context.

Because light exhibits wave-like properties in the same fashion as radio, it is also possible to measure the Doppler shift in light signals. As a result, light waves can be used to obtain relative velocity measurements for use with a relative velocity model. While it

has yet to be used in an indoor location system, Doppler shifts of light waves have been used in other fields of research, such as astronomy to observe the motion of stars, and meteorology to analyse the motion of weather systems.

2.3.2 Acoustic Phenomena

A number of indoor location systems use acoustic signals as their observational foundation. Acoustic signals are an attractive medium since they travel slowly compared with electromagnetic signals. For example, to achieve sub-metre distance measurements using electromagnetic signals, a system requires hardware that has a timing resolution of less than three nanoseconds. Achieving the same accuracy with acoustic signals only requires a resolution of three milliseconds—a million times slower. The difference means that acoustic measuring equipment can be cheaper than electromagnetic equipment because it does not require the same high clock resolution.

Many ultrasound based systems use a combination of ultrasonic and electromagnetic signals to determine range. The Active Bat system, for example, triggers an ultrasonic transmission from wearable tags using radio signals. The ultrasonic signals are then received and processed by a network of ultrasonic transducers embedded in the ceiling. By assuming that the travel and processing time of the radio signal is zero (actually, a small constant), time-of-flight can be calculated by subtracting the radio transmission time from the reception times of the ultrasonic signal. In essence, the system assumes that the transmission time of the ultrasonic signal is the transmission time of the RF signal.

Another combination system is the Constellation, employed as part of Intersense's family of CAVE virtual environments [44]. It combines ultrasound, infrared and inertial sensors to achieve high performance tracking. Infrared and ultrasound provide range measurements to correct the drift of the high frequency inertial sensors. The ranging works in reverse of the Active Bat. Specifically, the source of the trigger signal and the ultrasonic signal are swapped. In the Constellation, the mobile device carries an ultrasonic receiver and uses infrared to trigger ultrasonic transmissions from transducers in the infrastructure. Again, by assuming that the transmission time of the infrared signal is the transmission time of the ultrasonic signal, the system is able to calculate range.

Because of their designs, the Constellation and the Bat are both restricted in terms of their scalability. For example, if more than one mobile device were to be used with the Constellation, a collision avoidance scheme would have to be introduced to make sure that two different ultrasonic transmitters are not activated at the same time. The Bat must consider this as well. It must ensure that tags in the same area are not activated simultaneously so that the receiving system is confused. As the number of users of the system increases this mediation task becomes harder. Furthermore, the time between tag activations increases because the narrowband ultrasonic signals must be time multiplexed. Basically, the two-way communication between the mobile unit and the infrastructure restricts these systems in terms of their ability to scale with respect to the number of mobile devices.

The Cricket is different from the Active Bat and Constellation in that it uses one-way communication. Cricket beacons simultaneously transmit a radio signal and an ultrasonic signal that “listeners” use to calculate range. Once more, it is assumed that the radio signal arrives instantaneously, meaning that range can be calculated by subtracting the arrival time of the radio signal from that of the ultrasonic signal. Although it can be configured in reverse, the default configuration is to place the beacons in the environment and use the listeners as mobile devices [118].

A similar approach is taken by the “Low Cost Indoor Positioning System” developed at the University of Bristol [108]. With this system, the infrastructure broadcasts a radio signal followed by a sequence of ultrasonic signals that are separated by a constant time interval. With knowledge of this interval, mobile receiving devices are able to calculate their distance from each of the ultrasonic transmitters and, hence, determine their position. The radio signal basically provides a synchronisation mechanism that allows the mobile device to determine the transmission time of each of the ultrasonic signals.

A few location systems employ only acoustic media. One of these systems; presented by Hazas et al.; uses broadband ultrasound to overcome some of the issues that narrowband systems face [57, 54]. In particular, the system is able to deal with in-band noise, allows for multiple channel access, and provides for signal source identification. While the system uses an RF trigger signal, the authors propose that it can be used in an asynchronous pseudoranging configuration, allowing for radio free operation.

Despite its benefits, broadband ultrasound comes at the expense of simplicity and accessibility. For example, broadband ultrasound requires heavy signal processing and is currently dependent on custom built transceivers. In contrast, narrowband systems use simple signal processing methods and can be built with off-the-shelf components.

A system presented by Scott and Dragovic takes this design motivation to the extreme [113]. Their audio location system uses off-the-shelf PC microphones to track the source of audible sounds such as the snap of a finger. Software operating on a PC records the sound's time of arrival at each of the microphones by looking for sharp rises in the amplitude of their corresponding audio streams. The tracking algorithm performs a form of range-based localisation using the difference between each of the arrival times.

The same time-difference-of-arrival model is used by Duff et al. in a self-calibrating system that tracks ultrasonic tags [33]. Time-difference-of-arrival is similar to the ranging model given by Equation 2.2. The difference is that one of the equations is selected as the “zeroth” equation and is subtracted from the others. This makes the scalar to the left of the equal sign a range difference measurement, which is equivalent to time-difference-of-arrival multiplied by the speed of sound. In fact, the form of this equation is the same as that used by some closed-form solutions to the GPS pseudoranging equations [80, 40, 62]. However, the tracking systems presented by Scott and Duff use error minimisation techniques to solve their equations.

In the case of the BUZZ positioning systems, the use of narrowband ultrasound with stochastic state-space algorithms provides a compromise between accessibility and performance. The hardware components can be purchased off-the-shelf while the lightweight algorithms provide optimum positioning performance on a mobile device. Furthermore, by avoiding the use of radio signals, power consumption and form factor are improved.

In the next section, we explore a number of different algorithms for determining location.

2.4 Localisation Algorithms

In the previous two sections, we outlined a number of localisation models as well as some observational phenomena to realise them. In this section, we examine algorithms that combine the models and the observations to form a complete location sensing system.

2.4.1 Analytical Solutions

An analytical solution is the most straightforward way of determining position. Forming an analytical solution involves using a finite series of functions to transform the chosen model into a formula that provides a simple expression of the desired quantities. This new expression is then used to calculate estimates for position each time a measurement is taken.

To illustrate, we construct an example. We note that the techniques used in our example are not employed by the BUZZ and serve only as a reference for understanding analytical solutions. Consider a two dimensional angle based system with reference nodes located at $(-7, -1)$ and $(9, -3)$. Assume that the layout is similar to that shown in Figure 2.1, with measurements for θ_0 and θ_1 taken counter-clockwise from the east or $(1, 0)$ direction.

Before processing measurements, we manipulate the equations in Section 2.2.1 to solve for \mathbf{X} . We start by subtracting the first instance of Equation 2.1 for $i = 0$ from the second instance for $i = 1$. This gives:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} P_{1_x} \\ P_{1_y} \end{bmatrix} - \begin{bmatrix} P_{0_x} \\ P_{0_y} \end{bmatrix} + R_1 \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \end{bmatrix} - R_0 \begin{bmatrix} \cos \theta_0 \\ \sin \theta_0 \end{bmatrix} \quad (2.6)$$

where $\mathbf{P}_i = (P_{i_x} \ P_{i_y})^T$. The equality given by the second row of Equation 2.6 is then manipulated and a trigonometric identity is applied to provide an expression for R_1 .

$$R_1 = \frac{(P_{1_y} - P_{0_y}) \cos \theta_0 - (P_{1_x} - P_{0_x}) \sin \theta_0}{\sin(\theta_0 - \theta_1)} \quad (2.7)$$

Finally, we substitute this expression, and the positions of the reference nodes, into the instances of Equation 2.1 to solve for the desired variable \mathbf{X} :

$$\mathbf{X} = \begin{bmatrix} 9 + \left(\frac{-2\cos\theta_0 - 16\sin\theta_0}{\sin(\theta_0 - \theta_1)} \right) \cos\theta_1 \\ -3 + \left(\frac{-2\cos\theta_0 - 16\sin\theta_0}{\sin(\theta_0 - \theta_1)} \right) \sin\theta_1 \end{bmatrix} \quad (2.8)$$

In this form, the expressions provide us with a way of calculating position directly from the measured angles. For example, processing the measurements $\theta_0 = 36.87^\circ$ and $\theta_1 = 157.38^\circ$ results in a position estimate of $(-3, 2)$. Equation 2.8 is known as a closed-form solution.

Analytical approaches can be troublesome in practice. Specifically, it is not always possible to manipulate the model equations to express the unknown quantities in terms of the known quantities. In fact, the model equations for the relative velocity method described in Section 2.2 cannot be rearranged to provide a direct expression for position. In these non-linear cases, a different type of algorithm is required.

One downside to analytical solutions is that they do not take into account errors in the measurements. Measurement error is unavoidable in practice and results from noise and other unmodelled effects of the sensing system. If we were to plug measurement values into Equation 2.8, the errors in the measurements would propagate directly into the solution for position. If there is a high dilution of precision in the equation (for example, the nodes \mathbf{P}_i are not situated ideally) then small measurement errors can result in large position errors.

This also becomes an issue when the system is over-constrained. For instance, when there are more than two nodes contributing measurements to the solution in our example. In this case, the different errors from each measurement will result in different solutions for position, depending on which nodes were used with Equation 2.8. The net result is that successive measurements, even if they are taken while the mobile device is in the same position, will cause the solution to “jump around”.

There are ways to use these differing estimates for position to provide extra information to the solution, however. For example, if the error characteristics of each of the measurement sources is known, they can be factored in to provide an optimum compromise.

It is also possible to use information obtained from previous calculations to improve on the position estimate. We discuss algorithms with these capabilities in the next sections.

2.4.2 Simple Numerical Solutions

For single variate equations where analytical solutions do not exist, numerical algorithms may be applied. A single variate equation is one that can be represented as $f(x) = g(x)$, where x is the variable for which we are solving. Since this representation can also be expressed as $f(x) - g(x) = 0$, the process of solving for x is also called *root finding* as the solutions to the system provide the roots, or zeros, of this equation.

Numerical solutions are iterative algorithms that start with some initial information about the unknown variable x , and subsequently converge to a solution. Four common single variate numerical methods include the Bisection Method, the Secant Method, the False Position Method and the Newton Raphson-Method [100]. All of these require that the functions $f(x)$ and $g(x)$ be continuous, real-valued functions.

The Bisection, Secant, and False Position Methods are based on the intermediate value theorem. This theorem states that given an interval $[a, b]$ on x such that the continuous function $h(x) = f(x) - g(x)$ satisfies $h(a)h(b) < 0$, it holds that there is a root between a and b . In other words, if the function is continuous, and $h(a)$ and $h(b)$ are of opposite sign, then the function will cross through zero between a and b . The methods operate by iteratively reducing the size of the interval until the root is found.

The Newton-Raphson Method requires that the equation $h(x)$ be continuous and differentiable. By using a guess for x , the method iterates to the solution by finding the root of the line tangent to the curve at this guess. The root of the tangent is then used as the guess for the next iteration. The method essentially follows the slope of the curve towards the root. This implies that the more that the slope of the curve approximates that of a straight line, the less iterations are required to converge to a solution. Newton-Raphson is more efficient than the Bisection, Secant or False Position methods, but can fail to converge if the initial guess is not chosen well.

One short-coming to the single variate numerical solutions, is that they only find local roots to $h(x)$. They pursue the roots in the neighbourhood of the initial guess, or within

the specified interval. As such, the accuracy of the initial conditions is important to ensure that the intended solution is found. In practice, however, this initial information is not always available.

Another short-coming of the methods, as we have described them, is that they are only designed to work with one-dimensional, or single variable equations. Many of the problems posed in practice, however—including position finding problems—contain several variables. The Newton-Raphson Method can be extended into multiple dimensions. This form of the method involves multiplying with the inverse of the Jacobian matrix, which is constructed from the partial derivatives of the system of equations. The Jacobian is the multidimensional analog of the derivative. The inverse is taken because the Jacobian is the divisor in the calculation.

The Newton-Raphson Method can also be used to solve optimisation problems. Optimisation problems are those that seek to determine the value for x where the function $h(x)$ is a minimum or maximum. In multiple dimensions, this form of Newton's Method involves calculation of the Jacobian matrix as well as the inverse of the Hessian matrix (the Hessian is the multidimensional analog of the second derivative). Basically, optimisation with Newton's Method is the process of finding the zeros of the function's derivative. Since these zeros occur at values where the function forms a peak or a trough, the process is equivalent to solving for the maximum or minimum values. We look at minimisation further in the next section.

Simple numerical techniques suffer from some of the same disadvantages as analytical solutions. Namely, they do not explicitly deal with measurement error or over-constrained systems. In the next section, we explore algorithms that have been designed for these conditions.

2.4.3 Error Minimisation

Error minimisation algorithms work by minimising the difference between real-world observations and predictions provided by the model equations. These differences are called errors or residuals and they quantify how well a particular solution fits the measured data. By iteratively adjusting the unknown values in the system, error minimisa-

tion algorithms use the residuals to navigate the solution space. A zero valued residual indicates that the algorithm has found the precise solution.

Measurements in practice contain errors that arise from instrumentation noise and interference. For this reason, precise solutions with zero valued residuals—a condition assumed by the analytical and simple numerical methods previously described—are generally impossible with real-world data. Instead, error minimisation algorithms are designed to find the solution with the lowest possible residual (i.e. minimal error). Since error minimisation is the process of finding the optimal solution for the given data, these algorithms are also known as *optimisation algorithms*.

In a manner similar to the simple numerical solutions, error minimisation algorithms iterate the solution space and compute expected measurements for each estimate of the solution. Since the model equations are used to predict measurements, the equations need only express the measured values in terms of the unknown values—a form that is generally easier to achieve than that required by purely analytical solutions. A common way of formulating a minimisation problem is to sum the squares of the nonlinear equations. For example, if there are N nonlinear equations $\mathbf{h}_0(\mathbf{x}) \dots \mathbf{h}_{N-1}(\mathbf{x})$ such that $\mathbf{h}_i(\mathbf{x}) = \mathbf{f}_i(\mathbf{x}) - \mathbf{g}_i(\mathbf{x}) = 0$, we can form the following sum:

$$S = \sum_{i=0}^{N-1} (\mathbf{h}_i(\mathbf{x}))^2$$

Minimising this function gives the optimum solution for the unknown vector \mathbf{x} even if measurement errors exist such that $\mathbf{f}_i(\mathbf{x}) - \mathbf{g}_i(\mathbf{x}) \neq 0$. The method is referred to as *sum of squares* or *least squares* optimisation.

Error minimisation algorithms can be classified into two groups: gradient based or stochastically based methods [125]. Gradient based methods include algorithms such as the Method of Steepest Descent [100], Newton's Method, and a method tailored for least squares optimisation called Levenberg-Marquardt [90] (it is a combination of Newton's Method and Steepest Descent [100]). They involve the use of derivatives to observe the rate at which an area of the solution space converges towards the optimum solution. These gradients are followed towards the error minimum.

In contrast, stochastic algorithms probabilistically determine the transition of multiple individual solutions or hypotheses. They include algorithms such as Simulated Anneal-

ing [75] and Particle Swarm Optimisation [72, 35]. Stochastic algorithms can be used for problems where the derivative of the error function is difficult or impossible to obtain. They are also more robust to falling into local minima. Local minima are valleys in the solution space that have low error values, but are different to the intended global minimum. The use of multiple hypotheses allows the stochastic algorithms to span the solution space, thereby increasing their ability to differentiate the global solution from the local minima.

In terms of real-time position sensing, error minimisation techniques have been used in a number of systems. For example, they have been used with the Bat [128], the self-calibrating ultrasonic tracking system from the University of Bristol [33], and the audio location system by Scott and Dragovic [113]. These systems tend to be tracking systems, where computational resources are abundant and groups of simultaneous measurements are collected relatively infrequently. This trend follows from the fact that error minimisation is computationally intensive and works best when several measurements are provided as an observational “snap-shot”. The snap-shot ensures that the unknown parameters describing position are constant for the calculation, and that the problem is well-constrained.

For many location systems, error minimisation techniques are too resource hungry. This is especially true for mobile or wearable applications where computational resources are limited. Furthermore, minimisation algorithms do not use all of the information available to them, particularly information obtained from previous calculations. For example, if the position of a mobile device is calculated frequently, such that the device does not move far between calculations, then information from previous calculations can be utilised as part of the current calculation. Algorithms that use this notion of the device’s *state* are known as state space methods. We explore these in the following section.

2.4.4 State Space Methods and the Kalman Filter

State space methods are algorithms which can be used to estimate the state of deterministic systems, including physical processes. A physical process is defined as a continuous, time-evolving series of events that occurs in the natural world and follows the laws of physics. Examples include the swing of a pendulum or the level of liquid contained in

a vessel. In the realm of location sensing, we are interested in the process that involves the position and movement of mobile devices.

The *state* is a set of parameters that describe the instantaneous condition of a process at a particular point in time. State space methods operate by iteratively combining the previous estimate of the state with observed measurements. The measurements provide a window into the true behaviour of the process and allow the system to correct for predictive errors. Basically, the algorithms first *predict* the current state using a model of the process dynamics and the previous state, then *correct* the prediction by incorporating current measurements. This mechanism is repeated iteratively for the duration of execution.

Many state space algorithms are referred to as recursive time-domain filters since they use temporal models to map previous outputs back into the system. This predictive feedback mechanism is based on the Markov assumption that the current state contains all of the information needed to predict the next. This is powerful in that the system is not required to store long histories of state estimates, making it possible to provide estimates in real-time on practical hardware.

One of the most popular state space algorithms is the Kalman filter [70, 130]. It is an efficient algorithm that has been used in a number of real-time navigation and positioning systems. These include implementations for GPS receivers [48], inertial navigation [48], radar tracking [24], head tracking [43], and indoor positioning [44, 86], to name a few. The efficiency of the filter is derived from the assumption that all noise is Gaussian. This, combined with the further assumption that process and measurement models are linear, means that problems can be expressed stochastically using linear representations of mean and variance.

The Kalman filter is essentially an implementation of Bayes' rule in that it provides a way of updating the stochastic properties of the system given new measurement input [130]. The current state of a Kalman filter system is one that is conditioned on prior measurements as well as previous states. Put simply, the Kalman filter tracks a process by performing a weighted average of three entities: the information provided by the previous state, a model of how the process evolves over time, and observations of the process. The weights are represented in the form of covariance matrices which can be thought of as “confidence” measures. For example, a Kalman filter that is confident in its

previous estimate will weight this information highly when it is “averaged” with recent measurements. Conversely, the measurements will contribute more to the solution if the filter is less confident in the state.

Designing a Kalman filter to solve any problem requires the expression of two equations. These are known as the *state transition equation* and the *measurement equation*.

The state transition equation, also known as the difference equation, describes the evolution of the process over time. It is used to predict how the state changes between measurements. In terms of location sensing, this equation models the dynamics of the object being positioned and will be different for different systems. One way in which state transition models can differ is in whether or not there is control over the process dynamics. For example, input from the steering system in a car can be used to predict where the car is going. This is different to a human positioning system since the only way of gathering location information about people is through observation (the system has no direct control over a person’s movements). For these types of systems the state transition equation takes the form:

$$\hat{\mathbf{x}}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (2.9)$$

Here, the state transition matrix \mathbf{A}_k is used to calculate a prediction for the current state vector $\hat{\mathbf{x}}_k$ using the previous state vector \mathbf{x}_{k-1} . $\hat{\mathbf{x}}_k$ and \mathbf{x}_{k-1} are N sized vectors containing all of the modelled parameters of the process. \mathbf{A}_k is an $N \times N$ matrix. The N -sized vector \mathbf{w}_{k-1} is the zero-mean, normally distributed noise present in the process. The subscript k is used to denote the iteration or time-step of the filter. The hat notation ($\hat{\cdot}$) is used to denote a predicted quantity.

We note that for systems that *do* have control over system dynamics, such as the car in our example, Equation 2.9 contains another term on the right-hand side of the equal sign. This term, normally denoted $\mathbf{B}_k \mathbf{u}_{k-1}$, adds the effect of the control input vector \mathbf{u} via the input matrix \mathbf{B} . For the systems described in this thesis, however, the term is not relevant.

The measurement equation expresses the observations in terms of the state variables.

It allows the filter to predict measurements based on the output of the state transition equation. Examples of measurement equations for position sensing are those given in Section 2.2 where we discuss methods that relate position to measurements of angle, distance and velocity. The general form of the measurement equa-

tion is as follows:

$$\hat{\mathbf{z}}_k = \mathbf{H}_k \hat{\mathbf{x}}_k + \mathbf{v}_{k-1} \quad (2.10)$$

In this equation, the predicted measurement vector $\hat{\mathbf{z}}_k$ is calculated by multiplying the state prediction by the measurement sensitivity matrix \mathbf{H}_k . \mathbf{H}_k is an $M \times N$ matrix, where M is the number of measurements in $\hat{\mathbf{z}}_k$. \mathbf{v}_{k-1} is the zero-mean, normally distributed measurement noise that models the real-world sensor noise and interference.

The Kalman filter uses these two equations to calculate the difference between the predicted measurements and the actual measurements ($\mathbf{z}_k - \hat{\mathbf{z}}_k$). It combines this value—known as the *innovation* or measurement *residual*—with weights calculated from information about the measurement and process noise to optimally update the state.

In terms of its operation, the Kalman filter first calculates a prediction for the state using a form of Equation 2.9:

$$\hat{\mathbf{x}}_k = \mathbf{A}_k \mathbf{x}_{k-1} \quad (2.11)$$

This first step in the process of integrating a measurement can be performed before or after a measurement is obtained. If the state transition matrix \mathbf{A}_k depends on the time between measurements, and measurements arrive asynchronously, then $\hat{\mathbf{x}}_k$ will need to be calculated after the observation.

Second, the Kalman filter predicts the confidence in the state given the state prediction. The confidence in the state is represented by a covariance matrix \mathbf{P} , the *system covariance*. It is a measure of how accurate the filter regards the estimates in the state vector. It is predicted as follows:

$$\hat{\mathbf{P}}_k = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{Q}_k \quad (2.12)$$

In this equation, $\hat{\mathbf{P}}_k$ is known as the *a priori* estimate for the system covariance. This means that it represents the filter's confidence in the state before conditioning with the measurement. The vector \mathbf{Q}_k is the covariance matrix for the process noise \mathbf{w} , also known as the *process covariance*. It is added to the system covariance with the effect of decreasing the confidence in the state. This is intuitive, since when time passes without an observation of the real-world process, the filter should become less confident in the state.

In the third step, the Kalman filter calculates the *Kalman gain matrix*. The Kalman gain matrix \mathbf{K} is a weighting factor used to optimally integrate the measurements with the prediction for the current state. We show its calculation with two equations:

$$\mathbf{Y}_k = (\mathbf{H}_k \hat{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (2.13)$$

$$\mathbf{K}_k = \hat{\mathbf{P}}_k \mathbf{H}_k^T \mathbf{Y}_k \quad (2.14)$$

In the first equation, the *information matrix of innovations* \mathbf{Y}_k is calculated. It incorporates the weight of the measurement noise through addition of the *measurement covariance matrix* \mathbf{R}_k . \mathbf{R}_k is the covariance representation of the zero-mean measurement noise \mathbf{v} . In the second equation, the information matrix is multiplied with the measurement sensitivity matrix and the *a priori* system covariance. This gives the Kalman gain: the weighting factor that determines how the measurements will influence the state.

Before the Kalman gain is used, the filter calculates a prediction for the measurement using Equation 2.10:

$$\hat{\mathbf{z}}_k = \mathbf{H}_k \hat{\mathbf{x}}_k \quad (2.15)$$

We note that this equation can be evaluated earlier in the algorithm, if the need arises. For example, the measurement prediction $\hat{\mathbf{z}}_k$ can be used to detect and reject outlying measurements (we return to this in Section 3.3.2).

In the final steps of the iteration, the Kalman filter integrates the measurement with the state:

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k) \quad (2.16)$$

Here, the innovation in the state, provided by the measurements, is calculated by, first, taking the difference between the measurement vector \mathbf{z}_k and the measurement prediction vector $\hat{\mathbf{z}}_k$, and second, by multiplying by the Kalman gain. This information is then added to the predicted state to complete the integration of the measurements.

In light of the integrated measurements, the filter also uses the Kalman gain to update the current system covariance. It uses the following equation:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \hat{\mathbf{P}}_k \quad (2.17)$$

where \mathbf{I} is the identity matrix. After this operation, the system covariance represents the distribution of the error in the state, conditioned on the error distribution of the measurement.

At the end of each measurement-integrating iteration, the state vector \mathbf{x}_k and the system covariance matrix \mathbf{P}_k hold the optimum estimate for the distribution of the state. The process of integrating a measurement can be visualised in terms of a beating heart: the system covariance first “expands” with the passage of time (this is seen in the calculation of the *a priori* system covariance $\widehat{\mathbf{P}}_k$ in Equation 2.12), then “contracts” as the observation is integrated (Equations 2.13, 2.14 and 2.17). Since the process covariance \mathbf{Q}_k is related to time, the heart, or system covariance, grows continually with the passage of time; adding measurements at discrete points causes it to shrink. The smaller the size of the heart, the more confidence there is in the state. At start-up, when there is little confidence in the state and the heart is large, the contractions outweigh the expansions until the filter eventually reaches a steady state of similar volumed contractions and expansions. We have found that the heart-beat analogy is useful for visualising and discussing the time-progression of three-dimensional error distributions in our positioning systems.

The process equation and the measurement equation are both linear expressions. For non-linear systems, more general representations of these equations are used. For example, the process equation can be generalised as a function that relates the previous state to the current state.

$$\widehat{\mathbf{x}}_k = f(\mathbf{x}_{k-1}, \mathbf{w}_{k-1})$$

Similarly, the measurement equation can be generalised as a function relating the state to the measurements:

$$\widehat{\mathbf{z}}_k = h(\widehat{\mathbf{x}}_k, \mathbf{v}_{k-1})$$

There are versions of the Kalman filter that use these representations to adapt the filter to non-linear process and measurement equations.

One non-linear version of the Kalman filter has come to be known as the extended Kalman filter [120]. It linearises the Kalman filter by applying a first-order Taylor series approximation to the process and measurement equations. Specifically, it approximates \mathbf{A} and \mathbf{H} by evaluating the Jacobian of $f()$ at \mathbf{x}_{k-1} and $h()$ at $\widehat{\mathbf{x}}_k$, respectively.

Another non-linear version of the filter is called the Unscented Kalman filter [69]. Unlike the extended Kalman filter, no linearisation is performed. The method uses sampled representations of mean and covariance which are propagated through $f()$ and $h()$. The resulting samples are then used to reconstruct the mean and covariance of the original distributions on the output of these functions. The formulation of the unscented Kalman filter is different to that of the traditional and extended Kalman filters in that the main equations (Equations 2.11 through 2.17) are modified significantly.

The advantages of the unscented Kalman filter are two-fold. First, it is easy to implement since no Jacobian calculations are required. Second, it more accurately models non-linear transformations in the process and measurement equations. The downside, however, is that the sampling mechanism within the unscented Kalman filter can make it more resource intensive than the extended Kalman filter [79]. As a result, we have chosen the extended Kalman filter as the estimator for the BUZZ positioning systems. We discuss general implementation details in Chapter 3, while specific extended Kalman filter designs for the BUZZ are presented in Chapters 4 and 5.

2.5 Conclusions

This chapter has provided an introduction to location sensing by defining a number of concepts and terminology. The building blocks for creating a location system have been explored. We have discussed the spatial models for recovering position, the various methods for measuring quantities within these models, and the algorithms that bind them together. The BUZZ positioning systems' algorithms use two different spatial models, one based on the pseudoranging model and the other based on the relative velocity model. They use narrowband ultrasound as the basis for spatial observations. Gluing the models and the observations together is the extended Kalman filter, a state space algorithm.

In the next chapter we explore the properties common to the Synchronous and Asynchronous BUZZ.

Chapter 3

The BUZZ

This chapter provides a discussion of common characteristics of the BUZZ positioning systems. It begins with a description of the nature of locating people in Section 3.1. Section 3.2 examines the properties of ultrasound and the issues surrounding ultrasonics-only positioning. In Section 3.3, we outline our algorithms and discuss their deployment on wearable hardware. Section 3.4 concludes the chapter with a discussion of the techniques we use to evaluate our systems.

3.1 People Positioning

The goal of the BUZZ positioning systems is to position individual mobile devices that have been attached to people. The nature of this attachment is loosely defined. To-date we have used our systems with applications where the devices are placed on top of the head, held in the hand, or fastened to hand-held instruments such as light-sabres (see Section 1.2). In any case, we assume that the dynamics of a mobile device is coupled with the dynamics of the human users wearing them.

To design a system capable of positioning humans, we must first examine the properties of human movement. Unlike cars, aeroplanes and other large bodies, people do not have much mass. This means that they have less momentum when in motion and,

therefore, have a higher probability of exhibiting sporadic bursts of high acceleration. The movement of a person's limbs, for example, can produce large accelerations over relatively short distances. This is true despite the fact that humans move with relatively low velocities.

Cars, aeroplanes and robots have systems that control their behaviour and motion. Information from these control systems can be used by a location system to indicate how the moving body's position is changing—essentially, helping it to predict where the car, aeroplane or robot is, or is going to be. Humans do not have control systems that we can tap into. A location system that determines the position of people-attached mobile devices must be able to do so using only observational data.

Human manoeuvrability is a major factor that must be considered. While certain vehicles such as cars and robots are normally constrained to occupy certain spaces or pathways, humans are able to go almost anywhere. An in-car GPS navigation system, for example, can be programmed to utilise the assumption that cars stay on roads. Using a road-map database, such a system can provide a good estimate for the position of the vehicle even when the GPS estimate is out by many metres. When positioning humans, on the other hand, this technique is not generally possible; people can climb hills, cross roads, and scale walls.

The same is mostly true for indoor systems. Knowledge of the whereabouts of walls, ceilings and floors can be useful for correcting spurious estimates in a low-resolution, building-wide system. At higher resolutions, it is also possible to provide a map of furniture and other objects. However, given that humans are able to lean or climb over desks, sofas, and chairs, this approach is likely to be of little value. Needless to say, the task of creating models to aid in the positioning of people is difficult. There are many factors to consider and the application and environment are important.

Nonetheless, models can be used to increase the performance of a location system. One approach used by researchers in computer vision is to use models of the human structure to aid in camera-based tracking. Works by Wren et al. describe methods of tracking humans using information about joints and limbs [135, 134]. Their models allow a system to introduce constraints based on the mechanics of the human form, thereby improving the predictive components of the system.

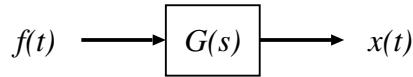


Figure 3.1: A generic control system with transfer function $G(s)$. The input to the system is the function $f(t)$, the output is $x(t)$.

The models that we use with the BUZZ positioning systems take a simple approach. They assume that the movement of a mobile device is a result of process noise, equally probable on all three spatial axes. Depending on whether the process model contains velocity or not, the Kalman filters attribute this process noise to acceleration or velocity, respectively. In essence, the models assume that the mobile device is free to move in all directions and is only bound by the relationship between position, velocity and acceleration.

We derive the values for the process covariance matrix \mathbf{Q} according to a white-noise integrator method presented by Brown and Hwang [15]. The method uses control systems theory to generate equations for the desired elements. Consider a generic control system such as the one given in Figure 3.1. This system has a transfer function $G(s)$, an input function $f(t)$ and an output function $x(t)$. In the time domain, the output can be obtained by convolving the input with the inverse Laplace transform of the transfer function. This is denoted as $g(t)$.

$$x(t) = f(t) * g(t) = \int_0^t g(u) f(t-u) du$$

The mean square value of the output is therefore:

$$\begin{aligned} E[x^2(t)] &= E\left[\int_0^t g(u) f(t-u) du \int_0^t g(v) f(t-v) dv\right] \\ &= \int_0^t \int_0^t g(u) g(v) E[f(t-u)f(t-v)] du dv \end{aligned}$$

In these equations, $E[x^2(t)]$ is used to denote the expectation value of $x^2(t)$. The term $E[f(t-u)f(t-v)]$ is the autocorrelation function of the input $f(t)$. It is abbreviated as $R_f(u-v)$ such that the final expression for mean square value is:

$$E[x^2(t)] = \int_0^t \int_0^t g(u) g(v) R_f(u-v) du dv \quad (3.1)$$

The relevance of Equation 3.1 is apparent when we assume that the input function $f(t)$ is a zero-mean Gaussian white noise process. This means that, since covariance can

be calculated from the relation $\text{cov}(x) = E[x^2] - (E[x])^2$ and $E[x] = 0$ (zero mean), the expected mean square value is the equal to the covariance. Also, since the input is *white* noise, we assume that it contains all frequencies with amplitude A . As a result, the corresponding autocorrelation function is merely the Dirac delta function: $R_f = A\delta(u - v)$.

Using Equation 3.1 we calculate the elements of \mathbf{Q} for two types of motion model: the *Position Model* and the *Position-Velocity Model*.

3.1.1 Process Covariance for the Position Model

The Position Model, or the P-model, assumes that the noise in the process is the result of velocity. In essence, it models velocity as zero mean, Gaussian white noise. In terms of the control system model, the transfer function is an integrator: $G(s) = s^{-1}$ and $g(t) = 1$. Brown and Hwang show that the covariance for each of the three dimensions of position is calculated as follows:

$$\text{cov}(x) = A_v \Delta t \quad (3.2)$$

Here, the spectral amplitude of the white noise velocity process is shown as A_v . We set this value according to rough estimates for the mobile device's velocity. We have found that a value of 1 ms^{-1} works well for most applications. The time range Δt is the time over which the noise is integrated. In terms of an operating Kalman filter, it is the time between measurements.

The diagonal elements of \mathbf{Q} corresponding to the (X, Y, Z) values of position are calculated using the above equation. The off-diagonal elements are zero since we assume that movement between the three spatial dimensions is uncorrelated.

3.1.2 Process Covariance for the Position-Velocity Model

The process covariance matrix \mathbf{Q} is larger for the Position-Velocity Model (PV-model) because it models the noise of three additional variables. These variables represent velocity of the mobile device in each of the three spatial dimensions. In this version of

the process model, acceleration is modelled as zero mean, Gaussian, white noise. For the covariance of position, the transfer function is a double-integrator with $G(s) = s^{-2}$ and $g(t) = t$. Plugging these functions into Equation 3.1, Brown and Hwang derive the position covariance elements:

$$\text{cov}(x) = \frac{A_a \Delta t^3}{3} \quad (3.3)$$

Again, Δt is the time between measurements. A_a is the spectral amplitude of the input noise. In practice, we have found that a value of $A_a = 1 \text{ ms}^{-2}$ works well to model the acceleration of the mobile device.

For the process covariance of velocity, the transfer function is a single integrator, similar to that of the Position Model. Thus, the equation for calculating the process covariance of the velocity is also similar:

$$\text{cov}(v) = A_a \Delta t \quad (3.4)$$

The only difference is that the spectral amplitude used is the one for acceleration and not velocity.

Since the position and velocity of the mobile device are not independent, we must also calculate the variance between them. This is the *cross covariance* between position and velocity, and is calculated as follows:

$$\text{cov}(x, v) = \frac{A_a \Delta t^2}{2} \quad (3.5)$$

The derivation of Equation 3.5 uses different functions for the time representation of the transfer function $g(t)$. Specifically, $g(t) = 1$ for the variable x and $g(t) = t$ for the variable v . Apart from these values, the off-diagonal elements of the \mathbf{Q} matrix are zero.

The output of a system that integrates Gaussian white noise is called a random-walk process. These processes are characterised by a mean of zero and a covariance that increases without bound. Their properties imply that a Kalman filter using random-walk process models must input measurements frequently enough, so as to keep the process covariance from “blowing up” the system covariance. This behaviour can be seen from the equations above. In particular, the covariance is proportional to powers of Δt , which is the time between measurements.

Modelling the movement of the mobile devices as random-walk processes is straightforward and computationally inexpensive. It is also easy to adjust the noise amplitude parameters to reflect the dynamics of the particular application. For example, a system that positions the head of tourists walking around a museum may be configured to assume that movement is less erratic than the movement of a device used in a game. In either case, the random-walk serves to provide a general time-based spatial constraint that is sufficient for positioning people.

3.2 Ultrasound

The BUZZ positioning systems use narrowband ultrasound as the only source of spatial observation. Our motivation is based on economics: by avoiding the use of other media we reduce the number of components on our wearable devices. This saves space, money and reduces power consumption. From this cost saving perspective, ultrasound itself carries a number of advantages:

- Audio signals are slow, meaning that inexpensive, low resolution circuitry can be used for timing.
- Narrowband transducers are inexpensive, which saves on the total component cost of the system.
- Narrowband ultrasonic components consume little power, resulting in longer lasting batteries for mobile devices.

Furthermore, narrowband ultrasonic transducers are currently available as off-the-shelf components. This means that we can build systems for use today, unlike other systems that rely on custom hardware.

The BUZZ systems explored in this thesis share the same basic architecture. Specifically, self-positioning mobile devices receive short bursts of narrowband ultrasound generated by beacons fixed in the infrastructure. These bursts, also known as *chirps*, are transmitted in a specific pattern that allows the mobile devices to calculate position. The pattern of the signals and the properties of the infrastructure are what differentiate the two systems presented in Chapters 4 and 5.

The similarities of the two systems lie in their observational frameworks: *the calculation of position is based only on the arrival times of incoming chirps*. We discuss the issues surrounding this measurement scheme in the next two sections.

3.2.1 Sensing Narrowband Ultrasound

We define a chirp as a $250\ \mu\text{s}$ burst of narrowband ultrasound centred around the 40 kHz frequency. We note that this definition of *chirp* is different from that used within the radar and sonar communities, where the frequency of the signal changes over time. Our chirps operate at a constant frequency of 40 kHz. They are generated and received by off-the-shelf transducers that provide a wide beam angle, high output and high sensitivity [108].

At 40 kHz, we achieve a good balance between the various properties of the acoustic medium. Signals with frequencies lower than 40 kHz begin to approach the audible range for humans and other animals. This not only means that the signals can be heard, but also means that there is more interference present in the environment. Conversely, at higher frequencies, attenuation becomes a factor. Audio signals attenuate at a rate proportional to the inverse of the square of their frequency [83]. In other words, signals with high frequencies can only travel short distances. We have found that 40 kHz provides a range of around 8 to 10 m depending on the transmission voltage. This range is sufficient for use within rooms as large as $10\ \text{m} \times 10\ \text{m}$.

We have observed the presence of ranging errors in our experiments, which are symptomatic of ultrasound and the ultrasonic transducers we employ. These errors have been seen to be as high as 10 to 15 cm in practice. We attribute them to three main causes.

The first cause is related to the signal travelling distance. As the chirp travels through the air, the energy within its waveform attenuates. This means that chirps that travel further take longer to stimulate a response from the receiver. The result is a chirp reception delay, as shown in Figure 3.2. The figure illustrates that the attenuated signal takes longer to reach the registration threshold such that the chirp appears to arrive slightly later. We note that this delay is a result of the way we have implemented our ultrasonic receiver hardware (see Section 3.3). A more sophisticated receiver, for example, could perform a reverse scan of the waveform to identify exactly when it arrived. However, we have avoided this more complex option in order to keep costs down. Additionally,

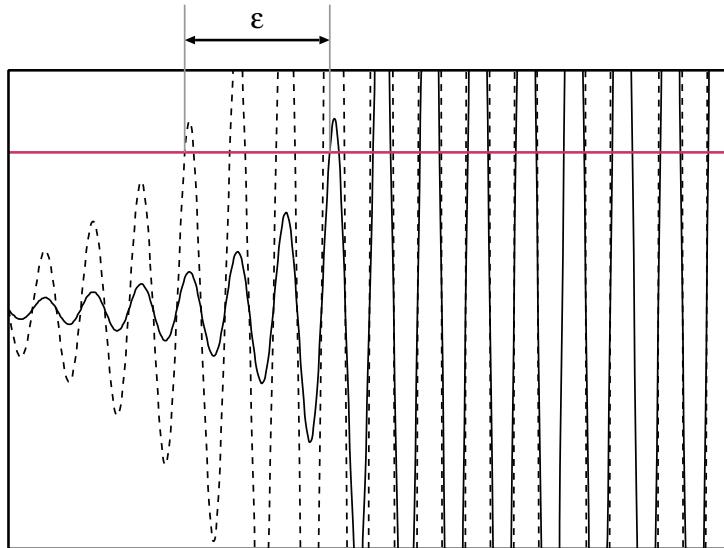


Figure 3.2: The amplitude of a strong signal (dashed) and an attenuated signal (solid) after amplification. The timing error ϵ is the difference between the time that each signal reaches the receiving circuit's threshold voltage for a digital "high" (red).

the distance related delay can be modelled within the positioning algorithm. Attempts at doing so have revealed some improvement, but we have found the extra complexity is unnecessary in practice.

The second cause of ranging errors is the anisotropic nature of the transducers. Basically, the behaviour of the transducers is dependent on the angle at which they are directed [78]. Larger angles between a transmitter and a receiver induce larger timing delays. As the mobile device moves around the positioning space, the angles between the transmitters and the device's receiver change. This causes a systematic error in the timing of the received signal, similar to the distance related error just described. While it is possible to measure the angle of the infrastructural transducers to achieve some gain through an "angle-versus-timing" model, we have found that this does not provide much improvement. This is especially true when the configuration effort required from the user is considered. It may be possible to automate the measurement of the angles through an auto-calibration procedure, but this is beyond the scope of this thesis.

The third cause of error is a result of our assumption that the speed of sound is constant. This is a false assumption, however, since the speed of sound changes as a function of

temperature and humidity (humidity has less of an effect than temperature). For example, one degree Celsius difference between system installations will cause about 2 mm error at a distance of 1 m, and 1 cm error at a distance of 5 m. Our algorithms assume that the speed of sound is 343 ms^{-1} , its value at 19°C and 0% humidity. To model the effect of temperature and humidity on the speed of sound would require the introduction of additional sensors. This technique has been employed in other systems [57], but we have decided against such an approach. We reason that the indoor environments in which we use our systems have a relatively similar temperature and humidity, and that the resulting errors will be minimal.

Our experience has shown that it is sufficient to neglect the systematic nature of the errors and assume that the measurement noise is zero mean, white and Gaussian. This assumption is inherent to the use of the Kalman filter as the basis for our algorithms. The Kalman filter expects the measurement error to have these properties, however, the systematic nature of these errors mean that they do not have a zero mean. The consequence of not modelling the errors depends on the arrangement of the beacons. If they are distributed well along all three axes such that dilution of precision is low, then the measurement error is effectively averaged out. If the dilution of precision is large, then the effects of the systematic errors will show up in the estimate for position.

In essence, we have made a compromise between the complexity and the accuracy of our positioning systems. If the measurement errors were the only source of error in the system, it may be worthwhile to model them. However, the effects of occlusions and reflections also add to the problem of measuring time-of-flight with ultrasound. They can either degrade, eliminate or interfere with the signals. As a result, we have found that modelling measurement errors does not actually do much to improve the performance of the algorithms when they are used with real-world applications. As our results in Chapters 4 and 5 show, a measurement covariance of 15 cm is effective for modelling the combined effects of the errors.

Another consideration to sensing narrowband ultrasound signals is the utilisation of the 40 kHz frequency channel. The fact that narrowband ultrasound occupies a single frequency band means that signals must be multiplexed in time. Ideally, each transmitter is given a monopoly on the ultrasonic channel for the duration of its activity in order to avoid collisions. Each of our systems employs a unique time division multiplexing scheme that attempts to maximise the use of this channel. Nevertheless, the time related

constraint of narrowband ultrasound puts an upper bound on the transmission period and, hence, measurement rate of our systems. We revisit this topic in Chapters 4 and 5.

3.2.2 Handling Anomalous Measurements

The use of narrowband ultrasound means that measuring equipment is susceptible to interference. As we have stated, this interference comes in the form of reflections—where a chirp “bounces” off of one or more surfaces before arriving at the receiver—and as in-band noise generated by mechanical activity in the space (jingling keys, coins, etc.). For methods that do not incorporate state, such as the error minimisation methods discussed in Section 2.4, spurious readings cause large but short-lived errors in position estimation. For a Kalman filter, on the other hand, the integration of spurious readings can, in the worst case, cause the system to diverge. A raw Kalman filter is not designed to handle spurious readings. It is an infinite impulse response filter (IIR) so the adverse effects of incorporating outlying measurements can have a lasting impact on performance [48]. It is therefore important for Kalman filtering algorithms to implement methods of rejecting bogus measurements.

It turns out that because Kalman filters use knowledge of the previous state, they are better able to identify outliers than algorithms that do not have any feedback. To spot anomalous readings caused by ultrasonic interference, the BUZZ systems use the Mahalanobis distance between the predicted and received measurement vectors. This metric, also called the χ^2 (chi-squared) statistic of the measurement residual, provides a method for identifying outliers given the current confidence in the state [48]. Using the Kalman filter notation from Section 2.4.4, the χ^2 statistic is calculated as follows:

$$\chi^2 = \frac{(\mathbf{z}_k - \hat{\mathbf{z}}_k)^T \mathbf{Y}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k)}{M} \quad (3.6)$$

Here, \mathbf{Y}_k is the information matrix as calculated in Equation 2.13. The vector $(\mathbf{z}_k - \hat{\mathbf{z}}_k)$ is the residual vector, and M is the size of the measurement vector \mathbf{z}_k . The effect of incorporating the information matrix is to provide a weighting of the filter’s confidence in the state. As a result, the χ^2 value for each measurement increases as the system covariance decreases. Put simply, as the confidence in the state grows, the algorithm becomes better at predicting measurements and can therefore be more rigorous in testing them against its predictions. Rejecting outliers is simply a matter of placing an upper

bound on this value and discarding violating measurements. For each BUZZ system, the χ^2 threshold has been determined experimentally.

The χ^2 method is effective for identifying most reflections and in-band interference. As long as these spurious signals do not directly affect the reception of a direct signal (collision at the receiver), readings for the interference will disagree sufficiently with the Kalman filter's predictions that it will be ignored. For interference that *does* collide with a direct signal, there is nothing that can be done. If the signal is corrupted to the extent that its χ^2 value is above the threshold, it will be lost. Otherwise, the corrupted signal will be incorporated into the state with the error propagating into the estimate for position.

Another issue surrounding ultrasound is signal occlusion. This occurs when a body of sufficient size blocks the direct path of a signal. The issue is particularly relevant for systems that involve people. People have limbs and clothing that can obstruct signals when the transducers are covered. They can also block the visibility of large spaces with their bodies, especially when the mobile device is hand-held (versus head or shoulder mounted). While the χ^2 thresholding method can deal with outlying signals, the only way to overcome occlusions is to use a sufficient coverage of beacons in the infrastructure. Ideally, the beacons should be laid out to ensure that most signals can reach the receiver even when others are occluded.

3.2.3 Beacon Density

In this section we provide an approximate beacon density calculation. It is intended to provide the users of our systems a rough guide for placing and separating the beacons within an indoor environment.

We assume that each of the beacons—in either of the BUZZ systems—are placed on the ceiling and face directly down towards the floor. This is not a recommended configuration (see Section 2.2.2) but allows us to simplify our density approximation. The chirps transmitted from a single beacon are assumed to form a cone, which expands downward at the transducer's beam angle θ . This behaviour is illustrated in Figure 3.3. Outside of the cone, chirps are inaudible by the receiver. The radius of the cone r is the audible range of the beacon in the XY plane and depends on the distance of the receiver from the

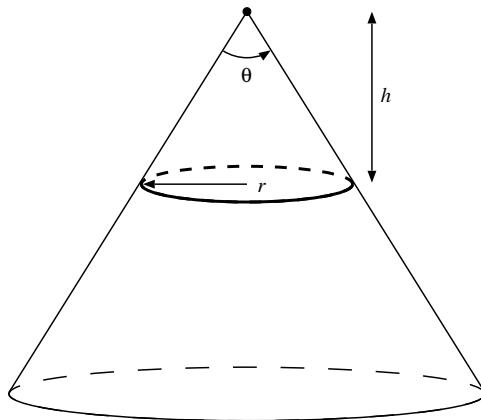


Figure 3.3: Audible cone of beacon with beam angle θ . The radius of the spot r depends on the distance from the beacon h .

ceiling h . The area of the “spot” created by the cone is calculated as:

$$A = \pi h^2 \tan^2\left(\frac{\theta}{2}\right)$$

We approximate the required beacon density ρ by dividing the required number of beacons N by this area.

$$\rho = \frac{N}{\pi h^2 \tan^2\left(\frac{\theta}{2}\right)}$$

This gives a measure for the number of beacons required per m^2 . We choose, however, to work with the inverse of ρ . This is an “area per beacon” measure and is more intuitive than the fractional values that calculations for ρ produce. For example, $1/\rho$ gives the user an idea of the area of positioning space that should be allocated to each beacon. The beacons can then be placed in the environment accordingly.

The transducers used with our beacons have a beam angle of 140° [108]. With this, Table 3.1 provides beacon density values for different values of N and h . The values of $N = 6$ and $N = 8$ correspond to the different requirements of the Synchronous and Asynchronous BUZZ, respectively. The values $h = 1 \text{ m}$ and $h = 2 \text{ m}$ are typical distances for a head- and hand-mounted receiver.

In practice, the arrangement of beacons will be different for each installation. It is recommended that beacons be placed at different heights within the positioning space and

	$N = 6$	$N = 8$
$h = 1 \text{ m}$	3.95	2.96
$h = 2 \text{ m}$	15.81	11.86

Table 3.1: Recommended beacon density values, given as m^2 per beacon.

that they be directed to provide maximum coverage of the space. For example, in smaller spaces it is possible to place the beacons on the walls as well as the ceiling. The beacons mounted on the walls should be directed so that most of the transmission cone occupies the space and is therefore more likely to be audible by the receiver. This is also recommended for beacons on the ceiling which are situated near the periphery.

3.3 Self Positioning Mobile Devices

The systems described in this thesis are positioning systems. In short, the algorithms that perform the task of calculating position do so on the hardware comprising the mobile device. Since these devices are mobile and are usually carried or worn by a user, we aim to make them as small as possible. As a result, computational resources such as battery-life, memory, and processing power are limited. To calculate position under these conditions requires the use of efficient and light-weight algorithms.

In the following sections, we describe the hardware comprising our mobile devices and outline the general structure of the algorithms that run on them.

3.3.1 Mobile Device as a Receiver

In this section we describe the hardware comprising our positioning systems. The software running on these systems are original works, however, the custom built hardware has been designed and manufactured by Randell [103, 106, 108]. Descriptions of the designs are provided for completeness.

Our mobile devices are passive, ultrasound receiving units. A single unit consists of a wearable computer attached to an ultrasonic receiver. The receiver has a PIC microcontroller that records the timing of the ultrasonic chirps generated by the infrastructure.

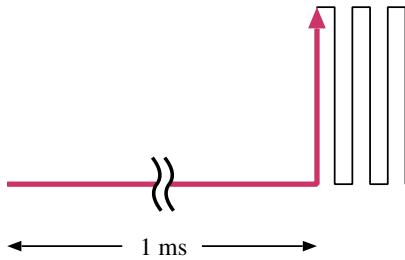


Figure 3.4: A chirp is characterised as at least 1 ms of “silence” followed by a rising edge on the digital input pin of the PIC. The rising edge marks the arrival time of the signal.

The chirps are picked up by the receiver’s transducer and pass through an amplification circuit before reaching the PIC. The output of the amplification stage is connected to a digital input pin on the PIC, which is sampled at around 800 kHz.

We have programmed the PIC to recognise the arrival of a chirp as a 1 ms period of quiet (digital zero) followed by a rising edge (digital one), as shown in Figure 3.4. This simple correlation ensures that the PIC gets a precise lock on the start of an incoming signal. The absolute time of the transition is recorded as a four-byte time-stamp and is sent over a serial link to the wearable computer for processing. The time-stamp is taken from a real-time counter that begins counting at power-up. It has a frequency of 156250 Hz, which gives the receiver a resolution of $6.4\ \mu\text{s}$ or around 2 mm. The counter wraps every 7.64 hours.

For the wearable computer, we have used two different systems over the course of our research, the *Bitsy* and the *Gumstix*. For both wearables, all of our algorithms are implemented using the C programming language and compiled using a cross compiler.

The earlier of the two systems, the ADS Bitsy, is a 32-bit, 200 MHz StrongARM computer running the Linux operating system [6]. It has 64 MB of RAM, a PCMCIA slot and three serial ports. We configured the system as a generic wearable computer with an 802.11 wireless networking card and a custom interface board to access the serial ports. A nine-wire bus is used to connect various different sensors—including the ultrasonic receiver—to the interface board. The sensors and the Bitsy are powered with a nine-volt battery housed in a custom unit. A bare Bitsy consumes approximately 330 mA to 500 mA at idle and full CPU activity [5]. The size of the Bitsy wearable, encased in

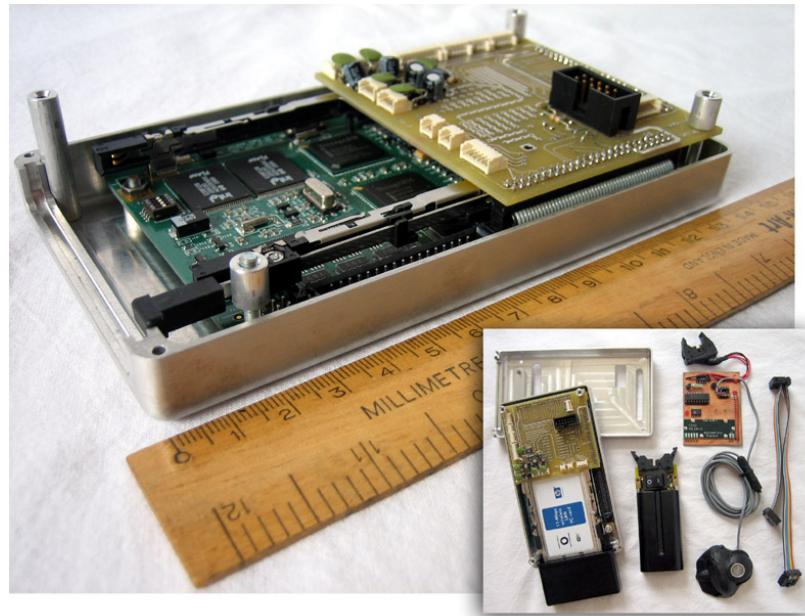


Figure 3.5: The ADS Bitsy with custom interface board housed in an aluminium case. *Inset:* A complete system with battery, receiver and nine-wire bus. Total weight: 560 grams. Power consumption: 330-500 mA.

a custom aluminium shell, measures $183 \times 83 \times 33$ millimetres. Figure 3.5 shows the Bitsy, the nine-wire bus with connectors, a battery unit, and an early ultrasonic receiver board/transducer. The entire setup shown in the inlay image weighs 560 grams.

The latest wearable system that we employ uses a small, 32-bit, single-board computer called a Gumstix [49]. The Gumstix has a 200 or 400 MHz Intel XScale microprocessor and an on-board Bluetooth module for wireless networking. It has 64 MB of RAM, runs Linux, and can be configured to have two or three serial ports. We connect the Gumstix to our sensors and a lithium-polymer battery using an eight-wire bus similar to the nine-wire bus of the Bitsy. The power consumption of a 200 MHz Gumstix is 30-40 mA while idle and 120 mA with the CPU usage at 100% [50]. Together with a custom bus interface board and a Bluetooth aerial, the computer measures $80 \times 20 \times 25$ millimetres. The inlay in figure 3.6 shows a complete Gumstix system including a battery and the latest receiver. The combined weight of these components is only 54 grams. The total component cost for the system is around \$150 USD.

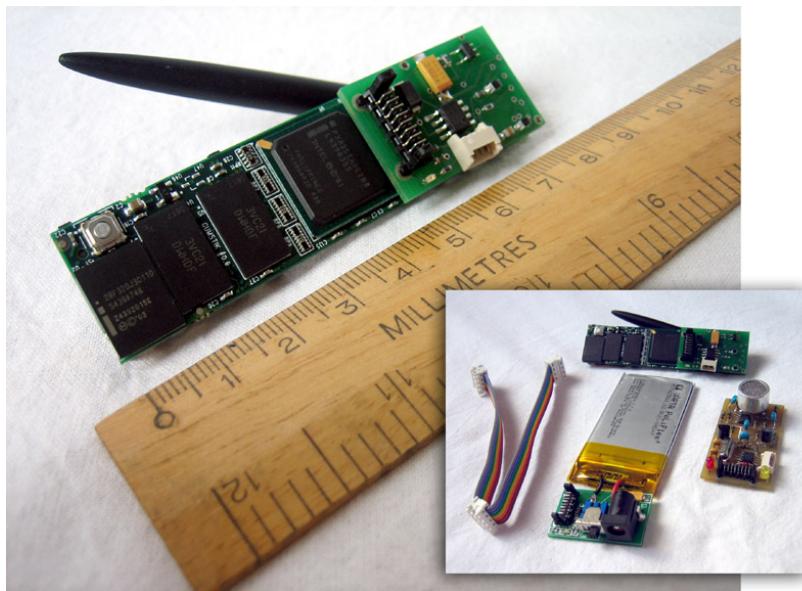


Figure 3.6: A Gumstix with an interface board and a Bluetooth aerial. *Inset:* A complete system with battery, receiver and eight-wire bus. Total weight: 54 grams. Power consumption: 30-120 mA (200 MHz model).

Due to its size and weight advantages, the Gumstix is currently our platform of choice. However, compared to the specifications of modern personal computers, the Gumstix is not a powerful machine. This is further highlighted by the fact that both the Gumstix and the Bitsy do not have any hardware floating point support. This means that floating point operations—which are necessarily performed in software—consume more processor cycles than they would on a PC. The bottom-line is that resources on our wearables are limited, therefore, the algorithms that run on them must be designed with this in mind.

3.3.2 Light-weight Positioning Algorithms

In this section we describe how we use the Kalman filter to implement our positioning algorithms. We first discuss its advantages over other popular filtering methods in terms of its computational efficiency. This is an essential consideration for deploying our algorithms on our wearable computers.

At its core, the Kalman filter represents noise and estimation error using only two statistical parameters: mean and variance. This is contrasted with a family of filters—called Sequential Monte Carlo methods—that have become increasingly popular for estimating partially observable processes [31, 42]. Monte Carlo estimators, such as the particle filter, “simulate” distributions using a number of samples (particles) embodying estimates for all or part of the state. In essence, these algorithms represent the system error distribution through the particle density—the more particles that are used, the more accurately this distribution is modelled.

To propagate the distribution at each observation time-step, a particle filter performs the following actions:

- particles are pushed through the process and measurement equations to provide measurement estimates
- these estimates are compared to the actual measurements to calculate a metric for measurement support
- new particles are randomly sampled from the old ones based on this support; greater support results in more child particles

Over time and with sufficient measurements, this process forces the particles to “condense” towards the most probable solution(s). The behaviour of the particles has inspired another name for this type of filtering within the computer vision community: *Condensation* [68].

While Monte Carlo methods are able to model non-white, non-Gaussian probability densities, they do so at the price of efficient resource usage. For many problems, accurately modelling the error distribution requires a large number of particles, usually on the order of hundreds to several thousands. Each of these particles consumes processor cycles and memory, thereby limiting the use of these algorithms on low performance machines.

On the whole, Kalman filters use less resources than particle filters. However, use of the conventional Kalman filter is limited to linear problems with white, Gaussian noise. To cope with problems that do not have these properties, many systems use the extended Kalman filter [120]. It linearises the Kalman filter by applying a first-order Taylor series

approximation to the process and measurement equations. As such, the design of an extended Kalman filter requires the calculation of partial derivatives to form Jacobian matrices. While not all problems can be formulated to use an extended Kalman filter (i.e. where the Jacobian is difficult, expensive or impossible to calculate), it has been used successfully with the BUZZ positioning systems. Further details are provided in Chapters 4 and 5.

Our Kalman filters implement a resource-saving technique known as serial or *single-constraint-at-a-time* (SCAAT) measurement processing [129, 48, 15]. With this method, single measurements are integrated with the filter serially rather than all at once. It is a counter-intuitive method especially from an analytical or an error minimisation point of view. Analytical and error minimisation solutions require that all known values, within a system of equations, be present concurrently. SCAAT, on the other hand, uses the stochastic, state space property of the Kalman filter to add observations to the state sequentially.

SCAAT can be applied wherever measurements can be decorrelated. This process involves diagonalising the measurement covariance matrix \mathbf{R}_k . Since the measurements in our systems are already uncorrelated, SCAAT is ideally suited as a way of improving the performance of the Kalman filter. The method improves performance in three ways. The first is that, with SCAAT, measurements can be added to the filter when they are observed. This is advantageous for systems such as ours, where observations trickle in over time. It allows the filter to more accurately reflect the process by avoiding assumptions of measurement simultaneity which are required for batch processing methods.

The second improvement comes from the fact that SCAAT allows the algorithm to isolate outlying measurements. Because measurements are processed individually, outliers can be identified even if the observations occur simultaneously. Recall from Section 3.2.2 that we use the χ^2 metric to determine the quality of the measurements based on the confidence in the state. This metric is a scalar value calculated from the measurement residual vector ($\mathbf{z}_k - \hat{\mathbf{z}}_k$) and represents the *combined* fitness of all of the measurements in the measurement vector. As a result, a decision to reject a set of measurements based on their χ^2 value can cause the system to reject good measurements as well as bad. Processing each measurement individually avoids this. Since a SCAAT system can produce a χ^2 value for each observation, decisions about quality can be made on a measurement-by-measurement basis.

The third benefit to SCAAT is that it is more computationally efficient than batch processing the measurements. While this seems counter-intuitive, the reason for its efficiency comes from the way in which the information matrix is calculated. The key is in Equation 2.13, which involves a matrix inversion. In a SCAAT implementation where $M = 1$, this step becomes a simple scalar division:

$$Y_k = \frac{1}{\mathbf{H}_k \hat{\mathbf{P}}_k \mathbf{H}_k^T + R_k} \quad (3.7)$$

Note that the measurement sensitivity matrix \mathbf{R}_k and the information matrix \mathbf{Y}_k have become the scalars R_k and Y_k . Therefore, in a filter that processes measurements serially, the matrix inversion process only requires a simple division operation. For a conventional Kalman filter, however, the inversion operates on a matrix of size $M \times M$, where M is the number of measurements. Inverting a generic matrix such as this is a non-trivial operation that can consume a large number of processor cycles. Common matrix inversion methods include Gauss-Jordan elimination, Gaussian elimination, or LU decomposition.

While the largest performance gain is through the inversion operation, other Kalman filter equations are also simplified with SCAAT. For example, χ^2 calculations are reduced to three scalar multiplications:

$$\chi^2 = (z_k - \hat{z}_k)^2 Y_k \quad (3.8)$$

All equations involving the measurement sensitivity matrix \mathbf{H}_k (which is now a vector), the measurement vector \mathbf{z}_k (now a scalar), and the measurement covariance matrix \mathbf{R}_k (also a scalar) are simplified.

As we have stated, even for systems where observations occur simultaneously, the performance increases from SCAAT are significant enough to warrant introducing extra operations to decorrelate measurements. In our research, we have experimented with both conventional and SCAAT Kalman filters and have found that SCAAT provides at least an order of magnitude reduction in processing time. As a result, we have chosen to implement each of the BUZZ systems presented in this thesis using SCAAT.

3.4 Evaluation Metrics

In general, evaluating the performance of a position sensing system is a non-trivial task. The primary reason is that it is difficult to obtain a ground-truth for comparing position estimates. This is even more complicated when the performance of the system needs to be evaluated in the dynamic, non-stationary case. Stated simply, we consider the task of evaluating positioning systems to be a “chicken and egg” problem. The question is: *how do we evaluate our positioning systems without a position sensing system already in place?*

We have decided to employ three different techniques for evaluating our systems. The first uses a simulation environment that allows us to test the affects of a number of different environmental and system parameters. The second is known as the Path Comparison Method, where the performance of the algorithms is compared with real-world paths traversed by the mobile device. A camera based method is the third evaluation technique. It uses a high speed camera to create a ground-truth against which we compare the output of the algorithms.

In the next sections, we detail our evaluation techniques and provide an overview of some common evaluation practises.

3.4.1 Evaluation Concepts and Techniques

To specify the accuracy and precision of a location system, evaluators use two main metrics for spatial error analysis: Circular Error Probable (CEP) and Distance Root Mean Square Error (DRMS) [64]. These are defined below.

Circular Error Probable denotes a radius from the true position within which a certain probability of system estimates fall. These distance values are usually given as 50% and 95% precision figures. CEP is also referred to as the Circle Error Probability or, for the 50% case, the Circle of Equal Probability. Spherical Error Probable (SEP) is the analog of CEP in three dimensions.

Distance Root Mean Square Error is the square root of the averaged squared distances between the true position and the system estimates. It is essentially the standard deviation of the distance errors within the horizontal plane. In three dimensions, DRMS is also known as Mean Radial Spherical Error (MRSE).

These metrics are radial (circular/spherical) approximations to the elliptical error densities observed in evaluation experiments. Unlike ellipse or ellipsoid measures, which are expressed in terms of covariance matrices, CEP and DRMS are expressed as scalars. Although this loses some detail in the statistics, it makes for more convenient and intuitive error reporting.

Where possible, we visualise the error probable metrics as *cumulative distribution functions* (CDF). These functions provide a useful visual representation of the errors in one or more experiments. They are characterised as plots with distance measures on the X -axis and fractions of occurrences on the Y -axis. To determine the error at a certain fraction or percentile, one merely locates a distance value along the X -axis by extending a vertical line from the curve at the percentile in question.

All error values are determined through experiment. An experiment is essentially the process of collecting and comparing position estimates to corresponding positions in the real world. We refer to these “real” positions collectively as the experimental *ground-truth*.

Obtaining a ground-truth for an experiment is a matter of knowing *where* the mobile device was located *when* the corresponding estimates were taken. For stationary tests, it is common to first map out a number of points within the positioning environment, then place the mobile receiver at each point so that position estimates can be recorded. Often, the process of mapping these points is an arduous task involving precise calibration of measuring tapes and other equipment.

Despite the effort involved, stationary experiments do not describe the time-related properties of the systems under scrutiny. Specifically, errors calculated at individual test points during a stationary evaluation are unable to provide insight into how well the same system will perform when the mobile device is moving. Instead, performance must be evaluated through experiments where time is included in the ground-truth. For

example, if the positioning system outputs a position estimate at time t , this should be compared to the position ground-truth taken at the same time, t .

One method for doing this is to use a camera, or a set of cameras, to video the movement of the receiver during the experiment. By synchronising the timing of the camera to the positioning system and extracting the position from the video, it is possible to determine the precise path followed by the mobile device. Another possibility is to use a motorised vehicle to move the mobile device around a predetermined test track. This requires the use of a switching system to determine the time that the device visited certain points on the track. For example, Smith et al. use a miniature train set and an optical track counter to evaluate the dynamic performance of the Cricket [118]. Such an approach allows for repeatability, which is useful for comparing different positioning systems. Its downside is that the dynamics explored are very specific, namely, that of a train moving around a track.

While both methods provide ground-truths for comparison, they each require a significant amount of effort. Furthermore, as these ground-truth-finding techniques are, essentially, position sensing systems in themselves, they will contain a certain degree of error. Balancing this error with the effort and cost of the ground-truth mechanism is a major challenge for system evaluators. We have chosen to take a simple and inexpensive approach to evaluating the BUZZ. We present our techniques in the following sections.

3.4.2 Simulations

In designing both the Synchronous BUZZ and the Asynchronous BUZZ we started by testing our algorithms against the output of a simulator. This allowed us to do experiments under controlled conditions where ground-truths are precisely known. The simulations allowed us to control a number of different variables including:

- beacon placement and density,
- mobile device path,
- sensor noise,
- reflections, and

- occlusions

The simulator essentially mimics the action of the receiver hardware by outputting chirp reception times. A predetermined path of the mobile device is used to calculate these times based on the pattern of transmission from the simulated beacons. Gaussian sensor noise is added to the output according to a standard deviation specified at start-up. The frequency of reflections and occlusions are also specified as parameters before the simulations are run.

The path of the mobile device is simulated using a *constrained random-walk* for acceleration on all three axes. Basically, the position and velocity of the device are derived from zero-mean random Gaussian acceleration integrated over a sequence of small time windows. The seeds to the random number generator are set as a parameter in the simulator to allow for repeatable output. The random-walk is constrained in that we control the maximum position and velocity of the device. Velocity is kept within predefined limits by throttling the acceleration. In order to prevent the mobile device from going outside the boundaries of the positioning volume (e.g. walls), we reverse engineer the kinematics of the device's trajectory. For example, when the device enters a bordering space near the boundary, we reduce the acceleration so as to prevent it from leaving the boundary. When it is confirmed that the device has re-entered the core of the positioning volume, the acceleration is adjusted again so that it comes to rest at a random point near the centre of the room. From this point, the acceleration is again generated from the Gaussian random number generator.

Figure 3.7 shows the behaviour of the simulator on the *XY* axes for a 30 second trial. It can be seen that the position of the virtual mobile device slows down and reverses direction before it reaches the boundary of the space. We have found that the random-walk method generates a wide range of possible paths with varying velocities. Using acceleration to control the dynamics also makes for a smooth and realistic path.

One of the first things we discovered through simulation is how the placement of beacons matters. In general, the error in the position estimate produced by a system is related to errors in measurement. As described in Section 2.2.2, dilution of precision can be used to describe the amount that measurement errors are amplified in the estimate. For example, poor geometrical layouts of reference nodes can exaggerate errors, as can the position of the mobile device with respect to these nodes. These factors increase the dilution of

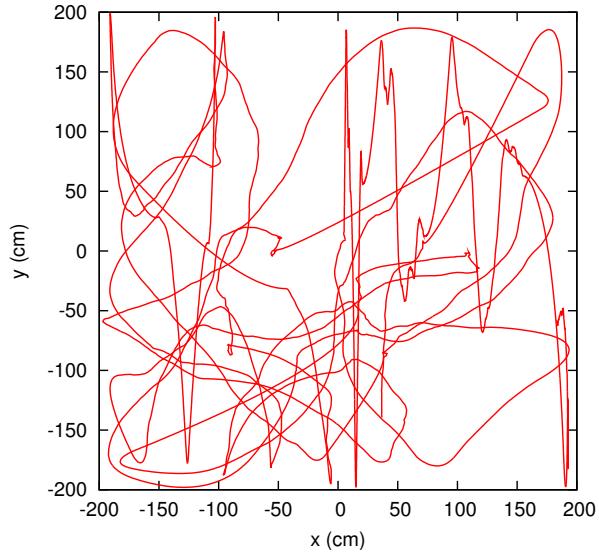


Figure 3.7: Simulator position output on the XY axes for 30 seconds. Note how the virtual mobile device slows down and changes direction before it reaches the walls.

precision such that when the measurement errors are propagated through the algorithm, they result in larger errors in the output.

For our systems, dilution of precision increases when the beacons are not sufficiently distributed in the environment. One particular example occurs when all of the beacons are placed on or near the ceiling. Such planar geometries cause the error along the vertical axis to be larger than those along the horizontal axes. In general, a wide spread of beacons along all three axes provides the best performance for the BUZZ.

The simulations have also allowed us to observe the performance of our systems in extreme conditions. For example, we have run experiments to estimate the limits of the algorithms with respect to receiver dynamics—i.e. different velocities and accelerations. Other experiments have explored the extent to which the systems can operate with varying levels of sensor noise, reflections, and occlusions.

The simulator has been written in the C programming language. It is executed from the command-line where the various system and environmental parameters are set. As a research contribution, the simulator gives a location system designer access to data that mimics the real-world movement behaviour of humans. Although it currently only

outputs measurements in the form of chirp reception times, it can be easily altered to output other measurements such as distances, pseudoranges or GPS coordinates.

We provide detail about the simulation performance of the BUZZ positioning systems in the next chapters. First, we discuss our techniques for evaluating the BUZZ in the real-world.

3.4.3 Real World Evaluation

Stationary evaluation of the BUZZ in the real-world is done in a conventional manner: estimates are collected at chosen points in the positioning space, and are then compared against the corresponding ground-truths. To evaluate dynamic performance, we have taken a novel approach that combines two different methods. The first method involves analysing the path created by a stream of position estimates. Put simply, we calculate error by comparing the path created by our algorithms to a well-defined ground-truth path. However, as this method neglects the time-based properties of the positioning error, we introduce the second method of evaluation. This method, which uses a camera to capture a ground-truth, is similar to the method suggested Section 3.4.1. It is made less complex, however, by the fact that we only use a small two-dimensional testing plane.

In the next few sections we provide specifics of each of the dynamic evaluation methods. First, however, we note that the different accuracies of the Synchronous BUZZ and the Asynchronous BUZZ call for different approaches. For example, the Synchronous BUZZ is more accurate and therefore requires a finer ground-truth and stricter analysis. To evaluate this system, we perform a stationary evaluation, an evaluation of the output path, and a camera based evaluation. The Asynchronous BUZZ, on the other hand, is only evaluated using a form of the path comparison method. We reason that the other methods do not add anything to our assessment of this system. The issue is discussed further in Chapter 5.

Path Comparison Method

The path comparison method requires that the trajectory of the mobile receiver be known. This is achieved in different ways, depending on which of the BUZZ systems is under evaluation. For the Asynchronous BUZZ, the mobile receiver is placed on top of a wheeled office chair and pushed to different points in the room. The placement of the receiver on the chair is calibrated to correspond directly with the X - Y coordinates of markers on the floor. The location of these markers are precisely known as is the height of the receiver from the floor. This information tells us the exact position of the receiver when the chair is located at each of the markers.

The setup for the Synchronous BUZZ is similar. The mobile receiver is fixed to the back of the office chair such that the diameter of the circle it traces, when the chair is spun, is known. When the positioning algorithm begins executing, the chair is moved to the locations of the markers on the floor. At each location, the chair is spun so that the receiver traverses the path of a circle centred around the marker on the floor. The procedure gives a more intricate path than the non-spinning method used with the Asynchronous BUZZ. This provides us with more detail to compare the output of the Synchronous BUZZ against.

The spinning-chair method provides three interesting points of analysis. First, even though the ground-truth is unknown while the mobile device is in-between markers, its path is precisely known when the chair is spun. Given these circular paths as ground-truths, we are able to assess the accuracy of the Synchronous BUZZ by observing the deviation of its estimates from the circles. Second, the method allows us to estimate the dynamic performance of the system in different parts of the room. We can observe the local performance of the algorithm in a single act of spinning the chair, but we can also observe performance at different locations by analysing the differences between sets of “spin” outputs. Third, since the algorithms’ motion models do not support continuous acceleration (such as that of a spinning chair), the method can be considered a worst case scenario.

The method of statistical evaluation for the Synchronous BUZZ is illustrated in Figure 3.8. The circular path of the mobile device is shown in Figure 3.8(a) where the black dots indicate the true position of the device when an estimate was produced. The estimates are shown as blue crosses while the centre of the circle—the location of the

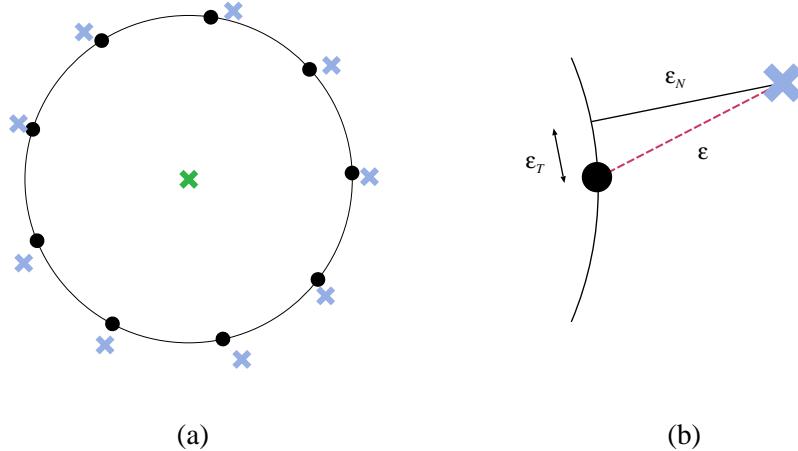


Figure 3.8: (a) Black dots indicate the true location of the mobile device as it traverses a circular path. The blue crosses are the estimates produced by the positioning system. (b) A close-up shows how the Projected Standard Deviation metric uses the normal distance to the circle ε_N rather than the distance to the actual ground-truth ε . The tangential error ε_T is assumed to be approximately equal to ε_N .

marker—is depicted by a green cross. Since we do not have access to the position of the black dots, we instead use the normal distance to the circle. This is shown in Figure 3.8(b), where ε is the true distance error and ε_N is the distance error normal to the closest point on the circle. Calculating ε_N is simply a matter of subtracting the radius of the circle from the distance between the marker and the estimated point.

In order to use ε_N to approximate ε , we use the Projected Standard Deviation method presented by Randell and Muller [109]. It estimates ε by assuming that the error tangential to the path ε_T is approximately equivalent to the normal error ε_N . This allows us to form the following approximation:

$$\varepsilon \approx \sqrt{\varepsilon_N^2 + \varepsilon_T^2} \approx \sqrt{2\varepsilon_N^2} = \sqrt{2}\varepsilon_N \quad (3.9)$$

Basically, the $\sqrt{2}$ factor compensates for any tangential error that may be present. By performing this calculation for each estimated point in the experiment, we essentially calculate an approximation of the total positioning error produced by the system.

This approximation means that the Path Comparison Method will not be able to register errors that are tangential to the path. For example, any latency in the position output will

not be accounted for in the results. Nonetheless, the technique provides us with a metric for comparing the algorithms of Synchronous BUZZ. The method can be employed in any part of the testing environment without the use of complex rigs or predefined test tracks. This flexibility is important for being able to test the systems in different parts of the room, as is described in Chapter 4.

Extending the evaluation to three dimensions involves incorporating the vertical component of the estimated position values. Since the height of the mobile device is kept constant on the top of the chair, the ground-truth along the vertical axis is known. Generally, system evaluators incorporate vertical error into evaluation results using two different types of metric. The first uses a linear representation of the CEP and DRMS errors known as Linear Error Probable (LEP) and Root Mean Square (RMS) Error, respectively. They are calculated in the same manner as their horizontal counter-parts, but use the one-dimensional distance from the horizontal plane intersecting the mobile device. Vertical LEP and RMS are stated along-side CEP and DRMS to provide a picture of the vertical and horizontal performance of the system separately. This is how performance figures for GPS receivers are stated. In general, the vertical DOP for GPS is quite high so it makes sense to keep the two-dimensional and vertical figures separate.

The second set of metrics that involve vertical error—and the ones we employ in Chapters 4 and 5—are the 3D error figures given by SEP and MRSE. They estimate the 3D distance between an estimate and the “true” position. Figure 3.9 shows how the vertical distance error ϵ_v is related to the 3D distance error ϵ_{3D} for our path comparison method. Again, we approximate this value by using the Projected Standard Deviation of the error in the horizontal plane.

$$\epsilon_{3D} = \sqrt{2\epsilon_N^2 + \epsilon_v^2} \quad (3.10)$$

Akin to the two-dimensional metric, this equation provides us with a method for approximating the three-dimensional distance to the ground-truth. With a collection of these error values, we are able to calculate both the SEP and the MRSE performance metrics.

Camera Method

To supplement the Synchronous BUZZ performance figures given by the Projected Standard Deviation method, we perform an experiment that involves using a high-speed cam-

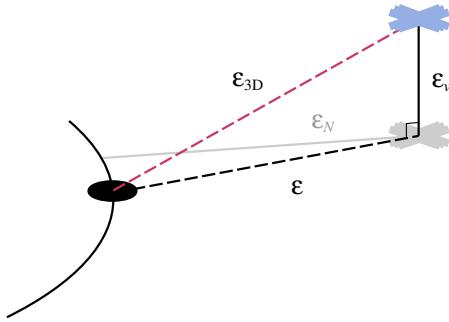


Figure 3.9: In three-dimensions, the error incorporates the known vertical error ϵ_v . The true error ϵ_{3D} is approximated using ϵ_v and the horizontal error ϵ estimated with the Projected Standard Deviation method.

era to capture the ground-truth. To make the evaluation as simple as possible, we mount the camera on the ceiling and record the position of the receiver as it moves across a calibration grid on the floor. Timing synchronisation between the receiver and the camera is achieved by capturing a monitor displaying chirp reception times within the video. Figure 3.10 shows a single frame from a video recorded with our camera. The monitor displaying the receiver’s counter values is in the bottom right corner. The real-world axis is shown in red, while the trajectory of the receiver is depicted by the green arrow.

The video is recorded at 180 frames-per-second in order to accurately capture the reception times of the 30 Hz signals. We minimise latency by connecting the receiver circuit directly to a PC instead of to a wearable computer. This means that the latency experienced by the system is only the time it takes for the receiver to transmit six bytes at 19200 baud (3.125 ms), rather than the unpredictable network latency involving WiFi or Bluetooth (which can range in the hundreds of milliseconds).

The motion of the receiver in the video is analysed using a tool created by the Motion Ripper Group at the University of Bristol, UK. This tool, similarly referred to as the *Motion Ripper*, allows a user to mark X - Y pixel coordinates for multiple features within selected frames. The marked frames, called “key” frames, are then used to interpolate the pixel coordinates for frames between. The system has been used by the Motion Ripper Group to capture 3D motion of insects and humans [47]. For our purposes, the software provides pixel representations of the receiver’s position for every frame in the video. Determining the real world values from these pixels is done by performing a

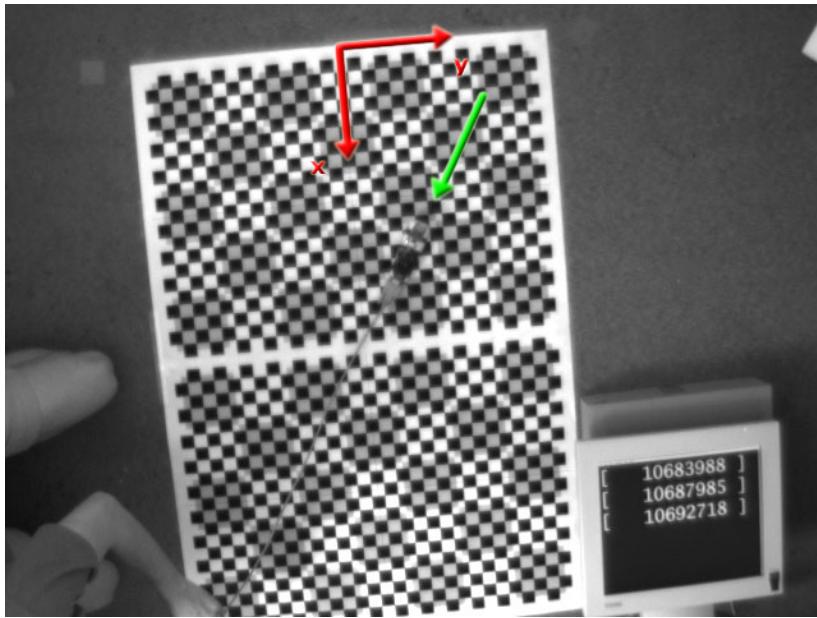


Figure 3.10: Frame of video from camera evaluation

perspective transformation using sets of pixels correlated to points on the calibration grid. This is the same method used by Criminisi et al. [27] to measure real-world planes using perspective images and is explained in further detail by Wren [23].

Extracting the timing part of the ground-truth is achieved by correlating frame numbers with the receiver counter values. This is done by recording the frame number in which each chirp reception time appears on the monitor. However, our method requires an extra processing step since the equipment we employ to display the reception times introduces a bias in the data. Specifically, the times exhibit a bursty behaviour when recorded by the camera. This behaviour is illustrated in Figure 3.11. Instead of observing a linear relationship between the X and Y axes in the plot, the figure shows that the PC outputs measurement values at a fixed rate, then catches up through large “jumps” along the X -axis.

We expect that the behaviour is a result of the way that the Linux PC buffers the reception times before they are displayed on the monitor. Specifically, the software running on the PC, which receives the data over the serial line, writes the counter values to the `stdout` output stream. The operating system then displays them within the terminal window shown on the monitor (the large fonts are achieved by adjusting the terminal properties).

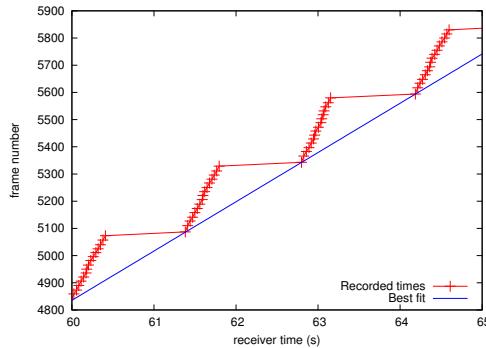


Figure 3.11: Measurement buffering

Despite the application explicitly flushing the `stdout` buffer after every print operation, the operating system buffers the display to the terminal window. One way to get around the problem is to use a more direct route to the graphics hardware. For example, it is possible to use a graphics library such as OpenGL to display the reception times.

Nonetheless, we get around the issue by determining a minimum-latency relationship between chirp reception times and frame numbers. Basically, we calculate a best fit over the burst-points on the measurement stream by making the assumption that they contain the least latency. This fit is shown as the green line in Figure 3.11. The parameters estimated by the best fit calculation—which uses data from 10000 frames—allow us to determine frame numbers from chirp reception times. With the parameters we are able to look-up the ground-truth for any time-stamped position estimate.

3.5 Conclusions

In this chapter, we have outlined the characteristics that are common to both the Synchronous BUZZ and the Asynchronous BUZZ. We have explored the issues surrounding the problem of positioning people. The process models used by the BUZZ, known as the P-model and the PV-model, have been introduced and their respective process covariance values have been derived for the extended Kalman filter.

We have justified the use of narrowband ultrasound as a medium for observing spatial relationships. The issues surrounding ultrasonic measurements include the systematic

errors caused by attenuation, transducer misalignment, and assumptions about the speed of sound. We have described the χ^2 method by which the BUZZ systems recognise and reject anomalous measurements. A set of beacon density recommendations have also been derived.

We have detailed the hardware used by both positioning systems and provided justification for the use of the SCAAT extended Kalman filter. The chapter is concluded with a description of the methods used to evaluate the BUZZ positioning systems. These methods include the use of a simulator that generates ground-truth positions using a constrained random-walk. For real data, the Path Comparison Method allows us to compare the behaviour of the algorithms against each other in different parts of the room, while the Camera Method provides a way of evaluating the time-based properties of the Synchronous BUZZ.

In the next chapter we examine the Synchronous BUZZ in depth.

Chapter 4

The Synchronous BUZZ

The design of the Synchronous BUZZ sprang from research involving an earlier ultrasonic positioning system created at the University of Bristol [108]. We discovered that we could improve upon this preceding system in two main ways. First, by eliminating radio signals from the design we are able to reduce the size and weight of the ultrasonic receiver considerably. Large RF chips and cumbersome antennae, that quite often consume more than fifty percent of the receiver configuration, are no longer required. The receiver is reduced to a single ultrasonic transducer and accompanying amplification circuitry, as described in Section 3.3.

The second improvement achieved with a radio free solution is the elimination of clock synchronisation errors. These errors are particularly evident in systems that use radio signals to indicate the start of a chirp transmission sequence, as is the case with the Synchronous BUZZ's predecessor. Unlike other systems that use radio to indicate the transmission of a single chirp [101, 128], the initial Bristol system uses it to indicate the start of a multiple-chirp transmission cycle. The system calculates range through the use of a hard-coded agreement, between the beacon control unit and the receiver, as to the length of delay between transmission of the radio signal and the subsequent chirps. We have been able to demonstrate that the actual delay used by the control unit and the delay observed by the receiver are not consistent, especially for chirps late in the sequence. The observation is a direct result of different clock frequencies on the transmitting and receiving devices. The frequency of crystal oscillators can differ by

factors of 10^{-6} to 10^{-4} [126], producing errors in the calculation of beacon-receiver distances. Additionally, some RF chips use complex error correction protocols that take a non-deterministic amount of time to complete. This adds further error to the distance calculations. By eliminating the radio synchronisation mechanism and the associated assumption of equal clock frequencies, we are able to eliminate the errors in the distance calculations.

In the following sections we provide detail into the design and implementation of the Synchronous BUZZ and evaluate it with the methods outlined in the previous chapter. The chapter concludes with a discussion of the system's performance and other considerations to its use.

4.1 Connected Infrastructure

The Synchronous BUZZ operates by exploiting a periodic chirp transmission pattern governed by the beacon control unit in the infrastructure. This pattern allows the mobile device to model the unknown transmission times that are necessary for calculating position.

The transmission control unit consists of a PIC microcontroller and signal amplification circuitry. Beacon transducers placed on the walls and ceiling of the environment are wired to the control unit. The cost of this transmitting system is around \$150 USD [108].

The PIC on the control unit is programmed to activate each beacon in a cyclic, round-robin fashion. The time period between each activation is constant such that the transmission time of each beacon can be modelled by the following equation.

$$E_k = E_{k-1} + P_{k-1} \quad (4.1)$$

Here, E_k represents the transmission time of the k th chirp and P is the time between transmissions. The chirp reference k begins incrementing from the first chirp that the receiver receives and, since we use SCAAT, it is also synonymous with the time-step of the Kalman filter. The beacon that transmits chirp k can be calculated by $k \bmod N$,

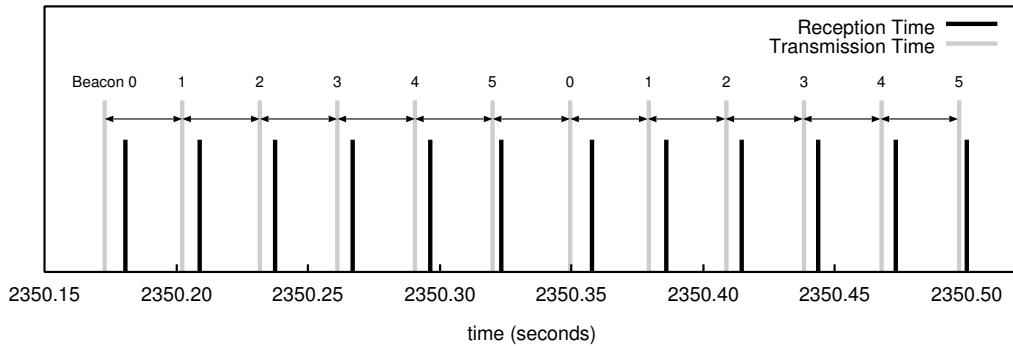


Figure 4.1: The transmission and reception times of a six beacon system over two cycles of a positioning trial. The time between the reception time and the transmission time is the distance-related time-of-flight.

where N is the number of beacons. The two variables E and P completely model the behaviour of the transmitting system.

Essentially, the mobile device performs positioning by observing the deviation from the repeating transmission pattern. The reception time of the chirps with respect to their transmission times is illustrated in Figure 4.1. This visualisation shows two transmission cycles of a six beacon system. The fact that the transmissions are equally spaced is what allows the algorithm to infer the transmission times from one observation to the next using Equation 4.1. These times are also modelled with respect to the local clock on the receiving device, which means that the algorithm automatically compensates for drift and skew from the clock on the control unit. For instance, in this trial the control unit was programmed with a 30 ms transmission period P . At the time shown in the figure, the receiver estimated P to be 29.500901117 ms, a 1.7% difference.

Each time that the infrastructure is erected it must be configured so that the position of each beacon is precisely known. For the experiments in this chapter as well as the next, each beacon position is measured by hand using measuring tapes and plumb bobs. While there are methods for performing this type of calibration automatically for range based systems [34, 114], they are not currently suitable for the type of systems we explore here.

In the next section we look at how the transmission model is combined with position state variables to form the positioning algorithm.

4.2 Positioning Algorithm

4.2.1 Kalman filter Parameters

The positioning algorithm uses an extended Kalman filter to perform the task of stochastically merging observations of chirp reception times with a model of the underlying process. Recall from Section 2.4 that the measurement equation for our filter must express the measurements as a function of the state. To do this, we first express the chirp reception time in an equation involving time-of-flight.

$$R_k = E_k + t_k$$

Here, R_k is the chirp reception time, E_k is the transmission time as defined in the previous section, and t_k is the time of flight for chirp k . We then employ the ranging model described in Section 2.2.2 to introduce an expression for t_k using the position of the mobile device \mathbf{X} and the position of the originating beacon \mathbf{T}_i ($i = k \bmod N$). The measurement equation becomes:

$$R_k = E_k + \frac{\|\mathbf{X}_k - \mathbf{T}_i\|}{v_s} \quad (4.2)$$

where v_s is the speed of sound. Equation 4.2 is the general measurement equation for our algorithm. It is used by the extended Kalman filter to predict the current measurement R_k using the state variables E_k and \mathbf{X}_k .

The process equations for the transmitter related variables have already been partially expressed by Equation 4.1. We must also add an equation to tell the algorithm that the transmission period is constant:

$$P_k = P_{k-1} \quad (4.3)$$

Together, Equation 4.3 and Equation 4.1 completely describe the time varying behaviour of the transmitting process.

As discussed in Section 3.1, there are two different ways of formulating the process equations for the dynamics of the mobile device. The first form of the equations assumes that the process noise is attributed to the velocity of the filter and therefore predicts that the position of the mobile device does not move between measurements. This is the

P	chirp transmission period
E	chirp transmission time
\mathbf{X}	3D position vector
\mathbf{V}	3D velocity vector (PV-model only)

Table 4.1: Variables comprising the state vector \mathbf{x} , the unknown / estimated values in the system. The subscript k is used to denote the current measurement and iteration.

Position Model or the P-model:

$$\mathbf{X}_k = \mathbf{X}_{k-1} \quad (4.4)$$

The other form of the equations assumes that the process noise is attributed to acceleration. Modelling the dynamics in this way requires the introduction of variables describing the velocity of the mobile device \mathbf{V} . With these variables added to the state vector the process equations form the basis for the Position-Velocity Model or PV-model:

$$\begin{aligned} \mathbf{V}_k &= \mathbf{V}_{k-1} \\ \mathbf{X}_k &= \mathbf{X}_{k-1} + \Delta t \mathbf{V}_{k-1} \end{aligned} \quad (4.5)$$

Basically, the PV-model uses the time between measurements Δt to predict the current position of the device given its currently estimated velocity. In general, the PV-model provides for smoother positioning output than the position only model, but is prone to instability if the estimate for velocity diverges. We show the experimental performance of both of these process models in Sections 4.3 and 4.4.

Table 4.1 summarises the variables in the state vector \mathbf{x} of the extended Kalman filter. Depending on whether the Position model or the Position-Velocity model is used, this vector will contain either five or eight variables. The process and measurement equations for the filter are summarised in Tables 4.2 and 4.3. As per our discussion in Section 3.3, the extended Kalman filter uses these equations to perform the following steps:

- Given the reception time of the current chirp k , the filter predicts the current state \mathbf{x} , and calculates the process noise covariance \mathbf{Q} using the elapsed time Δt .
- The predicted state is plugged into the measurement equation to calculate the predicted reception time \hat{R}_k for the current chirp.
- The χ^2 metric is calculated using the measurement residual. This is the difference between the measurement and the prediction $(R_k - \hat{R}_k)$.

Process Equations	Measurement Equations
$P_k = P_{k-1}$ $E_k = E_{k-1} + P_{k-1}$ $\mathbf{X}_k = \mathbf{X}_{k-1}$	$R_k = E_k + \frac{\ \mathbf{X}_k - \mathbf{T}_i\ }{v_s}$

Table 4.2: Position model equations

Process Equations	Measurement Equations
$P_k = P_{k-1}$ $E_k = E_{k-1} + P_{k-1}$ $\mathbf{V}_k = \mathbf{V}_{k-1}$ $\mathbf{X}_k = \mathbf{X}_{k-1} + \Delta t \mathbf{V}_{k-1}$	$R_k = E_k + \frac{\ \mathbf{X}_k - \mathbf{T}_i\ }{v_s}$

Table 4.3: Position-velocity model equations

- If the χ^2 value is within the acceptance threshold the measurement is integrated with the filter

When the integration of the measurement is complete, the filter waits for the next incoming chirp and performs the steps again.

4.2.2 Getting a Fix: Initialisation and Stability Monitoring

In order to properly process the measurements, the source of the chirps, i.e. the beacon numbers, must be determined. Because our implementation does not provide a simple way of identifying this directly from the measurements, we use the characteristics of the Kalman filter to perform the task. This is carried out in an initialisation stage of the filter, which we explain below.

The Synchronous BUZZ positioning algorithm is implemented using a general state machine framework, as is shown in Figure 4.2. The first state the algorithm enters on start-up is an initialisation state called *Find Rotation* where it attempts to lock on to the sources of the incoming chirps (akin to the fix-acquisition phase of a GPS receiver). To do this, N (the number of beacons) extended Kalman filters are executed in parallel. Each is assigned a particular rotation number that indexes a list of beacons sorted by the order of their activation. For example, filter 0 assumes that the current chirp

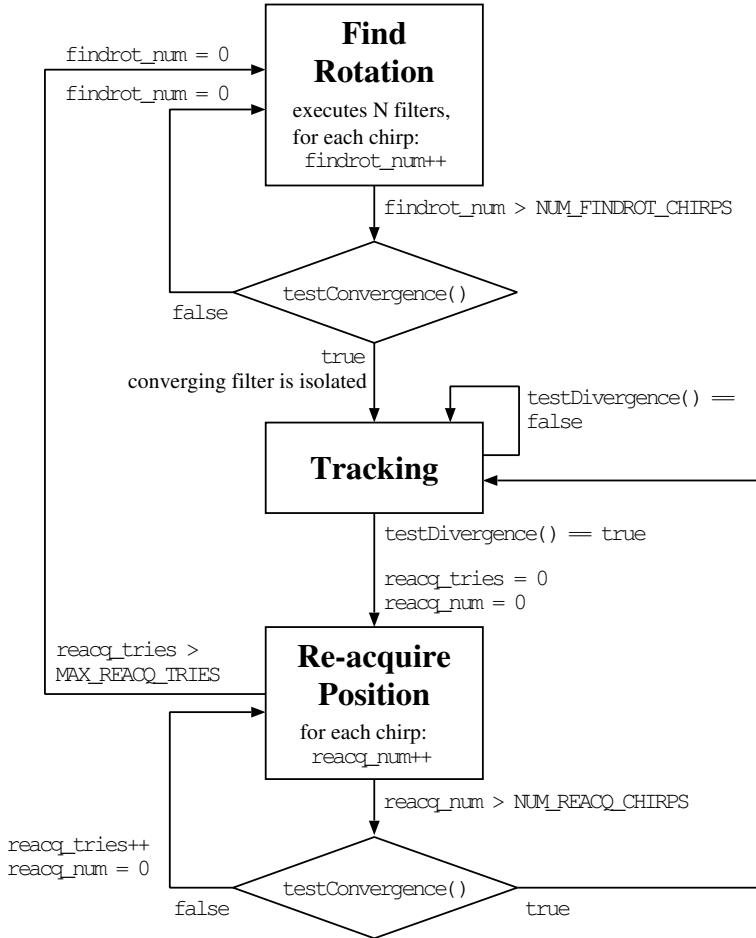


Figure 4.2: State machine framework for the Synchronous BUZZ. The *Find Position* state is where the algorithm isolates the filter that is correctly indexing the beacons. This is done by monitoring the system covariance of each filter as well as their average χ^2 values.

originated from beacon 0, filter 1 assumes it originated from beacon 1, and so on. By monitoring the system covariance matrix \mathbf{P} and the average χ^2 values for each filter using the `testConvergence()` function, the algorithm is able to establish which filter is indexing this list correctly.

Basically, after some initial start-up time, the winning filter converges and its system covariance and χ^2 values decrease and stabilise. The rest of the filters diverge such that their covariance and χ^2 values become increasing large. The `testConvergence()` function merely identifies the filter which has system covariance and average χ^2 val-

ues below experimentally determined thresholds. When it has done this, the algorithm declares the converging filter as the winner, discards the other filters, and enters the *Tracking* state.

Solving for the correct beacon index in this way requires that the chirp transmission pattern be geometrically asymmetric. This means that there should only be one unique sequence of chirps in the beacon list for a particular pattern and layout. For example, if the beacons are arranged in a rectangular geometry and they transmit in a circular fashion, then there will be multiple solutions for position where the observed distance to the beacons is consistent. Essentially, such a set-up would cause mirrored positions corresponding to each of the possible solutions. Under these conditions there is no way to determine which filter is the correct one.

The fact that the Kalman filter is an iterative estimator also means that the state variables must be seeded with some reasonable values before the algorithm begins executing. In practice, the mobile device is held in a specified location at start-up until the algorithm gets a position fix and has determined the correct beacon index. The transmission period must also be seeded with a reasonable value. This is stored in a configuration file with the beacon positions and is read by the algorithm at start-up. In practice, the algorithm takes roughly 5 to 10 seconds to successfully enter the *Tracking* state.

In the *Tracking* state, the algorithm continues to monitor itself using the system covariance \mathbf{P} and the χ^2 values through the `testDivergence()` function. This function also looks for long periods of missing chirps and systematically occluded beacons, which occur when the filter has lost its fix. If any of these measures indicate a level of sufficient uncertainty, the algorithm abandons its current fix on the position variables \mathbf{X} (and \mathbf{V} if the PV-model is used) and re-initialises their values and confidences to the last known “good” position. This is where the algorithm enters the *Re-acquire Position* state. If the filter cannot re-establish a lock on position within a certain number of tries, the winning filter is abandoned and the algorithm starts again from the *Find Rotation* state. This fix re-acquisition method has proven effective for recovering from the effects of interfering noise and signal occlusion.

In the next sections we detail and discuss a number of different experiments. We start by looking at the results of some simulations.

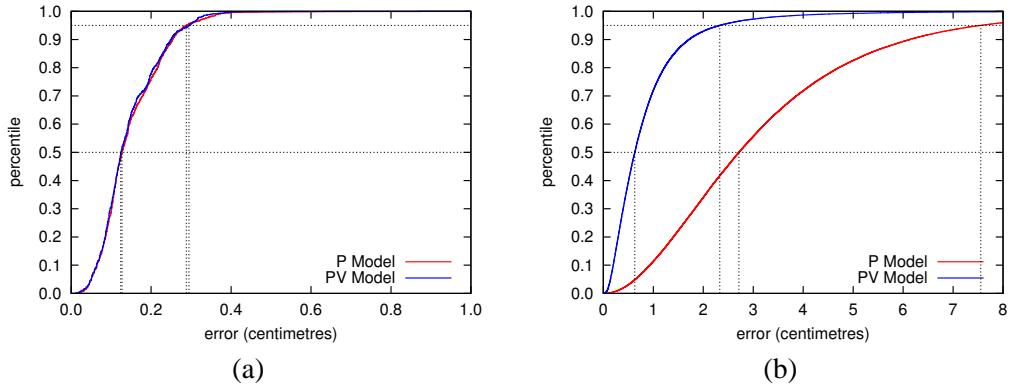


Figure 4.3: 3D error CDFs using ideal data from (a) some stationary simulations, and (b) a dynamic simulation.

	Stationary		Dynamic	
	P	PV	P	PV
SEP (50,95)	0.13	0.29	0.12	0.29
Horiz. CEP (50,95)	0.09	0.18	0.09	0.18
Vert. LEP (50,95)	0.08	0.25	0.08	0.25
MRSE	0.13	0.13	3.94	1.27
Horiz. DRMS	0.06	0.06	2.61	0.76
Vert. RMS	0.12	0.11	2.95	1.01

Table 4.4: Simulation results for ideal data (values in centimetres)

4.3 Evaluation through Simulation

We have performed a number of simulations to gauge the performance of our system under various conditions. Figure 4.3 and Table 4.4 summarise results obtained using ideal simulation data: zero sensor noise, no reflections or occlusions, and an ideal transmitter configuration. The period of chirp transmission is 30 ms. The experiment serves as a benchmark against which we compare the results given in the remainder of the thesis. Specifically, the output from this experiment is the best performance that can be achieved by our system.

Figure 4.3(a) shows three CDFs highlighting the 50% and 95% SEP values for 64 stationary trials. The position of the mobile device for each trial was varied such that all of the $(4 \times 4 \times 2.5)$ m³ positioning volume was covered. Each trial represents 5 minutes of position data. Figure 4.3(b) gives the 3D CDF for a 60 minute dynamic trial with

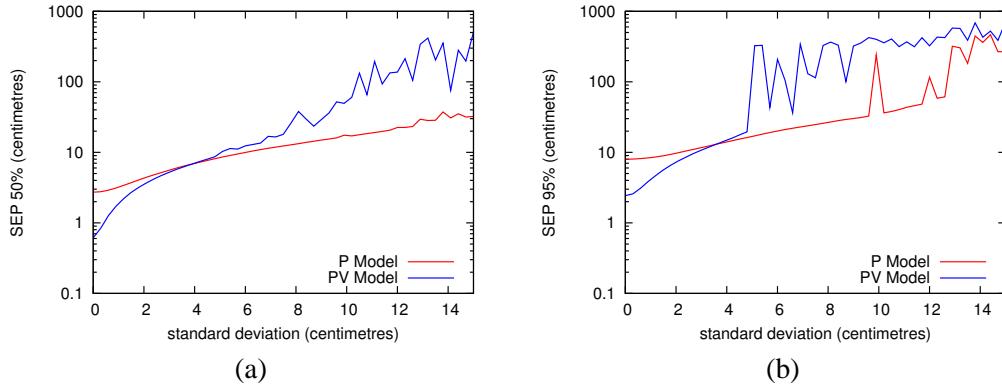


Figure 4.4: (a) 50% and (b) 95% SEP versus sensor error standard deviation.

the path generated using the random walk method described in Section 3.4. The standard deviation of acceleration for this trial was 9 ms^{-2} on each axis. The average and maximum speeds achieved were 0.56 ms^{-1} and 1.6 ms^{-1} , respectively.

In the following subsections, we examine the effect that various operating conditions have on the performance of the Synchronous BUZZ.

4.3.1 Sensor Noise

Figure 4.4 plots sensor noise against the three-dimensional dynamic performance of our two models. The diagram shows the results from 51 simulations, each using a different value for the standard deviation of the Gaussian sensor noise. The X-axis of the plot gives the standard deviation in centimetres, a distance measure. However, sensor noise is experienced by the receiver in terms of timing errors, which the simulator calculates by dividing by the speed of sound. We choose to represent sensor noise in terms of distance in order to make it easier to conceptualise.

The eight beacons used in the simulations are configured as per the previous ideal experiment (Figure 4.3). For each simulation, the same 10 minute random walk for the mobile device is used. As with the ideal simulation, the standard deviation of the acceleration was 9 ms^{-2} . The average and maximum speeds achieved were 0.54 ms^{-1} and 1.6 ms^{-1} , respectively.

It is apparent from Figure 4.4 that the Position-Velocity model is more accurate than the Position model at lower noise amplitudes. However, at standard deviations of greater than around 3 cm, the P-filter begins to outperform the PV-filter. Since velocity errors are accumulated in the estimate for position, increased sensor noise results in scenarios where the PV Kalman filter diverges and position effectively “runs away”. In these situations, the algorithm eventually enters the recovery state described in Section 4.2.2. It remains in this state until it can re-acquire a fix on position.

We stress that the nature of the sensor errors observed in practice is different than the Gaussian errors produced by our simulation. In the real-world, errors happen systematically as a result of the varying orientation and range of the mobile device with respect to the beacons (see Section 3.2.1). Nonetheless, the results of this simulation give us an indication of how each of the algorithms respond when they are faced with circumstances where there are larger sensor errors.

To keep the experiments uniform, we use a standard deviation of 1 cm for the sensor noise in the remaining simulations. This value is a conservative estimate for the sensor noise measured in our lab. In ideal conditions with no interference, the measured sensor noise standard deviation is approximately 2 mm. This corresponds to the resolution of the clock on the receiver, which samples the signal at a frequency of 156250 Hz.

4.3.2 Occlusions and Number of Beacons

Here we explore how the number of beacons placed in the infrastructure affects performance. Recall from Section 2.2.2 that at least four reference nodes are required in order to solve for position in a three-dimensional pseudoranging system. Despite the fact that the Synchronous BUZZ uses two variables to model time within the system (rather than just one as is the case with GPS), it is still essentially a pseudoranging system. Specifically, it solves for position as well as a timing offset given by the difference between the clocks on the receiving and transmitting systems. As a result, the Synchronous BUZZ requires measurements from a minimum of four beacons to operate.

To test the performance with respect to the number of beacons, we simulated a number of dynamic trials where the quantity of beacons and the probability of occlusion for each beacon were varied. Figure 4.5 shows the 50% SEP results for the (a) P-filter and (b)

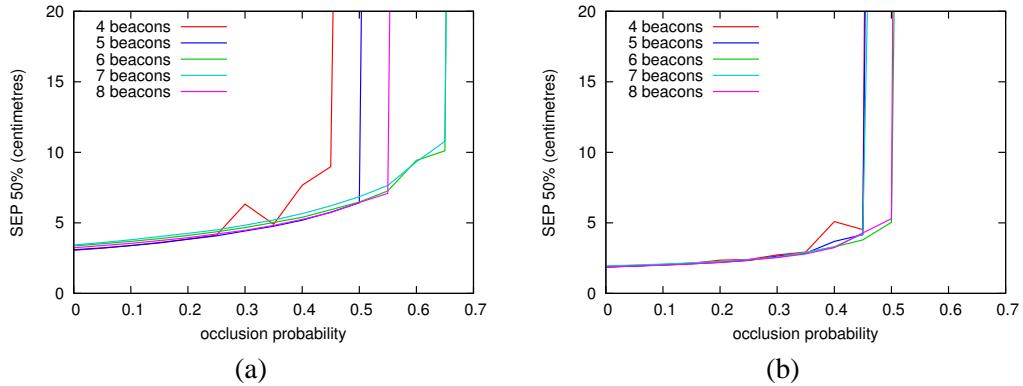


Figure 4.5: The effect of occlusions on 50% SEP for the (a) P-filter and the (b) PV-filter.

PV-filter. In these simulations, the path of the mobile device was consistent for each 10 minute trial, with the same movement as in the sensor noise experiment. We observe that as the probability of occlusion increases for each of the beacon configurations, there is a point where there are not enough measurements available to solve the problem. As a result, the 4-beacon configuration diverges at lower probabilities than the other set-ups. For configurations greater than four, we observe that the PV-filter is less able to handle the occlusions compared with the P-filter for the same reason that it performs poorly in the face of large sensor noise. That is, errors in velocity, this time as a result of a lack of measurements, cause the position estimate to diverge.

The occlusions in this simulation are introduced randomly on a per-beacon basis. In the real-world, however, occlusions are more systematic. For example, an object in the environment will occlude one or more beacons for a sustained period of time rather than flickering in and out randomly as is the case with our simple simulation. A more sophisticated simulator could imitate reality more closely by modelling the size, position and number of objects, but is beyond the scope of this thesis.

In general, each occluding object effectively removes one or more constraints from the system, meaning that the more beacons there are, the better the system is able to cope with occlusions. This also applies in situations where beacons are out of range or the mobile device is outside of the beam envelope. For the systems that we have erected to-date, we have used between six and eight beacons. The experiments in the remainder of this thesis use eight.

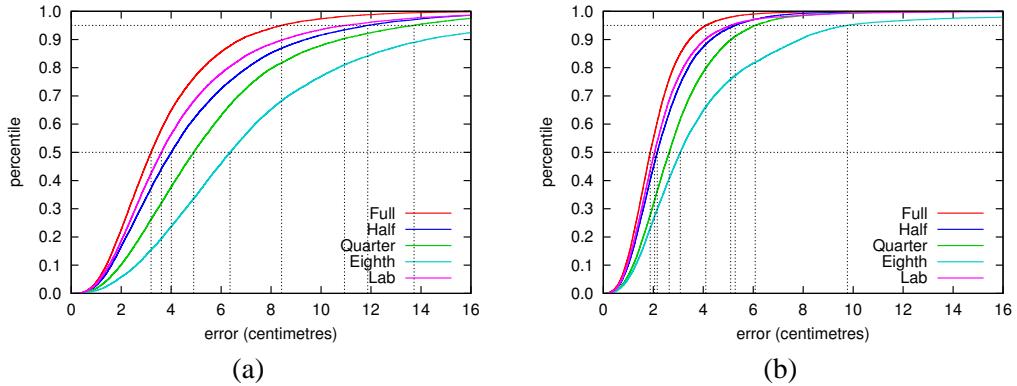


Figure 4.6: The performance of the (a) P-filter (b) PV-filter for different beacon configurations.

4.3.3 Beacon Placement

To examine how the dilution of precision affects the performance of the system, we ran simulations using different beacon configurations. Figure 4.6 shows the results of five different eight-beacon geometries. These geometries are explained below.

Full: This configuration covers the entire positioning space. The beacons are well spread on all three axes.

Half: In this configuration only half of the space is covered by the beacons. They are well spread in the Y and Z axes but are compressed along the X -axis.

Quarter: This geometry covers only one quarter of the full volume. The beacons are compressed along the X and Y axes while the Z -axis has coverage equivalent to the full configuration.

Eighth: In this configuration only one-eighth of the full positioning volume contains beacons. The beacons are compressed along all three axes.

Lab: This is the configuration used in our lab and in the real-world experiments of the section following.

The mobile receiver traced the same 10 minute path in each of the simulations. Figure 4.6(a) gives the results for the P-filter and Figure 4.6(b) provides the PV-filter results.

As predicted, the performance of our algorithms decreases when poor geometries are used. We observe that the Lab geometry is better than that of the Half geometry but not as good as the Full geometry. Although we have attempted to place the beacons in our lab at ideal locations, the Full geometry has a higher virtual “ceiling” meaning that the DOP along the Z-axis is lower and therefore produces better results.

4.4 Real World Evaluation

In this section we evaluate the performance of the Synchronous BUZZ in the real-world. We perform three different types of experiment to quantify the accuracy and precision of the system. The first experiments are the stationary experiments where the mobile device is placed at known locations in the positioning volume for extended lengths of time. The second experiments compare the output of the algorithm against a known path traced out by the device. In the third experiment, we analyse the accuracy of the system using a camera to record the path of the mobile device. The period of beacon activation for all of the experiments is 30 ms.

4.4.1 Stationary Performance

To explore the stationary performance of the system we placed the mobile receiver at 18 different positions in the lab. Nine of these positions were on the floor and nine were placed at a height of 1.06 m. The positions were chosen in order to adequately cover the horizontal XY plane. The ground-truth for each location was measured by hand and refined using a trilateration algorithm. Between 10 to 15 minutes of chirp reception times were recorded at each position.

The combined 3D error CDF for the experiment is shown in Figure 4.7. We can see that the performance of the P- and PV-filters is comparable, as predicted by the stationary simulations. We also notice that the CDF curves have slight ripples in them. This feature exists as a result of placing the receiver at discontinuous locations in the room. Specifically, if we examine the results from each location individually, we observe that the positioning error changes as a function of the receiver’s location.

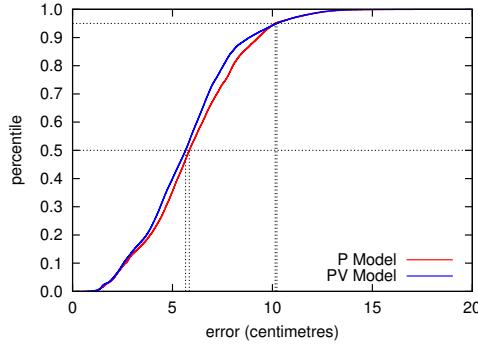


Figure 4.7: 3D error CDF (SEP) for 18 10-minute stationary trials

Figure 4.8 highlights the influence of receiver location by showing positioning results for two different points in the room. The plots on the left give position estimates at a point near the centre, while the plots on the right give estimates for a point near one of the walls. By comparing the CDF curves with their corresponding estimate clouds, we observe that the 50% SEP values provide a measure for the systematic error in the system (accuracy), while the slope of the curve provides an indication of the variance within the estimate clouds (precision). The former measure can also be estimated by taking the difference between the 95% and 50% values.

In terms of the effects of location, we notice that the spread of the estimates, as well as their bias from the ground-truth (shown as the circled dot), increases for points on the periphery. The CDFs reflect this as well: the curve for the location closest to the wall is shifted further to the right and its slope is more shallow than the point near the centre of the room.

We attribute this behaviour to an increase in both dilution of precision and sensor error. At the outermost locations, the dilution of precision is higher in the XY plane, meaning that sensor noise contributes more to the positioning error. In addition, since beacon-receiver ranges are greater and transducer misalignment angles are wider, the systematic errors present in the measured reception times are more pronounced (see Section 3.2.1).

We also observe that, at locations on the periphery, occlusions (which happened occasionally during the trial due to people walking through the room) have a more adverse effect on the output of the filters. Their influence is depicted by the estimates situated

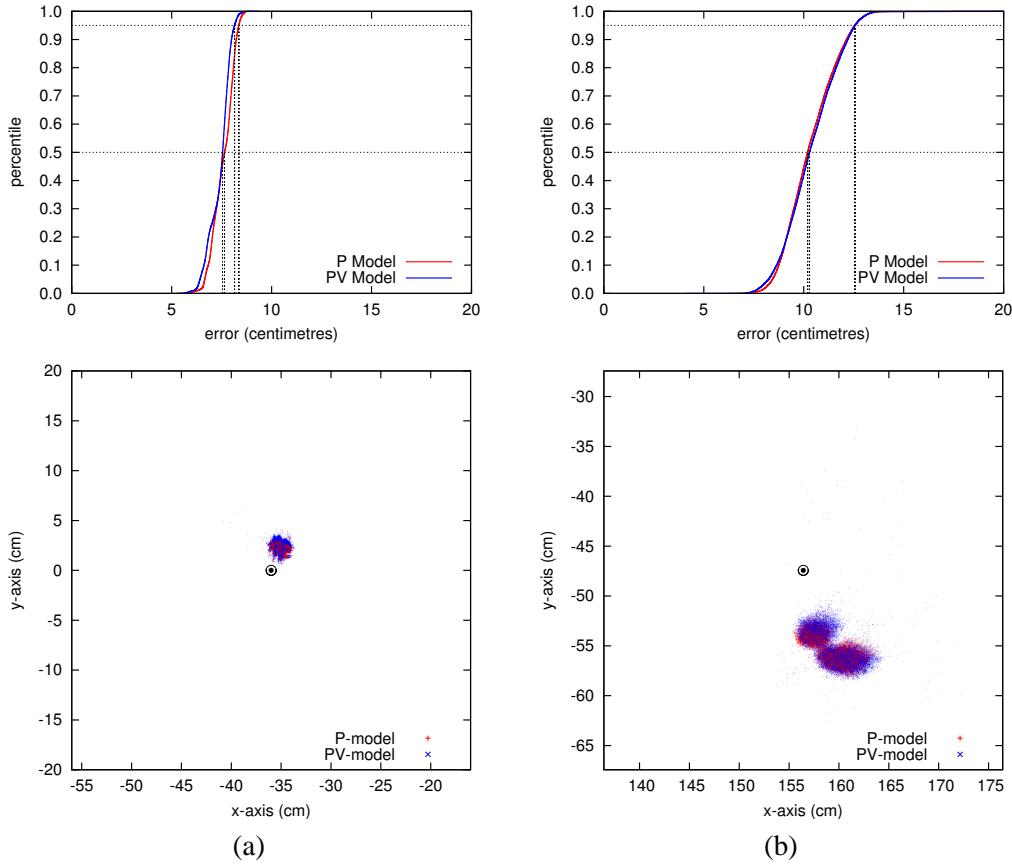


Figure 4.8: Stationary performance at two locations: (a) near the centre of the room, and (b) near a wall. At both points the mobile device was resting on the floor.

furthest from the main clusters in Figure 4.8. We can see that the error in these outlying estimates is greater for the receiver location on the outskirts of the positioning volume.

We explain the differing effects of occlusions in terms of the mechanics of the Kalman filter. The time-division nature of chirp transmission means that if a chirp is occluded, the system must wait for the next beacon to transmit before it is able to input a measurement. Within this time, the uncertainty in the state grows according to the parameters of the process covariance. This causes two things to happen. First, measurements that arrive after the occlusion will be more heavily weighted and, second, the χ^2 test will become more relaxed. The result is that not only will measurements with larger error have a greater probability of being accepted, but they will also have a greater effect on the state. The consequence is that occlusions cause the state to jump away from the correct position until the system covariance stabilises again.

4.4.2 Path Comparison Analysis

In order to gauge the performance of the positioning system while the mobile device is moving, we describe two different experiments using the path comparison method outlined in Section 3.4. During the first experiment, we moved the swivel chair to points in the room and spun the back slowly on its base. The PV-filter estimates that the speed of the receiver averaged 0.31 ms^{-1} and had a maximum speed of 0.69 ms^{-1} . In the second experiment, we executed the same movements but this time we spun the chair-back at a higher speed. The PV-filter estimates the average and maximum speeds to be 0.59 ms^{-1} and 1.72 ms^{-1} , respectively.

Like the stationary experiments, we examine two different points in the room to isolate the effect of the mobile device's location on performance. Figure 4.9 shows the results from both the low- and high-speed experiments at a point near the centre of the room. The plots on the bottom row show the paths that the algorithms estimate during the trial, with the black circle indicating the ground-truth. The plots in the top row give their corresponding Projected Standard Deviation error CDFs.

We make two main observations about these results. The first observation concerns the difference between what our simulations predict about the relative performances of the P- and PV-filters, and the results we see here. Namely, we notice that the models perform similarly in these experiments while the dynamic simulation results, shown previously in Figure 4.3(b), predict that their differences should be greater. We postulate that the lack of difference is a result of our evaluation method. Since the Projected Standard Deviation Method assumes that the error tangent to the path is equivalent to the normal error, any large tangential errors will be minimised while smaller tangential errors are exaggerated. We believe that the PV-filter actually produces smaller tangential errors than the P-filter for our dynamic trials, but that our evaluation method is unable to register this difference.

The second observation is taken by examining the difference between the low-speed results, shown in Figure 4.9(a), and the high-speed results, shown in Figure 4.9(b). What we see is that the performance of the P-filter is almost unaffected between the two trials. The performance of the PV-filter, on the other hand, degrades noticeably in the high-speed trial. We attribute this result to the path followed by the receiver. Specifically, the tight circles that the mobile device traverses as it is spun on the back of the office chair, causes it to accelerate constantly. For the P-filter, this is not much of an issue

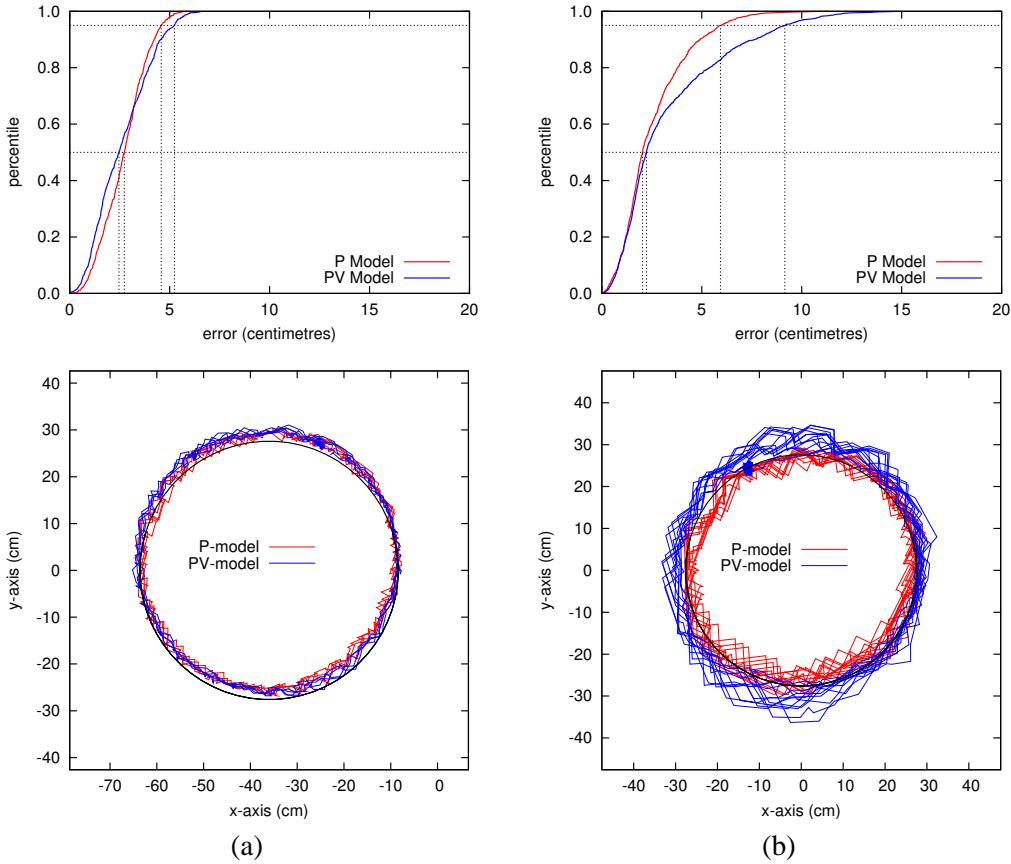


Figure 4.9: Spinning chair performance near the centre of the room. Two different experiments are shown: (a) low speed (b) high speed

since the process covariance has been tuned with knowledge of the fact that the receiver is mobile. In other words, the systematic process error produced through movement has been compensated for by placing higher weight on the measurements and less weight on the state. This is what allows the algorithm to “move” despite the fact that the model continuously predicts that the receiver is stationary.

The configuration is different for the PV-filter. By including velocity in the process equations, the PV-filter is able to more closely approximate the dynamics of the mobile device. The resulting increase in confidence for the time-based predictions mean the Kalman filter is effectively tuned to give the state a higher weight. The net effect is that it is more difficult for good measurements to bring the PV-filter back once it has lost its fix on position. We see this occur in times of high sensor noise—for example, in our sensor noise simulations (Figure 4.4) and at receiver locations near the outside of the

positioning volume (the large PV-filter outliers in Figure 4.10(b))—and at times where the assumptions of the receiver’s dynamics do not match its true motion.

The second situation is highlighted by our experiments here. In particular, the PV-filter expects a *zero-mean* acceleration of the mobile device but is instead faced with a continuous, non-zero acceleration. The position errors that arise because of the PV-filter’s assumption are especially pronounced during the high-speed trial where systematic sensor errors are large. Basically, the PV-filter’s “over confidence” in the state is what causes it to go off track, while the P-filter is more easily corrected by subsequent measurements. The CDF plots reflect this behaviour by showing that the relative performances of the algorithms at different percentiles are exaggerated. Starting with the stationary results and subsequently observing the low- then high-speed results, we see that the gaps between the P-filter and PV-filter curves increase. Specifically, the PV-filter performs better at lower percentiles where the estimates contain low systematic error (the receiver is travelling with a constant velocity, for example). In contrast, the P-filter performs better at higher percentiles where all of the measurements, including those with high systematic errors, are included.

To illustrate how the location of the receiver affects the performance of the system during the dynamic trials, we provide results from a point close to a wall. These are shown in Figure 4.10. As with the stationary experiments, we notice that the error is larger at this location. Specifically, we observe a distinct systematic offset in the output, where the paths created by the algorithms are shifted lower than the ground-truth by about 6 cm. Again, we attribute the behaviour to increased DOP as well as increased systematic sensor noise. The beacons furthest from the receiver, on the other side of the room, have larger misalignment angles in this experiment. This creates a bias in the measurements, which effectively pushes the positioning results away from the ground-truth.

Like the results from the point in the centre of the room, we observe a similar increase in error for the faster moving receiver. Again, the CDF plots illustrate that the P-filter fairs better at higher percentiles, while the PV-filter performs better at lower percentiles.

Figure 4.11 provides the cumulative performance for the dynamic experiment over all parts of the room. Table 4.5 summarises the stationary as well as the combined low-speed and high-speed dynamic positioning results. The same relationship between the P- and PV-filter at different percentiles is observed.

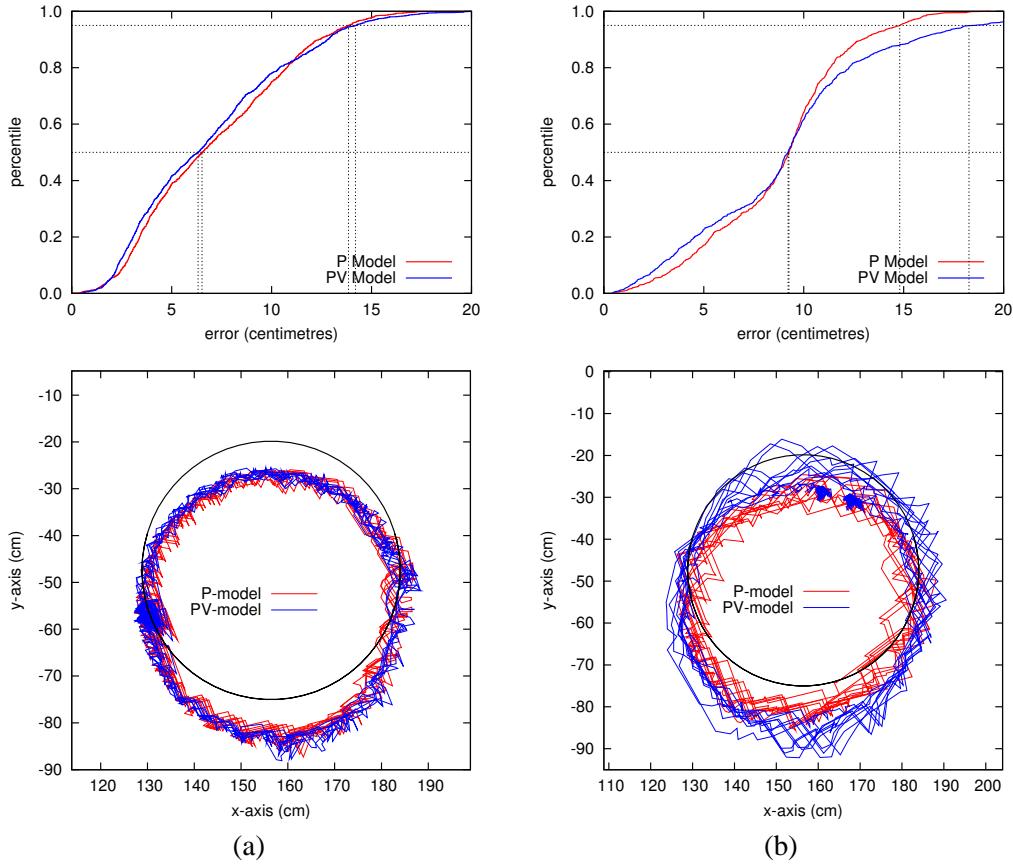


Figure 4.10: Dynamic positioning results for an outermost location. (a) Low speed. (b) High speed.

	Stationary		Dynamic	
	P	PV	P	PV
SEP (50,95)	5.84	10.14	5.65	10.23
Horiz. CEP (50,95)	3.64	9.48	3.58	9.60
Vert. LEP (50,95)	3.93	7.48	3.55	6.92
MRSE	2.73	2.58	2.19	2.10
Horiz. DRMS	2.18	2.13	1.92	1.92
Vert. RMS	1.65	1.46	1.04	0.85

Table 4.5: Performance figures for the Synchronous BUZZ in centimetres. The error calculated for the dynamic trials is done using the Projected Standard Deviation method.

From the evidence provided by the simulations as well as the results at lower percentile values in the CDFs, we expect the PV-filter to outperform the P-filter when the receiver

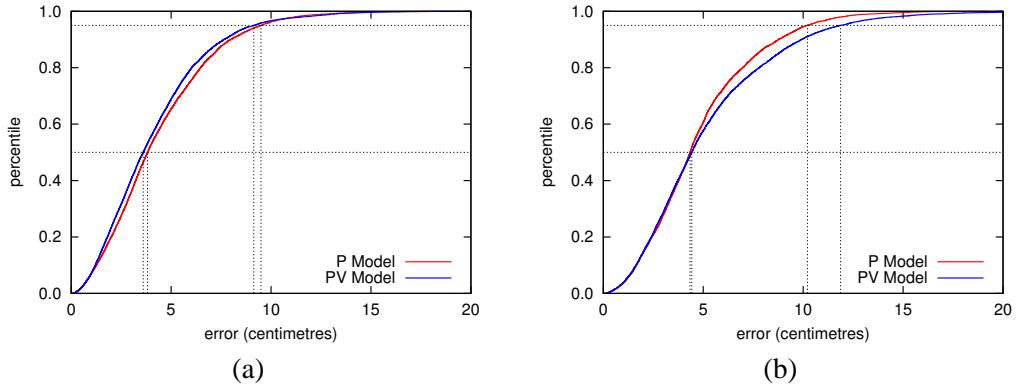


Figure 4.11: Total dynamic performance CDFs for the (a) low speed trial and (b) the high speed trial

is moving with a constant velocity. We examine this hypothesis further in our third experiment.

4.4.3 Video Ground-truth

In this experiment we use the camera method described in Section 3.4.3 to evaluate the algorithms' performance with respect to position and time. The receiver is pulled across the calibration grid for a distance of roughly 1.2 m in eight separate movements. We attempted to vary the speed of the device between movements while at the same time keeping it constant for each movement. Figure 4.12 shows the 3D positioning error and the instantaneous ground-truth speed for four of the trials.

We note that the error of both the P- and PV-filters is proportional to the receiver's speed. We also observe that for the sections of the movement where speed is high but relatively constant, the PV-filter performs better than the P-filter. This is attributed once again to the different ways in which the two models weight the measurements and the state. For the P-filter, we see that the measurements have more of an impact on the output, shown by larger peaks and valleys on the curves. As the receiver moves, the system covariance increases due to larger discrepancies between the state and the measurements. This causes the Kalman filter to weight the measurements higher such that the errors introduced by each measurement make an increasing impact on the estimate. The mea-

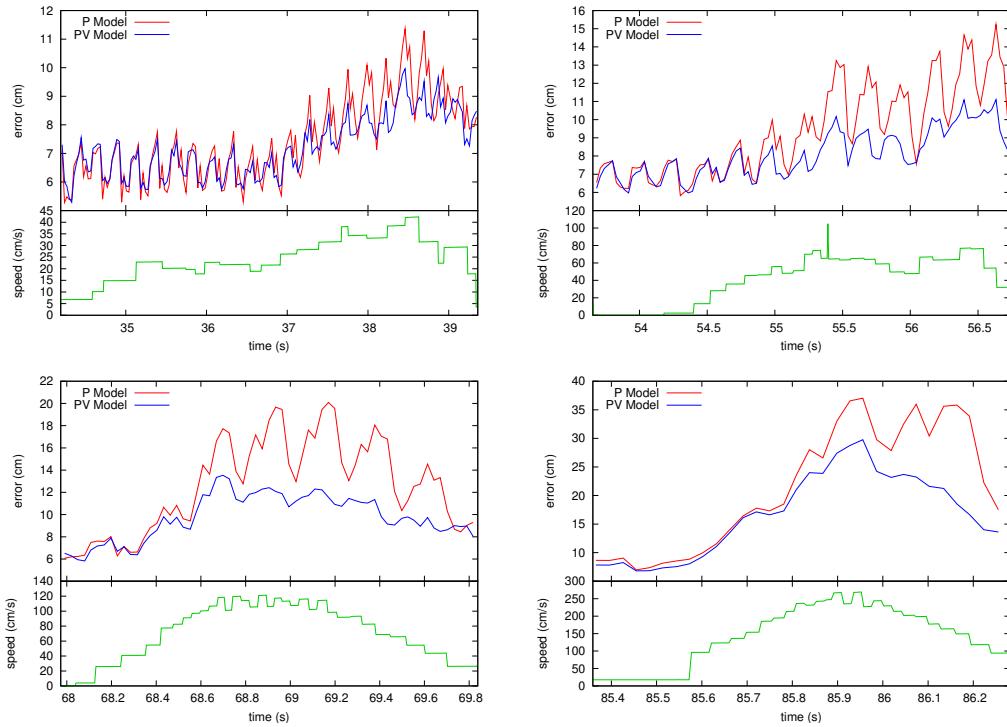


Figure 4.12: 3D error and speed versus time for four movements across the calibration grid.

surement error from each beacon can be seen in the repetitive eight-point pattern on the P-filter error curves.

While the error curves of the PV-filter have a similar pattern, the PV-filter places a higher weight in the state such that the overall error is more consistent and the curves are smoother. Unlike the previous experiment, the motion of the receiver is more in-line with the PV-filter’s assumptions. In other words, the acceleration of the device in this experiment is closer to zero throughout, allowing the PV-filter to better predict the state between measurements and overtake the P-filter in its performance.

To summarise the results of this experiment, we provide the CDF for all eight movements in Figure 4.13. As predicted, we observe that the PV-filter performs better than the P-filter at the higher percentiles. We also notice that a bias occurs such that the minimum error given by both algorithms is around 6 cm. We discuss the origin of this result in Section 4.5.3.

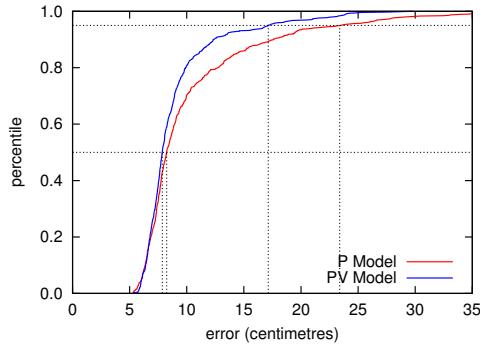


Figure 4.13: Combined 3D error CDF for all eight movements in the camera experiment

4.5 Discussion

4.5.1 Real-world Errors

The results of our simulations give us an idea of the theoretical best performance of the Synchronous BUZZ. In dynamic simulations where the measurements are free of sensor noise and there are no occlusions and no interference, the P-filter achieves a 95% SEP of 7.56 cm, while the PV-filter achieves a 95% SEP of 2.33 cm. In the real-world, however, systematic sensor errors cause performance to degrade. This results in actual 95% SEP values that fall within an accuracy of 10 to 20 cm.

The two major causes of real-world systematic sensor error are transducer misalignment and the attenuation of ultrasound. As these are both functions of the receiver's location in the room, the algorithms experience different levels of sensor noise as the mobile device travels throughout the positioning space. The result is areas within the space, particularly those on the periphery, that have higher positioning error.

Compounding the effect of the systematic errors is the fact that DOP also changes as the receiver moves. Location-varying DOP is a characteristic of indoor positioning systems such as the Synchronous BUZZ. Unlike GPS where distances to the reference nodes are much greater than the distance normally travelled by a GPS receiver, the volume covered by an indoor positioning system can be easily traversed by the mobile device. For example, as our mobile device moves from the centre of the room to an area close to one of the walls, the relative beacon position shifts from one that surrounds the device on

all axes to one that only covers half of the volume. To achieve the same effect with a GPS receiver, a person would have to carry the device through the earth’s upper atmosphere to a space near the orbiting trajectory of the satellites; something not easily achieved by the average wearable computer user.

Another source of positioning error in the real-world involves assumptions about the speed of sound. If the value used by the algorithms is inaccurate—because the temperature in the room has changed, for example—then the pseudoranges calculated by the measurement equation will contain errors. Like the effects of misalignment and attenuation, these errors will cause the position estimates to deviate from the true position of the receiver.

4.5.2 Comparative Performance of the Algorithms

The difference between the error characteristics of the P-filter and the PV-filter is a result of systematic process noise. This process noise is caused by discrepancies between each algorithm’s dynamic model and the actual movement of the receiver. For example, the PV-model algorithm is superior in situations where the receiver is moving with a constant velocity. In this situation, the actual dynamics of the mobile device match the assumptions of the algorithm’s process model. While the P-filter cannot achieve the same level of accuracy for this type of movement, it outperforms the PV-filter at higher levels of process noise (i.e. acceleration) and sensor noise. This is due to the fact that the P-filter places heavier weight on the measurements, making it less vulnerable to corrupt values in the state.

Given the different strengths and weakness of the algorithms, one possible approach is to combine the two models into a single algorithm. For example, it may be possible to run the two algorithms side-by-side and use the output from the one that is giving the best solution. However, in practice determining the best solution is not a straightforward task. One possible method may be to analyse the system covariance of the filters (i.e. choosing the one with the most confidence), but this is left for future work.

One parameter that affects the influence of process noise is the chirp transmission period. It controls how often the algorithm gets to sample the real world. Indeed, if the receiver were to receive measurements at a higher frequency, the performance of both

the P- and PV-filters would improve. Doing so would essentially reduce the amount of predicting that the Kalman filters currently need to do to “connect-the-dots” between the measurements.

However, there is a lower limit to the transmission period employable with the Synchronous BUZZ. This is a result of the fact that the system must wait for all reflections produced by a chirp to dissipate before transmitting another. As the presence of reflections is largely related to the textures and materials of objects in the environment, the optimum period for any installation varies. We currently use a default period of around 30 ms which works well in our lab. We have been able to decrease this to 25 ms in some set-ups, while being forced to increase it to 50 ms in others.

4.5.3 Errors in Calibration and Ground-truths

We note that part of the error observed in our experiments may be caused by the fact that beacon positions were manually calibrated. While we attempted to measure the position of each beacon accurately, it is possible that our manual calibration procedure (measuring tapes and plumb-bobs) introduces positioning errors as large as a few centimetres.

We also note that the errors observed in our experiments can also be caused by inaccuracies in the ground-truths. These inaccuracies do not affect the actual positioning performance, but they do show up as errors in our results. In the Path Comparison experiments, errors in the hand-measured marker positions will cause biases in the output. We assume that the systematic errors present in these results are partly due to inaccuracies in these measurements.

Similarly, it is possible for ground-truth errors to cause biases or systematic errors in the output of the camera experiment. We have identified contributing factors to ground-truth error in this experiment as follows:

best-fit latency: we assume that the relationship between frame numbers and measurement times contains zero latency. However, if the burst points used for the best-fit calculation all have a non-zero latency (i.e. there is a delay between the arrival of the measurements causing buffer overflow and the time that they are displayed),

the best-fit line will be shifted up along the Y -axis. The net effect is to introduce latency in the ground-truth.

lens effects: it can be seen from Figure 3.10 that some non-linear distortion occurs in the video, supplemental to the effects of perspective. This means that errors in the ground-truth will change as the receiver moves across the camera's field of view.

parallax: due to our 2D assumptions about the footage captured by the camera, the height at which the mobile device was mounted (6.8 cm) will cause a location based discrepancy in the ground-truth.

In general, the perspective and camera induced errors are related to the position of the receiver within the scene. In contrast, the latency errors will have an effect on the time related aspects of the results. For example, these errors will be larger when the receiver is moving at higher speeds.

We note that the camera experiment also registers two other types of latency which affects real-world performance. The first is the *input latency*, which results from the 19200 baud serial link between the receiver circuit and the computer. It takes at least 3.125 ms for a six-byte chirp reception time to reach the computer, equating to an error of roughly 3 mm when the mobile device is travelling at 1 ms^{-1} . The second type of latency is the *output latency*. It is caused by the time it takes the Kalman filters to process a measurement and produce an output. However, in our camera experiment, we assume that the measurement arrival time and the position output time are synonymous. This means that results given in our evaluation are actually more favourable than what would be experienced in a real world application.

4.5.4 Resource Usage and Deployment

Output latency is related to the computational load of the algorithms on the CPU of the wearable computer. The more instructions that the Kalman filter requires to perform a time-step, the larger the latency. For each time-step, the PV-filter requires 1761 floating-point multiplication operations, while the P-filter requires 474. If we assume that, of all the operations, multiplication consumes the largest proportion of clock cycles (we neglect division since each algorithm has the same number and there are only two per

time step), these figures imply that the PV-filter has a latency that is approximately 3.5 times larger than the P-filter. This is true regardless of the platform the algorithms are run on.

In terms of the average load of the algorithms on the CPU—which is important if other applications are to be run on the wearable alongside the positioning algorithms—this is inversely related to the transmission period P . Decreasing P increases the frequency of measurement processing time-steps and puts a greater average load on the processor. However, at a value of $P = 25$ ms, which is the lowest period that we have worked with, the algorithms run comfortably on the Gumstix.

Given that the Gumstix is able to run eight Kalman filters in parallel during the initialisation sequence, we estimate that the PV-filter consumes less than one-eighth of the computational resources. Using the 3.5 factor for the relative loads of the algorithms, we estimate that the P-filter consumes less than 4% of the resources on the Gumstix. Also, we note that these figures are for non-optimised code. For example, the performance of the algorithms can be further improved by simplifying multiplication operations involving ones and zeroes.

With respect to the power consumption of the mobile devices, both filters run for over 8 hours on a Gumstix powered by a 3.7 V 750 mAh Lithium Polymer battery (the configuration in Figure 3.6). This is with the period P set to 30 ms. The similarity of the algorithms’ running times suggest that the computations consume only a small fraction of the total power distributed on the mobile device. We note that the length of the battery-life gives the Synchronous BUZZ close to one full day of continuous operation.

4.6 Conclusions

In terms of the use of the Synchronous BUZZ from a user’s perspective, we note that the restriction of having to place the mobile device within a specified location at start-up can be inconvenient. GPS, for example, does not require the user to be at, say, the Equator when the GPS receiver is switched on. Using closed-form solutions to the pseudorangeing equations [9, 17, 80], GPS is able to circumvent the problem by calculating initial seeds for position directly from the measurements. If a closed-form solution to

our pseudoranging equations can be found, a similar technique may be applied to the Synchronous BUZZ. However, such a solution remains an open problem. Nonetheless, given that the spaces covered by the positioning system are relatively small compared to GPS, we have found that the start-up location restriction has not been much of an issue in our installations to date.

From the results obtained in our experiments we make some recommendations for the use of the P- and PV-filters. The PV-filter is most effective when the receiver is mobile and acceleration and sensor noise is infrequent and small. One possible application of the PV-filter is to position tourists in a museum type setting. In these spaces people tend to move slowly and smoothly and would be ideal for the PV-filter. On the other hand, the P-filter performs better than the PV-filter in applications where acceleration and noise is constant and/or large. Game type applications where the receiver is moved erratically—for example, to defend oneself from Stormtrooper blaster fire—may benefit from using the P-filter.

Chapter 5

The Asynchronous BUZZ

In this chapter we present the Asynchronous BUZZ—a positioning system that maintains our cost-saving ethos through a design requiring minimal set-up and maintenance. The system’s main advantage, and the property that sets it apart from the Synchronous BUZZ and other location systems employing wired infrastructures, is that it can be used in sensitive environments where it is necessary to minimise visual impact.

Our design is based on independent ultrasonic beacons. The motivating factor for using independent beacons is that, in our experience of setting up and managing various system installations, wires linking devices within the infrastructure are generally problematic. They are particularly undesirable in environments where aesthetics are important. These environments include living rooms, museums, or historic buildings such as the Elizabethan House described in Section 1.2. In some environments, such as an office, wires are not a problem; they can be hidden within suspended ceilings or within conduits on the walls. Regardless, we believe the most effective solution is to use independent devices that can be affixed unobtrusively.

An example of a system that has achieved a wire-free solution is the Cricket [101]. As we mentioned in Chapter 2, the Cricket employs independent beacons that transmit both radio and ultrasound signals. The radio signal is used to identify the beacon and to provide timing synchronisation, while the ultrasound signal is used to calculate distance.

The Asynchronous BUZZ is meant to be simpler than the Cricket. Specifically, it is based on beacons that transmit ultrasound signals only. The advantage of not using RF is that we further reduce the cost, power consumption, and size of the beacon. The disadvantage is that we no longer have an obvious way of calculating distances, nor an obvious means of identifying signal sources. We demonstrate how we solve both of these problems in the following sections.

5.1 Independent Beacons

The infrastructure of the Asynchronous BUZZ has been designed to be simple, low-power, and free of wires. It consists of a number of small, independent and asynchronous beacons that transmit chirps at unique periods. We have identified two different approaches to perform positioning using these signals: a relative velocity method and a pseudoranging method. These are outlined in the following subsection.

5.1.1 Positioning Methods

As the mobile device moves around the positioning volume, it observes shifts in the periodicity of each of the beacons. This observation results from the Doppler effect: when the device moves towards a beacon, the received signal train will be compressed; when it moves away, the signals will become more separated. The amount that the chirps shift is proportional to the distance the receiver has moved over the period.

$$\Delta d_i = v_s \Delta P_i$$

Here Δd_i is the movement of the receiver relative to beacon i and v_s is the speed of sound. With this distance, the velocity of the receiver, relative to the beacon, can be calculated by dividing by the time of the observed period, $P_i + \Delta P_i$.

$$v_i = \frac{v_s \Delta P_i}{P_i + \Delta P_i} \quad (5.1)$$

This equation forms the foundation for the first approach based on the relative velocity method.

Our second approach is similar to that employed by the Synchronous BUZZ. Specifically, we model the transmission time for each beacon and employ the pseudoranging method to solve for position.

$$R_i = E_i + \frac{\|\mathbf{X} - \mathbf{T}_i\|}{v_s} \quad (5.2)$$

Notice the subscript i on the transmission time E . Unlike the Synchronous BUZZ, the transmission pattern of each beacon is independent and must therefore be modelled separately.

We return to the implementation of these two positioning approaches in Section 5.2. First we discuss the issues surrounding the design and operation of the beacons.

5.1.2 Beacon Transmission Considerations

In this section, we provide arguments for our choice in the number of beacons and their periods.

Number of Beacons

From the discussion in Section 2.2.3, we assume that we require at least six beacons to perform positioning with the relative velocity model. This stems from the fact that Equation 2.4 has six unknown variables. While it is difficult to know the exact number of constraints required (since the equations are non-linear), we can put a lower bound on the number of required beacons at six.

When the beacon requirement problem is analysed from the pseudoranging perspective, it appears that the system is under-constrained regardless of the number of beacons. This can be seen from Equation 5.2, where the number of unknown variables increases with each beacon added. The method we use to get around this issue is dealt with in Section 5.2.3. We claim here that the number of beacons required for the pseudoranging model is less than six, similar to the requirements of the Synchronous BUZZ. However, in practice we use eight beacons with the Asynchronous BUZZ. This provides enough constraint to satisfy both the relative velocity model and the pseudoranging model. It also provides some redundancy that allows the system to recover from situations where

signals go missing. This happens frequently with the Asynchronous BUZZ due to the asynchronous nature of beacon transmissions. It is also important that the number of beacons is not too high. Increasing the number of beacons increases the probability of chirp collisions. We have found that eight gives the right balance between redundancy and chirp density.

Transmission Period Difference

The transmission period of a beacon is used by the receiver to identify the signals it has transmitted. The period of the beacons must therefore be sufficiently different from one another. In order to ensure that each beacon has a relatively equal probability of contributing to the positioning solution, the periods must also be sufficiently similar. To balance these two constraints, we configure each beacon to have the same base periodicity (so that they chirp with a similar duty cycle), but configure them to differ by an ample amount (so that they can be uniquely identified).

The observed magnitudes of the periods are in flux while the receiver is moving. This must be accounted for when choosing the period separation. Consider two beacons i and j with periods P_i and P_j . Assume that $P_i > P_j$. We want to be able to differentiate between the two periods when the receiver is effecting changes in the periods through its movement. In the worst case scenario, the receiver will move directly towards beacon i and directly away from beacon j such that $v_i = -v_j$ (see Equation 5.1). This will shrink the period P_i by ΔP_i and lengthen P_j by ΔP_j . To make sure that the periods remain differentiable we assert that the following inequality must hold:

$$P_i + \Delta P_i > P_j + \Delta P_j$$

To determine a relationship between the receiver's velocity and the beacon periods, we solve Equation 5.1 for ΔP and substitute into the above inequality. This yields:

$$\begin{aligned} v_{\max} &< \frac{P_i - P_j}{P_i + P_j} v_s && \text{for } P_i > P_j \\ &< \frac{|P_i - P_j|}{2 \cdot \max(P_i, P_j)} v_s && \text{in general} \end{aligned}$$

This means that, given ideal conditions where there is no measurement noise, the velocity of the receiver must be less than v_{\max} to observe the difference between P_i and P_j . Rearranging the above expression gives:

$$|P_i - P_j| > v_{\max} \frac{2 \cdot \max(P_i, P_j)}{v_s}$$

which can be used to determine the period difference $|P_i - P_j|$ from the assumed maximum velocity of the receiver. For example, if the beacons have a base periodicity of around 500 ms and if we chose the maximum velocity of the receiver to be 2 ms^{-1} , then the periods must differ by around 6 ms.

For the beacons in the Asynchronous BUZZ, we have chosen 8 ms as the separation value. This equates to a maximum velocity of around 2.7 ms^{-1} .

We note that knowledge of the dynamics of the receiver also helps to identify the source of the chirps. If the velocity and the location of the receiver can be predicted, then the values ΔP_i and ΔP_j can also be predicted. Hence, the modified periods can be more easily identified. This technique is employed by the Asynchronous BUZZ's Kalman filters to improve their ability to identify chirp sources. In the literature, the problem of associating measurements with their sources is often referred to as the *data association problem*. It is present in applications such as radar target tracking, where identifying the source of received signals is non-trivial [10].

Base Transmission Period

Determining the optimum base transmission period for the beacons is a matter of balancing a number of factors. These factors include the need to minimise both the probability of collisions and the power consumption of the beacons—which are achieved with large periods—and the need to increase the frequency of measurements to ensure a high sampling rate—which is achieved with smaller periods.

To reason about possible choices for the base periodicity, we consider a beacon in a one-dimensional space. The signals it emits occupy a two-dimensional communication channel in space and time. When a beacon emits a signal, it propagates through both of the dimensions. The area that it occupies in this channel is related to the speed of

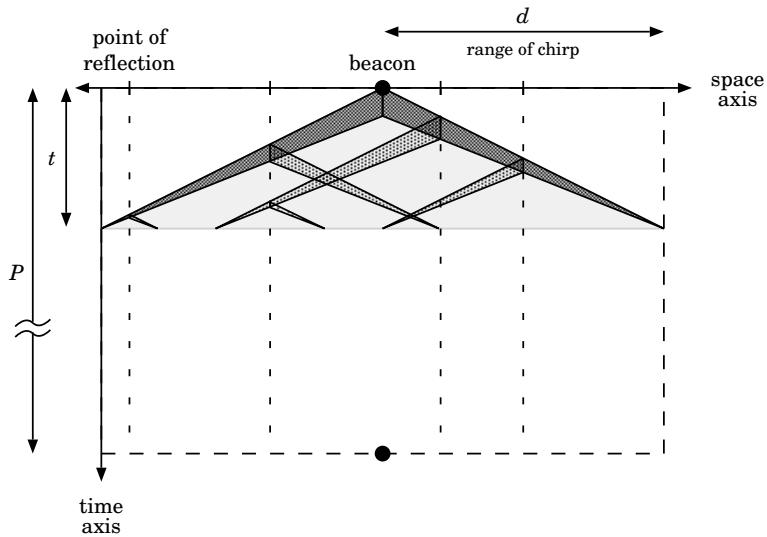


Figure 5.1: The propagation of an ultrasonic signal through a two dimensional channel spanning space and time. The slope of the top lines are equal to the speed of sound.

sound and the distance that the signal travels. Figure 5.1 illustrates the two-dimensional behaviour. The horizontal axis is the space dimension; it is traversed in both directions by the signal starting at the beacon. Time is indicated by the vertical axis, which is only traversed in one direction. The slope of the upper line on each of the signal bands is determined by the speed of sound. The width of each band decreases along the vertical axis due to attenuation as the signal travels through the air. The distance that the signals travel before they are attenuated beyond perception by the receiver is represented by the variable d . We assume that the reflected signals, shown as bands emanating from “points of reflection”, travel the same distance as the direct signals (i.e. no energy is lost in the reflection).

We note that if there were no reflective surfaces, the only signals present in the environment would be the direct signals. However, in many common environments, multiple reflections contaminate the channel such as is shown in Figure 5.1. Increasing the number of reflection points increases the occupancy of the channel. A conservative estimate for the amount of channel occupied by one signal can be approximated using the shaded triangle shown in Figure 5.1. The area of this part of the channel is calculated as follows:

$$\frac{2dt}{2} \implies \frac{d^2}{v_s}$$

Here, t is the height of the triangle and d is the range of the signal. The height is calculated from the speed of sound v_s as $t = d/v_s$.

For one signal, the total size of the channel is calculated using the periodicity of the beacon. It is a rectangle formed by the width of the triangle $2d$ and the time between transmissions P . The area of this rectangle allows us to calculate the fraction of the channel occupied by a single beacon.

$$f_{\text{occ}} = \frac{\frac{d^2}{v_s}}{2dP} = \frac{d}{2Pv_s}$$

For the Asynchronous BUZZ, we have chosen a base period of 500 ms. This period is low enough to provide frequent measurements ($8/0.5 = 16$ Hz on average) and high enough to keep the probability of collision within reason. The maximum range of our beacons is around $d = 8$ m. Therefore, the fraction of occupancy for one beacon is calculated as $f_{\text{occ}} = 2.33\%$.

We analyse our decision for the size of the period by deriving an equation for channel occupancy when eight beacons are present. When one beacon is present, the probability of the channel being empty at any point is $1 - f_{\text{occ}}$. The probability of the channel being empty when there are eight beacons is calculated as $(1 - f_{\text{occ}})^8$. Subtracting this from one gives an expression for the probability of occupancy:

$$1 - \left(1 - \frac{d}{2Pv_s}\right)^8 \quad (5.3)$$

Substituting $d = 8$ m and $v_s = 343$ ms⁻¹ results in a probability of 17% for a period of 500 ms. We note that this figure provides a measure for the amount of channel usage and is strongly correlated with the probability of collision. For example, at any point in the channel, there is a 17% chance of it being occupied with eight beacons chirping at 500 ms. We interpret this as meaning that each chirp has roughly 17% chance of colliding with another signal.

To illustrate the relationship between channel occupancy and beacon periods, we provide a plot of Equation 5.3 in Figure 5.2. It can be seen that at periods less than 500 ms the probability of occupancy increases dramatically. While periods greater than 500 ms create less chance for collision, they also decrease the measurement frequency. The

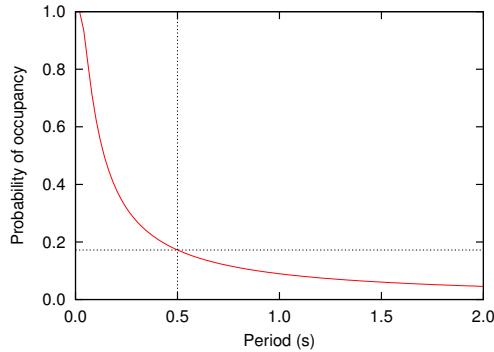


Figure 5.2: Channel occupancy versus base periodicity for eight beacons

figure shows that our choice of 500 ms optimally satisfies both criteria by sitting on the elbow of the curve.

We roughly estimate that, in terms of the number of constraints added to the system by each measurement, the collision rate is within acceptable limits. For example, if we assume that each of the measurements contributes an equal amount to the solution at equal time intervals, and that the collision probability is uniformly distributed among these constraints, then we can roughly calculate the effective number of constraints used by the system. We approximate this as $8 - 8 \times 0.17 = 6.62$, which is greater than the number of unknowns in the system from the velocity model perspective. In other words, even with measurements lost to collisions, the beacons provide a sufficient number of constraints to solve for position.

Combining the chosen period difference and base periodicity, the period of each of the eight beacons is:

$$P_i = 500 + 8i + \varepsilon_i \quad \text{ms} \quad i = 0, 1 \dots 7$$

Here, ε_i represents the error introduced by the different clocks used by the beacons, and ranges between factors of 10^{-6} and 10^{-4} [126]. This means that the periods of the beacons, as observed by the receiver, can differ from their programmed integer values by as much as $56\mu\text{s}$. The presence of the errors is beneficial because it increases the lowest common multiple of the periods. For example, consider the worst case scenario when two chirps from the 500 and 540 ms beacons collide at some point in the channel. For this to occur, the beacons must be turned on at times that are whole fractions of their periods, which is also dependent on the receiver's location along the space axis. If $\varepsilon_i = 0$, then the beacons would collide at this point every $\text{LCM}(500, 540) = 13.5 \text{ s}$. From this

perspective, making the beacon periods co-prime is the solution to minimising repeated collisions. However, the presence of the error causes colliding transmissions to drift apart over time so that repeating collisions occur less frequently with our transmission scheme.

5.1.3 Beacon Hardware

We describe the hardware comprising our beacons in this section. As with the discussion in Section 3.3.1, the software running on the beacons is original work, however, the hardware designs and prototypes are courtesy of Randell [103, 91]. Discussion of beacon circuit design is provided for completeness.

Each beacon comprises eight components:

- an open face ultrasonic transducer
- a radial inductor
- a PIC micro-controller
- a ceramic resonator
- 3 capacitors
- a solar panel of size $(3 \times 4) \text{ cm}^2$

Optionally, the beacons can be powered by a 3 V battery, obviating the need for the solar cell and one of the capacitors. The total component cost for one of the beacons is around \$10 USD. Figure 5.3 shows two of the beacons, one with a solar panel and the other with a battery.

The periodic chirps are entirely generated in software. In order to create a 40 kHz signal, the PIC pulls two of its output pins up and down in a complementary fashion. For example, when pin 1 is up, pin 2 is down and vice versa. This generates a 6 V peak-to-peak swing from a 3 V supply. The ferrite cored inductor is tuned to resonate with the piezoelectric transducer resulting in a signal of over 12 V peak-to-peak at 40 kHz. The

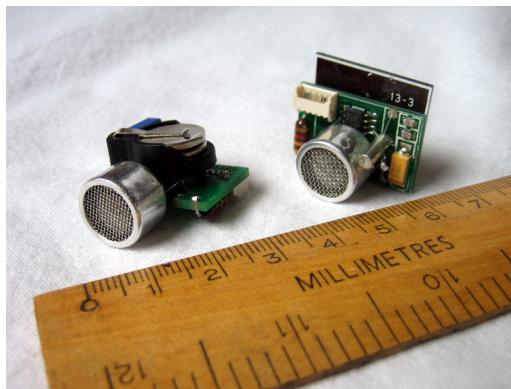


Figure 5.3: Asynchronous beacons

consequent sound pressure level (SPL) is in excess of 105 dBm for each $250\mu\text{s}$ chirp. The transducer is the same as used with the Synchronous BUZZ. It has a wide beam angle of 140° at -12 dB.

Flipping beacon output pins into one state and back again in this fashion takes two instructions. Given that each instruction consumes four clock cycles, we need to run the PIC at at least (or multiples of) 320 kHz to generate a 40 kHz signal. The system also needs to sleep for around 500 ms before transmitting the next chirp. The “official” method for implementing a low-power suspend mode is to use the PIC’s on-chip RC watchdog timer. However, this method is temperature sensitive and can cause timing to degrade. Instead, our program performs a tight loop of NOP instructions and uses the interrupt timer to break out for the next chirp.

In order to run the system from a solar panel, it is essential that we take power consumption down to less than $100\mu\text{A}$. We do this by running the PIC with a slow resonator. We currently use a 640 kHz ceramic resonator and are in the process of finalising the design of a beacon that uses a 32 kHz resonator. With this new design, we use the PIC’s on-board 4 MHz clock to generate the 40 kHz signal and the 32 kHz clock to precisely regulate the period. If this design proves effective, it will be possible to reduce the size of the solar panel further.

The range of the beacon unit is over 7 m, and needs to be mounted close to a light source; preferably fixed to the inside of a shade or reflector. The range of the signal can be extended by using a solar panel that produces 6 V rather than 3 V. Although this is

just a matter of changing the wiring, manufacturers currently have no interest in building small panels with high voltage and low amperage.

5.2 Positioning Algorithm

The Asynchronous BUZZ uses extended Kalman filters to implement both positioning models. We note that, like the Synchronous BUZZ, the Asynchronous BUZZ uses a state machine framework, although a much simpler one. Specifically, it only has an initialisation state and a tracking state. As there is currently no way of recovering from a situation where the system has lost its fix, the algorithm is implemented without a position re-acquisition state. We discuss this further in Section 5.3.1. In the next section, we describe the initialisation state that prepares the extended Kalman filters for execution.

5.2.1 Initialisation

The initialisation sequence serves two purposes: to lock on to the source of the received chirps and to estimate their transmission periods. It is executed while the mobile device is stationary.

At startup, the initialisation algorithm requires loose approximations for the beacon periods. As chirps arrive, it performs a time domain autocorrelation over the history of reception times. It does this by maintaining a history of chirp reception times that is several seconds long. When the algorithm receives a chirp, it scans through the chirp history, subtracting each of the previous reception times from the current reception time. This is illustrated in Figure 5.4. If any of the resulting time ranges are similar to one of (or multiples of) the known periods, the algorithm stores the matching beacon number and the time ranges within a data structure. At the end of the history scan, the variance within the time ranges for each recorded beacon number is calculated. The current chirp is committed as originating from the beacon number with the smallest variance, as long as the variance falls within a threshold. The algorithm does this by marking the current chirp, as well as the historical chirps forming the associated time ranges, with the beacon number. It also updates an estimate for the period of the beacon by taking an average of

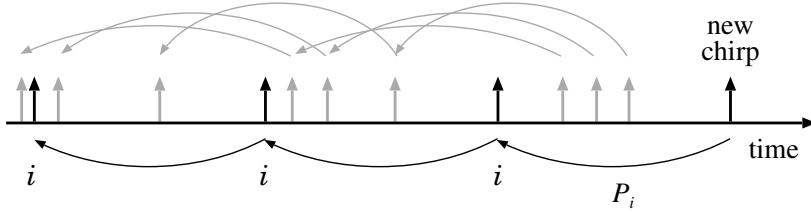


Figure 5.4: The initialisation sequence scans the chirp history to find a source match for the most recent chirp

all of the recorded time ranges. This extra information is used to increase the accuracy and efficiency of labelling future chirps.

The initialisation algorithm terminates when it has identified and labelled a predefined number of chirps. At this point, the positioning algorithm has an accurate approximation for the beacon periods, measured on the receiver's local clock. It also knows the previous reception times for each beacon and is thus able to begin predicting the arrival of future chirps. This information is passed to the Kalman filtering part of the algorithm to start positioning. The user is able to move the mobile device once this has occurred.

Depending on the number of chirps used, the initialisation phase can take anywhere from several seconds to several minutes to complete. One approach that works well in our experience, is to run the initialisation sequence for an extended period of time and record the results for use in future positioning sessions. This is possible since the time dependent variables in the system—the clocks on the beacons and receivers—do not change between sessions within the same installation. To this end, we have created a separate algorithm, called *pfind*, that starts by using the initialisation sequence to lock on to a small number of chirps, then instantiates a set of simple Kalman filters to estimate the periods of each of the beacons. When *pfind* has run for, say, five minutes, the receiver knows the beacon periods to a high degree of accuracy. Given these accurate periods, the positioning algorithm need only execute the initialisation sequence to lock on to the source of the chirps. This reduces the time that the mobile device is required to remain stationary when the positioning algorithm is launched.

Data from the *pfind* program is used with the real-world experiments provided in this chapter. The data—composed of period estimates and variances for each beacon—is calculated using a five minute recording of chirp reception times. For the positioning

algorithms, the number of required labelled chirps in the initialisation sequence is set to five per beacon. With this setting, initialisation takes around five seconds to complete before positioning begins.

5.2.2 Doppler Filter

The first method we employ for position recovery is the Doppler Filter. It uses the relative-velocity approach to estimate the position of the ultrasonic receiver. It is implemented using an extended Kalman filter in a manner similar to the Synchronous BUZZ. The filter's state contains six variables representing position and velocity in three dimensions. The process equations implement the PV-model where both position and velocity are propagated in time.

The measurement equation is derived from Equation 5.1 and Equation 2.4 such that the reception times are expressed in terms of the state variables.

$$R_i = R_{i_{\text{prev}}} + P_i v_s \left(v_s - \frac{\mathbf{X}_k - \mathbf{T}_i}{\|\mathbf{X}_k - \mathbf{T}_i\|} \cdot \mathbf{V}_k \right)^{-1} \quad (5.4)$$

Equation 5.4 states that the current reception time for beacon i , R_i , is equal to the previous reception time for this beacon, $R_{i_{\text{prev}}}$, plus the beacon period P_i scaled by a factor related to position and velocity. Starting with the reception times collected by the initialisation sequence, this equation is able to iteratively predict the reception times of each of the signals. These predictions are what allow the algorithm to identify the source of each of the chirps as they arrive.

We note that, unlike the Synchronous BUZZ, the Doppler Filter does not model the behaviour of the beacons. Equation 5.4 is constructed independently of absolute transmission times, which are represented in the Synchronous BUZZ by the variable E . Therefore, the filter does not need to maintain their state by including them in the state vector. To operate, it only requires prior knowledge of the beacon locations and their approximate periodicities P_i .

To avoid chirp-beacon association errors, measurements that arrive close to the predicted reception times for more than one beacon (i.e. collisions) are ignored. We have tried a number of methods that attempt to rectify these ambiguous situations—including im-

plementing a Multi-hypothesis Kalman filter [10]—but all required large amounts of overhead without much gain. We have observed in practise that the frequency of collision is low enough to safely ignore. As a result, our algorithms are more light-weight and are easily able to run in real-time on a low power processor.

Compared to the Doppler positioning systems that use RF, our method can be considered a “velocity averaging technique”. While measurements of Doppler shift for RF signals are provided over the wavelength of the signal (on the order of nanoseconds for SARSAT which operates at 406 MHz), our system takes measurements over a much longer period (500 ms). This means that, rather than measuring an instantaneous relative velocity, the system measures an average relative velocity over a half-second time period.

The result is that the Doppler algorithm exhibits a latency in its estimate for position. Basically, the effect of using average measurements means that any changes in the mobile device’s position takes time to appear in the output. If the mobile receiver is moving along the path of a circle, for example, the system will use chirp reception times for two different points on the circle to measure the relative velocity. If the receiver maintains a constant speed on the circle, each measured velocity will therefore be calculated as the receiver’s velocity when it was located at a point halfway between the two points of chirp reception. The resulting measurement integration step will estimate position as somewhere near this point, rather than at the point of the most recent chirp arrival. The overall effect is to produce estimates that are aliased versions of the true path, offset in both time and space.

We estimate that the average latency of the Doppler Filter is roughly half of the base transmission period. However, this will depend on the weight given to the measurements, which is calculated using the time between measurements. The asynchronous behaviour of the system makes this difficult to predict.

The Doppler Filter assumes that the period of each of the beacons is constant. Specifically, it does not model the periods as state variables within the Kalman filter as is done with the Synchronous BUZZ. In fact, doing so results in an unobservable system, where the system covariance increases without bound. Adding these variables to the state vector introduces too many degrees of freedom, such that the measurements do not sufficiently constrain the system. Instead, we use constant valued representations for P_i , as determined by the initialisation sequence and the pfind algorithm.

Process Equations	Measurement Equations
$\mathbf{V}_k = \mathbf{V}_{k-1}$ $\mathbf{X}_k = \mathbf{X}_{k-1} + \Delta t \mathbf{V}_{k-1}$	$R_i = R_{i_{\text{prev}}} + P_i v_s \left(v_s - \frac{\mathbf{X}_k - \mathbf{T}_i}{\ \mathbf{X}_k - \mathbf{T}_i\ } \cdot \mathbf{V}_k \right)^{-1}$

Table 5.1: Doppler method extended Kalman filter equations

Process Equations	Measurement Equations
$E_i = E_{i_{\text{prev}}} + P_i$ $\mathbf{X}_k = \mathbf{X}_{k-1}$	$R_k = E_i + \frac{\ \mathbf{X}_k - \mathbf{T}_i\ }{v_s}$

Table 5.2: Pseudorange method equations

Table 5.1 summarises the process and measurement equations for the extended Kalman filter of the Doppler method.

5.2.3 Pseudorange Filter

Performing positioning via the pseudoranging method is more complex for the Asynchronous BUZZ than it is for the Synchronous BUZZ. The reason is that instead of modelling only two variables to describe transmission behaviour (P and E), the Asynchronous BUZZ must model N transmitting systems, where N is the number of beacons.

The process and measurement equations used by the filter are given in Table 5.2. The measurement equation is an expression for the time-of-flight of a chirp transmitted from beacon i , located at position T_i . It is similar to the measurement equation used with the Synchronous BUZZ except for the presence of the subscript i on the transmission time E . Essentially, the filter models each of the separate transmitting systems by keeping track of their transmission times.

We note that the periods P_i are constant values, and are not contained within the state. Also, the beacon transmission times are not incremented in the same fashion as the Synchronous BUZZ (note the missing k subscript in the first process equation). This is due to the asynchronous transmission pattern, where it is not possible to merely increment the transmission time E_i by the period P_i each time a chirp is received. Instead, E_i is incremented by P_i when the reception time of the current chirp has gone past the last

reception time for beacon i plus an additional $P_i/2$. This keeps all of the E_i values up to date and ensures that the filter is able to predict the arrival time of each of the chirps.

Like the Doppler algorithm, the predictive nature of the filter provides a method for determining the source of the chirps that takes into account the current position of the receiver. Again, chirps with ambiguous sources are ignored.

In total, the Kalman filter tracks $N + 3$ variables in its state vector, where N is the number of beacons. As mentioned in Section 5.1.2, this is a system with more unknowns than equations, implying that the system is not solvable in the conventional sense. However, the Pseudorange Filter is able to perform positioning if it is initialised with a reasonable seed for the position vector \mathbf{X} . A good position seed allows the filter to determine reasonable values for the beacon transmission times at startup. It does this by plugging the seed for \mathbf{X} , along with the first chirp reception times R , into the measurement equation and solving for E_i . From this point, the filter need only keep the transmission times relatively stable for positioning to be possible. So, even while the constraints on the transmission times E_i are weak—i.e. the only shared variables between equations are \mathbf{X} —the system is able to use the chirp reception times to determine the movement of the receiver. We observe the effect of seeding on the Pseudorange Filter in our results in Section 5.4.3.

In terms of the differences between the Doppler Filter and the Pseudorange Filter, we note that the Pseudorange Filter does not model velocity. Although this is possible, we have chosen to keep the number of degrees of freedom to a minimum by implementing a position-only dynamic model (see the second process equation in Table 5.2). The pseudoranging method also avoids the “averaging” characteristic of the Doppler method, in that it does not use two chirp reception times in the process of integrating one measurement. The pseudoranging equations directly express how the state of the process is related to single observations. As a result, the aliasing behaviour expected from the Doppler method is absent from the Pseudorange Filter. Its performance should therefore be generally better than the Doppler Filter. We explore this further in Sections 5.3 and 5.4.

5.2.4 Chirp-Beacon Association

In order to integrate measurements with either of the Kalman filters, we must be able to determine the source beacon for each chirp. The method used to do this with the Doppler and pseudoranging models is to completely couple the identification of chirp sources with the positioning algorithm. This is done automatically through the Kalman filter's measurement prediction steps in that the prediction for the next chirp is based on the dynamics of the receiver via the measurement equation. The χ^2 method is then used to compare the likelihood of a received chirp belonging to each beacon. If a single beacon can be identified, the chirp is integrated with the filter as originating from that beacon.

For the Doppler method, the identification of chirp sources is slightly more explicit since two chirps are used within the measurement equation. This has implications for the algorithm when chirps go missing due to occlusions or collisions. For example, if a chirp goes missing, $R_{i_{\text{prev}}}$ will be unknown for the next chirp originating from the same beacon, making the measurement equation incalculable. To overcome this, our algorithm reconstructs the missing $R_{i_{\text{prev}}}$ using information about the last known reception time and the state of the receiver dynamics. Ideally, we want to use an estimate for the state which is closest to the time that the missing chirp *should* have arrived. This allows us to accurately reconstruct the $\Delta P + P$ offset to be added to the last known reception time. In the case of a collision, where chirps are ignored because there is more than one likely source, the algorithm saves the state so that it can be used during the reconstruction. If no saved state is available when the missing chirp has been identified (at $P/2$ past its expected arrival), the algorithm uses the current state.

Another method of handling missing chirps is to merely wait until two subsequent chirps are available for use with the measurement equation. While this seems like a valid approach in that it does not involve “faking” receptions times, the method increases the time between Kalman filter measurement updates. This, in turn, reduces the sampling frequency of the system and decreases its ability to track the position of a dynamic receiver. We have found that the chirp reconstruction method achieves the appropriate balance between measurement integrity and maintaining measurement integration rates.

We have observed that the Doppler algorithm can get into a state where chirps from a particular beacon are ignored for prolonged periods of time. This happens when sensor

noise, collisions, or occlusions cause the algorithm to continually reconstruct missing chirp reception times. The resulting error that creeps into $R_{i_{\text{prev}}}$ in turn causes arriving chirps to be continually rejected. To break out of the cycle, we use a more permissive χ^2 threshold to identify chirps that have not been claimed by the lower threshold. If there is no ambiguity (e.g. only one beacon falls within the threshold), we label these chirps as originating from the beacon in question. In this way, the filter is able to use the labelled reception time in the measurement equation of the beacon's next chirp.

Our classification method is considered to be strongly coupled with the state. This stems from the fact that the accuracy of the classification process is largely dependent on the quality of the state. Another contrasting approach is to employ a classifier that is independent of the positioning algorithm. We have attempted this approach by constructing a classifier consisting of N particle filters, each used for identifying chirps from a designated beacon [91]. A separate particle filter uses the results of the classifier to perform positioning. In this system, there is no feedback between the two parts of the algorithm. For example, the classifying filters do not use the dynamics estimated by the positioning filter to improve their predictions.

In general, we have found that a decoupled classifier produces consistent but average identification rates. A coupled classifier, on the other hand, provides near perfect classification rates but only when the filter is in a good state. Basically, if the state goes off track, the coupled classifier degrades as well, causing the system to diverge.

Due to its ability to handle non-Gaussian, multi-modal distributions, the particle filter solution is able to cope with the higher misidentification rates offered by its decoupled classifier. Our Kalman filter solutions, on the other hand, cannot deal with misidentification rates as high as those produced by this classifier. In a brief experiment, where we configured the Kalman filters to accept the classification data from the particle filters, the misclassified chirps caused the filters to diverge. As discussed in Section 3.2.2, the Kalman filter is an IIR filter, meaning that incorporating outlying measurements (from misclassification) produces lasting errors in the system. Our experiment shows that the filters are unable to recover from these unfavourable states.

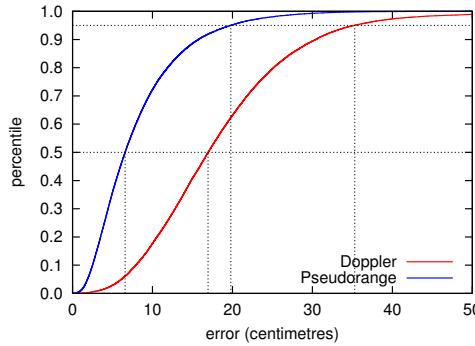


Figure 5.5: Dynamic simulation results with ideal data.

	Doppler	Pseudor.
SEP (50,95)	17.31 35.03	6.57 19.80
Horiz. CEP (50,95)	13.53 30.75	4.16 14.49
Vert. LEP (50,95)	6.86 23.57	3.64 14.75
MRSE	20.63	9.99
Horiz. DRMS	17.13	7.10
Vert. RMS	11.50	7.02
Chirp-beacon association statistics		
Correct	0.966	0.974
Wrong	1.83×10^{-5}	1.83×10^{-5}
Ignored	3.25×10^{-2}	2.42×10^{-2}

Table 5.3: Dynamic simulation results for ideal data (values in centimetres) and chirp-beacon association statistics (fraction of total chirps transmitted).

5.3 Evaluation through Simulation

In this section we evaluate the Asynchronous BUZZ using a number of different simulations. Figure 5.5 and Table 5.3 provide results from a 60-minute dynamic simulation using ideal conditions. As with our results for the Synchronous BUZZ, we provide the 3D error CDFs in Figure 5.5, but we also supplement this with chirp-beacon association statistics at the bottom of Table 5.3. Here, the fraction of total chirps transmitted is provided for the number of chirps correctly classified (“correct”), the number of chirps that were misclassified (“wrong”), and the number of real chirps that were ignored due to collision (“ignored”).

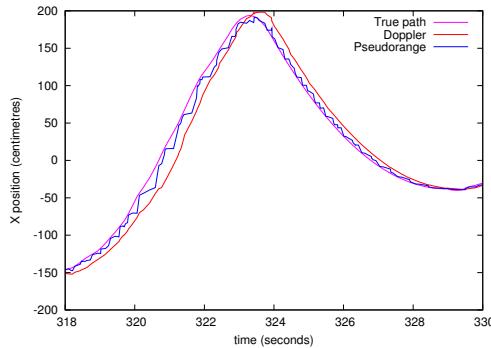


Figure 5.6: Position along X-axis versus time. The Doppler Filter lags behind the Pseudorange Filter due to way that it measures relative velocity over a long period.

The chirp reception times generated by the simulator contained no sensor noise, there were no occlusions or reflections, and the layout of the eight-beacon infrastructure was ideal. The path traversed by the receiver was generated using the random walk method described in Section 3.4. The maximum and average speeds achieved by the receiver were 1.6 ms^{-1} and 0.56 ms^{-1} , respectively.

Figure 5.6 is provided to show the latency of the Doppler Filter due to the long period over which it measures relative velocity. The figure shows the X-coordinate position of the receiver as it moves within a 12 second time window. We note that it takes slightly longer for the position estimate of the Doppler Filter to catch up to that of the Pseudorange Filter. The algorithm also overshoots the path of the receiver at the top of the peak shown (it takes longer to adjust to the change in direction). We attribute the difference between the performance of the two algorithms given in Figure 5.5 and Table 5.3 to this averaging behaviour.

The results we have obtained for this simulation serve as a benchmark for the results provided in the remainder of the chapter.

5.3.1 Sensor Noise

To observe how sensor noise affects the performance of our algorithms, we have created 51 10-minute simulations, each with a different value for the standard deviation of random Gaussian sensor noise. While the sensor noise experienced in practice is more

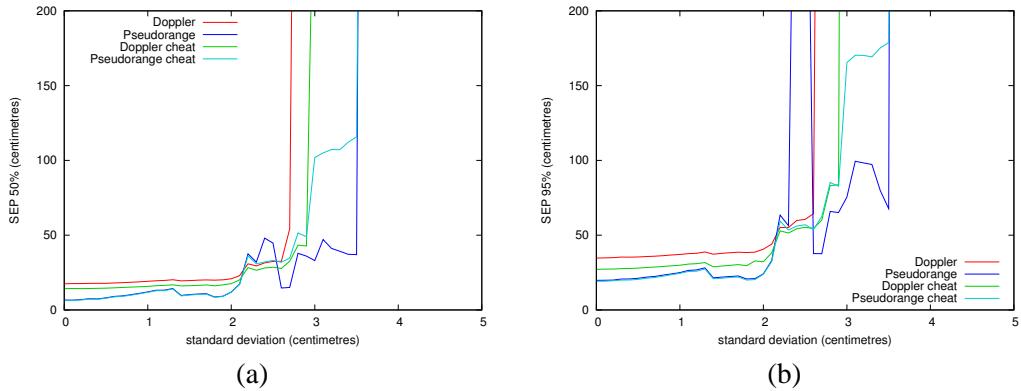


Figure 5.7: (a) 50% and (b) 95% SEP versus sensor error standard deviation.

systematic than Gaussian (see Section 3.2.1), we aim to explore how the two models behave in conditions of increasing sensor errors. The distance based error values range from 0 to 5 cm. As with the similar experiments done with the Synchronous BUZZ, we use a distance representation for the sensor noise in order to make it easier to conceptualise. The results of the experiment are shown in Figure 5.7.

We observe that, compared with the Synchronous BUZZ, the Asynchronous BUZZ is more sensitive to sensor noise. We note that the algorithms begin to break down at around 2 to 4 cm whereas both the P- and PV-models of the Synchronous BUZZ are still accurate at noise levels beyond this. To verify whether or not misclassification was the cause for divergence at the higher noise levels, we ran the experiment using algorithms with perfect classification. Their outputs are shown in the plot as lines labelled ‘‘cheat’’. We note that both versions of the algorithms perform similarly, suggesting that chirp-beacon misclassification is not a major factor.

Instead, we attribute the behaviour in this experiment to the extra degrees of freedom in system. Compared to the Synchronous BUZZ, the Asynchronous BUZZ is less constrained in terms of how the measurements provide observations of the underlying process. This is true for both the Doppler Filter and the Pseudorange Filter. In a sense, the algorithms have less of a handle on the state variables. This means that larger sensor noise is more likely to create outliers; measurements that the system cannot assimilate without adversely affecting the system. The Asynchronous BUZZ’s sensitivity to outliers can be seen from the performances of the original and cheating versions of the algorithms in Figure 5.7. What we notice is that the original algorithms, which perform their own classification using the χ^2 threshold, actually outperform the algorithms that cheat. This

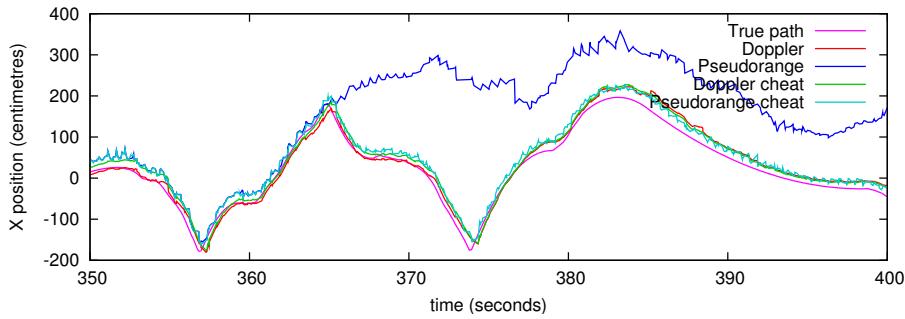


Figure 5.8: X-axis estimate for position showing divergence of the Pseudorange Filter. The sensor noise standard deviation is 2.4 cm.

is down to the different ways in which the algorithms handle outliers. Specifically, the cheating algorithms integrate all measurements, while the original algorithms disregard measurements that are above their χ^2 threshold. As the sensor noise increases, the measurements are more likely to be considered outliers. The original algorithms identify these outliers, while the cheating versions integrate them such that they diverge at lower levels of sensor noise.

With higher levels of sensor noise, the fraction of measurements considered to be outliers increases. This means that the algorithms either discard the measurements because they are above the χ^2 threshold, or integrate them because the system covariance is high. In the first instance, if the number of outliers is too high, the Kalman filters will not receive enough measurements to track the process. In the second instance, the algorithms have a high probability of diverging.

Another difference between the Synchronous BUZZ and the Asynchronous BUZZ is that the Asynchronous BUZZ does not implement any stability monitoring and recovery. This means that when the algorithms fall off track, they diverge and remain that way for the duration of the simulation. This can be seen from the 95% SEP plot in Figure 5.7(b). It shows the Pseudorange Filter going off course at a sensor noise standard deviation of 2.4 cm. Despite being fairly accurate for most of the simulation, it loses its fix on position just over half way through. Figure 5.8 shows this occurring along the X-axis of the position estimate.

We believe that it is possible to identify these situations by monitoring the system covariance values, and to recover by resetting the covariance and the initial state (similar

to what is done by the Synchronous BUZZ). However, we predict that it will not be a trivial procedure due to the fact that chirp-beacon association is heavily coupled with the system state. As a result, it may be necessary to employ the initialisation sequence again to lock-on to the chirp sources. This could be an annoying feature of the system since the initialisation sequence requires the mobile receiver to remain stationary at a predetermined location.

5.3.2 Monte Carlo Exploration

As mentioned in the previous section, the extra degrees of freedom in the Asynchronous BUZZ make it a more difficult problem to solve than the Synchronous BUZZ. To gain an understanding of how well the beacons and their measurements constrain the system, we provide an exploration of the solution space of the Doppler model. This is done through a number of Monte Carlo simulations. Our aim is to observe how different factors affect the Asynchronous BUZZ's ability to solve for position.

In the experiment, we randomly sample position and velocity pairs (\mathbf{X}, \mathbf{V}) and compute their associated ΔP values with respect to a set of beacons. This is done by equating Equation 5.1 with Equation 2.4 and solving for ΔP :

$$\Delta P_i = P_i \left(v_s \left[\frac{\mathbf{X} - \mathbf{T}_i}{\|\mathbf{X} - \mathbf{T}_i\|} \cdot \mathbf{V} \right]^{-1} - 1 \right)^{-1} \quad (5.5)$$

The computed ΔP values are compared to those generated from a reference position-velocity pair. The aim of the experiment is to determine how the Doppler equation constrains values for position and velocity. For example, (\mathbf{X}, \mathbf{V}) pairs varying from the reference pair should generate ΔP values that are different from those generated with the reference pair. The degree to which the ΔP values differ will indicate how well the equations constrain position and velocity.

The experiment assumes that we can take simultaneous measurements for ΔP and that we can identify the source of each of the measurements with 100% accuracy. While this is not possible in practice, it simplifies the experiment. Given that these conditions are a best case scenario, we should expect a degraded performance in practice.

To observe the impact of beacon placement, we explore the solution space created by two different eight-beacon configurations. The first configuration situates the beacons on the corners of a $4 \times 4 \times 4$ metre cube at locations $(-2, -2, -2), (-2, -2, 2), \dots (2, 2, 2)$. The second configuration simulates the set-up that we have in our lab, with the eight beacons placed on the walls and ceiling.

The reference pair is placed near the centre of the space, at location $(-0.5, 0.5, 0.4)$ m and we assume that it is moving with a velocity of $(0.5, 0.7, 0.1)$ ms $^{-1}$. This allows us to compute a reference ΔP_i^{ref} that corresponds to each beacon i . For example, ΔP_0^{ref} for the beacon at $(-2, -2, -2)$ is 1.1 ms and ΔP_7^{ref} for the beacon at $(2, 2, 2)$ is -1.2 ms. The collection $(\Delta P_0^{\text{ref}}, \Delta P_1^{\text{ref}}, \dots, \Delta P_7^{\text{ref}})$ forms a vector of reference measurements, denoted $\Delta \mathbf{P}^{\text{ref}}$.

For each beacon configuration, we sample a large number of random pairs $(\mathbf{X}, \mathbf{V})_j$ in the six dimensional space spanning location and velocity. The pairs are uniformly distributed in location (with bounds $[-5..5]$) and velocity (with bounds $[-3..3]$). For each pair $(\mathbf{X}, \mathbf{V})_j$ we calculate the ΔP_i^j for each beacon i and compare them with the reference measurements by taking the magnitude of their difference:

$$\delta^j = \|\Delta \mathbf{P}^{\text{ref}} - \Delta \mathbf{P}^j\|$$

This δ measure is the Euclidean distance between the reference ΔP vector and the candidate vector. It provides us with a measure of the total ΔP error, which is converted to a distance by multiplying by the speed of sound.

Ideally, there is only one point in the space with a zero δ value: the reference point itself. Position-velocity pairs that are further away from the reference pair in the location space should have higher δ values, while pairs that are nearby have lower δ values. We want this to be true despite the presence of sensor noise; which effectively desensitises our system to changes in ΔP measurements. In terms of the solution space, sensor noise “lifts” the error level at which are able to search for the solution. We clarify this statement in the following subsections.

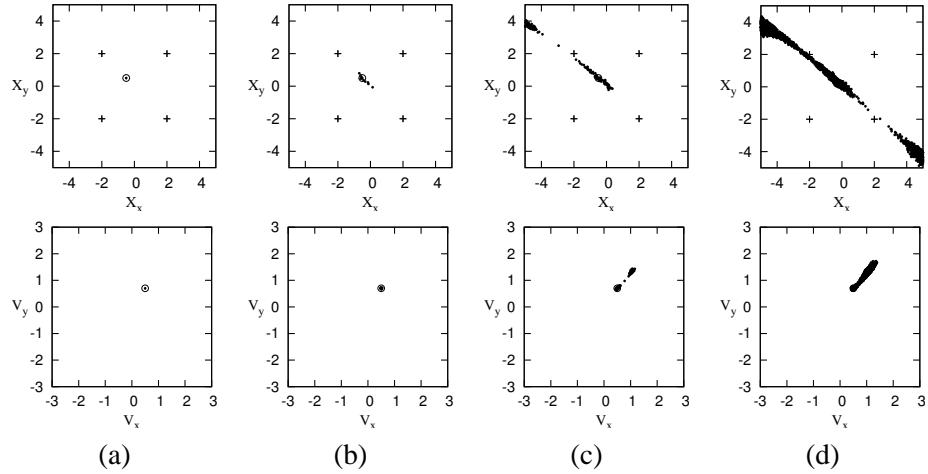


Figure 5.9: Position and velocity samples with δ less than (a) 2 cm, (b) 4 cm, (c) 6 cm, and (d) 8 cm using the cubic beacon configuration

Results on cubic layout of beacons

Figure 5.9 depicts the results of the cubic beacon experiment. Each plot in the figure shows the X - Y placement of position (in the top row) and velocity (in the bottom row). The Z -axis has been left out to keep the visualisation simple. The dot surrounded by a circle is the reference point, while all other dots are the randomly generated position-velocity pairs. For the plots detailing position, the stars depict the position of the beacons in the X - Y plane. For this experiment, only four beacons can be seen due to overlap caused by the aerial view.

The figure has been separated into columns (a) through (d), which correspond to different thresholds on δ . Column (a) includes all pairs that have δ values less than 2 cm, and Column (d) includes all pairs that have δ values less than 8 cm. It can be seen that, as the threshold for δ increases, there is more of the 6D space that is covered by candidates for the reference pair solution.

Viewing the location space, one can see that the candidates seem to occupy, roughly, a plane (which extends in the Z direction—not visible in the plots). It appears that the plane is perpendicular to the velocity of the receiver. In the velocity space, the candidates appear as an ellipsoid of velocities that have the same direction as the reference velocity, but with higher magnitude (speed).

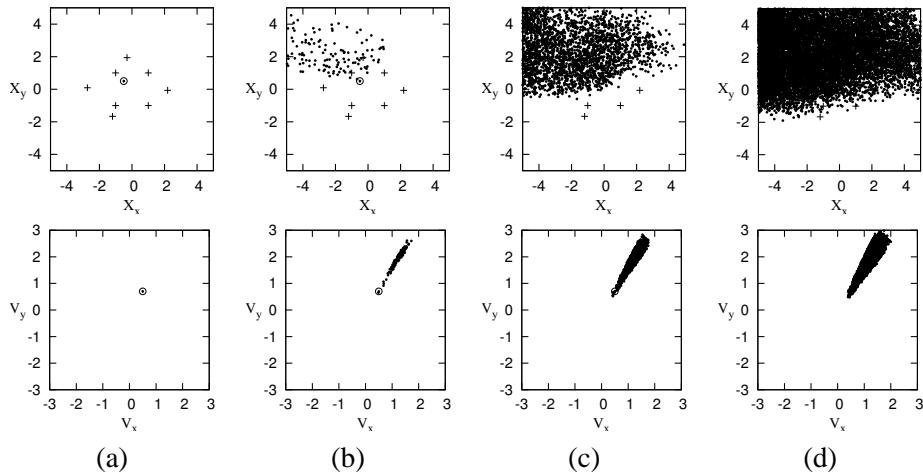


Figure 5.10: Position and velocity samples with δ less than (a) 2 cm, (b) 4 cm, (c) 6 cm, and (d) 8 cm using the lab beacon configuration

Results on laboratory layout of beacons

The results from simulations using the beacon configuration employed in our lab are given in Figure 5.10. In this configuration, there is a smaller variation in the Z-coordinates for the location of the beacons. As a result, we observe a much more unpleasant distribution of candidate solutions over the X and Y axes. Although it cannot be seen in the 2D plots provided, most of the candidate solutions exist in the lower half of the volume, under the floor. As with the cubic configuration, the candidates in the velocity space are approximately constrained to an ellipsoid extending in the same direction of the reference velocity.

Results along a line in space

In order to further explore the presence of candidates in the cubic beacon experiment, we have made three one-dimensional cuts through the 6D space and plotted the error against the distance traversed along these cuts. This is shown in Figure 5.11. Each cut follows the same path through the 3D location space, intersecting the reference point. The lines follow the diagonal band of points displayed in Figure 5.9(d). The horizontal line in Figure 5.11 is the corresponding 6 cm δ threshold. Each of the three lines depicting the

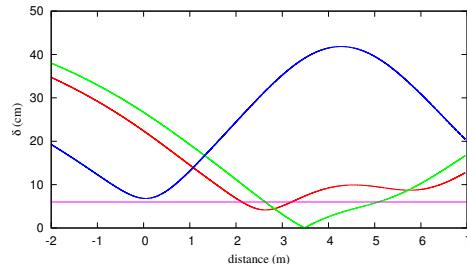


Figure 5.11: The distance δ between the reference pair and candidate pairs taken along three lines in the 6D space spanning position and velocity. The result shows that for large sensor noise (given by the straight red line) the range of probable position candidates is large.

path through the 6D space have a fixed but differing velocity. We observe that one of the lines reaches zero-error. This is the line with velocity set to the reference velocity and, as expected, it hits zero when it goes through the reference point.

The results show that our Doppler model only applies weak constraints to position, especially in the presence of sensor noise. The wide and shallow troughs, formed by the minima in Figure 5.11, illustrate that there are large parts of the solution space that satisfy the equations. These are the areas of the troughs that fall beneath the 6 cm mark. Even with a perfect estimate for velocity (the line that reaches zero), the range of possible positions along the line 3D space spans roughly 2 m.

Our conclusion is that, compared with the Synchronous BUZZ, high levels of sensor noise severely restricts the algorithms' ability to hone in on the true solution for position. This agrees with results in the previous section, where we observe how relatively low levels of sensor noise cause the algorithms to fail. Although we do not provide an experiment for the pseudorange model, the evidence from the sensor noise experiments indicate that results will be better, but not to the same quality of the Synchronous BUZZ.

5.3.3 Occlusions

In this section, we provide results from a simulation where the probability of occlusion was varied. These are shown in Figure 5.12. The same path was used for each of the 71 10-minute trials. Again, we used eight beacons, there were no reflections, and sensor

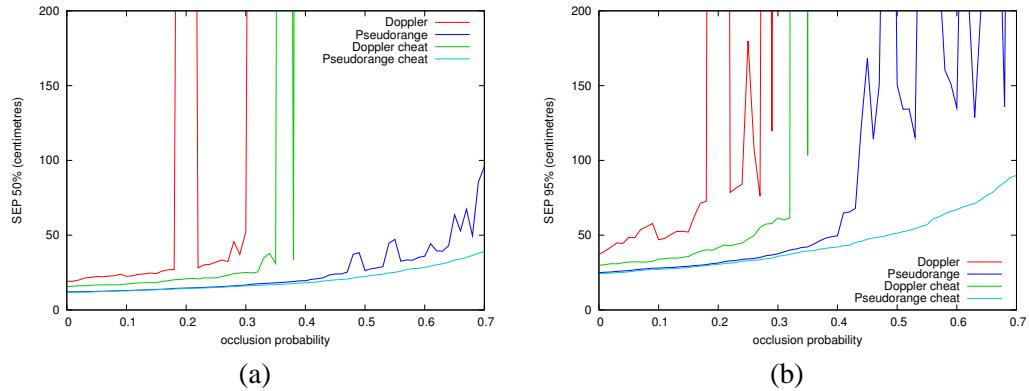


Figure 5.12: (a) 50% and (b) 95% SEP versus occlusion probability.

noise was set to 1 cm standard deviation. Like the sensor noise experiment, we provide results for both algorithms as well as versions with perfect chirp classification (“cheat”).

What we observe from the results is that the Pseudorange Filter outperforms the Doppler Filter in the face of occlusions. We attribute this to two reasons. First, as we observe in experiments with the Synchronous BUZZ, the use of the PV-model by the Doppler Filter makes it more prone to divergence. If the filter begins to lose its fix on the state, errors in velocity accumulate quickly in the position estimate. The P-model used by the Pseudorange Filter does not suffer from this. The effect of the different dynamic models is exemplified by the performance of the cheating algorithms. Despite both algorithms having access to perfect chirp classification, the cheating Doppler Filter diverges while the cheating Pseudorange Filter remains stable. Once the cheating Doppler filter loses its fix, it “shoots” off into an irrecoverable state.

The second reason for the poorer performance of the Doppler Filter is its requirement of two reception times to integrate a measurement. Despite the algorithm attempting to reconstruct reception times for missing chirps, an increase in the number of occlusions results in a decrease in the sampling frequency. This, in turn, results in an overall increase in the error in the system. The effect can be seen in Figure 5.12 in the difference between the Doppler Filter and the cheating Doppler Filter. Basically, the cheating algorithm outperforms the original since it is able to integrate more measurements.

Figure 5.13 is provided to show how the Doppler Filter and the Pseudorange Filter perform with respect to chirp classification. In particular, we observe that the Doppler Filter’s ability to identify chirps breaks down at an occlusion probability of around 0.3.

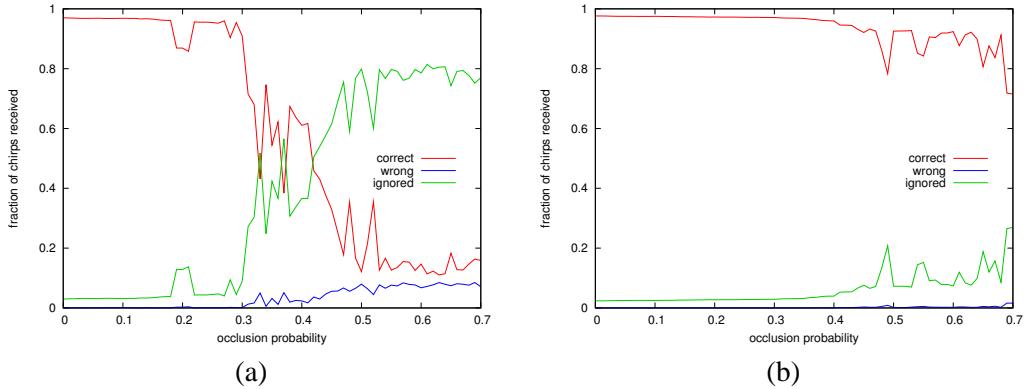


Figure 5.13: Chirp-beacon association statistics versus occlusion probability for the (a) Doppler Filter, and (b) Pseudorange Filter.

The Pseudorange Filter in Figure 5.13(b), on the other hand, performs significantly better. The tight coupling between classification and the state means that once the state begins diverge, chirp classification also begins to fail. The two parts of the algorithm feed into one another such that a breakdown in one will cause the breakdown of the other.

5.3.4 Reflections

Due to its asynchronous operation, the Asynchronous BUZZ is unable to control the timing of chirp transmission to avoid collisions with reflections. As such, the more reflections that are present, the more that collisions will occur. To observe the effect of reflections on the performance of the algorithms, we ran a number of simulations that added reflections to the channel.

For each transmitted chirp, two reflections were randomly added to the chirp train 3 ms (1 metre) and 6 ms (2 metres) after the chirp reception time. The probability of the first reflection was set by a parameter in the simulator. For the second reflection, this probability value is halved in order to simulate the effects of attenuation. While the reflection timings are not truly realistic—for example, the offset from the direct chirp will depend on the mobile device’s relation to surfaces in the room—they do successfully add interference to the measurement stream.

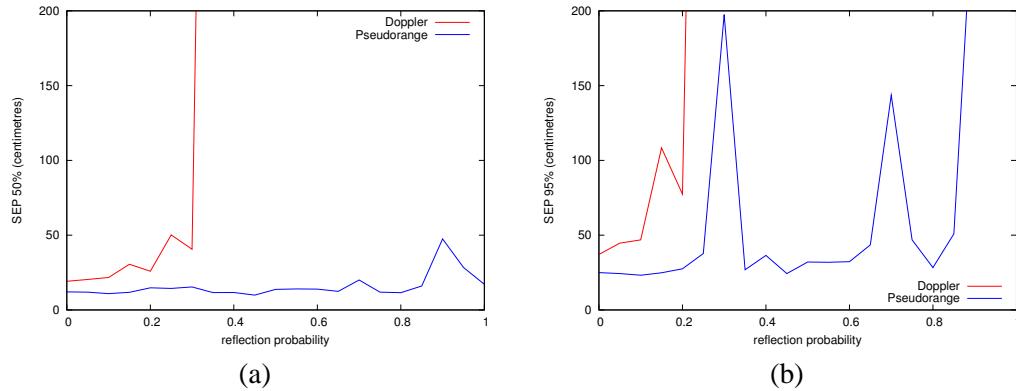


Figure 5.14: (a) 50% and (b) 95% SEP versus reflection probability.

Figure 5.14 shows the 50% and 95% SEP results for 20 simulations where the reflection probability ranged from 0 to 1. As per our previous experiments, eight beacons were used, sensor noise was set to 1 cm standard deviation, and there were no occlusions. Once again, we observe that the Pseudorange Filter outperforms the Doppler Filter. We see that the Pseudorange Filter is able to remain relatively stable for all of the experiments, while the Doppler Filter diverges at a low reflection density. The two peaks on the Pseudorange line in Figure 5.14(b) are the result of the algorithm losing its fix on position part way through the simulation.

The chirp-beacon association statistics are shown in Figure 5.15. Again, we notice that there is a high correlation between the performance of the algorithms and the chirp association statistics. This is highlighted by the two peaks on the curve of the Pseudorange Filter at probabilities of 0.3 and 0.7 in Figure 5.14(b). We can see that these points correspond to dips in the “correct” curve and peaks in the “wrong” and “ignored” curves in Figure 5.15(b). Basically, the more that the algorithms ignore or confuse measurements the higher the error in the system becomes. In the case of the Doppler Filter, the use of the PV-model, combined with the requirement for two reception times, means that these situations are likely to be detrimental.

5.4 Real World Evaluation

In this section we analyse the performance of the positioning algorithms using real-world data. Each experiment starts with the mobile device located at the centre of the

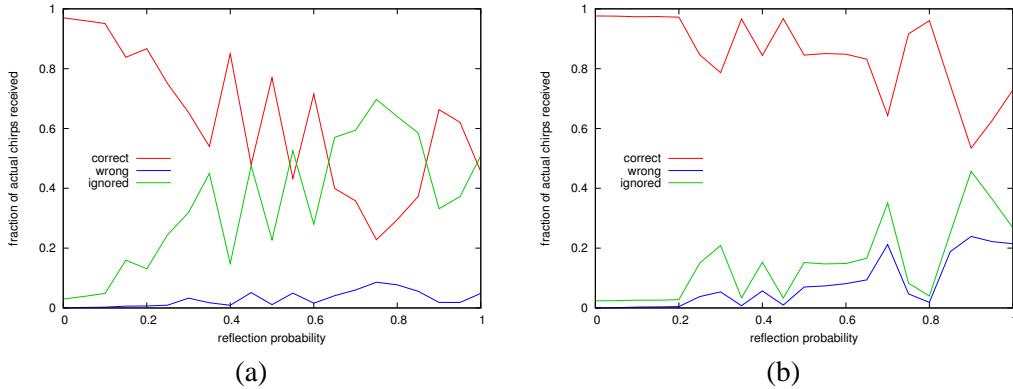


Figure 5.15: Chirp-beacon association statistics versus reflection probability for the (a) Doppler Filter, and (b) Pseudorange Filter.

room. When the initialisation sequence acquires a lock on the chirp sources, the device is then moved over a predetermined test track. To do this, the receiver and wearable computer are situated on the seat of a back-less office chair such that the receiver's X and Y coordinates are aligned with the centre of the chair. During the experiment, the centre of the chair is manually wheeled over floor markers that constitute the test track

Due to the coarse scale of accuracy of the Asynchronous BUZZ, we do not perform a rigorous statistical evaluation of each of the algorithms. We instead choose to make a rough estimation of their performances from the paths that they traverse, and explore and discuss the different characteristics of their output. In essence, we believe that determining whether our algorithms are accurate to, for example, 32 cm or 33 cm adds no value to our evaluation.

We use two different experiments to explore the performance of the Asynchronous BUZZ. The first compares and contrasts the output from three different trials around our test track. The second investigates the behaviour of each algorithm when the position seed for the Kalman filter is varied. Each experiment uses values for the transmission periods P_i that are measured over a relatively long period of time. The values are calculated using five minutes of measurements via the pfind algorithm outlined in Section 5.2.1. The output from this program is detailed in the next section.

i	P	$\text{cov}(P)$	% seen
0	0.500425239729	2.92×10^{-15}	89.9
1	0.508073649879	2.98×10^{-15}	91.2
2	0.515284973571	3.15×10^{-15}	89.8
3	0.523225534779	3.83×10^{-15}	64.3
4	0.531646794763	3.44×10^{-15}	91.1
5	0.539023432771	4.41×10^{-15}	63.1
6	0.547995818198	3.71×10^{-15}	91.8
7	0.556200229854	3.92×10^{-15}	90.5

Table 5.4: Output of the pfind algorithm

5.4.1 Initialisation Data

Table 5.4 summarises the output of the pfind algorithm after processing five minutes of chirp reception times. The P column gives the final value for the beacon periods as estimated by each of the Kalman filters. The final covariance for each filter is provided in the column with the heading $\text{cov}(P)$. The column marked “% seen” gives the percentage of chirps observed by the filters. This value is calculated by dividing the number of chirps integrated with the Kalman filter by the number of chirps it should have seen. The total percentage of observed chirps over all beacons is 84.0%.

We note that the total percentage of observed chirps corresponds with our predictions for collision probability. In Section 5.1.2, we estimated that there is approximately 17% chance of a chirp being lost through collision. Here, we calculate that $100\% - 84\% = 16\%$ of chirps are lost during the execution of the pfind algorithm. In our experience with the beacons in our laboratory environment, this figure has been consistent.

The estimated values for P_i are used with the experiments in the remainder of this section. These experiments use the same receiver circuit as the one used with the pfind algorithm. This ensures that P_i values are correct.

5.4.2 Three Trials

Our first experiment involves the comparison of outputs from three separate trials. In the first trial, the mobile device was moved around the test track 9 times in 219 seconds. In

the second and third trials the device was moved around the track 12 times in 253 seconds and 7 times in 233 seconds, respectively. This equates to, roughly, 24, 21 and 33 seconds per lap for each trial.

Figure 5.16 shows the output of the Doppler Filter and the Pseudorange Filter in each of the three experiments. The test track is shown by the black line. We observe that the Pseudorange Filter clearly outperforms the Doppler Filter in all three trials. Apart from the second trial where it loses its fix half-way through, the Pseudorange Filter reconstructs the path traversed by the mobile device relatively well. We estimate from these plots that it has an accuracy of less than 15 cm for most of the estimates, degrading to around 50 cm in certain sections of the path. In the case where the Kalman filter loses its fix, the system covariance values rise sharply. As mentioned previously, it may be possible to use this fact to identify divergence and perform a recovery operation.

We attribute the poorer performance of the Doppler Filter to the presence of sensor noise and reflections. The simulations in Section 5.3.1 and Section 5.3.4 illustrate the filter's extreme sensitivity to these factors. We estimate that real-world sensor noise, collisions caused by reflections, and the weak constraints of the model combine to reduce its ability to track the position of the device. In general, the algorithm appears to estimate the direction of travel well but that the position and speed are less accurate. For example, the paths created are of relatively similar shape, only distorted in size.

We attribute this distortion to the systematic errors related to the receiver's position in the room. These are the same errors discussed with respect to the Synchronous BUZZ. In particular, errors caused by dilution of precision, distance related attenuation, and transducer misalignment are all larger near the periphery of the space.

To understand how these errors affect the output of the Doppler Filter, we construct a 2D example. Consider the case when a receiver is far away from a beacon and the misalignment angle, distance, and DOP are large. This point is shown in Figure 5.17 as point *A*. We assume that, at this point, there is a timing error of δt seconds. Now assume that the mobile device is in transit to point *B*, where it receives another measurement. Here, the misalignment angle is low, as is the beacon-receiver distance (although it changes less than the angle). We assume that the measurement error at point *B* is zero. To calculate ΔP we subtract the first reception time and the period from the second reception time:

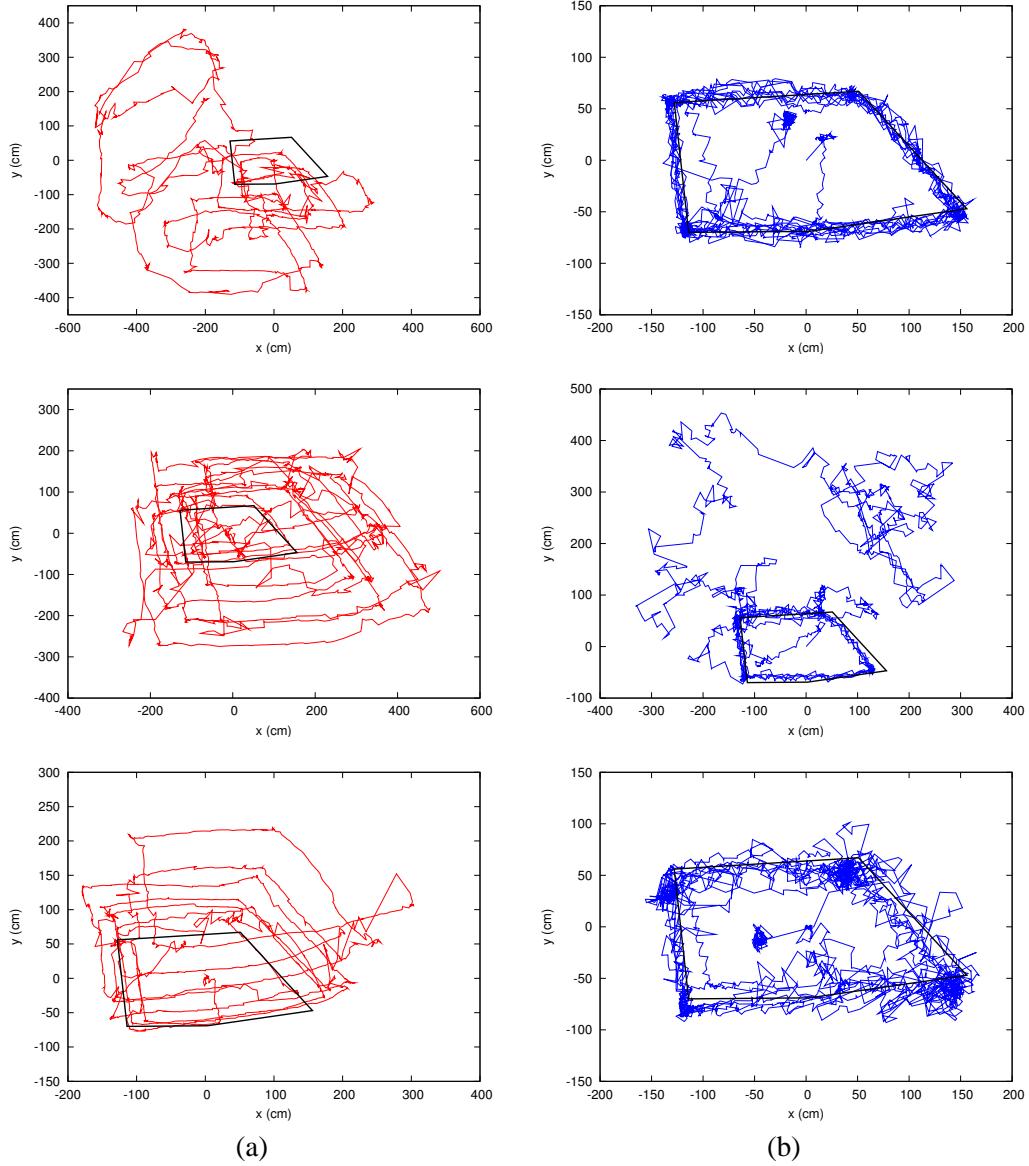


Figure 5.16: Path estimated by the (a) Doppler Filter and (b) Pseudorange Filter for three trials.

$$\Delta P = R_2 - R_1 - P$$

However, if we include the error caused by the larger distance and angle, R_2 will contain the offset δt :

$$\Delta P_{\text{new}} = R_2 - (R_1 + \delta t) - P = \Delta P - \delta t$$

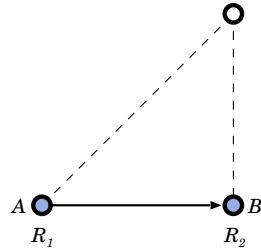


Figure 5.17: As the receiver moves towards the beacon, timing delays caused by attenuation and transducer misalignment decrease.

This means that the value of ΔP will appear to be smaller when the systematic error is present. The reverse is true if the receiver is travelling in the opposition direction, i.e. ΔP will appear to be larger by δt .

What this means in terms of the Doppler model is that the receiver will appear to move more slowly as it travels towards a beacon, and more quickly as it travels away. This systematic error makes its way into the state of the filter, causing the algorithm to over- or underestimate the velocity of the mobile device. The error in velocity is then accumulated in the position estimates such that the Doppler filter continuously over- or under-shoots parts of the path during the experiment.

We have discussed this behaviour previously with respect to the process models used by the Synchronous BUZZ. Specifically, systematic measurement errors cause more of a disturbance in process models containing higher order terms. This is due to the fact that the extended Kalman filter places more weight on the state when terms such as velocity are present in the model. Therefore, when the state becomes corrupted as a result of the systematic errors, it is more difficult for the filter to correct itself using future measurements.

Since the Pseudorange Filter uses a position-only process model, it does not suffer as badly from the systematic errors. The fact that it also uses a different measurement model also means that the systematic errors contribute differently to the estimates. Rather than updating position and velocity, as is the case with the Doppler Filter, the measurements of the Pseudorange Filter update position and beacon transmission times. This means that systematic measurement errors are absorbed in the transmission times, which do not affect position through the process model. In particular, the error in the transmission times do not accumulate as position errors in the same way as errors in velocity.

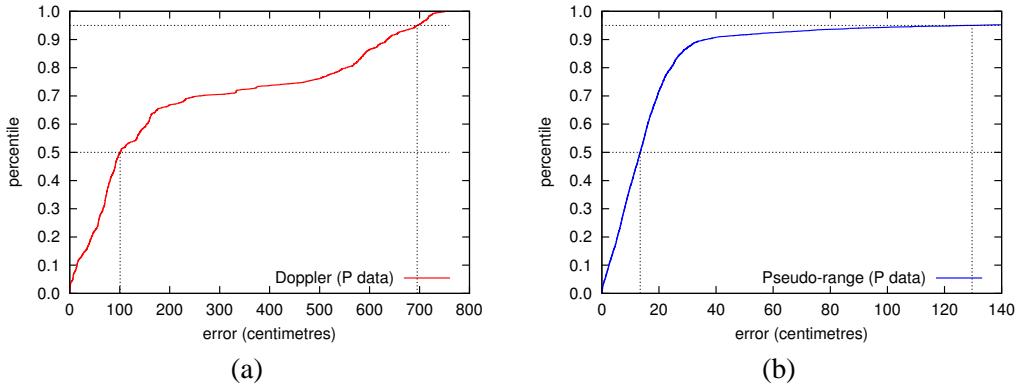


Figure 5.18: Vertical error CDFs for the (a) Doppler Filter and the (b) Pseudorange Filter.

To give a rough idea of the accuracy of the two algorithms, we provide position error CDFs for the vertical (Z) axis. This is possible since the height of the receiver is constant throughout the trials. The two CDFs are shown in Figure 5.18. They reflect what we observe in Figure 5.16. Specifically, the Doppler Filter performs poorly, while the Pseudorange Filter has a median value of 13.4 cm. We can also see that most of the estimates produced by the Pseudorange Filter are within around 50 cm, and that the sharp rise in error comes from the situation when it loses its fix during the second trial.

5.4.3 Position Seed

All of the experiments described thus far have used a good seed for the initial state of the Kalman filter. In other words, we have started the filter off with a position near to the actual starting position of the mobile device. In this section, we perform experiments to determine how errors in the position seed affects performance.

To do so, we use the same three trials from the previous section, but start each of the filters off with the state set to $(100, 100)$ cm. This is different to the actual starting point of $(0, 0)$ cm. Figure 5.19 shows the output from both of the algorithms, where the seed for the filters is depicted by the circle. We notice that the overall performance of the Doppler Filter is basically unaffected; it appears to improve for the first trial, but deteriorates for the second trial.

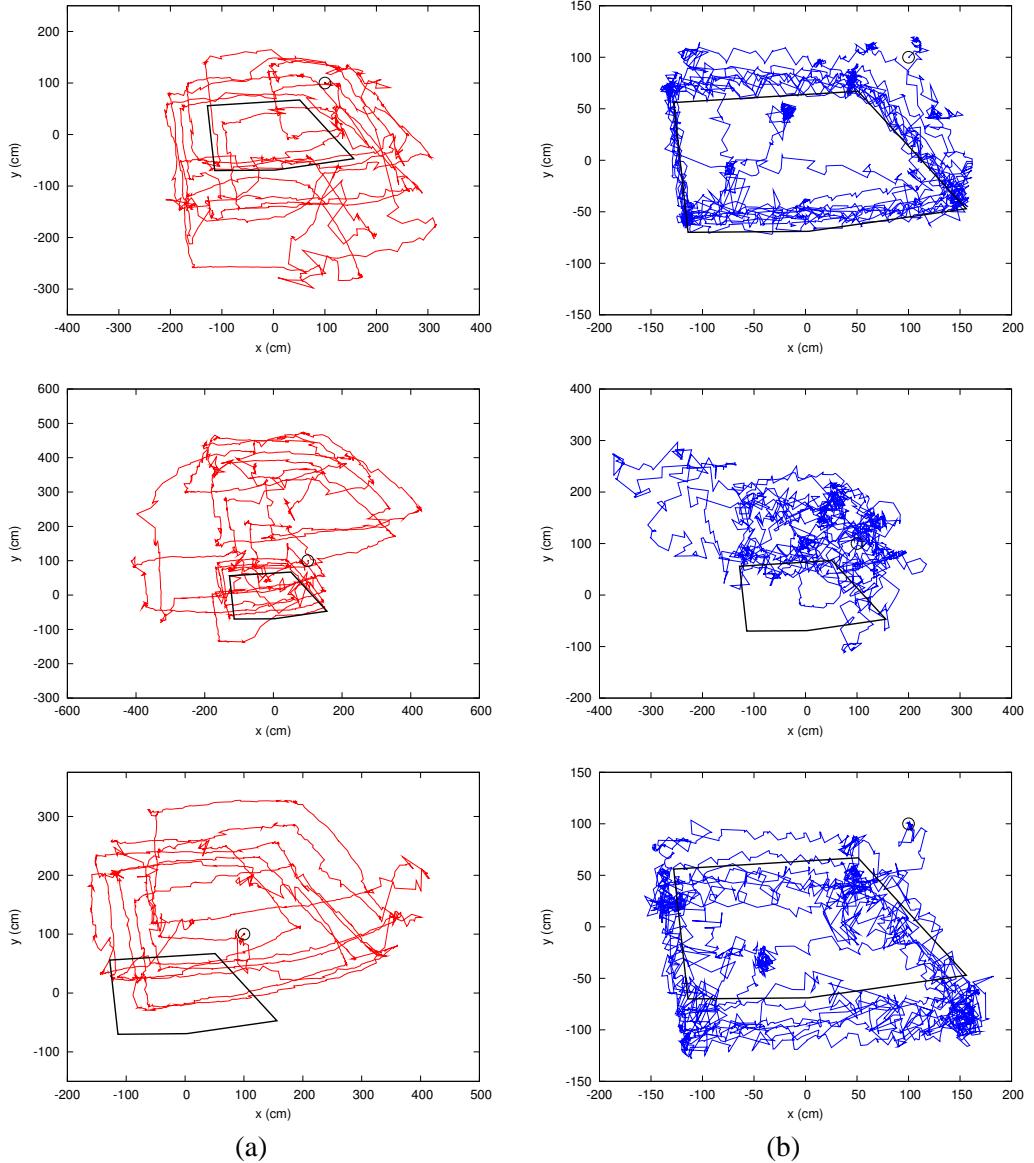


Figure 5.19: Path estimated by the (a) Doppler Filter and the (b) Pseudorange Filter when the position seed is set to (100, 100) cm.

Compared with the results in Figure 5.16, the Pseudorange Filter performs less well with a poor position seed. We observe in the second trial that the filter is unable to find a fix at all. In the first and third trials, the algorithm still manages to trace the path of the ground-truth, however, we notice that it is slightly offset along the Y-axis. We attribute this to the fact that the beacons are not spread as wide along the Y-axis as they are on the X-axis. The result is a higher dilution of precision, and therefore larger position errors,

in the Y -direction (this error pattern can be seen in all of the real-world experiments described in this, and the previous, chapter).

Compared with the Synchronous BUZZ, the transmission times of each of the beacons is less constrained by the measurement equations. For example, the only common variables shared between them are those that describe the receiver position. This means that E_i do not converge as strongly as they do with the Synchronous BUZZ, and are therefore more sensitive to errors in the position seed. What we observe from this experiment is that the algorithm takes an extended period of time to lock on to the transmission time values. From the data, we estimate that it takes roughly 30 to 60 seconds for the algorithm to converge to the appropriate times. While we can see that the algorithm can, for the most part, cope with poorly seeded position values, we recommend that, for use in practice, the error in the seed be as low as possible.

5.5 Movement Limitations

One of the defining features of the Asynchronous BUZZ is that the chirps only provide information about position when the mobile device is moving. This can be seen from Equation 2.4, the basis for the Doppler method. Basically, if the receiver is not moving then measurements for ΔP will be zero. Since the unit vector $(\mathbf{X} - \mathbf{T})/\|\mathbf{X} - \mathbf{T}\|$ cannot be zero, the velocity \mathbf{V} must therefore be zero. If this is true, the position of the mobile device \mathbf{X} can take on any value and the equation will still hold.

To explore how a stationary receiver affects the performance of the algorithms, we ran a 15-minute simulation where the receiver was in motion for the first five minutes, stationary for the middle five, and in motion again for the remaining five minutes. The simulation uses the same eight-beacon configuration as the benchmark simulations (Figure 5.5) and there were no occlusions or reflections. The sensor noise was set to a standard deviation of 1 cm.

The 3D positioning error plotted over the duration of the simulation is shown in Figure 5.20. The three time periods are separated by vertical lines. Output from two versions of each algorithm—with and without accurate P values from pfind—are provided. For the trials using coarse estimates for P , we notice that the error steadily increases

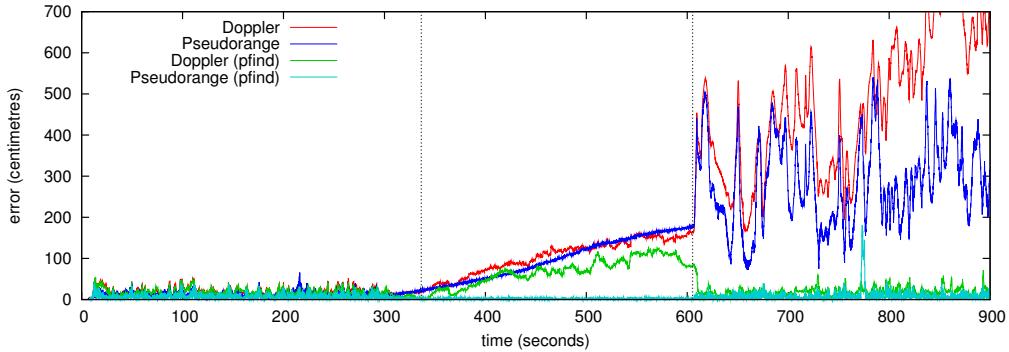


Figure 5.20: 3D position error versus time. The receiver is stationary in the middle time period.

for both filters while the receiver is motionless, then becomes unstable when it begins moving again. The lack of information provided by the measurements causes the state to drift off track. As predicted for the Doppler Filter, the covariance for position rises quickly when the receiver comes to a stop, while the covariance for velocity remains stable. For the Pseudorange Filter, the system covariance creeps up as a result of sensor noise and errors in P_i . Without measurements, these errors begin to accumulate in the transmission times E_i such that the beacon-receiver ranges are sent off-course. When information starts coming in again as the receiver moves, the drifted states of both systems no longer agree with the measurements. As a result, the filters diverge.

One possible way of overcoming this issue is to only accept measurements that have enough “information” in them. To implement this, we propose that low information measurements could be identified as having low χ^2 values. If such a measurement is recognised, the system would be updated by keeping the system covariance and the state the same, while updating the time-related variables in the system. For the Doppler Filter, the chirps would continue to be labelled but the position and velocity of the receiver would not be updated. It is also possible to monitor the velocity related state variables of the Doppler Filter to identify when the receiver is stationary. In the case of the Pseudorange Filter, updating the state would involve keeping the position constant and deriving the transmission times from the position. This would prevent the error in the transmission periods from accumulating in the transmission times.

We note that it takes some time for the error to climb to an irrecoverable state. In our experience with the algorithms, short periods of motionlessness—of around 30 seconds

to a minute—do not cause much trouble. Also, the rate at which the Pseudorange Filter degrades is related to the error in P_i . If we use the pfind algorithm to obtain more precise values for P_i , the time it takes for the algorithm to reach an irrecoverable state will increase. As can be seen from the line labelled “Pseudorange (pfind)” in Figure 5.20, the error in the Pseudorange Filter does not increase when precise values of P_i are used. We also observe that using these values improves the performance of the Doppler Filter, as per the line labelled “Doppler (pfind)”. The improvement occurs as a result of decreased system covariance at the end of the first time frame. Basically, while the error still increases throughout the length of the stationary period, it does not get high enough to be irrecoverable.

5.6 Discussion and Conclusions

We have observed a number of differences between the Synchronous BUZZ and the Asynchronous BUZZ. In general, the Asynchronous BUZZ is less accurate and less stable in live experiments. The weak constraints of both the Doppler and Pseudorange models make it difficult for the extended Kalman filters to get a good fix on position. This is made increasingly challenging by colliding chirps, sensor noise, and an asynchronous measurement stream.

There are also constraints on the operation of the system. For example, it is necessary to make sure that the receiver is in motion throughout the duration of execution. While small time frames of motionlessness are not damaging, longer time frames cause the filters to drift and subsequently diverge. In addition, the user must also make sure that the position state variables are well seeded at startup. This ensures that the algorithms achieve a lock on the process as early as possible, improving the stability and accuracy of the systems.

It is clear from our results that the Pseudorange Filter is superior to the Doppler Filter for use in practice. Unfortunately, measurement errors present in the real-world as well as collisions and occlusions produce adverse effects in the Doppler filter. These effects render it mostly useless as a positioning algorithm in practice.

It may be that the extended Kalman filter is the wrong type of estimator for this type system. The ambiguity of measurement sources, large systematic errors, and weak constraints mean that the filter is always teetering on the edge of divergence. If something goes wrong—for example, a chirp is misclassified or there is a time period when a large number of signals are occluded—the extended Kalman filter can easily slip into an irrecoverable state. A multi-modal estimator, such as the particle filter, may be better able to cope with these situations. The drawback to such algorithms, however, are the large resource demands that they impose.

The number of multiplications required by the filters for each received measurement depends on whether the algorithm integrates the measurement or ignores it. For the Doppler Filter, the number of multiplications per measurement is 1377. This goes up to 1779 if the measurement is integrated. The respective values for the Pseudorange Filter are 4823 and 6561. The difference between the filters' number of multiplications is attributed to the size of their state vectors. The Doppler Filter has six state variables, while the Pseudorange Filter has $N + 3$, where N is the number of beacons. For eight beacons, the Pseudorange filter has a state vector that contains 11 variables. Due to the constraints of the channel, this value should not increase. Specifically, adding more beacons to the positioning space is inadvisable since it will increase the number of collisions.

We estimate that the power consumption of the Asynchronous BUZZ will be larger than that of the Synchronous BUZZ. Despite having a lower average chirp period, the Asynchronous BUZZ must perform prediction calculations for each signal that arrives, including those that are reflections. The Synchronous BUZZ does not do this since, after its initialisation stage, chirp beacon association is fully determined. Additionally, the Asynchronous BUZZ has a larger number of multiplications per signal, which further contributes to power consumption. We have not calculated the exact battery-life of the mobile device since doing so would require the device to be in constant motion for several hours. We roughly estimate that the batteries will last half as long as the Synchronous BUZZ (approximately 3.5 hours). The exact calculation is left for future work.

The use of the Asynchronous BUZZ in a real-world environment is also future work. We have yet to put either of the algorithms to full use in a location based application. It remains to be seen whether the compromises on performance and operation are worth the benefits of the Asynchronous BUZZ's low power, low profile infrastructural hardware.

Nonetheless, we have shown that it is possible to use asynchronous ultrasonic beacons as a basis for a positioning system. The introduction of a covariance monitoring and recovery facility, such as the one used by the Synchronous BUZZ, may remove the issues regarding instability and divergence. However, as discussed in Section 5.3.1, this may be a nuisance if the user is forced to put the mobile receiver at a predefined location.

Chapter 6

Conclusion

In this thesis we have presented two indoor ultrasonic positioning system architectures. They are unique in that they only employ narrowband ultrasound as a signalling mechanism. Each system has been designed with a specific application in mind. The Synchronous BUZZ is aimed at applications that require higher quality positioning accuracy. Its measured three-dimensional SEP figures are 4 cm, 50% and 10 cm, 95%. It operates at an update rate of 33 Hz. The Asynchronous BUZZ is designed for applications that put more emphasis on aesthetics than accuracy. It has a wire-free infrastructure which means that it can be more easily deployed in places such as museums and historic buildings. It provides position estimates within 50 cm of accuracy, where most estimates are accurate to around 20 cm.

The systems have been evaluated using a number of novel techniques. The techniques are simple to employ yet provide an effective analysis of the properties of the systems. We have used them to analyse the different algorithms developed for each system. For the Synchronous BUZZ, the strengths and weaknesses of the P-filter and the PV-filter have been discussed. We have also done the same for the Asynchronous BUZZ with the Doppler Filter and the Pseudorange Filter.

In the remaining sections in this chapter, we highlight our research contributions, discuss the limitations of the BUZZ systems and outline directions for future research.

6.1 Contributions

We have demonstrated that narrowband ultrasound can be used to perform indoor positioning on wearable computers. Our approach is novel in that it is done without the aid of any electromagnetic synchronisation mechanism such as RF or infrared. This means that our devices are smaller and consume less power. The systems we have described use transmission patterns that communicate timing information from the infrastructural systems to the receiving mobile devices. With knowledge of these patterns, our algorithms use time-of-flight based ranging models as well as the relative velocity model to compute position.

The Synchronous BUZZ has been designed with an emphasis on performance. It aims to get the most use out of the 40 kHz narrowband ultrasonic transmission channel by tightly multiplexing its signals. It is able to do this through the use of a central unit in the infrastructure that controls the activity of transmitting transducers wired to it. We believe that the Synchronous BUZZ's design allows for optimum performance for a positioning system that relies solely on ultrasonic measurements (i.e. no inertial sensors). To date, it has been used successfully with gaming applications and art installations. The system is also cheap compared to systems with similar performance. The component cost for the infrastructure is approximately \$150 USD, while a complete Gumstix wearable system is also around \$150 USD.

The Asynchronous BUZZ has been designed with an emphasis on aesthetics. By using a wire-free infrastructure the system can be installed in settings where a wired system is prohibited. The Asynchronous BUZZ operates using beacons that transmit chirps at fixed but unique periodicities. This allows the receiving hardware to differentiate the source of the signals as well as measure the change in the transmission periods as it moves around the room. The system is unique in that the beacons are extremely low-power while at the same time providing enough information to calculate position to within 20 cm to 50 cm. Again, no electromagnetic signals are used. The component cost for each beacon is around \$10 USD (the cost for the receiving system is the same as for the Synchronous BUZZ).

We have used a number of novel evaluation techniques to quantify the performance of both of the systems. A simulator has been created to test each system under various

environmental and operational conditions. The simulator uses a constrained random-walk to generate realistic paths for the receiver. Sensor noise, reflections, occlusions as well as beacon locations can all be controlled within the simulator's environment.

In the real world, the Path Comparison method provides us with a way of determining relative performance between different algorithms and different parts of the room. The Camera Method allows us to analyse the time related performance of the Synchronous BUZZ, which is not captured by the Path Comparison Method. Based on our results, we have provided recommendations for the use of the different algorithms for each of the systems.

6.2 Limitations

The use of iterative state space methods with the BUZZ systems means that the main task of the algorithms is to maintain an estimate for the state. As a result, the state estimate for each algorithm must be initialised as well as monitored throughout the duration of execution. The serial nature of the measurement streams, combined with unknown beacon-chirp association, means that there is no straightforward way to initialise either set of algorithms. The solution that we have used so far is to force the user to place the mobile device at a fairly specific, albeit user-defined, location at start-up. From this point, the extended Kalman filter is able to get a lock on position as well as other variables.

We note that this restriction can be a problem for certain applications and spaces. If either system were to be deployed in a large room, for example, it may be difficult to coordinate the initialisation of devices at specific locations. Furthermore, if a mobile device were to lose its fix and needed to be restarted, forcing users to return to a location could be a nuisance.

It may be possible to eliminate the initialisation problem with the Synchronous BUZZ if a closed form solution to our pseudoranging equations can be found. Additionally, we note that once the mobile device has found a lock on the position and transmission variables, it should never actually lose its fix. Specifically, given that P is a constant value, it is possible to work out the distances to each beacon by extrapolating E into the

future. By performing a trilateration operation on these distances, the position of the device is recoverable. If the receiver is moving there will be some error in the position calculation due to assumptions of simultaneity. However, the result should be good enough to reinitialise the extended Kalman filter and reestablish the fix. This approach would improve on the current method of trying to reinitialise using the last best estimate for position.

In the case of the Asynchronous BUZZ, the experiments in Section 5.4.3 highlight the importance of a good start-up seed. There appears to be no way to get around the initialisation or fix-loss problem without starting the algorithms over again at a specific location. It was hoped that the Doppler model would provide a method of calculating position directly from the measurements (i.e. without having to solve for any variables describing the properties of the beacons). However, the systematic sensor noise present in the measurements causes the Doppler Filter to diverge. As a result, the Doppler Filter is ineffective as a positioning algorithm in the real-world.

Another limitation of the Asynchronous BUZZ is its dependence on motion to be able to calculate position. While the extent to which this limitation affects usability is still unknown, we predict that it should not be too much of an issue. In particular, mobile devices are intended to be mobile and we expect that they will not be stationary for longer than several seconds in a typical application. If the receiver does stop moving, the simulations in Section 5.5 show that the algorithms are able to cope with up to 15 minutes of motionlessness.

As out-of-the-box systems, the Synchronous BUZZ and the Asynchronous BUZZ still require additional work. In particular, the set-up and calibration of the systems is complicated and currently requires the expertise of an engineer. The most difficult part of the process is the measuring of beacon locations in the infrastructure. Manual approaches using measuring tapes, plumb-bobs or theodolites are time consuming and inefficient. The most effective method of calibration would be one that is performed automatically. While there is work being carried out in this field, for example research by Duff [32], positioning and auto-calibration systems have yet to be combined effectively as one package.

6.3 Further Research

Observing the Asynchronous BUZZ in practice will highlight how well it operates in real-world scenarios. We plan to use it within an environment such as the Red Lodge in Bristol, which is characterised as an aesthetically sensitive space prohibiting the use of wired beacons. The Asynchronous BUZZ is ideal for this type of space and we are interested to see how well it fares as the basis for a location based tour guide. We are currently discussing with the curator plans for deployment of a full working system within the Great Oak Room. Our intention is to observe

- how robust the system is over a prolonged period of use,
- how much difficulty the initialisation requirements pose, and
- how the algorithms cope with the movements of real users.

The development of such a system will also require some consideration as to how the users will view or listen to the content. We envisage the use of a palm sized display attached to the current mobile device combined with a set of headphones. An ideal place for the ultrasonic receiver would be on top of the head, attached to the headphones. This should keep it constantly in view of the beacons hidden throughout the room.

With respect to applications of the type intended for the Red Lodge—tourist applications within historic buildings or museums—it would be an advantage to use a positioning system that can span multiple rooms. While the Synchronous BUZZ was only designed with one-room set-ups in mind, multiple installations of the system, one in each room, could be used to create a building-wide system. As the user moves from room-to-room, the algorithms could switch between infrastructures and coordinate frames, performing a type of hand-off. However, this is a complicated approach and requires a large amount of configuration. The problem of determining the location of the mobile device from a cold-start initialisation is also made more complex due to the fact the algorithms need to know which beacon infrastructure is currently in view.

A wire-free system, such as the Asynchronous BUZZ, may be more applicable to multi-room applications. If the configuration was designed carefully, the mobile device may

be able to determine the room in which the user is located based on the beacon periodicities it observes. For example, if the eight beacons placed in one room have different periods to the eight placed in another, the device may be able to identify which eight it is currently receiving. Whether or not this is feasible, and whether positioning can be performed after the room has been determined, is an interesting topic for future research.

Another area of future research is in fusing the positioning systems with inertial sensors, such as accelerometers and gyroscopes. It is well known that these types of sensors complement positioning systems well due to the characteristics of their errors [48]. In combined systems, the inertial sensors provide a way of interpreting motion between ultrasonic measurements, reducing the error introduced by prediction. Likewise, the drift of the inertial sensors is corrected by measurements within an absolute reference frame given by the range or pseudorange measurements of the positioning system.

Indeed a number of location systems, such as the Constellation [44], have been integrated with inertial sensors to achieve impressive results. For the Synchronous BUZZ, inertial sensors in the mobile device would provide the algorithms with the ability to estimate position as well as rotation, giving output with six-degrees-of-freedom. The extra sensory information would allow the system to model location and orientation related errors in the pseudorange measurements. This would reduce the systematic errors observed in our experiments in Section 4.4. Additionally, the inertial sensors could increase the update rate of the system to further improve accuracy. However, the increased processing would consume more of the mobile device’s resources and would have to be balanced with the demands of the application using it.

For the Asynchronous BUZZ, inertial sensors may help to “connect the dots” between long periods without measurements. The extra information may be able to help in disambiguating chirps that arrive at the same time since the motion of the receiver will be known. This should decrease the number of ignored chirps and enhance the system’s robustness. Inertial sensors would also be able to tell the algorithm when the mobile device is stationary, helping it to overcome the constant-motion limitation. This does not require a sophisticated inertial sensor either. It need only be able to distinguish between moving and stationary states of the mobile device.

Given that the BUZZ systems, especially the Synchronous BUZZ, have such a small resource footprint, it is worthwhile considering whether the algorithm would work on a

lower specification device than the Gumstix. For example, the use of a digital signal processor based system or even a custom designed chip should provide for a more efficient, smaller device. Similar to modern GPS receivers, the entire system could be housed in one package. This would further serve to make the systems suitable for deployment as an off-the-shelf product.

Bibliography

- [1] *International Snow Science Workshop (ISSW 2000), 2-6 October 2000, Big Sky MT, USA*, 2000.
- [2] *7th International Symposium on Wearable Computers (ISWC 2003), 21-23 October 2003, White Plains, NY, USA*. IEEE Computer Society, 2003.
- [3] *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*. IEEE Computer Society, 2003.
- [4] K. Aksnes, P.H. Andersen, and E. Haugen. A Precise Multipass Method for Satellite Doppler Positioning. *Celestial Mechanics and Dynamical Astronomy*, 44(4):317–338, 1988.
- [5] Applied Data Systems Inc. Bitsy’s User Manual. Addendum 110111-1003A-2, March 2002. http://www.applieddata.net/developers/documents/110111-1003A-2_excerpt_Bitsy_power_specs.pdf.
- [6] Applied Data Systems Inc. Website - Bitsy Specifications. http://www.applieddata.net/products_bitsySpec.asp, June 2006.
- [7] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM*, pages 775–784, 2000.
- [8] H. Balakrishnan, R. Baliga, D. Curtis, M. Goraczko, A. Miu, N. Priyantha, A. Smith, K. Steele, S. Teller, and K. Wang. Lessons from Developing and Deploying the Cricket Indoor Location System, November 2003.
- [9] S. Bancroft. Algebraic Solution of the GPS Equations. *IEEE Transactions on Aerospace and Electronic Systems*, 21(7):56–59, January 1985.
- [10] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association*. Academic Press Professional, Inc., San Diego, CA, USA, January 1988.
- [11] Michael Beigl, Stephen S. Intille, Jun Rekimoto, and Hideyuki Tokuda, editors. *UbiComp 2005: Ubiquitous Computing, 7th International Conference, UbiComp*

2005, Tokyo, Japan, September 11-14, 2005, Proceedings, volume 3660 of *Lecture Notes in Computer Science*. Springer, 2005.

- [12] Marek Bell, Matthew Chalmers, Louise Barkhuus, Malcolm Hall, Scott Sherwood, Paul Tennent, Barry Brown, Duncan Rowland, and Steve Benford. Interweaving Mobile Games with Everyday Life. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 417–426, New York, NY, USA, 2006. ACM Press.
- [13] Barry Brown, Matthew Chalmers, Marek Bell, Malcolm Hall, Ian MacColl, and Paul Rudman. Sharing the Square: Collaborative Leisure in the City Streets. In *ECSCW 2005: Proceedings of the Ninth European Conference on Computer-supported Cooperative Work, 18-22 September 2005, Paris, France*, pages 427–447, 2005.
- [14] Barry Brown, Ian MacColl, Matthew Chalmers, Areti Galani, Cliff Randell, and Anthony Steed. Lessons from the Lighthouse: Collaboration in a Shared Mixed Reality System. In Cockton and Korhonen [26], pages 577–584.
- [15] Robert Grover Brown and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, Inc, third edition, 1997.
- [16] Wolfram Burgard, Dieter Fox, Hauke Jans, Christian Matenar, and Sebastian Thrun. Sonar-Based Mapping of Large-Scale Mobile Robot Environments using EM. In Ivan Bratko and Saso Dzeroski, editors, *ICML*, pages 67–76. Morgan Kaufmann, 1999.
- [17] J. Chaffee and J. Abel. On the Exact Solutions of Pseudorange Equations. *IEEE Transactions on Aerospace and Electronic Systems*, 30(4):1021–1030, October 1994.
- [18] Y.T. Chan and F.L. Jardine. Target Localization and Tracking from Doppler-Shift Measurements. *IEEE Journal of Oceanic Engineering*, 15(3):251–257, 1990.
- [19] Y.T. Chan and J.J. Towers. Passive Localization from Doppler-Shifted Frequency Measurements. *IEEE Transactions on Signal Processing*, 40(10):2594–2598, 1992.
- [20] Y.T. Chan and J.J. Towers. Sequential Localization of a Radiating Source by Doppler-Shifted Frequency Measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 28(4):1084–1090, 1992.
- [21] Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday, and Christos Efstathiou. Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences. In *CHI*, pages 17–24, 2000.
- [22] A. Chiuso, H. Jin, P. Favaro, and S. Soatto. MFm: 3-D Motion From 2-D Motion Causally Integrated Over Time. In *Proceedings of the 6th European Conference on Computer Vision (ECCV 2000)*, pages 734–750, 2000.

- [23] Christopher R. Wren. Perspective Transform Estimation. <http://alumni.media.mit.edu/~cwren/interpolator/>, November 2006.
- [24] C. K. Chui and G. Chen. *Kalman Filtering with Real-time Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1987.
- [25] Woo Cheol Chung, D.S Ha, and D. Ni. Performance Evaluation of an Ultra Wideband Radiolocation System with Directional Beacon. In *Radio and Wireless Conference, 2004 IEEE*, pages 59–62, September 2004.
- [26] Gilbert Cockton and Panu Korhonen, editors. *Proceedings of the 2003 Conference on Human Factors in Computing Systems, CHI 2003, Ft. Lauderdale, Florida, USA, April 5-10, 2003*. ACM, 2003.
- [27] Antonio Criminisi, Ian D. Reid, and Antonio Zisserman. A Plane Measuring Device. In Adrian F. Clark, editor, *BMVC*. British Machine Vision Association, 1997.
- [28] A.J. Davison, Y. González Cid, and N. Kita. Real-Time 3D SLAM with Wide-Angle Vision. In *Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon*, July 2004.
- [29] Andrew J. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *ICCV* [3], pages 1403–1410.
- [30] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.
- [31] Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, January 2001.
- [32] Paul Duff. *Auto-calibrating Ultrasonic Positioning Systems*. PhD thesis, University of Bristol, 2007. In progress.
- [33] Paul Duff, Michael R. McCarthy, Angus Clark, Henk L. Muller, Cliff Randell, Shahram Izadi, Andy Boucher, Andy Law, Sarah Pennington, and Richard Swinford. A New Method for Auto-calibrated Object Tracking. In Beigl et al. [11], pages 123–140.
- [34] Paul Duff and Henk L. Muller. Autocalibration Algorithm for Ultrasonic Location Systems. In *ISWC* [2], pages 62–68.
- [35] R. Eberhart and J. Kennedy. A New Optimizer Using Particle Swarm Theory. In *Sixth International Symposium on Micro Machine and Human Science (MHS'95)*, pages 39–43, 1995.

- [36] Austin I. Elazar and Ronald Parr. DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 1135–1142. Morgan Kaufmann, 2003.
- [37] Eiman Elnahrawy, Xiaoyan Li, and Richard P. Martin. The Limits of Localization Using Signal Strength: A Comparative Study. In *Proceedings of The First IEEE International Conference on Sensor and Ad hoc Communications and Networks*, October 2004.
- [38] Equator IRC. Website. <http://www.equator.ac.uk/>, August 2006.
- [39] B.T. Fang. Trilateration and Extension to Global Positioning System Navigation. *Journal of Guidance, Control, and Dynamics*, 9(6):715–717, 1986.
- [40] B.T. Fang. Simple Solutions for Hyperbolic and Related Position Fixes. *IEEE Transactions on Aerospace and Electronic Systems*, 26(5):748–753, September 1990.
- [41] Martin Flintham, Steve Benford, Rob Anastasi, Terry Hemmings, Andy Crabtree, Chris Greenhalgh, Nick Tandavanitj, Matt Adams, and Ju Row-Farr. Where Online Meets on the Streets: Experiences with Mobile Mixed Reality Games. In Cockton and Korhonen [26], pages 569–576.
- [42] Dieter Fox, Jeffrey Hightower, Lin Liao, Dirk Schulz, and Gaetano Borriello. Bayesian Filtering for Location Estimation. *IEEE Pervasive Computing*, 2(3):24–33, July-September 2003.
- [43] E. Foxlin. Inertial Head-tracker Sensor Fusion by a Complementary Separate-bias Kalman Filter. In *IEEE Virtual Reality Annual International Symposium*, pages 185–194, April 1996.
- [44] Eric Foxlin, Michael Harrington, and George Pfeifer. Constellation: A Wide-range Wireless Motion-tracking System for Augmented Reality and Virtual Set Applications. In *SIGGRAPH*, pages 371–378, 1998.
- [45] W.H. Foy. Position-Location Solutions by Taylor-Series Estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 12(2):187–194, March 1976.
- [46] I. Getting. The Global Positioning System. *IEEE Spectrum*, 30(12):36–47, December 1993.
- [47] David Gibson, David Oziem, Colin Dalton, and Neill Campbell. Capture and Synthesis of Insect Motion. In *Symposium of Computer Animation*. ACM Siggraph, July 2005.
- [48] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley, Inc., 2001.
- [49] Gumstix Inc. Website. <http://www.gumstix.com/>, June 2006.

- [50] Gumstix Inc. Gumstix Power Spec. http://docwiki.gumstix.org/Power_spec, April 2007.
- [51] D. Hähnel, D. Fox, W. Burgard, and S. Thrun. A Highly Efficient FastSLAM Algorithm for Generating Cyclic Maps of Large-Scale Environments from Raw Laser Range Measurements. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, 2003.
- [52] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W4: Real-Time Surveillance of People and Their Activities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):809–830, 2000.
- [53] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The Anatomy of a Context-Aware Application. In *MOBICOM*, pages 59–68, 1999.
- [54] Mike Hazas and Andy Hopper. Broadband Ultrasonic Location Systems for Improved Indoor Positioning. *IEEE Trans. Mob. Comput.*, 5(5):536–547, 2006.
- [55] Mike Hazas, John Krumm, and Thomas Strang, editors. *Location- and Context-Awareness, Second International Workshop, LoCA 2006, Dublin, Ireland, May 10-11, 2006, Proceedings*, volume 3987 of *Lecture Notes in Computer Science*. Springer, 2006.
- [56] Mike Hazas, James Scott, and John Krumm. Location-Aware Computing Comes of Age. *IEEE Computer*, 37(2):95–97, 2004.
- [57] Mike Hazas and Andy Ward. A Novel Broadband Ultrasonic Location System. In Gaetano Borriello and Lars Erik Holmquist, editors, *Ubicomp*, volume 2498 of *Lecture Notes in Computer Science*, pages 264–280. Springer, 2002.
- [58] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek F. Abdelzaher. Range-free Localization Schemes for Large Scale Sensor Networks. In David B. Johnson, Anthony D. Joseph, and Nitin H. Vaidya, editors, *MOBICOM*, pages 81–95. ACM, 2003.
- [59] John Hereford and Bruce Edgerly. 457 kHz Electromagnetism and the Future of Avalanche Transceivers. In *ISSW* [1].
- [60] Jeffrey Hightower and Gaetano Borriello. Location Systems for Ubiquitous Computing. *IEEE Computer*, 34(8):57–66, August 2001.
- [61] Jeffrey Hightower, Roy Want, and Gaetano Borriello. SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength. UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, February 2000.
- [62] K.C. Ho and Y.T. Chan. Solution and Performance Analysis of Geolocation by TDOA. *IEEE Transactions on Aerospace and Electronic Systems*, 29(4):1311–1322, October 1993.

- [63] L. Hogle, K. Markin, and J.W. Bradley. Navigation Systems Performance Versus Civil Aviation Requirements. *Proceedings of the IEEE*, 71(10):1208–1213, October 1983.
- [64] D.Y. Hsu. *Spatial Error Analysis: A Unified, Application-oriented Treatment*. IEEE Press, 1999.
- [65] Richard Hull, Josephine Reid, and Erik Geelhoed. Creating Experiences with Wearable Computing. *IEEE Pervasive Computing*, 1(4):56–61, 2002.
- [66] IEEE 802.15.4a WPAN Task Group. Website. <http://www.ieee802.org/15/pub/TG4a.html>, November 2006.
- [67] International Satellite System for Search and Rescue. Website. <http://www.cospas-sarsat.org/>, May 2006.
- [68] Michael Isard and Andrew Blake. Condensation – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
- [69] S.J. Julier, J.K. Uhlmann, and H.F. Durrant-Whyte. A New Approach for Filtering Nonlinear Systems. In *Proceedings of the 1995 American Control Conference*, pages 1628–1632, June 1995.
- [70] Rudolph E. Kalman. A New Approach to Linear Filtering and Prediction. *Journal of Basic Engineering (ASME)*, D(82):35–45, March 1960.
- [71] Dongwoo Kang, Young Namgoong, Suckchel Yang, Sungsoo Choi, and Yoan Shin. A Simple Asynchronous UWB Position Location Algorithm Based on Single Round-trip Transmission. In *The 8th International Conference on Advanced Communication Technology, 2006 (ICACT 2006), 20-22 February 2006*, volume 3, pages 1458–1461, February 2006.
- [72] J. Kennedy and R. Eberhart. Particle Swarm Optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.
- [73] J.V. King. Cospas-Sarsat: an International Satellite System for Search and Rescue. *Space Communications*, 18(3-4):139–150, 2002.
- [74] J.V. King. Overview of the Cospas-Sarsat Satellite System for Search and Rescue. *Online Journal of Space Communication*, 4, 2003.
- [75] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.
- [76] Masakatsu Kourogi, Takeshi Kurata, and Katsuhiko Sakaue. A Panorama-based Method of Personal Positioning and Orientation and Its Real-time Applications for Wearable Computers. In *ISWC*, pages 107–114. IEEE Computer Society, 2001.

- [77] John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-Camera Multi-Person Tracking for EasyLiving. In *VS '00: Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, page 3. IEEE Computer Society, 2000.
- [78] John S. Lamancusa and J. Fernando Figueroa. Ranging Errors Caused by Angular Misalignment Between Ultrasonic Transducer Pairs. *Journal of the Acoustical Society of America*, 87(3):1327–1335, March 1990.
- [79] Jr. LaViola, J.J. A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion. In *Proceedings of the 2003 American Control Conference*, pages 2435–2440, June 2003.
- [80] J.L. Leva. An Alternative Closed-Form Solution to the GPS Pseudo-Range Equations. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1430–1439, October 1996.
- [81] Teng Li, Anthony Ekpenyong, and Yih-Fang Huang. A Location System Using Asynchronous Distributed Sensors. In *INFOCOM*, 2004.
- [82] D.E. Manolakis. Efficient Solution and Performance Analysis of 3-D Position Estimation by Trilateration. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1239–1248, October 1996.
- [83] W. Manthey, N. Kroemer, and V. Mágori. Ultrasonic Transducers and Transducer Arrays for Applications in Air. *Measurement Science and Technology*, (3):249–261, March 1992.
- [84] D. Marshall, T. Ward, and S. McLoone. From Chasing Dots to Reading Minds: The Past, Present, and Future of Video Game Interaction. *ACM Crossroads*, 13(2), Fall 2006.
- [85] Martin Rieser. Hosts Website. <http://www.martinrieser.com/Hosts.htm>, September 2006.
- [86] Michael R. McCarthy and Henk L. Muller. RF Free Ultrasonic Positioning. In *ISWC* [2], pages 79–87.
- [87] Florian Michahelles, Peter Matter, Albrecht Schmidt, and Bernt Schiele. Applying Wearable Sensors to Avalanche Rescue: First Experiences with a Novel Avalanche Beacon. *Computers & Graphics*, 27(6):839–847, 2003.
- [88] Keith Mitchell, Duncan McCaffery, George Metaxas, Joe Finney, Stefan Schmid, and Andrew Scott. Six in the City: Introducing Real Tournament - a Mobile IPv6 Based Context-aware Multiplayer Game. In *NETGAMES*, pages 91–100. ACM, 2003.

- [89] Mobile and Wearable Computing Website, University of Bristol. A Walk in the Wireless Woods. <http://wearables.cs.bris.ac.uk/public/wawiwo.htm>, May 2006.
- [90] J. J. Moré. The Levenberg-Marquardt Algorithm: Implementation and Theory. In G. Watson, editor, *Lecture Notes in Mathematics, v630*, pages 105–116. Springer Verlag, 1978.
- [91] Henk L. Muller, Michael R. McCarthy, and Cliff Randell. Particle Filters for Position Sensing with Asynchronous Ultrasonic Beacons. In Hazas et al. [55], pages 1–13.
- [92] Asis Nasipuri and Ribal Najjar. Experimental Evaluation of an Angle Based Indoor Localization System. In *Proceedings of the Second Workshop on Wireless Network Measurements (WiNMee 2006), Boston, MA, USA*, April 2006.
- [93] Dragos Niculescu and Badri Nath. VOR Base Stations for Indoor 802.11 Positioning. In Zygmunt J. Haas, Samir R. Das, and Ravi Jain, editors, *MOBICOM*, pages 58–69. ACM, 2004.
- [94] David Nistér. Preemptive RANSAC for Live Structure and Motion Estimation. In *ICCV* [3], pages 199–206.
- [95] NOAA Satellite and Information Service. Website. <http://www.sarsat.noaa.gov/>, May 2006.
- [96] US Department of Defence and Department of Transportation. 2001 Federal Rationavigation Systems. Public Report DOT-VNTSC-RSPA-01-3.1/DOD-4650.5, National Technical Information Service, Springfield Virginia 22161, 2001.
- [97] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal de Lara. Accurate GSM Indoor Localization. In Beigl et al. [11], pages 141–158.
- [98] B.W. Parkinson and S.W. Gilbert. NAVSTAR: Global Positioning System - Ten Years Later. *Proceedings of the IEEE*, 71(10):1177–1186, October 1983.
- [99] Shwetak N. Patel, Khai N. Truong, and Gregory D. Abowd. PowerLine Positioning: A Practical Sub-Room-Level Indoor Location System for Domestic Use. In Paul Dourish and Adrian Friday, editors, *Ubicomp*, volume 4206 of *Lecture Notes in Computer Science*, pages 441–458. Springer, 2006.
- [100] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, second edition, 1992.
- [101] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket Location-Support System. In *MOBICOM*, pages 32–43, 2000.

- [102] Mark Pupilli and Andrew Calway. Real-Time Camera Tracking Using a Particle Filter. In *Proceedings of the British Machine Vision Conference*, pages 519–528. BMVA Press, September 2005.
- [103] Pyxis Design. Website. <http://www.pyxisdesign.net/>, November 2006.
- [104] Julian Randall, Oliver Amft, Jürgen Bohn, and Martin Burri. LuxTrace - Indoor Positioning Using Building Illumination. *Personal and Ubiquitous Computing Journal*, 2006.
- [105] Julian Randall, Oliver Amft, and Gerhard Tröster. Towards LuxTrace: Using Solar Cells to Measure Distance Indoors. In Thomas Strang and Claudia Linnhoff-Popien, editors, *LoCA*, volume 3479 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 2005.
- [106] Cliff Randell, Paul Duff, Michael McCarthy, and Henk Muller. HeadRacer: A Head Mounted Wearable Computer with Ultrasonic Position Sensing. In *Seventh International Conference on Ubiquitous Computing - Demonstration*, September 2005.
- [107] Cliff Randell, Erik Geelhoed, Alan J. Dix, and Henk L. Muller. Exploring the Effects of Target Location Size and Position System Accuracy on Location Based Applications. In Kenneth P. Fishkin, Bernt Schiele, Paddy Nixon, and Aaron J. Quigley, editors, *Pervasive*, volume 3968 of *Lecture Notes in Computer Science*, pages 305–320. Springer, 2006.
- [108] Cliff Randell and Henk L. Muller. Low Cost Indoor Positioning System. In Gregory D. Abowd, Barry Brumitt, and Steven A. Shafer, editors, *Ubicomp*, volume 2201 of *Lecture Notes in Computer Science*, pages 42–48. Springer, 2001.
- [109] Cliff Randell and Henk L. Muller. Exploring the Dynamic Measurement of Position. In *ISWC*, pages 117–126. IEEE Computer Society, 2002.
- [110] Josephine Reid, Richard Hull, Kirsten Cater, and Constance Fleuriot. Magic Moments in Situated Mediascapes. In *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology ACE 2005*. ACM, June 2005.
- [111] Kay Römer. The Lighthouse Location System for Smart Dust. In *MobiSys*. USENIX, 2003.
- [112] Andreas Savvides, Chih-Chieh Han, and Mani B. Srivastava. Dynamic Fine-grained Localization in Ad-Hoc Networks of Sensors. In *MOBICOM*, pages 166–179, 2001.
- [113] James Scott and Boris Dragovic. Audio Location: Accurate Low-Cost Location Sensing. In Hans-Werner Gellersen, Roy Want, and Albrecht Schmidt, editors, *Pervasive*, volume 3468 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2005.

- [114] James Scott and Mike Hazas. User-Friendly Surveying Techniques for Location-Aware Systems. In Anind K. Dey, Albrecht Schmidt, and Joseph F. McCarthy, editors, *Ubicomp*, volume 2864 of *Lecture Notes in Computer Science*, pages 44–53. Springer, 2003.
- [115] Daniel P. Siewiorek, Asim Smailagic, Junichi Furukawa, Andreas Krause, Neema Moraveji, Kathryn Reiger, Jeremy Shaffer, and Fei Lung Wong. SenSay: A Context-Aware Mobile Phone. In *ISWC* [2], pages 248–249.
- [116] K. Siwiak. Ultra-wide Band Radio: Introducing a New Technology. In *The 53rd IEEE Semianual Vehicular Technology Conference, VTC2001 Spring*, volume 2, pages 1088–1093, 2001.
- [117] Asim Smailagic, Daniel P. Siewiorek, Uwe Maurer, Anthony Rowe, and Karen P. Tang. eWatch: Context Sensitive System Design Case Study. In *ISVLSI*, pages 98–103. IEEE Computer Society, 2005.
- [118] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka B. Priyantha. Tracking Moving Devices with the Cricket Location System. In *MobiSys*. USENIX, 2004.
- [119] R. C. Smith and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1987.
- [120] H.W. Sorenson. Kalman Filtering: Theory and Application. *IEEE Press*, 1985.
- [121] Squid Soup. Website. <http://www.squidsoup.org/>, September 2006.
- [122] P.V. Sudaroli, S. Kannan, and N. Nagarajan. User Position Estimation for Search and Rescue using Satellite in LEO Constellation. *IEE Proceedings on Radar, Sonar and Navigation*, 144(1):22–28, February 1997.
- [123] F. Tschirky, B. Brabec, and M. Kern. Avalanche Rescue Systems in Switzerland: Experience and Limitations. In *ISSW* [1].
- [124] Ubisense Limited. Ubisense - Precise Real-time Location. <http://www.ubisense.net>, May 2006.
- [125] F. van den Bergh. *An Analysis of Particle Swarm Optimizers*. PhD thesis, University of Pretoria, 2001.
- [126] John R. Vig. Introduction to Quartz Frequency Standards. Technical Report SLCET-TR-92-1 (Rev. 1), Army Research Laboratory, Electronics and Power Sources Directorate, Fort Monmouth, NJ 07703-5601, U.S.A., October 1992.
- [127] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

- [128] Andy Ward, Alan Jones, and Andy Hopper. A New Location Technique for the Active Office. *IEEE Personal Communications*, 4(5):42–47, October 1997.
- [129] Greg Welch and Gary Bishop. SCAAT: Incremental Tracking with Incomplete Information. In *SIGGRAPH*, pages 333–344, 1997.
- [130] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, 27599-3175, April 2004.
- [131] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nard Colucci. The HiBall Tracker: high-performance wide-area tracking for virtual and augmented environments. In *VRST*, pages 1–10, 1999.
- [132] M.Z. Win and R.A. Scholtz. Impulse Radio: How It Works. *IEEE Communications Letters*, 2(2):36–38, 1998.
- [133] M.Z. Win and R.A. Scholtz. Ultra-wide Bandwidth Time-hopping Spread-spectrum Impulse Radio for Wireless Multiple-access Communications. *IEEE Transactions on Communications*, 48(4):679–689, April 2000.
- [134] Christopher R. Wren. Perception for Human Motion Understanding. Technical Report TR2004-108, Mitsubishi Electric Research Laboratories, August 2004.
- [135] Christopher Richard Wren, Ali Azarbajayani, Trevor Darrell, and Alex Pentland. Pfnder: Real-Time Tracking of the Human Body. In *FG*, pages 51–59. IEEE Computer Society, 1996.
- [136] Christopher Richard Wren and Emmanuel Munguia Tapia. Toward Scalable Activity Recognition for Sensor Networks. In Hazas et al. [55], pages 168–185.
- [137] J.M. Zagami, S.A. Parl., J.J. Bussgang, and K.D. Melillo. Providing Universal Location Services Using a Wireless E911 Location Network. *IEEE Communications Magazine*, 36(4):66–71, April 1998.
- [138] Yilin Zhao. Standardization of Mobile Phone Positioning for 3G Systems. *IEEE Communications Magazine*, 40(7):108–116, July 2002.