An inexpensive hyperbolic positioning system for tracking wildlife using off-the-shelf hardware

SW Krüger

22036784

Dissertation submitted in fulfilment of the requirements for the degree Masters in Computer and Electronic Engineering at the Potchefstroom Campus of the North-West University

Supervisor: Prof ASJ Helberg

Co-supervisor: Dr PP Krüger

May 2017

Abstract

In this dissertation, a design is presented for a Time Difference of Arrival (TDOA) positioning system that addresses a need in the market for an inexpensive and low-power wildlife positioning system. With this system, the position of a transmitter is determined cooperatively by a group of receiver stations from the differences in the time at which a short-lived transmission reaches the receivers. This enables the use of simple, inexpensive tag devices for which the energy consumption per position estimate is less than a hundredth of a GPS-enabled tag's energy consumption.

TDOA positioning requires precisely synchronised clocks at the receivers in order to relate the arrival time at different receivers to one another, which increases the cost and complexity of the receivers. A design with novel techniques that enable the use of simple, low-cost receivers with unsynchronised, inaccurate clocks is presented in this study. Arrival time estimates from different receivers are calibrated in software with the aid of periodic transmissions from one or more beacon transmitters. Furthermore, OOK modulation is used to enable fast frequency offset recovery.

A prototype implementation of the design was developed with easily reproducible receiver stations constructed from low-cost, off-the-shelf, general-purpose hardware modules. An inexpensive software defined radio device, the RTL-SDR, was used and signal processing was performed in software. Techniques for improving the precision and for reducing the computational requirements of the signal processor were devised, analysed and compared. Software was developed for performing fast real-time signal processing on an inexpensive single-board computer, the Raspberry Pi 3. Moreover, software was developed for calculating position estimates from arrival time estimates and for analysing the data. The code has been released as open-source software to allow collaboration with research groups working in similar directions and to facilitate further experimentation.

A pilot field test was conducted with two receivers spaced 9 km apart. The standard deviation of the TDOA estimates was found to be 11.5 ns, which is equivalent to a precision of 3.5 m for two-dimensional position estimates. In comparison with similar positioning systems, the system presented in this dissertation has a lower power consumption and is more than an order of magnitude cheaper, while similar positioning accuracy is being achieved.

The techniques and software that were developed is not limited to wildlife tracking, but can be adapted for other applications such as livestock monitoring, asset tracking and passive radar.

Keywords: *TDOA, radio tracking, localisation, wildlife tracking, subsample interpolation, synchronisation, RTL-SDR*

Samevatting

’n Hiperboliese posisioneringstelsel vir die opsoring van wild met goedkoop, algemeen beskikbare hardeware

In hierdie verhandeling word ’n verskil-in-aankomstyd (VAT) posisioneringstelsel aangebied wat ’n behoefte in die mark vir ’n goedkoop en lae-energie wildopsporingstelstel aanspreek. Met hierdie stelsel word die posisie van ’n sender gemeenskaplik deur ’n groep ontvangerstasies bepaal deur gebruik te maak van die verskil in die tyd waarteen ’n kort uitsending vanaf die sender die ontvangers bereik. Dit maak die gebruik van eenvoudige en goedkoop opsporingstoestelle moontlik waarvan die energieverbruik per posisieskatting minder as ’n honderdste van dié van ’n GPS-geaktiveerde opsporingstoestel is.

VAT-posisionering vereis noukeurige sinkronisasie tussen die ontvangers sodat aankomstye by verskillende ontvangers met mekaar vergelyk kan word, wat die koste en kompleksiteit van die ontvangers verhoog. ’n Ontwerp met nuwe tegnieke wat die gebruik van eenvoudige, laekoste ontvangers met ongesinkroniseerde, onakkurate klokfrekwensies moontlik maak, word in hierdie studie voorgelê. Aankomstydskattings van verskillende ontvangers word in sagteware gekalibreer met behulp van periodiese uitsendings vanaf ten minste een bakensender. Verder word die dragolf met aan-af-sleuteling gemoduleer sodat die draerfrekwensie maklik deur die ontvangers bepaal kan word.

’n Prototipe van die stelsel is ontwikkel met ontvangerstasies wat uit goedkoop, algemeen beskikbare, veeldoelige hardewaremodules saamgestel is. ’n Goedkoop sagteware-gedefinieerde radiotoestel, die RTL-SDR, is gebruik en seinverwerking is as sagteware toegepas. Tegnieke om die noukeurigheid van die seinverwerker te verbeter en om die verwerkingsvereistes te verminder is uitgedink, geanaliseer en vergelyk. Sagteware is ontwikkel om vinnige intydse seinverwerking op ’n goedkoop enkelbordrekenaar, die Raspberry Pi 3, uit te voer. Verder is daar ook sagteware ontwikkel om posisieskattings vanaf aankomstydskattings te bereken en om data te analiseer. Die kode is as oopbronsagteware beskikbaar gestel om samewerking met navorsingsgroepe wat in soortgelyke rigtings werk, toe te laat, en om verdere eksperimentering te vergemaklik.

’n Toetslopie is uitgevoer met twee ontvangers wat 9 km uitmekaar geplaas is. Die standaardafwyking van die VAT-skattings was 11.5 ns, wat ooreenstem met ’n noukeurigheid van 3.5 m vir twee-dimensionele posisieskattings. Die stelsel wat in hierdie verhandeling voorgelê word, is goedkoper en meer energie-doeltreffend as soortgelyke stelsels, terwyl die akkuraatheid van die posisieskattings vergelykbaar is.

Die tegnieke en sagteware wat ontwikkel is, is nie beperk tot wildopsporing nie, maar kan vir ander toepassings soos veemonitering, bate-opsporing en passiewe radar aangepas word.

Acknowledgements

Soli Deo Gloria!

To

my supervisor, Prof. Helberg, for his patience, suggestions, advice and guidance throughout the years,

my co-supervisor, Dr. Paulus Krüger, for his technical guidance, assistance and innovative ideas,

Francois Botha and Jan-Adriaan Cordier from Wireless Wildlife, for the equipment I could use for the study and for their assistance with the field tests,

my parents, for their unceasing love and support and for the opportunity to further my education,

Karen Krüger, for proofreading the work,

thank you, I am deeply grateful.

Great are the works of the Lord,
studied by all who delight in them.
— Psalm 111:2

Contents

1 Introduction 1

1.1 Context . . . 1

1.1.1 Radio positioning . . . 1

1.1.2 Wildlife radio tracking . . . 2

1.1.3 Hyperbolic positioning . . . 4

1.1.4 Commercial-Off-The-Shelf hardware . . . 6

1.1.5 Related work . . . 6

1.2 Research problem . . . 7

1.3 Research goals . . . 7

1.4 Document structure . . . 8

2 Principles of TDOA radio positioning 9

2.1 Positioning principles . . . 10

2.1.1 Positioning . . . 10

2.1.2 Electromagnetic propagation measurements . . . 11

2.1.3 Radio positioning system classification . . . 13

2.1.4 Parameter estimation . . . 14

2.2 TDOA position estimation . . . 15

2.2.1 Introduction to TOA positioning . . . 15

2.2.2 Introduction to TDOA positioning . . . 16

2.2.3 Analytic solutions . . . 18

2.2.4 Numeric solutions 21

2.3 Dilution of precision (DOP) 22

2.3.1 Introduction to DOP 22

2.3.2 Mathematical definition 23

2.3.3 Jacobian matrix for 2D TDOA positioning 25

2.3.4 PDOP heat maps 25

2.3.5 Conclusion 27

2.4 Arrival time estimation 30

2.4.1 Introduction 30

2.4.2 Positioning signal pulse 30

2.4.3 Cramér–Rao lower bound 31

2.4.4 Spread spectrum 32

2.5 Chapter summary 32

3 Arrival time estimation using DSSS **35**

3.1 An overview of DSSS 35

3.1.1 Introduction 35

3.1.2 Spreading codes 36

3.1.3 Matched filter 38

3.2 Subsample interpolation 40

3.3 Modulation schemes 43

3.3.1 Phase shift keying (PSK) 44

3.3.2 On–off keying (OOK) 47

3.3.3 Frequency shift keying (FSK) 49

3.4 Signal detection 49

3.4.1 Introduction to signal detection theory 50

3.4.2 A detector for a signal in AWGN 52

3.5 Carrier detector 53

3.5.1 Approximate positioning signal as a single tone 54
3.5.2 ML carrier frequency estimation from the DFT 55
3.5.3 Single-tone signal detection 56
3.6 Chapter summary 57

4 Design 59

4.1 Design premises 59
4.2 System architecture 61
4.3 Transmitter 62
4.4 Receiver 64
4.4.1 Slicer 65
4.4.2 Carrier frequency recovery 65
4.4.3 SOA estimator 71
4.5 Positioning server 78
4.5.1 Aggregator 79
4.5.2 Identifier 79
4.5.3 Matchmaker 80
4.5.4 TDOA estimator 81
4.5.5 Position estimator 87
4.6 Chapter summary 87

5 Implementation with RTL-SDR and Raspberry Pi 89

5.1 Transmitter 89
5.1.1 Tag device 89
5.1.2 OOK at high baud rate 90
5.1.3 Software 91
5.1.4 Energy consumption 91
5.1.5 Accurate chip rate 92
5.2 Receiver hardware 92

5.2.1 RF receiver 92
5.2.2 Signal processor 94
5.2.3 Preamplifier 95
5.3 Signal and detection processing software 96
5.3.1 Software architecture 97
5.3.2 Tools 100
5.3.3 Fast carrier detection 100
5.3.4 Fast detector 102
5.4 Prototype positioning system 102
5.4.1 Overview 102
5.4.2 System software 103
5.4.3 Summary of parameters 105
5.5 Chapter summary 105

6 Results 107

6.1 Sanity check 107
6.2 Assessment and comparison of estimation methods 112
6.2.1 Methodology 112
6.2.2 Correlation peak interpolation 114
6.2.3 Carrier peak interpolation 119
6.2.4 TDOA estimation 122
6.3 Pilot field test 127
6.3.1 Methodology 127
6.3.2 Position estimates 128
6.3.3 Carrier frequency 130
6.4 Prototype receiver station 131
6.4.1 Software performance 131
6.4.2 Power consumption 133

6.4.3 Integration test 133

6.5 Verification and validation 134

6.5.1 Definition 134

6.5.2 Methodology 135

6.5.3 Verification 135

6.5.4 Validation 136

7 Conclusion 139

7.1 Overview of work 139

7.2 Key findings 141

7.3 Comparison to related work 142

7.4 Contributions 144

7.5 Recommendations for future work 146

7.6 Concluding remarks 148

List of Figures

2.1 Illustration of the difference between self-positioning and network-positioning systems 14

2.2 Illustration of TDOA in two dimensions 17

2.3 Ambiguous position in two dimensions with three receivers 18

2.4 Geometric illustration of PDOP for a fixed receiver configuration 23

2.5 Heat maps of PDOP for TDOA positioning 29

3.1 Autocorrelation of a length-13 Barker code 37

3.2 Example showcasing the output of a matched filter that matches a length-31 Gold code to a received signal 38

3.3 Output of the correlator when the correlator's step size and the chip boundaries are misaligned by half a sample 39

3.4 Periodogram of a PSK-modulated Gold code with 2047 chips and a chip rate of 1 MHz, normalised to unity power 45

3.5 Example showcasing the effect of a frequency offset between the template and received signal on the output of the correlator 47

3.6 Periodogram of an OOK-modulated Gold code with 2047 chips and a chip rate of 1 MHz, normalised to unity power 49

4.1 Top-level system architecture 61

4.2 Expanded top-level system architecture 61

4.3 Functional system architecture diagram: Transmitter 62

4.4 Functional system architecture diagram: Receiver 64

4.5 Functional flow block diagram: Signal processor 64

4.6 Functional flow block diagram: Carrier frequency recovery 66

4.7 Functional flow block diagram: SOA estimator 72

4.8 Number of complex multiplications per correlator output sample for different block lengths and different template lengths when using the overlap–save method 75

4.9 Functional flow block diagram: positioning server 79

5.1 Transmitter board 90

5.2 Close-up of modification for OOK modulation at a high baud rate 91

5.3 Three different RTL-SDR variants 94

5.4 Preamplifier without its enclosure, shown connected to an RTL-SDR 96

5.5 The flow of data through a pipeline of software modules connected together through their CLI interfaces 98

5.6 Off-the-shelf devices for four receiver stations 102

5.7 Photograph of a receiver station 103

6.1 Estimate of the power spectral density (PSD) of the received signal before carrier frequency recovery 108

6.2 Peak samples of the FFT with a sinc-like interpolation function superimposed . 108

6.3 Magnitude of the time-domain samples after carrier frequency recovery 109

6.4 Extract from the correlator output showcasing the distinct correlation peak and a few multipath components 109

6.5 Gaussian curve fit to three samples surrounding the correlation peak 110

6.6 Comparison between the cross-correlation peak, time-shifted to align with the interpolation peak, and the template's autocorrelation peak 110

6.7 Template signal superimposed on the received positioning signal at the estimated arrival time 111

6.8 Comparison of RMSE of simulated time delay estimates at different SNRs for various correlation peak interpolation methods 116

6.9 RMS of the frequency estimation error at different SNRs for various carrier frequency offset interpolation methods, resulting from a simulation by which a template signal is shifted in frequency 120

6.10 Residuals of a linear and quadratic function fit to a scatter plot of the SOA values of beacon transmissions at one receiver against the corresponding SOA values at another receiver for an extract of the road test data 123

6.11 Comparison of the TDOA estimation errors for an extract of the laboratory test data when only one beacon detection is used to relate SOA values, in contrast with the errors when interpolating between two beacon detections 125

6.12 Position of the mobile transmitter over time during the pilot test, estimated using TDOA positioning from the data that was captured 128

6.13 Carrier frequency offsets of the transmitters that were observed at each of the receivers, estimated from the DFTs of the data blocks 131

List of Tables

5.1 Parameters that were used during the test of the prototype positioning system 105

6.1 Resulting estimated precision and accuracy of the TDOA estimates and runtime of signal processing software when the signal and detection processing software is executed with different correlation peak interpolation methods, evaluated on two sets of captured test data 117

6.2 Estimated precision and accuracy of TDOA estimates resulting from various carrier frequency interpolation methods when evaluated on the test data 121

6.3 Estimated precision and accuracy of TDOA estimates and runtime of signal processing software for different frequency shift algorithms 122

6.4 Standard deviation and mean error of TDOA estimates that are calculated from captured test data using different methods for calibrating SOA values with beacon detections 126

6.5 Standard deviation of the TDOA estimates for different window sizes 127

6.6 The number of position estimations, the mean position, and the standard deviation of the position estimates of the mobile transmitter at the stops between the two receivers 129

6.7 Comparison between the mean position estimates of the TDOA system for each stop and the position estimates calculated from coordinates that were recorded with a GPS device 130

6.8 Results from a performance test of the single-threaded signal processing software on a Raspberry Pi 3 132

7.1 Comparison between the properties of the positioning system presented in this dissertation and the properties of two related systems 143

List of Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
COTS	commercial off-the-shelf
CRLB	Cramér–Rao Lower Bound
DFT	Discrete Fourier Transform
DOP	Dilution of Precision
DSSS	Direct Sequence Spread Spectrum
DTFT	Discrete-Time Fourier Transform
FFT	Fast Fourier Transform
GPSDO	GPS Disciplined Oscillator
LLS	Linear Least Squares
LS	Least Squares
LOS	Line-Of-Sight
MLE	Maximum Likelihood Estimator
NLLS	Nonlinear Least Squares
OOK	On–Off Keying
PDF	Probability Density Function
PDOP	Positional Dilution of Precision

PSK	Phase-Shift Keying
PRN	Pseudo Random Noise
RMSE	Root-Mean-Square Error
SBC	Single-Board Computer
SOA	Sample-of-Arrival
SDR	Software-Defined Radio
SIMD	Single Instruction, Multiple Data
SNR	Signal-to-Noise Ratio
TCXO	Temperature Compensated Crystal Oscillator
TDOA	Time Difference of Arrival
TOA	Time of Arrival
WNLLS	Weighted Nonlinear Least Squares

Chapter 1

Introduction

This chapter serves as the cornerstone of the research described in this dissertation. First, the context of the problem is presented to substantiate the significance of the problem and to guide the reader from the general field of study to the particular research problem. The research problem and objectives are then formulated. Lastly, the structure of the dissertation is outlined.

1.1 Context

1.1.1 Radio positioning

The use of radio waves for position determination, called *Radio Positioning*, can easily be taken for granted nowadays. Billions of people carry devices that can estimate the position of the user using radio positioning technologies. The best known, most prevalent and perhaps epitome of positioning systems is the Global Positioning System (GPS). Every day countless military, civil, commercial and scientific users rely on the position, velocity and time information provided by Global Navigation Satellite Systems (GNSSs) such as GPS [1]. The use of ground-based radio positioning systems such as Loran-C, Omega, and Decca fell rapidly with the introduction of GPS, which resulted in many of these mature systems being decommissioned. However, there are still numerous lesser-known radio positioning technologies that exist in the shadow of GPS. Some of these technologies are used for radio positioning in environments where GNSS signals are weak or cannot reach, such as inside buildings or dense urban environments, or as an alternative to GPS to preserve battery power. Furthermore, GNSSs are well suited for navigation applications, but other radio positioning systems could be more suitable for tracking applications. Navigation refers to ascertaining one's own position, whereas tracking refers to following the movements of someone or something else.

With GPS, a mobile unit determines its own position based on radio signals it receives from multiple satellites. The satellites have no knowledge of the users, but they simply broadcast the signals for anyone to use. The information regarding the mobile unit's position resides at the mobile unit itself. GPS is an example of a *self-positioning* system. In a *self-positioning system* a mobile unit estimates its own position based on signals it receives from transmitters at known locations [2]. In a *network-positioning* system, on the other hand, the mobile unit transmits signals and fixed stations with known positions receive the signals. The mobile units do not have knowledge of their own positions, but the position information resides at the network of receivers. Examples of common use cases of network-positioning systems include military surveillance, mobile phone tracking by the service provider's network infrastructure for emergency services [3], and in aviation for calculating the position of aircraft [4].

Which of the two approaches, self-positioning or network-positioning, is better suited depends on the application. Self-positioning systems are generally better suited for navigation applications, while network-positioning systems are generally a better fit for tracking applications [5]. Network-positioning usually involves a more complex system and infrastructure. However, it allows the mobile unit to be small, inexpensive, and with low energy requirements [6]. These advantages make network-positioning compelling for wildlife radio tracking, which is the primary application we focus on in this dissertation.

1.1.2 Wildlife radio tracking

Since the introduction of the first workable system in the 1960s, radio positioning has become a widespread technology for tracking the movement and behaviour of wild animals [7]. Various motives exist for tracking wildlife. While scientists are mostly interested in the data being collected, private owners of game farms or estates are usually more concerned about the security of the animals and may employ wildlife tracking to help prevent theft or poaching [8]. Game lodges and reserves can track wildlife to help tour guides lead tourists to the animals, or they can use it to help manage their vegetation and habitat resources. The list of animals that are equipped with radio tags and the diversity of the environments in which they are tracked are endless: from lions, elephants and rhinos in Africa, to birds migrating between continents, to fish and reptiles [7].

Design constraints

Even though the implementation of radio positioning is a widespread problem that is used in many different industries and for myriads of applications, wildlife tracking places additional constraints on the weight, size, installation and maintenance of the radio tags. One of the biggest requirements for wildlife tracking is to keep disturbances to the animal to a minimum. There are two primary ways in which wildlife tracking can interfere with the normal life of the animals, namely carrying the tag can be a hindrance to the animal, and installing the tag can be harmful [1]. Carrying a foreign object can easily impede small or light animals' locomotion abilities. This imposes a challenging restriction on the size and weight of tags attached to animals such as birds. Size and weight are less of a concern for large animals like rhinoceros, but capturing and anaesthetising the animal to install or replace a tag is costly, dangerous and may pose a risk to the animal. Thus, for animals that cannot be captured easily, the electronic tag should ideally outlive the animal or the period for which data needs to be captured to ensure that the animal is not exposed to multiple risky tag installations.

For both types of interference — that is, the load caused by the tag and the risky installation process — it is necessary to minimise the energy consumed by the tag's electronics to minimise disturbances to the animal. Tags that consume less energy enable longer service intervals for the same battery, which is advantageous for large animals. It also enables smaller batteries to be used, and thus smaller and lighter tags, which allows smaller animals to be tracked.

Cost is another constraint when tracking wildlife. If the tags are too expensive or if the system is too labour intensive, the study sample size that a limited budget can buy may be too small to be of value to researchers. Furthermore, it is difficult for a private owner or game reserve to justify the cost of a tracking system if the cost is not insignificant in comparison to the value of the animals.

Wildlife tracking technologies

Various technologies for tracking wildlife are in use today. One of the oldest, the conventional Very High Frequency (VHF) system, provides the simplest solution. However, it requires the attention of a human operator who has to identify the direction of the animal by rotating a hand-held antenna in the direction in which the received signal is the strongest. The direction can be used to home in on the animal, or measurements at different locations can be used to triangulate the position of the animal. The requirement of a human operator can be eliminated with the use of fixed receiver stations, commonly involving directional antennas. However, the accuracy of the position estimates may be insufficient, and the inefficient use of energy due to a signal with low information content limits the lifetime of the tag attached to the animal [1].

The miniaturisation of GPS receivers into single-chip systems and the accompanying reduction in size, cost and weight have recently made GPS-enabled tags a feasible and easy solution for tracking wildlife. It is difficult to compete against the accuracy, low cost, small chip size, low power consumption and global coverage offered by modern GPS receivers.

Even though GPS receivers keep getting smaller, more sensitive and more energy efficient, the positioning approach used by GPS, self-positioning, has some inherent drawbacks that make it unsuitable in applications where regular position updates, long service intervals or lightweight tags are desired. Firstly, since the tag receives the signals used for positioning, position estimation has to be performed by the tag itself.[1] Processing the broadband GPS

[1] It is, strictly speaking, not necessary to process the signal on the tag in real-time. Raw digital samples can be captured and processed off-line. However, the energy required to transmit the raw data to a base station for off-line processing is even higher than processing it on the tag itself.

signal is computationally expensive and requires fast signal processing hardware for which the power consumption is a concern. Furthermore, it takes several seconds and up to several minutes to acquire satellite signals and provide a position estimate, also called a *position fix*. The relatively high power consumption of the signal processor integrated over the time it takes to obtain a fix constitutes a demand for energy that is prohibitive when energy consumption per position update is a significant design constraint.

Secondly, once a GPS-enabled tag has obtained a fix, the position information resides on the tag and is thus still out of reach of the user. The information can be stored and retrieved later when the animal is recaptured or when the tracking device has been released by a drop-off mechanism. However, in most cases and especially for security applications, remote monitoring is an essential requirement. A radio communication channel is required to download the data from the tag and to report it to the user. The cost in energy to transmit data from the tag imposes an additional demand on the battery.

Wildlife tracking technologies other than VHF and GPS exist, but they have shortcomings in terms of accuracy or energy consumption, as discussed in [1]. Hyperbolic positioning systems provide an alternative solution and feature tags that are simpler, lighter, cheaper, and longer lasting.

1.1.3 Hyperbolic positioning

As mentioned in Section 1.1.2, the energy consumption of GPS-enabled tags prevents them from being used when minimum energy consumption is desired. As an example, consider the following back-of-the-envelope calculation of the energy consumption that can be expected for a single position estimate when using a GPS-enabled tag. This will be compared to the energy consumption in a network-positioning system in the subsequent example.

Example 1.1. Assume the tag makes use of OriginGPS's Nano Hornet ORG1411 GPS module [9] to acquire a position fix every few minutes. This GPS module has an integrated antenna, measures only $10 \times 10 \times 3.8\,\text{mm}$, and weighs $1.4\,\text{g}$. Assume a power supply voltage of $V_{CC} = 1.8\,\text{V}$. To save energy, the GPS module is only powered when a new position update is required, with the result that the GPS signals have to be reacquired for each fix. The GPS module consumes approximately $I_{acq} = 43\,\text{mA}$ during signal acquisition. Assume an average acquisition time of $t_{acq} = 5\,\text{s}$. The estimated energy consumed by the GPS module for a single fix, E_{acq}, is:

$$E_{acq} = V_{CC} \times I_{acq} \times t_{acq} = 387\,\text{mJ}.$$

Assume Silicon Labs' Si4010 chip [10] is used as RF transmitter to communicate the position information to a base station. If it takes 50 ms to transmit the data for a single position update, and if the chip consumes approximately 20 mA during transmission, the energy used to transmit a fix is 1.8 mJ. The total energy consumed to acquire and transmit a position update is 389 mJ. △

In contrast, we could move the complexity and computation from the tags to fixed infrastructure such as base stations. As mentioned in Section 1.1.1, network-positioning generally allows for a simpler mobile unit with lower energy requirements. Instead of the tag estimating its own position based on signals it receives from satellites, the process can be reversed. The tag can periodically transmit a positioning signal. A network of nearby receivers that listen continuously for tag transmissions can use this signal to compute the position of the tag. The tag can be much simpler than a GPS-enabled tag; the simplest design does not require much more than an RF transmitter. Consider the following rough calculation showcasing the estimated energy being consumed by a tag for a single position update when network positioning is used.

Example 1.2. Suppose the same RF transmitter is used as in Example 1.1. Suppose a short 1 ms positioning signal is transmitted each time a position update is required. The estimated amount of energy consumed for a single position update is then 36 µJ, which is about four orders of magnitude (10 000 times) less than the energy consumed by a GPS-enabled tag (Example 1.1). △

The position of the tag can be estimated by measuring the characteristics of the radio waves received by receivers at known locations. Several characteristics can be measured and used for positioning: the power of the received signal, the angle of arrival, the time at which the signal arrives, or a combination of these characteristics. Each has its strengths and weaknesses in different environments and for different applications. In this dissertation, we focus on using Time of Arrival (TOA) measurements only. TOA measurements do not require specialised directional antennas like angle-of-arrival measurements. Furthermore, transforming the measurements of the physical properties of a radio wave to an estimate of geometric distance is not nearly as dependent on the topography and environment as with signal strength measurements.

If the time of transmission and the time of arrival at a receiver are known, the distance between the transmitter and receiver can be calculated by multiplying the propagation time of the signal with the propagation speed. However, measuring the propagation time, i.e. the difference between the time of transmission and the time of arrival, requires precise synchronisation between the transmitter and the receivers. This requirement can be eliminated by estimating the position using the difference in the time of arrival. This technique is called *hyperbolic positioning* [11]. Multilateral hyperbolic positioning is a positioning technique whereby the difference in the time at which a signal from an emitter arrives at three or more receivers, called the Time Difference of Arrival (TDOA), is used to locate the position of the emitter.

Hyperbolic positioning has been studied and used for almost a century. The first hyperbolic systems, the British Gee and the American LORAN-A system, had their origins in the World War II era [5]. Inexpensive and powerful signal processing hardware brought on by technological advances make it feasible to deploy hyperbolic positioning in sophisticated applications such as wildlife tracking.

1.1.4 Commercial-Off-The-Shelf hardware

As mentioned in Section 1.1.2, cost is one of the most important constraints of wildlife tracking systems. However, there are no commercial products on the market that can be used to track wildlife using hyperbolic positioning. Designing and developing bespoke hardware is expensive and time-consuming. Furthermore, the low production volumes of a custom-built product lead to high per-unit costs.

In this dissertation, we investigate the use of commercial off-the-shelf (COTS) hardware for hyperbolic positioning. Instead of developing tailor-made hardware for the receivers, some of the complexity of the design is moved into software to allow the use of inexpensive and readily available general-purpose hardware. This approach can lead to a simple and versatile solution that is easy and cheap to reproduce.

1.1.5 Related work

Wildlife tracking systems based on multilateral hyperbolic positioning is not a new concept. A group at the Cornell University Laboratory of Ornithology developed a prototype system for automatic wildlife tracking based on multilateral hyperbolic positioning and employed it for tracking birds [12, 13]. The transmitters of the prototype system are affordable (about €90 per tag), lightweight (3 g to 5 g), small, and last for a few months when transmitting a positioning signal every second with a 235 mAh battery [14]. The position estimate is reported to be accurate to 9 m in theory, but tens of metres in practice.

Their system has a few shortcomings. The system is not available commercially, and a lack of publicly available design documents prevents it from being reproduced without developing it from scratch. The circuitry for radio reception and signal processing has been custom-designed, which makes it costly to develop and produce on a small scale. The receivers are rather unwieldy; they are large and weigh about 14 kg per unit without the weight of the two lead acid batteries used to power them. The relatively high power consumption of the receivers (16 W) is another shortcoming which can be improved upon.

Another wildlife tracking system based on hyperbolic positioning, called *ATLAS*, was designed and implemented by the Minerva Center of Movement Ecology at The Hebrew University of Jerusalem in conjunction with staff and students from Tel–Aviv University [15, 16]. The tags weigh about 1 g to 10 g, depending on the battery being used. They reported localisation errors with a standard deviation of about 5 m and a mean of between 5 m and 15 m [16].

The ATLAS receivers are almost entirely built from well-supported COTS hardware. However, the hardware is rather expensive. Each receiver station has an *Ettus Research USRP N200* radio with a *WBX* daughterboard and a GPS-disciplined reference oscillator [15], which amounts to about $2800 per unit. In addition to the radio, each receiver station also consists of a personal computer (PC) for signal processing, which makes it troublesome to deploy without mains electricity and shelters for the equipment. Even though the receivers consist mostly of COTS hardware, reproducing their system would require significant investment in redevelopment unless their software and detail design are released.

As was discussed above, related work exists for tracking wildlife using hyperbolic positioning. However, we follow a different approach with unique criteria. Optimising the performance of the system and achieving maximum accuracy is not our goal at this stage. We want to focus on a simple and easily reproducible receiver design that uses inexpensive COTS hardware, even if a little performance has to be sacrificed. We use basic principles as described in literature and techniques similar to those used in related work, but we expand on the techniques and integrate it in a novel manner for usage on simple, inexpensive hardware.

1.2 Research problem

The following research questions were formulated based on the context given in the previous section:

> Is it feasible to implement a hyperbolic radio positioning network for wildlife tracking using inexpensive general-purpose COTS hardware? How can it be implemented and what techniques can be used to mitigate the shortcomings of the inexpensive hardware? What is the accuracy that can be expected despite the constraints imposed by a simple design with inexpensive hardware?

We are primarily concerned with wildlife tracking as the application. However, the end product, an outdoor, low cost, low power and easily reproducible network-based positioning system capable of covering an area the size of a large game reserve or national park, applies to a more general problem than just wildlife tracking.

1.3 Research goals

The goals that were set to answer the research problem, are:

- to study, analyse and summarise the principles of hyperbolic positioning and the sources of error that have to be accounted for;
- to devise a design for a TDOA positioning system that compensates for the limitations of low-cost receiver hardware;
- to implement a prototype system as a proof of concept and to verify the design as a whole;

- to perform laboratory experiments and pilot field tests with the prototype system;
- to analyse the results of the experiments in order to assess the performance of the system, verify the correctness of the design, evaluate its feasibility, and identify challenges that are involved;
- to evaluate and validate aspects of the design that have a significant impact on the accuracy of the system based on simulations and empirical test data;
- and finally, to comment on the feasibility of the system and to make recommendations for improving the performance of the prototype system.

Goals that are explicitly not within the scope of this dissertation, also called *non-goals*, are:

- to perform tests with tags attached to animals — this can be done in a separate study after the prototype system has been developed, and
- to purposefully try to achieve or improve upon the highest accuracy reported by related work.

The performance goal is determined by the application of tracking large animals and evaluated relative to the accuracy of similar tracking systems. The prototype system should be accurate enough for tracking large animals, i.e. accurate to tens of metres, while further improvements and different use cases can be addressed by future work.

1.4 Document structure

This dissertation is organised as follows: This chapter, *Chapter 1*, brought the research into context, stated the problem that was investigated and summarised the goals that were set out for the research. Chapters 2 and 3 are hybrids of literature studies and theoretical analyses. *Chapter 2* concentrates on TDOA in general without considering a specific implementation. The technique is expressed mathematically and the aspects that influence the performance of TDOA positioning are described. *Chapter 3* builds on Chapter 2, but focuses on Direct Sequence Spread Spectrum (DSSS) and how it can be used to meet the two essential requirements for a TDOA positioning signal simultaneously, namely high bandwidth and high energy. *Chapter 4* details the design of the prototype system. *Chapter 5* explains how the system was implemented. *Chapter 6* reports on the experiments and tests that were carried out and the results obtained from them. It also serves to validate and verify the design and the implementation. Finally, *Chapter 7* concludes the dissertation with a discussion of the results and recommendations for future work.

Chapter 2

Principles of TDOA radio positioning

> *[The universe] cannot be read until we have learnt the language and become familiar with the characters in which it is written. It is written in mathematical language, and the letters are triangles, circles and other geometrical figures, without which means it is humanly impossible to comprehend a single word.*
>
> — Galileo Galilei

When designing a new positioning system, it is important to understand the fundamental principles that govern it. It is also necessary to comprehend, at least on an elementary level, how different parameters and various sources of errors would influence the performance of the system. Finding literature explaining the fundamental principles that affect the design of a positioning system and practical considerations that need to be taken into account is challenging. Myriads of books and research articles on communication systems are available, but most of them are only concerned with data communication and do not provide information about the use of radio signals for positioning. Existing literature on radio positioning tends to be too theoretical, abstract, and general for practical application, or too focused on a specific application. The literature is usually aimed at positioning in which the signals and protocols have been defined beforehand by an existing positioning system such as GPS, or an existing technology such as GSM, WiFi, or IEEE 802.15.4a.

The aim of this chapter is to analyse and provide a fundamental understanding of the principles of TDOA radio positioning and their interrelationships. Section 2.1 starts with general principles that apply to all radio positioning techniques. This includes placing TOA into context with other radio measurement techniques and providing a brief summary of parameter estimation. Section 2.2 describes TDOA position estimation by defining it, deriving an analytic solution for solving linearised equations, and discussing numeric solutions for solving the nonlinear equations directly. Section 2.3 defines and describes the relationship between TDOA measurement error and the resulting positioning error, and shows how the transmitter–receiver geometry dilutes the precision of the TDOA estimate. Section 2.4 describes arrival time estimation, including an analysis of the elements of a radio signal that constitute a good positioning signal, and an explanation of why spread spectrum techniques are beneficial for TDOA positioning. Finally, the chapter is concluded with a summary in Section 2.5.

This chapter examines TDOA positioning in general without considering how the TDOA measurements are being taken, while the next chapter looks at how the arrival time can be measured using DSSS signals.

2.1 Positioning principles

2.1.1 Positioning

Location is always expressed relative to something else. For instance, the location of a point on a two-dimensional Cartesian coordinate system is generally expressed relative to the origin by means of two quantities, x and y, called the Cartesian coordinates. The two coordinates are the signed distances from the origin to the perpendicular projections of the point onto the two axes of the coordinate system. Similarly, a position on earth can be described using three coordinates, namely latitude, which is expressed as an angle relative to the Equator; longitude, which is usually expressed as an angle relative to Greenwich meridian; and elevation, which is generally expressed as the height above sea level.

Location is not only expressed in coordinates relative to something else, but it is also always determined or estimated with measurements relative to one or more other objects with known positions. For example, the position of a point on a two-dimensional Cartesian grid printed on paper can be determined by measuring the distance to the origin with a ruler and the angle relative to one of the axes with a protractor. Similarly, the latitude of a position on the earth, and to a limited extend the longitude as well, can be estimated by measuring the angle between celestial bodies and the horizon with a sextant [17].

Measurements for positioning involve the observation of physical quantities [18]. For a physical property to be used for positioning, the property has to be dependent on the position of the object relative to the reference point. This can be expressed mathematically as a functional relationship:

$$f\colon \boldsymbol{x} \to q \tag{2.1}$$

where q is the value of the property and the vector $\boldsymbol{x}$ the coordinates of the object being localised relative to the reference being measured from.

After measurement, q is known, but the information we are interested in, the position $\boldsymbol{x}$, is still unknown. The function f will not be invertible if $\boldsymbol{x}$ is of multiple dimensions, for example, if we want the position in two- or three-dimensional space. Many positions will lead to the same measurement value, and it will not be possible to deduce the position from a single measurement. Instead, multiple independent observations are necessary to solve the position unambiguously. For the observations to be independent, the properties being observed should have different functional relationships (f) or different reference points.

One method to solve the position from a set of independent observations is to create a model for each type of observation in the form of a mathematical function f relative to a common reference point that imitates the actual relationship between position and measurement values in the physical world. A system of equations can then be formed from the observations, one for each measurement, and the position can be solved.

2.1.2 Electromagnetic propagation measurements

Positioning is not limited to observations with the eye. The location of an object can also be estimated through the use of electromagnetic waves. This is accomplished by exploiting the physical properties of electromagnetic wave propagation to provide estimated distance and angle measurements relative to one or more objects with known positions.

There are three basic properties of electromagnetic wave propagation that can be measured and used for position estimation, namely

- the propagation direction, measured as the *Angle of Arrival (AOA)* at the receiver;
- the propagation attenuation, measured as the *Received Signal Strength (RSS)*;
- the propagation delay, measured as the *Time of Arrival (TOA)* [19].

The values of these properties as measured by the receiver depend on the position of the receiver relative to the transmitter. This dependency can be used for position estimation. The position can be estimated given the assumed propagation model and the functional relation between measurements and position (Equation (2.1)). The different types of measurements are explained in more detail below.

Angle of Arrival (AOA) In the absence of any discontinuities in the propagation medium, the waveform arriving at the receiver will travel along the fastest path between the transmitter and the receiver. A measurement of the direction of the incident signal can be related to the geometric angle between the transmitter and the receiver. The AOA can be measured by noting the angle at which the received signal strength at the receiver is maximum or minimum while varying the radiation pattern of either the transmitting or receiving antenna, or by using an antenna array at the receiver and noting the difference in the time of arrival (or phase) at each of the array elements [19].

Received Signal Strength (RSS) The power of an electromagnetic signal decreases as the wave propagates further and further from the transmitter due to the inverse-square law of electromagnetic radiation. In free space, the square of the distance between a transmitter and receiver is inversely proportional to the power density of the electromagnetic wave at the receiver. Thus, the distance between the transmitter and receiver can be calculated if the transmission power and the RSS are known.

In a practical environment, the signal will travel along different paths to the receiver due to discontinuities in the propagation medium, such as reflections of the wave against obstacles in the environment. The receiver will not only measure the power of the wave that travelled along the shortest path but also the vectorial combination of time-delayed signals that travelled along different paths as well as interfering signals from other transmission sources. Subsequently, the power measured by a receiver will not decrease monotonically as the distance between the transmitter and receiver increases, and the relationship will change according to the environment. This has the effect that the position to RSS mapping is non-trivial and cannot be inverted. However, various models exist for obtaining a position estimate from RSS measurements using a database of reference measurements that were obtained beforehand [18].

Time of Arrival (TOA) Electromagnetic waves travel at a constant speed in a homogeneous medium, namely the speed of light. There is thus a linear relation between propagation time and propagation distance. The distance between a transmitter and receiver can be calculated by multiplying the time it takes a signal to travel along the shortest path from the transmitter to the receiver with the known propagation speed in the medium.

One measurement is not sufficient for locating an object in two- or three-dimensional space. Different types of measurements (AOA, RSS or TOA) or measurements between different transmitter–receiver pairs have to be combined for an unambiguous position estimate. Various configurations exist to provide sufficient information for solving the unknown variables, which are usually the coordinates of the receiver or the transmitter in Two-Dimensional (2D) or Three-Dimensional (3D) space. For example, the two-dimensional position of a transmitter can be determined from AOA measurements at two different locations, or from both an AOA and RSS measurement at a single location.

As stated in Section 1.1.3, we focus on the use of TOA measurements in this dissertation. The receiver hardware required for TOA measurements is relatively simple in comparison with the specialised directional antennas required for AOA measurements. Furthermore, the functional relationship between measurements and position for TOA-based positioning is not nearly as dependent on the topography and environment as with RSS-based positioning.

2.1.3 Radio positioning system classification

A multitude of radio positioning technologies and techniques are in use today. It is necessary to understand some of the classification terms being used to demarcate the type of positioning we are concerned with and to differentiate it from other forms of positioning.

Before describing the classifications, it is necessary to first define the terminology that is used in this dissertation for referring to the different kinds of entities that are involved in the positioning process. The definitions are given below.

Terminal Either side of the radio link being used for positioning: a transmitter, receiver or transceiver.

Mobile unit A terminal with unknown position that is the target of the positioning process, i.e. the device being tracked.

Base station A terminal with known, usually fixed, position, that acts as a reference point for positioning.

Beacon A terminal with known position with periodic transmissions that facilitate timing synchronisation.

One classification criterion is the measurement technique being employed, e.g. *TOA positioning.* TOA positioning can further be classified as *one-way* or *two-way.* In one-way positioning, also called *unidirectional* positioning, the positioning signal is sent in one direction from a transmitter to a receiver. All the mobile units take on the same role. The mobile units are all either transmitters or receivers, and the base stations take on the opposite role. In two-way positioning, the mobile units and base stations are transceivers, and distance is calculated by measuring the round-trip time of a positioning signal.

Another classification criterion, which is also described in the first chapter, is the location of the positioning information after localisation, which is either at the mobile unit of which the position is being determined or the base stations that act as reference points. The former is generally called *self-positioning*, and the latter *network-positioning* [2]. With self-positioning, the mobile unit estimates its position based on signals from multiple transmitters at known locations. Self-positioning is also called *mobile-based* or *unilateral* positioning [2, 5, 18, 19]. With network-positioning, the mobile unit transmits a signal, and multiple receivers at known locations estimate the position of the mobile unit cooperatively. Network-positioning is also called *multilateral* or *remote* positioning [2, 5, 18, 19]. The difference between self-positioning and network-positioning is illustrated in Figure 2.1.

Further dimensions of classification also exist, such as outdoors or indoors, terrestrial (ground-based) or satellite, global or regional. In this dissertation, our objective is to develop an outdoor ground-based multilateral one-way TDOA positioning system. The descriptions, criteria and

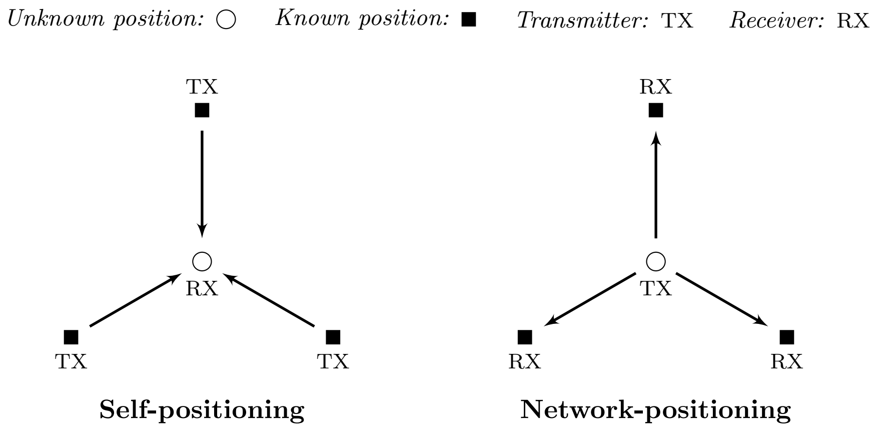


Figure 2.1: Illustration of the difference between one-way self-positioning and one-way network-positioning systems (based on [5, p. 255]).

performance measures in this dissertation assume this objective. For example, even though TOA can also be used for self-positioning, it is described from a network-positioning point of view in subsequent sections of this dissertation.

2.1.4 Parameter estimation

With radio positioning, the position of a mobile unit cannot be measured directly. We only have the opportunity to observe physical properties of the electromagnetic signal that depend on position. Moreover, the received signals that are being observed in order to derive the property values are corrupted by non-deterministic distortions called noise. The actual value of the property is unknown, but has to be estimated from noisy measurements. Estimating parameter values based on noisy measurements forms part of a more general problem in statistics, namely the problem of *parameter estimation.*

The estimation problem can be stated as follows [18]: Given a collection of k measured values, $\boldsymbol{m} = \langle m_1, \dots, m_k \rangle$, with values that depend on a collection of l parameters, $\alpha = \langle \alpha_1, \dots, \alpha_l \rangle$, find a function $\hat{\alpha}(\boldsymbol{m})$ that estimates the parameters α from the measurements, thus $\hat{\alpha}(\boldsymbol{m}) = \langle \hat{\alpha}_1(\boldsymbol{m}), \dots, \hat{\alpha}_l(\boldsymbol{m}) \rangle$. The function is called an *estimator* and its value an *estimate* [20].

One approach for establishing a model for parameter estimation is to use non-Bayesian statistics [20]. With this approach, it is assumed that a true but unknown value α exists for the parameters. The distortions that corrupt the measurements are considered as stochastic processes. Consequently, the measurements $\boldsymbol{m}$ are modelled as a vector of random variables, called a *random vector.* The same value of the parameters α can result in different measurement values $\boldsymbol{m}$ due to the random noise distorting the measurements. The likelihood that a collection of measured values is due to a given collection of parameters is given by the likelihood function

$$L_{\boldsymbol{m}}(\alpha) := p(\boldsymbol{m}|\alpha) \tag{2.2}$$

where $p(\boldsymbol{m}|\alpha)$ is the Probability Density Function (PDF) of the measurements $\boldsymbol{m}$ conditioned on the parameters α. A common method for estimating α from $\boldsymbol{m}$ is to find the estimate $\hat{\alpha}$ that maximises the likelihood function, thus

$$\hat{\alpha}(\boldsymbol{m}) = \arg\max_{\alpha} L_{\boldsymbol{m}}(\alpha) \tag{2.3}$$

This is known as the Maximum Likelihood Estimator (MLE). Note that, even though α is a constant vector, $\hat{\alpha}$ is a random vector since it is a function of the random vector $\boldsymbol{m}$.

Another common method for parameter estimation is the Least Squares (LS) method. Let the k-dimensional vector function $\boldsymbol{h}(\alpha)$ represent a model that relates parameter values to ideal errorless measurement values, thus

$$\boldsymbol{m} = \boldsymbol{h}(\alpha) + \epsilon \tag{2.4}$$

where the k-vector ϵ represents the unknown errors of the measurements. The Least Squares Estimator (LSE) of α finds the estimate $\hat{\alpha}$ that minimises the sum of the square errors between the predicted measurement values and the observed measurement values, thus

$$\hat{\alpha}(\boldsymbol{m}) = \arg\min_{\alpha} \sum_{i=1}^{k} (m_i - h_i(\alpha))^2 \tag{2.5}$$

or in vector–matrix notation

$$\hat{\alpha}(\boldsymbol{m}) = \arg\min_{\alpha} \left(\boldsymbol{m} - \boldsymbol{h}(\alpha)\right)^T \left(\boldsymbol{m} - \boldsymbol{h}(\alpha)\right). \tag{2.6}$$

This is known as the Linear Least Squares (LLS) problem if $\boldsymbol{h}$ is a linear equation and the Nonlinear Least Squares (NLLS) problem if $\boldsymbol{h}$ is a nonlinear equation.

The first use case of parameter estimation that we encounter in this dissertation, the problem of estimating position from TDOA measurements, is described next.

2.2 TDOA position estimation

2.2.1 Introduction to TOA positioning

If a constant propagation speed is assumed, and the time of transmission as well as the time the Line-Of-Sight (LOS) signal arrives at a receiver are known, the distance between the transmitter and receiver can be calculated by multiplying the propagation delay of the signal with the propagation speed:

$$r_i = \int_{t_0}^{t_i} c\, dt = c\,(t_i - t_0) \tag{2.7}$$

where r_i denotes the distance between the transmitter and receiver i, c the propagation speed, t_0 the time of transmission, and t_i the time of arrival at receiver i. The propagation speed of an electromagnetic waveform is equal to the speed of light in the propagation medium, which is constant in a homogeneous medium.

For network-positioning in 2D space, if $\boldsymbol{x} = \langle x, y \rangle$ is the (unknown) position of the transmitter, and $\boldsymbol{x}_i = \langle x_i, y_i \rangle$ the (known) position of receiver i, then Equation (2.7) can be written as:

$$\sqrt{(x - x_i)^2 + (y - y_i)^2} = c\,(t_i - t_0)\,. \tag{2.8}$$

This is the equation of a circle with radius $r = c\,(t_i - t_0)$ centred around $\boldsymbol{x}_i$, the position of receiver i. The transmitter may be located at any point along the perimeter of the circle. We may take measurements at multiple receivers to calculate the position unambiguously — a process known as trilateration. In 2D, this relates to the point or area of intersection of multiple circles.

Calculating the propagation delay $(t_i - t_0)$ requires the transmitter's clock measuring the time of transmission (t_0) to be synchronised with the receiver's clock measuring the time of arrival (t_i). The propagation speed of electromagnetic waves in air is about $3 \times 10^8\,\mathrm{m\,s^{-1}}$, which means that even a small clock error can have a devastating impact on the accuracy of the position estimate [11]. For example, a clock synchronisation error of 1 µs would result in a distance measurement error of 300 m. Synchronisation of the clocks is hard to achieve, and low-cost hardware components, which only provide reasonable short-term stability, cannot fulfil this requirement [18]. Not only do the transmitter and all receivers have to be synchronised, but the transmitter also needs to pass information to the receivers indicating when the transmission has started. Alternatively, the time of transmission can be used as an additional unknown variable and solved together with the position. For 2D positioning, solving the system of equations then relates to finding the intersection of cones [18]. Another approach is to eliminate the time of transmission from the equations using TDOA.

2.2.2 Introduction to TDOA positioning

The requirement for knowledge of transmission time, and thus also the requirement for synchronisation between the transmitters and the receivers, can be eliminated by calculating the position from the *difference* in propagation distance between pairs of receivers instead of the *absolute* propagation distance. From Equation (2.7), if r_i and r_j denote the distance between the transmitter and receiver i and j respectively, and t_i and t_j the TOA at the respective receivers, then:

$$r_i - r_j = c\,(t_i - t_0) - c\,(t_j - t_0) = c\,(t_i - t_j)\,. \tag{2.9}$$

Thus, the difference eliminates the transmission time t_0 from the equation, with the result that the difference in propagation distance can be calculated from the difference in the time of arrival. This technique is generally referred to as Time Difference of Arrival (TDOA).

In the case of network-positioning, if $\boldsymbol{x}$ represents the (unknown) position of the mobile unit,

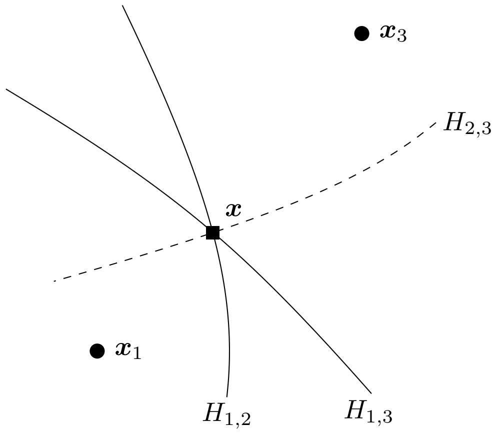


Figure 2.2: Illustration of TDOA in two dimensions. Measuring the difference in the time at which a signal from a mobile unit arrives at a pair of receivers yields a hyperbola branch along which the mobile unit is located. The intersection of the hyperbola branches from multiple TDOA measurements provides the position of the mobile unit.

and $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ the (known) positions of two base stations, then:

$$\|\boldsymbol{x} - \boldsymbol{x}_i\| - \|\boldsymbol{x} - \boldsymbol{x}_j\| = c\,(t_i - t_j) \tag{2.10}$$

where $\|\boldsymbol{x}\|$ denotes the Euclidean length of vector $\boldsymbol{x}$. Geometrically, in 2D space, this equation defines a locus of points of equal difference in distance to the two receivers being considered. This is the definition of one branch of a hyperbola with the two receivers as focal points (*foci*) [18]. Similarly, a hyperboloid is formed in 3D space.

For a single TDOA measurement, the position of the mobile unit is ambiguous; it can be located at any point along the hyperbolic curve or hyperboloidic surface. TDOA measurements between multiple independent pairs of receivers define different hyperbolas (in 2D) or hyperboloids (in 3D) of which the intersection yields the position of the mobile unit.

Consider the following example of TDOA positioning in 2D space, as illustrated in Figure 2.2:

Example 2.1. Assume there are three base stations positioned at $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ respectively, which are arranged as shown in Figure 2.2. Assume we have to find the position $\boldsymbol{x}$ of a mobile unit that transmits a positioning signal. Let $d_{i,j}$ be the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, thus

$$d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\,.$$

If the first base station receives the signal before the second base station with a TDOA such that $c(t_2 - t_1) = -0.25 \cdot d_{2,1}$, then the mobile unit is located somewhere along the hyperbola branch $H_{1,2}$ with equation

$$\|\boldsymbol{x} - \boldsymbol{x}_2\| - \|\boldsymbol{x} - \boldsymbol{x}_1\| = -0.25 \cdot d_{2,1}.$$

Note that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are the foci of $H_{1,2}$.

If the difference in TOA between the first and third receiver is such that $c(t_3 - t_1) = -0.1 \cdot d_{3,1}$, then it is known that the mobile unit is also located along the hyperbola branch $H_{1,3}$ with equation

$$\|\boldsymbol{x} - \boldsymbol{x}_3\| - \|\boldsymbol{x} - \boldsymbol{x}_1\| = -0.1 \cdot d_{3,1}.$$

The intersection of these two hyperbola branches yields the position of the mobile unit.

A third hyperbola branch, $H_{2,3}$, can be formed from the TDOA of the second and third base stations. However, it does not provide any further information since it is a linear combination of hyperbola branches $H_{1,2}$ and $H_{2,3}$:

$$\begin{aligned} r_3 - r_2 &= c(t_3 - t_2) \\ (r_3 - r_1) - (r_2 - r_1) &= c(t_3 - t_1) - c(t_2 - t_1). \end{aligned} \qquad \triangle$$

Example 2.1 illustrates how three base stations are used to locate the position of a mobile unit in two dimensions. However, the position in two dimensions is not always unambiguous when only three base stations are employed. Consider a different configuration: the example shown in Figure 2.3. The hyperbolas intersect at two points labelled $\boldsymbol{x}$ and $\boldsymbol{x}'$ respectively. A measurement from a fourth base station or *a priori* information is necessary to resolve the ambiguity.

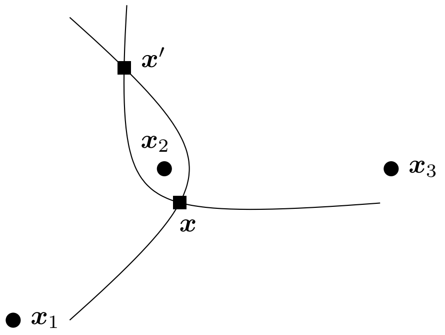


Figure 2.3: Ambiguous position in two dimensions with three receivers. A mobile unit at position $\boldsymbol{x}$ will yield the same TDOA measurements as a mobile unit at position $\boldsymbol{x}'$.

2.2.3 Analytic solutions

In Example 2.1, the position of a mobile unit was determined from TDOA measurements by intersecting curves graphically. It would be useful to have a closed-form equation that could be used to solve the position analytically. Thus, given $\boldsymbol{x}_i$ and t_i for $1 \leq i \leq N$, where N is the number of base stations, we want to solve the following system of equations:

$$\|\boldsymbol{x}-\boldsymbol{x}_i\| - \|\boldsymbol{x}-\boldsymbol{x}_j\| = c\,(t_i - t_j) \qquad \text{for } i = 1, \ldots, N; i \neq j. \tag{2.11}$$

However, this is a set of nonlinear equations, which is difficult to solve. A simple linearisation of the system of equations in terms of $\boldsymbol{x}$ follows. It is based on a derivation in [18], but has been rederived, expanded, and simplified.

Without loss of generality, we assume TDOA measurements are taken relative to the first receiver, thus $j = 1$. To simplify the equations and notation, the coordinate system is translated to set the origin at $\boldsymbol{x}_1$, and TOA measurements are taken relative to t_1, thus

$$\begin{aligned}\tilde{\boldsymbol{x}}_i &:= \boldsymbol{x}_i - \boldsymbol{x}_1\\ \tilde{t}_i &:= t_i - t_1.\end{aligned}$$

Equation (2.11) can now be written as:

$$\|\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{x}}_i\| - \|\tilde{\boldsymbol{x}}\| = c\tilde{t}_i = \tilde{r}_i \qquad \text{for } i = 2, \ldots, N.$$

Rearranging and squaring yields:

$$\|\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{x}}_i\|^2 = (\tilde{r}_i + \|\tilde{\boldsymbol{x}}\|)^2 \tag{2.12}$$

$$\|\tilde{\boldsymbol{x}}\|^2 + 2\tilde{r}_i\,\|\tilde{\boldsymbol{x}}\| + \tilde{r}_i^2 - \|\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{x}}_i\|^2 = 0. \tag{2.13}$$

Divide by $\tilde{r}_i$, assuming that $\tilde{r}_i \neq 0$:

$$\frac{\|\tilde{\boldsymbol{x}}\|^2 - \|\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{x}}_i\|^2}{\tilde{r}_i} + 2\,\|\tilde{\boldsymbol{x}}\| + \tilde{r}_i = 0. \tag{2.14}$$

Eliminate the term $2\,\|\tilde{\boldsymbol{x}}\|$ by subtracting Equation (2.14) for $i = 2$ from Equation (2.14):

$$\frac{\|\tilde{\boldsymbol{x}}\|^2 - \|\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{x}}_i\|^2}{\tilde{r}_i} - \frac{\|\tilde{\boldsymbol{x}}\|^2 - \|\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{x}}_2\|^2}{\tilde{r}_2} + \tilde{r}_i - \tilde{r}_2 = 0 \qquad \text{for } i = 3, \ldots, N. \tag{2.15}$$

In 2D space, if $\boldsymbol{x} = \begin{pmatrix} x & y \end{pmatrix}^T$:

$$\begin{aligned}\|\boldsymbol{x}\|^2 - \|\boldsymbol{x}-\boldsymbol{x}_i\|^2 &= x^2 + y^2 - (x - x_i)^2 - (y - y_i)^2\\ &= 2x_ix + 2y_iy - x_i^2 - y_i^2.\end{aligned}$$

If $\tilde{d}_i$ is the distance between receiver i and receiver 1, then

$$\|\tilde{\boldsymbol{x}}\|^2 - \|\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{x}}_i\|^2 = 2\tilde{x}_i\tilde{x} + 2\tilde{y}_i\tilde{y} - \tilde{d}_i^2. \tag{2.16}$$

Using the result of Equation (2.16) in Equation (2.15) and rearranging yields:

$$2\left(\frac{\tilde{x}_i}{\tilde{r}_i}-\frac{\tilde{x}_2}{\tilde{r}_2}\right)\tilde{x}+2\left(\frac{\tilde{y}_i}{\tilde{r}_i}-\frac{\tilde{y}_2}{\tilde{r}_2}\right)\tilde{y}=\frac{\tilde{d}_i^2}{\tilde{r}_i}-\frac{\tilde{d}_2^2}{\tilde{r}_2}-\tilde{r}_i+\tilde{r}_2 \qquad \text{for } i=3,\dots,N. \tag{2.17}$$

This can be written in vector–matrix notation as:

$$\underbrace{2\begin{pmatrix}\left(\frac{\tilde{x}_3}{\tilde{r}_3}-\frac{\tilde{x}_2}{\tilde{r}_2}\right) & \left(\frac{\tilde{y}_3}{\tilde{r}_3}-\frac{\tilde{y}_2}{\tilde{r}_2}\right)\\ \left(\frac{\tilde{x}_4}{\tilde{r}_4}-\frac{\tilde{x}_2}{\tilde{r}_2}\right) & \left(\frac{\tilde{y}_4}{\tilde{r}_4}-\frac{\tilde{y}_2}{\tilde{r}_2}\right)\\ \vdots & \vdots\\ \left(\frac{\tilde{x}_N}{\tilde{r}_N}-\frac{\tilde{x}_2}{\tilde{r}_2}\right) & \left(\frac{\tilde{y}_N}{\tilde{r}_N}-\frac{\tilde{y}_2}{\tilde{r}_2}\right)\end{pmatrix}}_{\mathbf{A}:=}\underbrace{\begin{pmatrix}\tilde{x}\\ \tilde{y}\end{pmatrix}}_{\tilde{\boldsymbol{x}}:=}=\underbrace{\begin{pmatrix}\frac{\tilde{d}_3^2}{\tilde{r}_3}-\frac{\tilde{d}_2^2}{\tilde{r}_2}-\tilde{r}_3+\tilde{r}_2\\ \frac{\tilde{d}_4^2}{\tilde{r}_4}-\frac{\tilde{d}_2^2}{\tilde{r}_2}-\tilde{r}_4+\tilde{r}_2\\ \vdots\\ \frac{\tilde{d}_N^2}{\tilde{r}_N}-\frac{\tilde{d}_2^2}{\tilde{r}_2}-\tilde{r}_N+\tilde{r}_2\end{pmatrix}}_{\mathbf{B}:=}$$

$$\mathbf{A}\tilde{\boldsymbol{x}}=\mathbf{B}$$

A unique solution in 2D space can be calculated if $N = 4$ and $\mathbf{A}$ is invertible:

$$\tilde{\boldsymbol{x}}=\mathbf{A}^{-1}\mathbf{B} \tag{2.18}$$

$$\boldsymbol{x}=\boldsymbol{x}_1+\mathbf{A}^{-1}\mathbf{B} \tag{2.19}$$

The system of equations is overdetermined when $N > 4$. In that case the matrix inverse in Equation (2.19) can be replaced with the pseudoinverse to yield the least squares solution that uses redundant measurements to improve the position estimate.

Equation (2.19) provides a system of linear equations that is computationally efficient and useful for solving or checking a position estimation quickly. However, it has a few shortcomings. Firstly, four base stations are required, even in cases where three base stations would have sufficed. Secondly, we assume ideal TDOA measurements in the derivation of the equations and ignore the effect of noise. Errors in arrival time measurements create uncertainty as to where the transmitter is located, and the linearisation process amplifies these errors [18].

A relatively simple closed-form solution was presented in this section, but various alternative closed-form solutions exist. For example, So and Chan [21] presents closed-form equations that can be employed for 2D positioning using only three base stations. More closed-form methods are described in [22–24].

Analytic solutions generally linearise the system of nonlinear equations given in Equation (2.11) to form a closed-form equation. Alternatively, the system of nonlinear equations can be solved directly. No analytical solution exists for solving the system of nonlinear equations and hence we have to revert to numeric methods to solve it iteratively [18].

2.2.4 Numeric solutions

Instead of solving the position analytically, it can be determined by finding the position estimate $\hat{\boldsymbol{x}}$ that minimises the disagreement between the TDOA values at that position as predicted by the propagation model and the actual TDOA measurements. If $\hat{t}_{i,j}$ is the actual TDOA measurement,

$$\hat{t}_{i,j} := \hat{t}_i - \hat{t}_j,$$

and $t_{i,j}(\boldsymbol{x})$ the predicted TDOA value at $\boldsymbol{x}$,

$$t_{i,j}(\boldsymbol{x}) = \frac{1}{c}\left(\|\boldsymbol{x} - \boldsymbol{x}_i\| - \|\boldsymbol{x} - \boldsymbol{x}_j\|\right),$$

then the objective is to minimise the sum of the squares of the differences between the actual and the predicted values. Thus, the objective is to minimise the cost function

$$L(\boldsymbol{x}) = \sum_{\substack{i=1 \\ i \neq j}}^{N} \left(\hat{t}_{i,j} - t_{i,j}(\boldsymbol{x})\right)^2$$

or the equivalent using TDOA distances,

$$L(\boldsymbol{x}) = \sum_{\substack{i=1 \\ i \neq j}}^{N} \left(\hat{r}_{i,j} - r_{i,j}(\boldsymbol{x})\right)^2$$

where

$$\begin{aligned} \hat{r}_{i,j} &:= \hat{r}_i - \hat{r}_j \\ &= c\left(\hat{t}_i - \hat{t}_j\right) = c \cdot \hat{t}_{i,j} \\ r_{i,j}(\boldsymbol{x}) &= \|\boldsymbol{x} - \boldsymbol{x}_i\| - \|\boldsymbol{x} - \boldsymbol{x}_j\|. \end{aligned}$$

The estimated position of the mobile unit is the position for which $L(\boldsymbol{x})$ yields the lowest value:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} L(\boldsymbol{x}). \tag{2.20}$$

Note that the method that has just been described is the method of LS. It is an NLLS problem as the model $t_{i,j}(\boldsymbol{x})$ is nonlinear. When errors are included in the model and error statistics are available, the covariance matrix of the error vector can be added to the cost function as a weight matrix, which transforms it into a Weighted Nonlinear Least Squares (WNLLS) problem [18].

Finding $\hat{\boldsymbol{x}}$ is not a simple task since the model $t_{i,j}(\boldsymbol{x})$ is nonlinear. The cost function may thus have local minima apart from the global minimum, i.e. the cost function is multi-modal. Equation (2.20) is an optimisation problem — a common problem for which a multitude of solutions exist. For example, one approach is global exploration, which includes techniques such as grid search and random search [25]. Another approach is to use an iterative local search scheme. Some commonly used iterative NLLS minimisation methods include the Gauss–Newton, the Steepest Descent, and the Levenberg–Marquardt algorithm [18].

Convergence to the global minimum may not be guaranteed, especially for local search schemes. However, global and local search algorithms can be combined, or the initial guess for the iterative algorithm can be derived from *a priori* information such as the previous position estimate, or from a suboptimal and potentially closed-form solution calculated from an approximation of the problem.

Different algorithms have different accuracies, computational requirements and restrictions. In general, a trade-off exists between position accuracy and computational requirements. For instance, algorithms based on the linearised set of equations are computationally less expensive than those based on the nonlinear equations, but their accuracies are generally lower. Several studies exist that compare different position estimation techniques, such as [24–26].

Once accurate TDOA estimates have been obtained, the problem of finding the position from the estimates is essentially the same for all TDOA systems. The position-finding algorithm being employed is not our primary concern at this stage, but whether accurate TDOA estimates can be obtained with cheap receiver hardware. It is, however, important to understand how position estimates are determined from TDOA estimates using analytic and numeric algorithms, as discussed above. It is also necessary to understand the relationship between TDOA accuracy and the eventual position accuracy, which is discussed next.

2.3 Dilution of precision (DOP)

2.3.1 Introduction to DOP

As discussed earlier, the position of the mobile unit is not measured directly, but a propagation model is used to estimate it from position-dependent measurements that are corrupted by noise. A common and important question in the field of radio positioning is what the resulting error in position would be for a given error in the measured value. This is quantified as a Positional Dilution of Precision (PDOP) value, which can conceptually be defined as:

$$PDOP = \frac{\Delta(Position)}{\Delta(Measurement)}.$$

The PDOP value essentially describes the sensitivity of the position estimate for errors in the measurement value. A more general measure is the Geometric Dilution of Precision (GDOP), also simply referred to as the Dilution of Precision (DOP). GDOP expresses how a change in the measured value would affect the unknown variables in the propagation model. In the case of GPS, which uses TOA positioning, the unknown variables are the coordinates of the mobile unit and the clock bias [11]. For TDOA positioning, the only unknown variables are

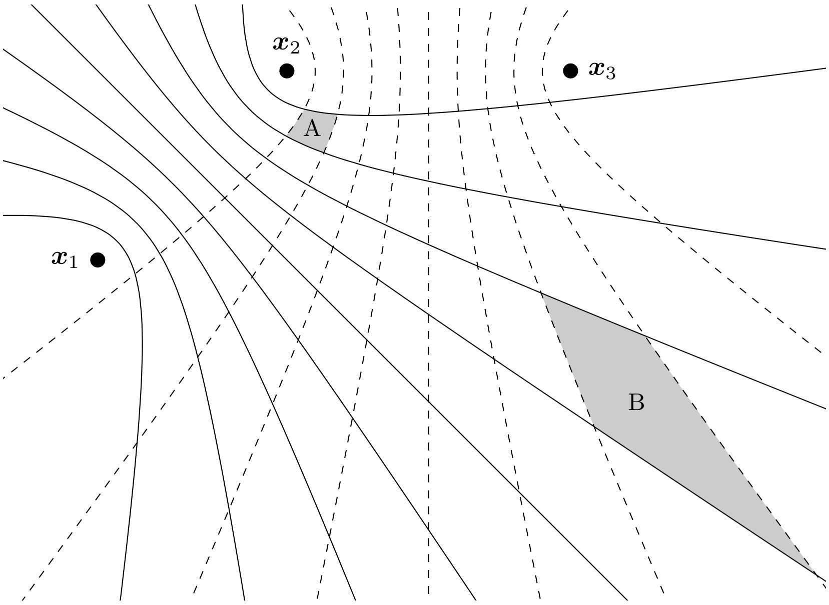


Figure 2.4: Geometric illustration of PDOP for a fixed receiver configuration. Even though the uncertainty in the TDOA measurement is equal, the area of uncertainty in position for a mobile unit located at B is significantly larger than for a mobile unit located at A. The PDOP at B is greater than at A.

the position coordinates. Hence, the GDOP is equal to the PDOP.

DOP is not a fixed value, but a function of the position of the mobile unit relative to the base stations. This is illustrated in Figure 2.4 for TDOA in 2D space. The hyperbolas divide the straight line between pairs of base stations into ten intervals of equal size. This spacing corresponds with an uncertainty of $0.1 \cdot d_{1,2}$ and $0.1 \cdot d_{2,3}$ in TDOA distance measurements between receivers 1 and 2 and receivers 2 and 3 respectively. The uncertainty in the TDOA measurement is the same at both positions A and B, but the area of uncertainty of a position estimate is clearly larger at B than at A. Thus, if a mobile unit is located in the centre of the shaded area at A and another unit in the centre of area B, and the TDOA distance estimates at both units are corrupted by a random error between $-0.05\,d_{i,j}$ and $0.05\,d_{i,j}$, then the estimated position of unit A will be anywhere within the shaded area at A, while the estimate of unit B will be within the larger shaded area at B. Consequently, the PDOP at B is greater than at A.

DOP is a function of the position of the mobile unit relative to the base stations, and thus also a function of the receiver configuration. At any given position, different receiver dispersions will yield different DOP values.

2.3.2 Mathematical definition

DOP has been defined conceptually and illustrated graphically in the discussion above. A more formal and concrete definition of DOP follows.

Let the vector $\boldsymbol{x}$ be the unknown variables in the positioning model, the vector ρ the measurements, and $d\rho$ the error in the measurements. For TDOA positioning, $\boldsymbol{x}$ represents the coordinates of the mobile unit and ρ the TDOA distance measurements. It can be shown that [27, p. 326], if the components of $d\rho$ are unbiased, independent and have identical distributions with a standard deviation of $\sigma_{d\rho}$, i.e.

$$\operatorname{cov}(d\rho) = \mathbf{I}\,\sigma_{d\rho}^2$$

then

$$\operatorname{cov}(d\boldsymbol{x}) = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\sigma_{d\rho}^2 \tag{2.21}$$

where $\mathbf{G}$ is the Jacobian matrix of ρ with respect to $\boldsymbol{x}$.

For 2D TDOA positioning, $\boldsymbol{x} = \begin{pmatrix} x & y \end{pmatrix}^T$. The expanded representation of the covariance of $d\boldsymbol{x}$ is then

$$\operatorname{cov}(d\boldsymbol{x}) = \begin{pmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{pmatrix}.$$

Let $\operatorname{cov}(d\boldsymbol{x}) = \mathbf{H}\,\sigma_{d\rho}^2$, i.e. $\mathbf{H} := \left(\mathbf{G}^T\mathbf{G}\right)^{-1}$. Then

$$\mathit{DOP} = \mathit{GDOP} = \mathit{PDOP} = \frac{\sqrt{\sigma_x^2 + \sigma_y^2}}{\sigma_{d\rho}} = \sqrt{\operatorname{trace}(\mathbf{H})}.$$

The vector ρ contains all TDOA distance measurements, thus

$$\begin{aligned} \rho = \boldsymbol{r}_\Delta &= \left(r_{i,j}\ \forall\, i; 1 \le i \le N, i \ne j\right) \\ &= c\boldsymbol{t}_\Delta = c\left(t_{i,j}\ \forall\, i; 1 \le i \le N, i \ne j\right) \end{aligned}$$

where $r_{i,j} = r_i - r_j$ is a TDOA distance measurement and $t_{i,j} = t_i - t_j$ a TDOA measurement. Note that the standard deviation of the TDOA measurement errors is related to the standard deviation of the TDOA distance measurement errors: $\sigma_{d\rho} = \sigma_{dr_\Delta} = c\,\sigma_{dt_\Delta}$.

GDOP can be separated into different DOP parts to characterise the accuracy of different components of $\boldsymbol{x}$. For 2D TDOA positioning, if x defines the east–west coordinate and y the north–south coordinate, then the East DOP (EDOP) and North DOP (NDOP) can be defined as

$$\begin{aligned} \mathit{EDOP} &= \frac{\sigma_x}{\sigma_{d\rho}} = \sqrt{\mathbf{H}_{11}} \\ \mathit{NDOP} &= \frac{\sigma_y}{\sigma_{d\rho}} = \sqrt{\mathbf{H}_{22}} \end{aligned}$$

where $\mathbf{H}_{ij}$ represents the entry in the i-th row and j-th column of matrix $\mathbf{H}$. It follows that

$$\mathbf{H} = \begin{pmatrix} EDOP^2 & \cdot \\ \cdot & NDOP^2 \end{pmatrix}$$

and

$$DOP = \sqrt{EDOP^2 + NDOP^2}.$$

2.3.3 Jacobian matrix for 2D TDOA positioning

The Jacobian matrix $\mathbf{G}$ has to be derived before the DOP values can be calculated. Assume 2D positioning and that TDOA measurements are taken relative to the first receiver, thus $j = 1$. Equation (2.11) can then be written as,

$$r_{i,1} = r_i - r_1 = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{(x - x_1)^2 + (y - y_1)^2} \qquad \text{for } i = 2, \ldots, N.$$

Note that

$$\begin{aligned} \frac{\partial}{\partial x} r_{i,1} &= \frac{2\,(x - x_i)}{2\sqrt{(x - x_i)^2 + (y - y_i)^2}} - \frac{2\,(x - x_1)}{2\sqrt{(x - x_1)^2 + (y - y_1)^2}} \\ &= \frac{x - x_i}{r_i} - \frac{x - x_1}{r_1} \end{aligned}$$

Similarly,

$$\frac{\partial}{\partial y} r_{i,1} = \frac{y - y_i}{r_i} - \frac{y - y_1}{r_1}$$

The Jacobian matrix of ρ with respect to $\boldsymbol{x}$ is

$$\mathbf{G} = \frac{\partial \rho}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{r}_\Delta}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{x - x_2}{r_2} - \frac{x - x_1}{r_1} & \frac{y - y_2}{r_2} - \frac{y - y_1}{r_1} \\ \vdots & \vdots \\ \frac{x - x_N}{r_N} - \frac{x - x_1}{r_1} & \frac{y - y_N}{r_N} - \frac{y - y_1}{r_1}. \end{pmatrix} \tag{2.22}$$

This matrix can be used to calculate the DOP for a given mobile unit position and base station configuration. Given the positions of the mobile unit and base stations, we can compute $\mathbf{G}$, then $\mathbf{H}$ and finally the DOP value.

2.3.4 PDOP heat maps

With all the necessary equations defined, it is now possible to calculate DOP values for TDOA positioning. DOP is a function of the mobile unit's position, as well as the positions of the base stations relative to each other, which will be called the receiver's *configuration.* Heat maps can help to gain insight into the characteristics of the DOP function, such as typical values that can be expected and positions that would yield lower errors than others.

Heat maps for four different receiver configurations are displayed in Figure 2.5. The heat maps were generated by dividing the 2D space into a grid, calculating PDOP values for a mobile unit located at each of the cells, and assigning colours to represent the PDOP values. The lighter the shading, the higher the PDOP value. The positions of the receivers are indicated by white squares. All TDOA measurements are taken relative to the receiver the farthest south–west. Note that units are omitted from the axes. Any unit of length can be attributed and scale applied to the axes since a DOP value does not depend on the absolute positions of the receivers and the mobile unit but on ratios of relative distances. The DOP value itself is unitless.

Figure 2.5a displays the heat map when there are three base stations configured in the shape of a triangle. Figure 2.5b has four base stations configured in the shape of a square. To discern low PDOP values, Figure 2.5c exhibits the PDOP values within the inside of the region bounded by the four receivers in Figure 2.5b. Figure 2.5d shows four base stations with a bad configuration that results in poor PDOP values.

The heat maps illustrate which positions would yield lower errors than others. For example, the PDOP for a mobile unit located at $(0.5, 0.5)$ in Figure 2.5a is 0.96. An error of 30 m in the TDOA distance estimate, i.e. 100 ns in the TDOA estimate, will yield an error of 29 m in the position estimate. The PDOP at $(1, 1)$ is 2.6, which means that the same TDOA error will be amplified to an error of 78 m in position.

There are a few characteristics of PDOP of TDOA position estimation that can be observed from the heat maps:

Number of base stations An increase in the number of base stations yields lower DOP values. For instance, the PDOP at $(0.5, 0.5)$ for the three-receiver configuration in Figure 2.5a is 0.96. The same position has a PDOP of 0.84 for the four-receiver configuration in Figure 2.5b. Subsequent additions of receivers yield increasingly better DOPs. However, the most significant improvement in DOP is observed when the number of receivers is increased from three to four, especially for mobile units located on the outside of the region enclosed by the base stations.

As evident from the white regions in Figure 2.5a, there are areas for which a mobile unit will experience extremely high DOP when only three base stations are employed. This can be attributed to the ambiguity in position that arises for TDOA positioning in 2D with only three receivers, as discussed in Section 2.2.2.

Configuration It is important to consider the positioning and dispersion of base stations relative to each other. For example, Figures 2.5b and 2.5d both contain four base stations, but significantly lower DOP values are observed in Figure 2.5b.

DOP deteriorates on the outside It can be observed that the DOP is low within the region enclosed by the convex hull of the base stations, but deteriorates rapidly outside of the convex hull. This can be attributed to the TDOA measurements being the differences between two TOA measurements.

Consider the following thought experiment as an informal explanation. For a small change in the position of a mobile unit located on the inside of the convex hull, there will be pairs of receivers for which the TOA measurement will increase at the one receiver and decrease at the other, resulting in a larger change in the difference between the TOA measurements. For a position on the outside of the convex hull, however, a change in position closer or further from the convex hull will, for most receiver pairs, either increase or decrease the TOA measurements at both of the receivers, resulting in a smaller change in the TDOA value. A small change in the TDOA measurement on the outside of the area enclosed by the receivers results in the same change in position that a larger change in TDOA measurement on the inside of the area would result in.

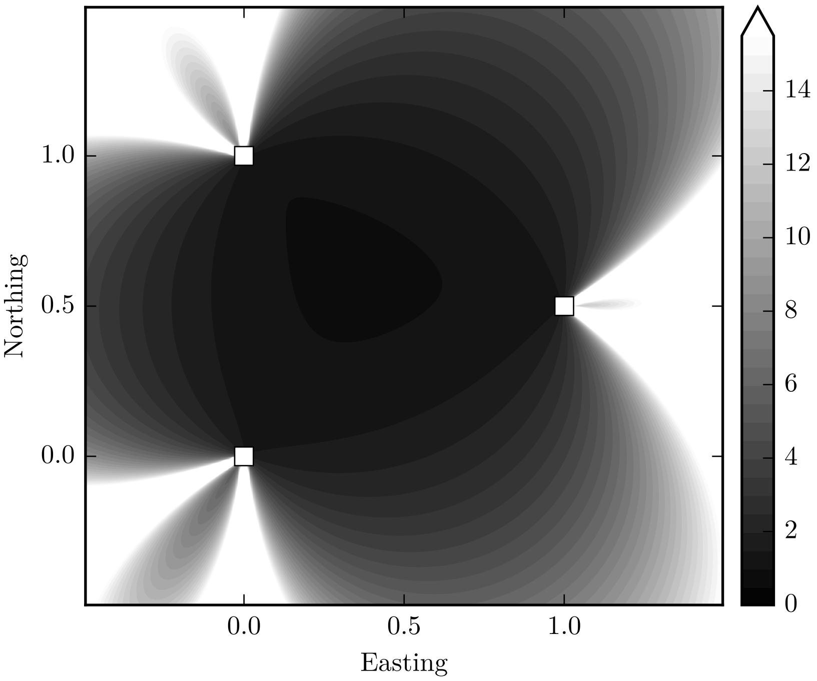
It is thus advantageous to employ base stations on the edges of the region that should be covered by a TDOA positioning system.

DOP can improve estimate A counter-intuitive and surprising observation from the heat map in Figure 2.5c is that the DOP value can be smaller than one, i.e. the positioning error can be less than the TDOA measurement error. Again, this can be attributed to the measurements being the difference of the distance between the mobile unit and pairs of receivers. When the position of a mobile unit that is located between a pair of receivers changes, the distance to the one receiver will increase and the distance to the other will decrease, resulting in up to twice the change in the difference between the distances. Thus, an error in the difference between the distances (the TDOA measurement) will be larger than the resulting error in position.

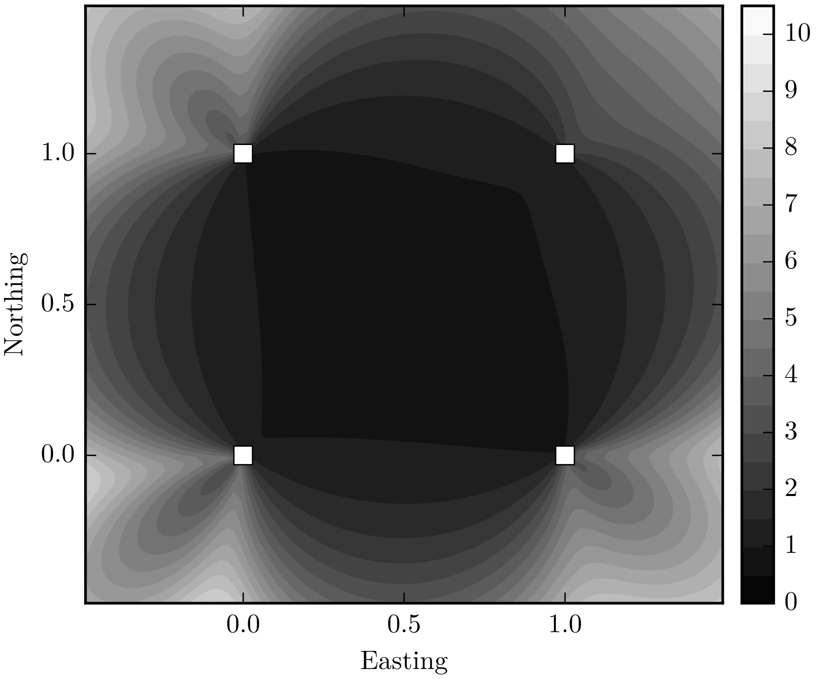
2.3.5 Conclusion

DOP characterises the contribution of the mobile unit – base station geometry to the positioning error. It can be thought of as a scaling factor that causes the amplification of measurement errors to the resulting errors in position to be more pronounced for mobile units in some locations than in others. It is important to give thought to the positions and the number of base stations when installing a TDOA positioning system since different base station configurations will yield different sensitivity to measurement errors at different locations.

The errors of the position estimates depend on the following two factors: the measurement errors and the mobile unit – base station geometry. The latter, the DOP, is the same regardless of the base station equipment or measurement techniques that are being used. The primary concern when designing a positioning system is thus not the accuracy of the position estimates, which depends on the geometry, but the accuracy of the TDOA measurements. With the knowledge of the relationship between measurement error and positioning error, we can now turn our attention to the accuracy of the TDOA measurements.

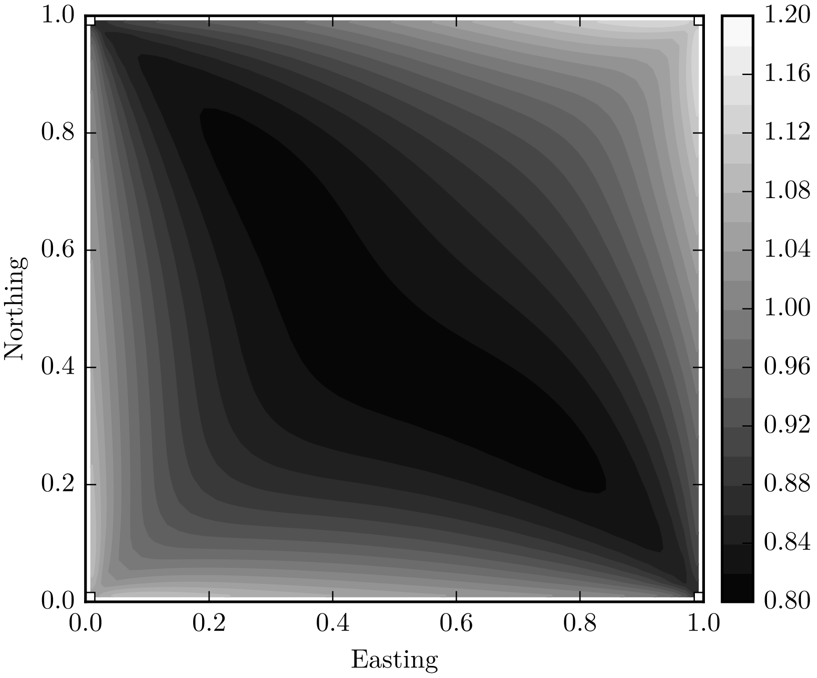


(a) Three base stations configured in a triangle.

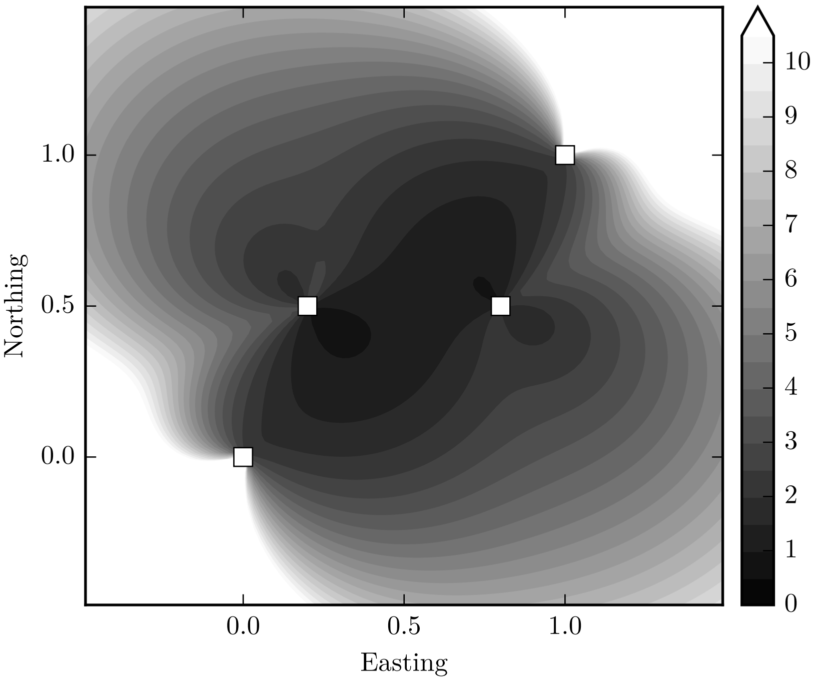


(b) Four base stations configured in a square.

(Figure continues on next page)



(c) Region enclosed by four base stations in a square.



(d) Four base stations with bad configuration.

Figure 2.5: Heat maps of PDOP for TDOA positioning with different receiver configurations. Receiver positions are indicated by white square markers. PDOP values are represented by a colour shading. The darker the shading, the lower the PDOP value.

2.4 Arrival time estimation

2.4.1 Introduction

An estimation problem that precedes the problem of estimating position from TDOA measurements is the measurement of the TDOA from observations made from the radio signal that is present at each of the receivers. TDOA estimation can be reduced to the more general problem of finding the arrival time of a radio signal at each receiver individually. Arrival time estimation is a problem shared by many applications including radar and GPS.

The arrival time estimation problem can be stated as follows. Suppose a time-limited, band-limited radio signal $s(t)$ of duration d is transmitted at time t_{tx}. Estimate the arrival time $t_{\text{tx}} + \tau$ from the observed signal $r(t)$ at the receiver, where τ is the propagation time of the signal along the direct line-of-sight path between the transmitter and the receiver. For a simple model of the propagation medium it can be assumed that $r(t)$ is a scaled and time-delayed replica of $s(t)$ with additive noise $n(t)$:

$$r(t) = A\, s(t - \tau) + n(t).$$

In the presence of multipath propagation, the received signal will consist of the superposition of multiple copies of $s(t)$ with different amplitudes and different propagation times greater or equal to τ.

In this section, we take a look at what constitutes a good positioning signal. Thus, what are essential characteristics of the signal $s(t)$ that will ensure that an accurate estimate of the arrival time can be obtained at the receiver in the presence of noise and multipath propagation?

2.4.2 Positioning signal pulse

Consider a simple positioning signal: a single pulse. A positioning signal with an instantaneous impulse will allow the receiver to record the precise instant at which the received signal's amplitude exceeds a set threshold. An infinitesimally thin spike is impractical, however, since it would require infinite bandwidth. The maximum bandwidth is limited by both regulations and the practicality of the implementation. A bandwidth-limited signal cannot change instantaneously but will bring about a rise time that is inversely proportional to the bandwidth [19]. With a slowly changing signal there is an uncertainty as to when the arrival time should be recorded. Furthermore, the longer signal edges caused by lower bandwidth can result in pulses arriving from indirect paths to interfere with the signal edges, adding to the uncertainty in arrival time.

High bandwidth is a desired characteristic of the positioning signal. Another desired characteristic is high energy, or more specifically, a high Signal-to-Noise Ratio (SNR). The magnitude of the pulse should be high enough to be distinguishable from the fluctuations of noise and to prevent false detections. The SNR also affects the accuracy of the arrival time estimation. Suppose the arrival time is recorded when the magnitude of the positioning signal reaches its maximum. Since the signal is band-limited, it takes time for its magnitude to change. There will thus be a region near the peak of the pulse for which the magnitude is close the peak magnitude. Additive noise can cause a point near the actual peak to have a greater magnitude than the peak itself and affect the point at which the arrival time is being recorded. The greater the power of the noise in comparison with the power of the positioning signal, the longer is the region over which noise can alter the position of the peak and thus the greater is the uncertainty in the arrival time estimate.

2.4.3 Cramér–Rao lower bound

Another way to look at the desired characteristics of a good positioning signal is through the Cramér–Rao Lower Bound (CRLB). The CRLB is commonly used in estimation theory for expressing a lower bound on the variance of an estimator. No unbiased estimator can perform better than the lower limit set by the CRLB.

A CRLB for delay estimation is derived in [18]. In summary, if the delay τ of a complex-valued band-limited baseband signal $s(t)$ is estimated from samples of the signal with complex-valued Additive White Gaussian Noise (AWGN),

$$r[l] = s(lT_s - \tau) + n[l], \qquad l = -L, \dots, L$$

where T_s is the sampling period and $n[l]$ represents the AWGN with zero mean and variance σ^2, then the variance of an unbiased estimator that estimates τ from the samples $\boldsymbol{r} = \begin{bmatrix} r[-L] & \cdots & r[L] \end{bmatrix}^T$ is at least

$$\mathrm{Var}[\tau(\boldsymbol{r})] \geq \frac{\sigma^2}{2\sum_{l=-\infty}^{\infty} \left| \frac{d}{d\tau} s(lT_s - \tau) \right|^2} \tag{2.23}$$

when $L \to \infty$.

It can be seen from Equation (2.23) that there is a linear relationship between the CRLB and the noise power. Thus, the minimum variance of the estimator will be cut in half if the SNR increases by a factor of 2 (3 dB). Also note that, for a set noise power, the CRLB is inversely proportional to the energy of the sampled derivative of $s(t)$. The more rapid the signal changes are, the greater the magnitude of the derivative will be and the lower the resulting CRLB [18].

The CRLB can be expressed in terms of the spectrum of $s(t)$ for further insight into how signal properties influence the CRLB. As shown in [18], if $S(f)$ is the Fourier transform of the baseband signal $s(t)$, if $s(t)$ is band-limited such that $S(f) = 0$ for $|f| > \frac{B}{2}$, and if the received signal is sampled at a sample rate of $f_s = B$, then the CRLB in terms of the spectrum of $s(t)$

is:

$$\mathrm{Var}[\tau(\boldsymbol{r})] \geq \frac{\sigma^2 T_s}{8\pi^2 \int_{-\frac{f_s}{2}}^{\frac{f_s}{2}} f^2 \left|S(f)\right|^2 df}. \tag{2.24}$$

The linear relationship between the CRLB and the noise power is again visible in Equation (2.24). Also visible is that an increase in the bandwidth of the signal will result in a lower minimum bound on the estimator's variance. Furthermore, note that the more energy is distributed at higher frequency components, the lower the CRLB will be and thus the better the performance of the delay estimator will be.

2.4.4 Spread spectrum

The wider the signal's bandwidth and the higher the SNR, the more accurate the arrival time estimate will be. The noise power is generally beyond control, and the positioning signal's bandwidth and the transmission power are limited by regulations. There is, however, another way to improve the SNR, namely increasing the transmission time. Care should be taken to ensure that the longer duration of the signal does not reduce its bandwidth. For example, simply increasing the duration of a pulse signal would reduce the bandwidth of the signal.

Spread spectrum techniques are commonly used in data communication to increase the signal's bandwidth beyond the minimum necessary for the given baud rate. This makes spread spectrum signals prime candidates for arrival time estimation. Spread spectrum techniques can provide a signal that meets both of the desired characteristics, namely high bandwidth and high energy. Spread spectrum also provides additional benefits, such as better resistance to interference, interception and fading [28, 29].

Common spread spectrum techniques are chirp, direct sequence, frequency hopping and time hopping spread spectrum. Each of the different spread spectrum techniques is used differently for distance measurements, and can even be combined to form hybrid techniques [19, 28]. In this dissertation, we consider DSSS since it can be processed digitally and since it is used by similar positioning systems [1, 11, 15]. Arrival time estimation using DSSS is described in the next chapter.

2.5 Chapter summary

In summary, location is always expressed and measured relative to one or more objects with known positions. With radio positioning, the physical properties of electromagnetic wave propagation is used to measure the position of an object. One of the properties that exhibit a position-dependent relationship is the propagation time of the wave. Multiplying the time it takes a signal to propagate from a transmitter to a receiver with the propagation speed yields the distance between the transmitter and the receiver. We focus on network-positioning, whereby a network of receivers with known positions is used to estimate the position of one or more transmitters.

The propagation time of a positioning signal cannot be calculated from the difference between the transmission time at the transmitter and the arrival time at the receiver since the transmission time is unknown. Transmission time can be eliminated from the equation by using the difference in the propagation time between the transmitter and two receivers, which is equivalent to the difference in the arrival time at the two receivers, called the Time Difference of Arrival (TDOA). When relating a TDOA value to position in 2D space, the value defines a hyperbola branch on which the transmitter is located. The differences in arrival time between other pairs of receivers are used to define more hyperbola branches. The intersection point of the hyperbola branches yields the position of the transmitter. The position of the transmitter is calculated by solving the system of nonlinear equations numerically, or the equations are linearised and solved analytically. In some cases, arrival time measurements at three receivers are sufficient for solving the position of the transmitter in 2D space, but measurements from at least four receivers are required to solve the position unambiguously.

Once accurate TDOA estimates have been obtained, the problem of finding the position from the estimates is essentially the same for all TDOA systems. Furthermore, the errors of the position estimates depend on two factors, namely the TDOA measurement errors and the mobile unit – base station geometry. The latter, the DOP, is the same regardless of the base station equipment or measurement techniques being used. The primary concern when designing a positioning system is thus not the accuracy of the position estimates, which depends on the geometry, but the accuracy of the TDOA measurements. Consequently, the focus of this dissertation is aimed at the process of and techniques for accurate TDOA estimation.

TDOA estimation can be reduced to the problem of finding the arrival time of a radio signal at each receiver individually. The higher the SNR and the wider the signal's bandwidth, the more accurate the arrival time can be estimated from samples of the received signal. The positioning signal should be selected carefully to meet both of these requirements, i.e. a wideband signal with high energy. Spread spectrum techniques can be used to attain these requirements. The use of one of the spread spectrum techniques, namely DSSS, for arrival time estimation is investigated in the next chapter.

Chapter 3

Arrival time estimation using DSSS

This chapter presents an analysis and summary of various aspects of arrival time estimation with DSSS signals and serves as the foundation of the design in the next chapter. First, an overview of DSSS is given in Section 3.1. Interpolation methods for improving the resolution of the arrival time estimate to a resolution that is smaller than the sample period, are then presented in Section 3.2. An analysis and comparison of the frequency spectra of different modulation schemes for modulating the DSSS code are provided in Section 3.3. The problem of deciding whether a positioning signal is present or not is discussed in Section 3.4. It is then shown in Section 3.5 that, for a code modulated with On–Off Keying (OOK), the carrier can be detected from the Discrete Fourier Transform (DFT) of the signal. Finally, the chapter is concluded with a summary in Section 3.6.

3.1 An overview of DSSS

To avoid duplication of existing literature that already provides good primers on the concepts surrounding DSSS, this section provides only a terse overview of DSSS as it relates to the problem of arrival time estimation. Refer to [1, 11, 19, 30] for more comprehensive information.

3.1.1 Introduction

Direct Sequence Spread Spectrum (DSSS) is commonly used in data communication to allow greater resistance to interference by spreading the transmitted signal over a bandwidth that is significantly greater than the minimum bandwidth required for communicating the message signal [19]. The DSSS signal is formed by using the message signal to modulate a predefined sequence of bits with a significantly shorter duration than that of the data symbols. The sequence of bits is called a *code*, and the bits are called *chips*. Each data symbol is effectively sliced into a sequence of bits of much shorter duration, resulting in a DSSS signal of which the bandwidth is larger than that of the message signal.

At the receiver, the message signal is reconstructed by counteracting for the effect of the code. This is done by lining up a local replica of the code, which will be referred to as the *template*, to the code embedded in the received signal and, at the position of each symbol, multiplying the local code signal and the received signal together. A bipolar code signal will ensure that the chips of the template counteract the chips in the received signal, resulting in the "despreaded" message signal. Note that the code itself is not considered as data since it is known in advance at the receiver.

To properly despread the message signal, it is necessary that the local replica of the code lines up with the expected code in the received signal. One method to synchronise the code phase is to cross-correlate the incoming signal with the local template signal. The cross-correlation function will exhibit maximum magnitude where the template aligns best with the incoming signal. A precise line-up yields the position, i.e. the arrival time, of the code that is embedded in the incoming signal. The ability to estimate the code phase of the incoming signal forms the basis of the use of DSSS for arrival time estimation and ultimately position estimation.

If the transmitted signal is used for the sole purpose of positioning, no data needs to be communicated. In that case, the message signal can consist of a single bit, resulting in a transmitted signal with a single occurrence of the code signal.

3.1.2 Spreading codes

The capability to synchronise to the code phase of the incoming signal lies within the properties of the code. To prevent false alignment, there should be low correlation between the code within the incoming signal and a misaligned template, as well as between the code and unintended signals. A sharp and distinct cross-correlation peak should be visible only when the template lines up precisely with an image of itself within the received signal. Since the template and the code within the incoming signal are identical, this property can be described in terms of the autocorrelation of the code.

Let $(c_j)_{j=1}^{N}$ be the code sequence. The autocorrelation coefficients are

$$a_i = \sum_{j=1}^{N-i} c_i c_{i+j}, \qquad 0 \leq i \leq N-1 \tag{3.1}$$

where i represents the lag in chips between the code and a time-shifted replica. A code with good autocorrelation properties for the purpose of synchronisation will exhibit off-peak autocorrelation coefficients that are significantly smaller in magnitude than the zero-lag coefficient, thus

$$|a_i| \ll |a_0| \qquad \forall\, i : 1 \leq i \leq N-1. \tag{3.2}$$

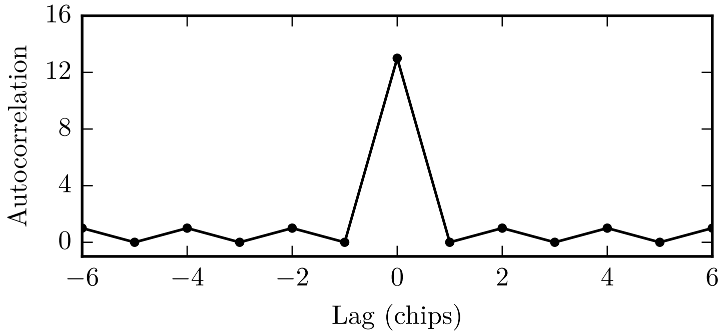


Figure 3.1: Autocorrelation of a length-13 Barker code.

It is this sharp autocorrelation peak at zero time lag that enables precise arrival time detection.

As an example of a code with good autocorrelation properties, consider the autocorrelation function of the Barker code of length 13 displayed in Figure 3.1. Barker codes are sequences with ideal autocorrelation properties, i.e. their off-peak autocorrelation coefficients are as small as possible. If chips take on values of $+1$ and -1, the maximum magnitude of the off-peak autocorrelation coefficients is one. However, it is conjectured that no Barker codes exist of length more than 13 [31]. Other sequences with good correlation properties should be employed for code sequences that are more than 13 chips long.

A random sequence of bits exhibits, statistically, good autocorrelation properties. However, a perfectly random code is impractical since the code needs to be reproducible by both the receiver and the transmitter. Instead, Pseudo Random Noise (PRN) codes are generally used. PRN codes appear similar to random sequences, but are generated deterministically and satisfy one or more of the standard tests for statistical randomness. A PRN code seems to lack any definite cyclical pattern, but it is of finite length and will repeat itself.

Examples of PRN sequences are memory codes, maximum length sequences, Gold codes and Kasami codes. Memory codes are random codes known to exhibit good autocorrelation properties that are stored in memory instead of being generated deterministically. A maximum length sequence, also called an m-sequence, is generated using a maximal Linear Feedback Shift Register (LFSR). Its length is maximal since it produces every binary sequence that can be represented by the shift register except for the case where all the registers are zero. Thus, if the LFSR has m registers, the m-sequence will enter a cycle and repeat itself after $2^m - 1$ chips. Different configurations of the LFSR will yield sequences with different properties.

Many code types provide families of codes with low cross-correlation between the codes in the family. The low cross-correlation allows multiple transmitters, each with a different code, to transmit on the same channel simultaneously without interfering with one another — a method commonly known as Code Division Multiple Access (CDMA). Code types such as Gold codes and Kasami codes are generated from m-sequences and provide sets of codes that are highly orthogonal to one another. The cross-correlation between two codes in the family is guaranteed to be below a certain threshold. The C/A codes used in the GPS L1 signals are, for example, Gold codes with a period of 1023 chips. Each satellite transmits on the same frequency, but with a unique code within a family of Gold codes.

Refer to [28] for a summary of different code types, their qualities, and how they are produced.

3.1.3 Matched filter

One strategy to search for and acquire the phase of the code within the received signal is to slide the template signal over the received signal, one time step at a time. The template is correlated with the received signal at each time step in order to check how well the received signal matches the template at that code phase. The template is considered to be aligned with the received signal when the output of the correlator exceeds a set threshold. Such a correlator that matches a template signal to a received signal is generally referred to as a *matched filter*.

Consider the example displayed in Figure 3.2. In this example, a length-31 Gold code is used as the template signal and a time-shifted replica of the template corrupted by AWGN as the received signal. The output of the matched filter is displayed, representing the correlation between the two signals as the template is shifted across the received signal. The signals are sampled at a sample rate of four samples per chip and the template is shifted one sample at a time. The correlator output exhibits a peak when the template has been shifted by 40 chips, indicating the position where the template is best aligned with the code of the incoming signal and providing an estimate of the signal's arrival time.

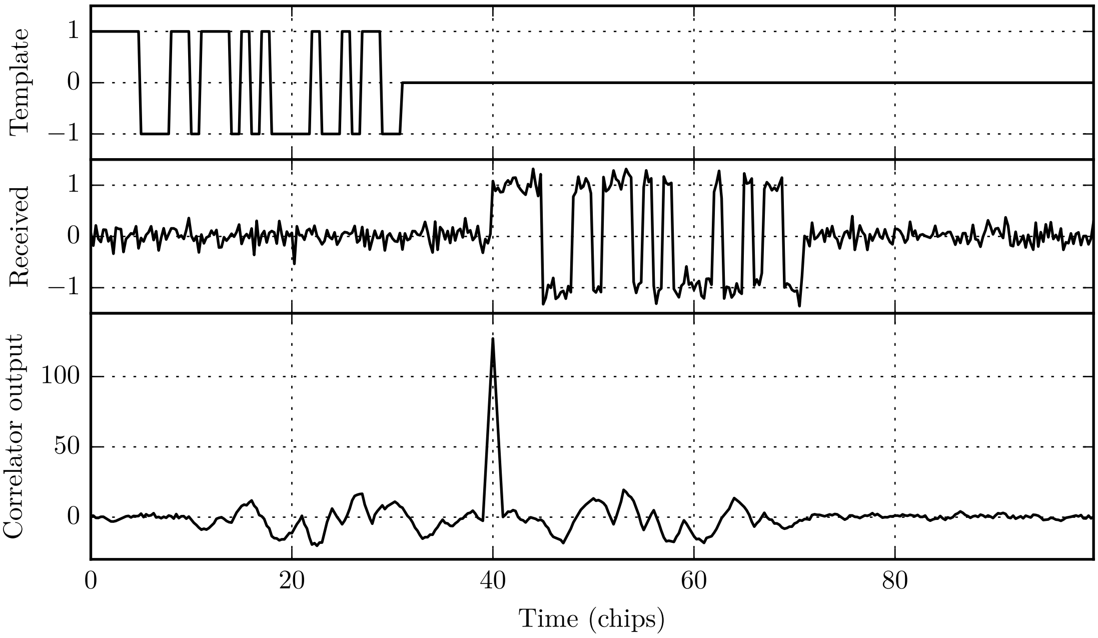


Figure 3.2: Example showcasing the output of a matched filter that matches a length-31 Gold code to a received signal.

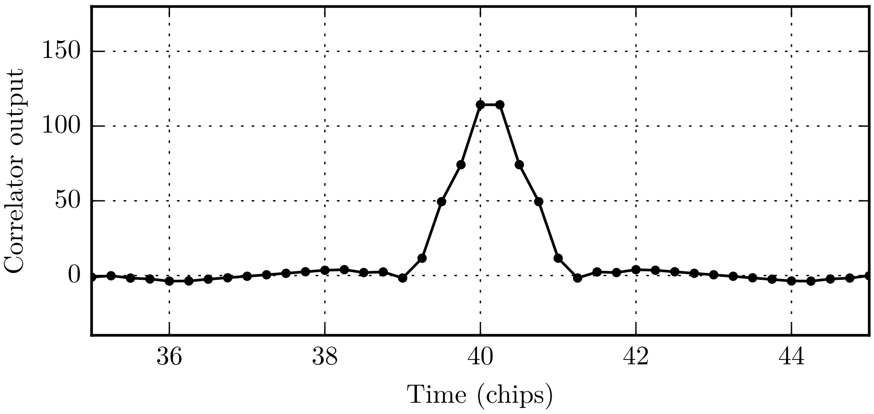


Figure 3.3: Output of the correlator when the correlator's step size and the chip boundaries are misaligned by half a sample.

Since the clocks at the transmitter and receiver are not synchronised, and since the correlation is evaluated at discrete time steps, the template may not line up perfectly with the code embedded in the incoming signal. For example, Figure 3.3 displays the output of the correlator in Figure 3.2 when the step size of the template and the phase of the chips in the received signal are misaligned by half a sample, yielding a blunt correlation peak and resulting in a quantisation error in the estimated code phase. The quantisation error may be tolerable if the spread spectrum signal is used solely for communication, but precise synchronisation is required for high-resolution arrival time estimation. The smaller the step size of the correlator, and consequently the higher the sample rate of a digital correlator, the smaller the quantisation error. A smaller step size, however, increases the computational complexity of the receiver.

If the transmitted signal consists of several instances of the code, i.e. if the message signal is several symbols long, precise synchronisation can occur in two stages. The first stage, *code acquisition*, occurs when the presence of a code is unknown and the code phase is completely unsynchronised. This involves coarse synchronisation of the code phase to within one chip by correlating the incoming signal with the template at discrete time intervals and finding the point of maximum correlation, as discussed above. Once the presence of the code is detected and the code phase coarsely synchronised, precise synchronisation commences with the second stage, called *tracking*. A closed-loop feedback loop such as a delay lock loop (DLL) is used to lock onto and fine-tune the code phase. If the code is transmitted continuously, as is the case with GPS, the step size of the correlator may be even longer than the chip duration since the template will eventually line up at least partially, after which the tracking stage can take over for precise synchronisation.

Tracking of the code phase allows for precise synchronisation, but it requires the transmitted signal to be long enough for the feedback loop to converge to a steady state. However, transmission time is a valuable resource in a low-power multilateral tracking system. To limit the energy consumption of the transmitter for a single position estimate, we look into the use of a short-lived positioning signal that consists of one or only a few repetitions of the code. A short-lived signal eliminates the use of a tracking loop as a viable option.

Another approach to circumvent the limited resolution of the code phase estimate due to the discrete step size of the correlator is to interpolate between the samples of the correlation peak. The next section is devoted to the topic of subsample interpolation.

3.2 Subsample interpolation

It was shown in the previous section that the time of arrival of a code embedded in a received signal can be estimated as the time at which the discrete cross-correlation between the received signal and a template signal yields a peak. The best resolution that can be achieved with this method is governed by the step size of the correlator, which is typically equal to the sample period. It is however possible to improve the resolution of the arrival time estimate to an accuracy that is better than the sample period by interpolating between the samples of the correlation peak.

It is rather surprising that none of the books on radio positioning that we came across mention subsample interpolation as a technique for improving the resolution of the arrival time estimate [11, 18, 19]. The reason is probably that these books are mostly concerned with position estimation using the signals of existing communication signals that are several symbols long or continuous signals such as GPS, for which closed-loop tracking is used for precise synchronisation. Literature on positioning systems that use signals of short duration for multilateral positioning mentions interpolation in passing and seemingly overlooks or understates the importance of interpolation [1, 16]. Literature on subsample interpolation for other applications such as radar or ultrasound is however available in abundance [32, 33]. Although the application differs, the problem remains the same: to estimating the delay between two signals. In our case, the two signals are the ideal template signal and a received signal consisting of a time-delayed replica of the template with unknown delay that is corrupted by noise. This section summarises a few simple interpolation methods.

The subsample interpolation problem can be stated as follows. Let $h[n]$ be the samples of a template signal with N samples; let $x[n]$ be the samples of the received signal; and let $y[n]$ be the output of the discrete correlator,

$$y[n] = \sum_{i=0}^{N-1} h^*[i]\, x[n+i] \tag{3.3}$$

where h^* denotes the complex conjugate of h. Let p be the integer-valued index of the maximum-magnitude sample of a peak within the correlation output, representing the coarse estimate of the code phase. Let δ be the difference between the actual code phase and the coarse code phase, i.e. the quantisation error of the coarse estimate, $-0.5 \leq \delta < 0.5$. The goal is to construct an estimator that estimates δ from the samples surrounding the correlation peak. We will refer to δ as the *peak offset*.

One way to interpolate between the discrete samples of the correlation peak is to model the expected shape of the peak as an analytic function, to fit the function to the discrete samples surrounding the peak, and to estimate the code phase from the extremum of the function. A computationally efficient method is to fit the function to the peak sample and its two neighbours. For example, a common interpolation method is to fit a parabola,

$$f_{\text{para}}(t) := at^2 + bt + c.$$

The estimated peak offset when the quadratic function is fit to three samples can be expressed as:

$$\hat{\delta}_{\text{para}} = -\frac{b}{2a} = \frac{|y_1| - |y_{-1}|}{4\,|y_0| - 2\,|y_{-1}| - 2\,|y_1|} \tag{3.4}$$

where $y_0 = y[p]$ is the value of the peak sample, and $y_{-1} = y[p-1]$ and $y_1 = y[p+1]$ the values of the samples to the left and right of the peak respectively. This interpolation method is generally referred to as *parabolic interpolation* or *quadratic interpolation* [33, 34].

Another analytic function that can be used to model the peak is the Gaussian function [33–35],

$$f_{\text{gauss}}(t) := a\,e^{-b(t-c)^2}.$$

The estimated peak offset when *Gaussian interpolation* is applied to the three samples surrounding the peak is:

$$\hat{\delta}_{\text{gauss}} = \frac{\ln|y_1| - \ln|y_{-1}|}{4\ln|y_0| - 2\ln|y_{-1}| - 2\ln|y_1|}. \tag{3.5}$$

Note that this is equivalent to fitting a parabola to the natural logarithm of the samples.

The shape of the correlation peak can also be approximated as the crest of a cosine function. The estimated peak offset with *cosine interpolation* is [33, 34]:

$$\begin{aligned}
\omega &:= \arccos\left(\frac{|y_{-1}| + |y_1|}{2\,|y_0|}\right)\\
\phi &:= \arctan\left(\frac{|y_{-1}| - |y_1|}{2\,|y_0|\sin(\omega)}\right)\\
\hat{\delta}_{\cos} &= -\frac{\phi}{\omega}.
\end{aligned} \tag{3.6}$$

The three three-point interpolation methods described above are all closed-form estimators that rely on an approximation of the shape of the correlation peak. Since the analytic function is an approximation, an approximation error will be inherent to the estimate. This error is especially noticeable when the SNR is very high, in which case the approximation error may be greater than the variation due to noise [33]. Another way to estimate the subsample arrival time is by means of iterative methods that do not rely on analytic functions.

For example, numerical optimisation can be used to iteratively search for the peak offset that maximises the magnitude of the correlation peak when the peak is interpolated from the DFT of the correlator output [33]. Let $x_p[n]$ be the extract of samples from the received signal that yields the correlation peak when being correlated with the template,

$$x_p[n] = x[p+n], \qquad 0 \leq n \leq N-1; \tag{3.7}$$

let $y_p[n]$ be the discrete circular cross-correlation between $x_p[n]$ and $h[n]$; and let $Y_p[k]$ be the DFT of $y_p[n]$. The interpolated value of the correlation function when being evaluated at offset δ relative to the coarse correlation peak is then

$$y_p(\delta) = \sum_{k=0}^{N-1} Y_p[k] e^{2\pi j \delta k/N}. \tag{3.8}$$

The peak offset can be estimated by using numerical methods to find the offset that maximises $y_p(\delta)$:

$$\hat{\delta}_{\text{maxi}} = \operatorname*{arg\,max}_{\delta \in [-0.5, 0.5]} |y_p(\delta)| \, . \tag{3.9}$$

Another iterative method, one that we have improvised without thorough mathematical analysis but that proved to deliver good results, is to fit the cross-correlation peak and a few samples on either side of the peak to the template's autocorrelation function. This is performed by finding the scaling factor and time-shift that would yield the smallest sum of square residuals between the autocorrelation samples and the amplitude-scaled, time-shifted cross-correlation samples, i.e. the LS fit. To ensure that samples closer to the peak, which have higher power and are less affected by noise, contribute more to the least squares solution, the residuals are weighted by the samples' distances from the peak.

More concretely, let $z[n]$ be the cross-correlation peak and m samples on either side of the peak,

$$z[n] = y[p+n], \qquad -m \leq n \leq m, \tag{3.10}$$

and let $Z[k]$ be the DFT of $z[n]$. Let $Z_\delta[k]$ be the DFT after a time-shift of δ has been applied, i.e. the generalised DFT,

$$Z_\delta[k] = Z[k] e^{2\pi j \delta k/N}, \tag{3.11}$$

and let $z_\delta[n]$ be the IDFT of $Z_\delta[k]$, i.e. $z[n]$ shifted in time. Moreover, let $a[n]$ be the autocorrelation coefficients of the length-N real-valued template signal $h[n]$,

$$a[n] = \begin{cases} \sum_{j=0}^{N-1} \left(h[j] \cdot h[N-j-1] \right) & 0 \leq n \leq m, \\ a[-n] & -m \leq n < 0. \end{cases} \tag{3.12}$$

The sum of weighted squared residuals when the cross-correlation samples are time-shifted by δ

and scaled in amplitude by a factor of A is then

$$S(A, \delta) = \sum_{n=-m}^{m} \frac{1}{n^2} (A \; |z_\delta[n]| - |a[n]|)^2. \tag{3.13}$$

The best fit yields the estimated peak offset:

$$\begin{bmatrix} \hat{A}_{\text{fit}} \\ \hat{\delta}_{\text{fit}} \end{bmatrix} = \arg\max_{A,\delta} S(A, \delta) \tag{3.14}$$

where the offset δ is constrained to the range $-0.5 \leq \delta \leq 0.5$.

Five interpolation methods were presented in this section, but myriads of alternatives exist as well. For instance, the time delay between two signals can be estimated from the phase shift in the frequency domain. This method generally exhibits lower biasing errors than the three-point interpolation methods [34]. However, since the selection of an interpolation method is not the primary goal of this dissertation, we only assess the five simple interpolation methods that were summarised in this section. Evaluation of other interpolation methods can be addressed in future work.

3.3 Modulation schemes

DSSS was introduced in Section 3.1 without a discussion on how the binary spreading code is conveyed as an analogue signal and modulated onto a carrier signal for transmission. Most of the literature on DSSS that we came across assumes Binary Phase Shift Keying (BPSK) without any mention of alternatives. Besides the phase, other properties of the signal that can be used to represent digital data are the signal's amplitude and frequency. The literature we found that makes mention of modulation schemes does so in passing or with data communication as application. Digital modulation schemes are commonly evaluated in terms of bit error rate, which provides valuable information for data communication, but not for radio positioning. The bit errors are not our concern since no data is being communicated, but we are concerned about the impact of the modulation scheme on the accuracy of the arrival time estimate.

Since the timing resolution depends on the signal bandwidth and not on the data rate, mapping more than one bit to a symbol is not beneficial for arrival time estimation. Thus, what matters for arrival time estimation is the baud rate, not the bit rate. Therefore, we only consider the binary forms of the modulation schemes.

In this section, the binary modulation schemes are evaluated in terms of their frequency spectra.

3.3.1 Phase shift keying (PSK)

The modulation scheme that is most commonly employed for DSSS is BPSK [19]. With BPSK, the phase of the carrier is shifted by 180° according to the state of the code chips. The phase components can, for example, be 0° for binary zero and 180° for binary one. At baseband, the modulated signal is a bipolar encoding of the code chips where a positive voltage denotes binary one and a negative voltage denotes binary zero.

Let $c_{\text{psk}}[n]$ be a sequence of length N_c that represents the code symbols with unit amplitude, thus $+1$ for binary one and -1 for binary zero. The analogue baseband signal for one period of the code modulated with BPSK can then be expressed as: [11]

$$x_{\text{psk}}(t) = \sum_{n=0}^{N_c-1} c_{\text{psk}}[n] \operatorname{rect}\left(\frac{t - nT_c}{T_c}\right) \tag{3.15}$$

where the rectangular function is defined as:

$$\operatorname{rect}(t) := \begin{cases} 1 & |t| < \frac{1}{2}, \\ 0.5 & |t| = \frac{1}{2} \\ 0 & |t| > \frac{1}{2}. \end{cases} \tag{3.16}$$

The energy spectral density of Equation (3.15) is: [11]

$$|X_{\text{psk}}(f)|^2 = T_c^2 N_c \operatorname{sinc}^2(T_c f) \, |X_{\text{code}}(f)|^2 \tag{3.17}$$

where $X_{\text{psk}}(f)$ denotes the Fourier transform of $x_{\text{psk}}(t)$ and where X_{code}, which we will call the *code transform* is defined as

$$X_{\text{code}}(f) := \frac{1}{\sqrt{N_c}} \sum_{n=0}^{N_c-1} c_p[n] \, e^{-j2\pi f n T_c} \tag{3.18}$$

and where sinc is the normalized sinc function,

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} & x \neq 0 \\ 1 & x = 0. \end{cases} \tag{3.19}$$

To analyse the properties of the spectrum mathematically without evaluating it for a specific code, the PRN code can be approximated as noise under the assumption of a long random code, and $c_p[n]$ can be replaced with a random variable. The expected magnitude of the code transform for a random code is: [11]

$$\mathbf{E}\{|X_{\text{code}}(f)|\} = 1. \tag{3.20}$$

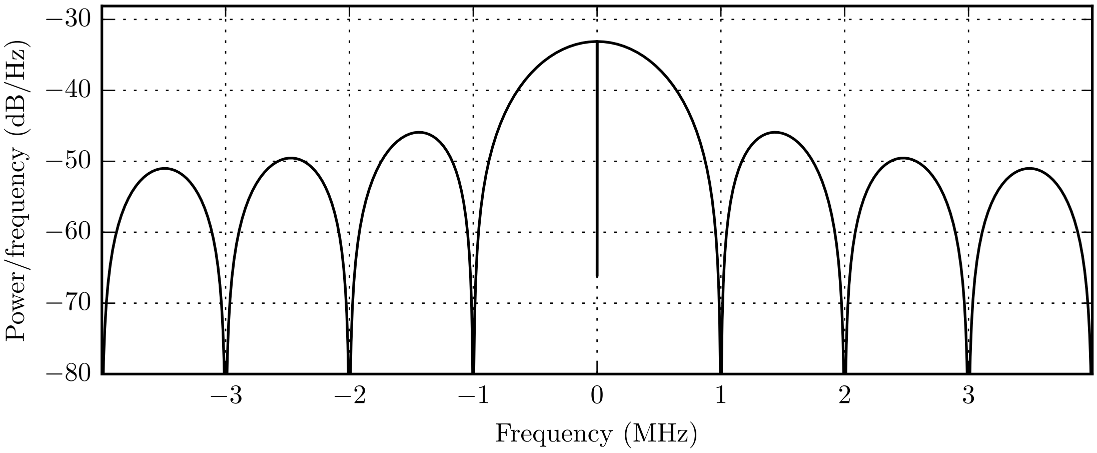


Figure 3.4: The periodogram, which is an estimate of the power spectral density, of a PSK-modulated Gold code with 2047 chips and a chip rate of 1 MHz, normalised to unity power.

The energy spectral density in Equation (3.17) then becomes:

$$\mathbf{E}\left\{|X_{\text{psk}}(f)|^2\right\} = T_c^2 N_c \operatorname{sinc}^2(T_c f). \tag{3.21}$$

It follows that the power spectral density is:

$$S_{\text{psk}}(f) = \lim_{T_d \to \infty} \frac{\mathbf{E}\left\{|X_{\text{psk}}(f)|^2\right\}}{T_d} = T_c \operatorname{sinc}^2(T_c f) \tag{3.22}$$

where $T_d = T_c N_c$ is the duration of the signal. It is evident from the equation that the maximum power density is:

$$\max\left(S_{\text{psk}}(f)\right) = T_c. \tag{3.23}$$

The total power is:

$$P_{\text{psk,total}} = T_d = N_c T_c \tag{3.24}$$

and the power spectral density normalised to unity power is:

$$S_{\text{psk,norm}}(f) = \frac{S_{\text{psk}}}{P_{\text{psk,total}}} = \frac{1}{N_c} \operatorname{sinc}^2(T_c f). \tag{3.25}$$

The maximum power spectral density as a fraction of the total power is thus:

$$\max\left(S_{\text{psk,norm}}(f)\right) = \frac{1}{N_c}. \tag{3.26}$$

The estimated power spectral density (periodogram) of a length-2047 Gold code that is modulated with PSK at a chip rate of $f_c = \frac{1}{T_c} = 1\,\text{MHz}$, normalised to unity power, is displayed in Figure 3.4. Note that the shape of the spectrum resembles that of a squared sinc function, which corresponds with Equation (3.21). Furthermore, note that the maximum power spectral density in the figure agrees with Equation (3.26):

$$10 \log_{10} \frac{1}{N_c} = -33\,\mathrm{dB/Hz}. \tag{3.27}$$

From Figure 3.4 as well as the squared sinc function in Equation (3.22), it is clear that the signal's power is spread over a wide range of frequencies. In fact, the power spectrum extends to infinity due to the rapid transitions between subsequent chips in Equation (3.15). However, roughly 90 % of the sinc function's power is contained within its main lobe:

$$\begin{aligned} \int_{-\infty}^{\infty} \operatorname{sinc}^2(x)\, dx &= 1 \\ \int_{-1}^{1} \operatorname{sinc}^2(x)\, dx &= 0.90. \end{aligned} \tag{3.28}$$

As evident from Equation (3.22) and visible from Figure 3.4, the main lobe spans a passband bandwidth of 2 MHz when the chip rate is 1 MHz. Thus, the signal will suffer minimal energy loss if the signal is band-limited to 2 MHz.

The random or PRN code causes the signal's power to span a wide range of frequencies without a distinct narrowband peak with high spectral density at any particular frequency. Compare that to an unmodulated carrier where the power is concentrated at the carrier frequency. As evident from Equation (2.24), signal power at the carrier frequency does not contribute to a better arrival time estimate. It is thus advantageous in terms of the accuracy of the arrival time estimate that PSK modulation suppresses the carrier. Furthermore, the lack of narrowband peaks reduces interference with other transmitters and systems that are sharing the channel.

Carrier frequency mismatch

The disadvantage of the suppressed carrier is that it makes carrier frequency recovery harder. Factors such as clock errors cause a mismatch between the local oscillator frequencies of the transmitter and receiver. Accurate carrier frequency offset recovery is vital for arrival time estimation. Consider Figure 3.5, which displays the cross-correlation between a template and a received signal for a length-31 Gold code. The received signal is identical to the template except for a frequency mismatch. Despite the absence of noise, the correlation between the received signal and the template is low where the two signals line up. The frequency mismatch causes the sign of the received signal to be inverted halfway through its length, which causes the cross-correlation of the first half to cancel out the cross-correlation of the second half, eliminating the correlation peak.

If A is the power of the autocorrelation peak of a random code, then the expected power of the correlation peak resulting from the cross-correlation of the template with a frequency-shifted replica is: [11]

$$\mathbf{E}\left\{P_{\text{corr-peak}}\right\} = A \operatorname{sinc}^2(\Delta f\, T_d) \tag{3.29}$$

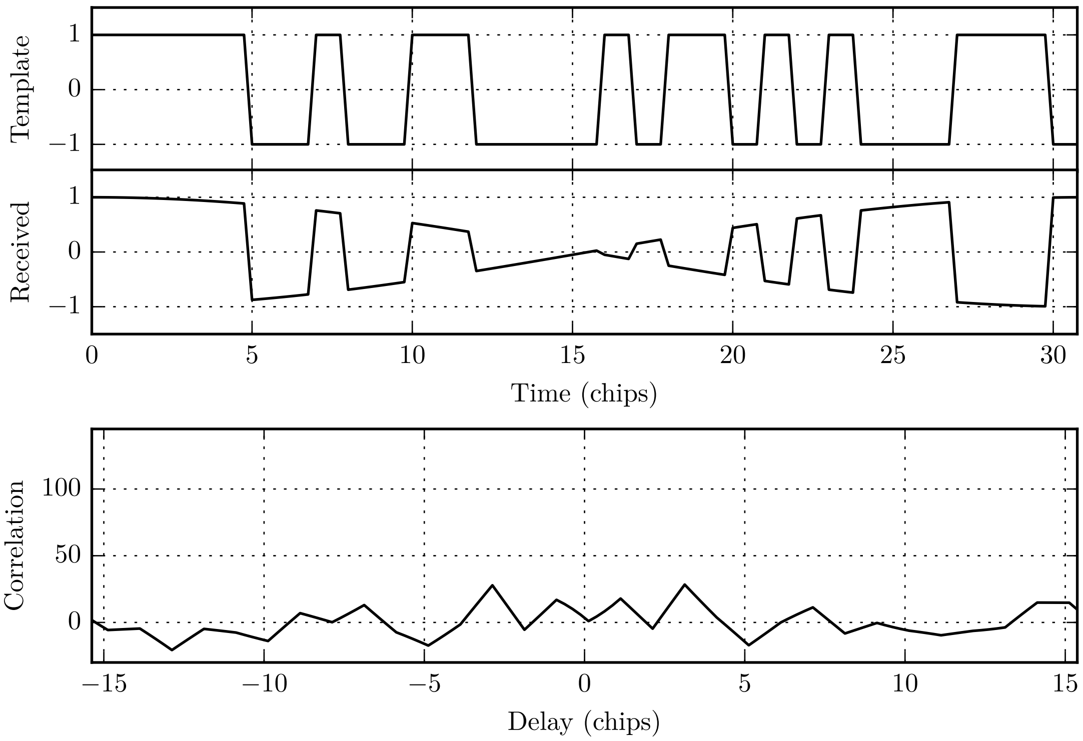


Figure 3.5: An example showcasing the effect of a frequency offset between the template and received signal on the output of the correlator. The correlation peak is eliminated despite the absence of noise. The figure is adapted from [1].

where Δf is the frequency offset and T_d the duration of the code signal. As evident from the equation above, reducing the duration of the code will reduce the impact of the frequency offset. However, the impact of a frequency mismatch can be substantial even for a relatively short code signal. For example, if a length-2047 Gold code is transmitted at a chip rate of 1 MHz, then the duration of the signal is 2 ms. A frequency offset of only 222 Hz will halve the power of the cross-correlation peak. Another approach to reduce the impact of the frequency offset is to limit the offset with the use of accurate clocks. However, if the code mentioned above is modulated onto a 434 MHz carrier, the frequency error between the receiver's clock and the transmitter's clock should be smaller than 0.5 ppm, which is challenging when cost and size are important design constraints and when the devices are exposed to a wide range of temperatures. If neither the signal duration nor the frequency offset can be held within acceptable limits, the detection process becomes a two-dimensional search over arrival time and frequency offset, which increases the computational requirements of the receiver [1].

3.3.2 On–off keying (OOK)

On–Off Keying (OOK) denotes a straightforward amplitude modulation technique. A binary one is represented by the presence of the carrier and a binary zero by its absence. Modulation is thus as simple as switching an unmodulated carrier between "on" and "off". At baseband, the modulated signal is a unipolar encoding of the code chips where a positive voltage denotes binary one and a negative voltage denotes binary zero.

Let x_{ook} be the OOK-modulated analogue baseband signal for one period of the code, which is the same as Equation (3.15) except that c_{psk} is substituted by the code symbols for OOK modulation, i.e. +1 for binary one and 0 for binary zero. For spectral analysis it is convenient to view the OOK-modulated signal as the linear combination of the PSK-modulated code signal and an unmodulated carrier:

$$x_{\text{ook}}(t) = 0.5\,x_{\text{psk}}(t) + 0.5\,x_{\text{carrier}}(t) \tag{3.30}$$

where $x_{\text{psk}}(t)$ is as defined in Equation (3.15) and $x_{\text{carrier}}(t)$ is an unmodulated carrier signal,

$$x_{\text{carrier}}(t) := \text{rect}\left(\frac{t - T_d/2}{T_d}\right). \tag{3.31}$$

The duration of the unmodulated carrier signal is equal to the duration of the code signal, i.e. $T_d = N_c T_c$.

The energy spectral density of $x_{\text{carrier}}(t)$ is

$$|X_{\text{carrier}}(f)|^2 = T_d^2 \,\text{sinc}^2\left(T_d f\right), \tag{3.32}$$

which, for a long code ($N_c \gg 1$), exhibits a narrow main lobe in comparison with the wide main lobe of $|X_{\text{psk}}|^2$. For example, for a length-2047 code with a chip rate of 1 MHz, the main lobe of the carrier spectrum spans 1 kHz, while the main lobe of the PSK-modulated signal is spread over 2 MHz. Furthermore, the spectrum of the carrier exhibits a peak with high spectral density,

$$\max|X_{\text{carrier}}(f)|^2 = T_d^2 = N_c^2 T_c^2. \tag{3.33}$$

In summary, the OOK-modulated signal can be viewed as the linear combination of a narrowband signal with high spectral density and a wideband signal with low spectral density. For a random code, half of the code bits will be ones. Consequently, about half of the signal's energy will be contained in the narrowband main lobe of the unmodulated carrier component, while the other half is spread over a wide range of frequencies. This unmodulated carrier's power does not contribute to the accuracy of the arrival time estimate (see Equation (2.24)). This power could have been distributed at higher frequencies for a more accurate arrival time estimate, e.g. by using Phase-Shift Keying (PSK) modulation. However, the distinct narrowband peak simplifies carrier recovery, as will be shown in Section 4.4.2.

Figure 3.6 displays the estimated power spectral density of a length-2047 Gold code modulated using OOK at a chip rate of 1 MHz, normalised to unity power. Note that this power spectrum is identical to the power spectrum of the PSK-modulated signal in Figure 3.4, except that the power density is 3 dB weaker at all the frequency components except the carrier frequency,

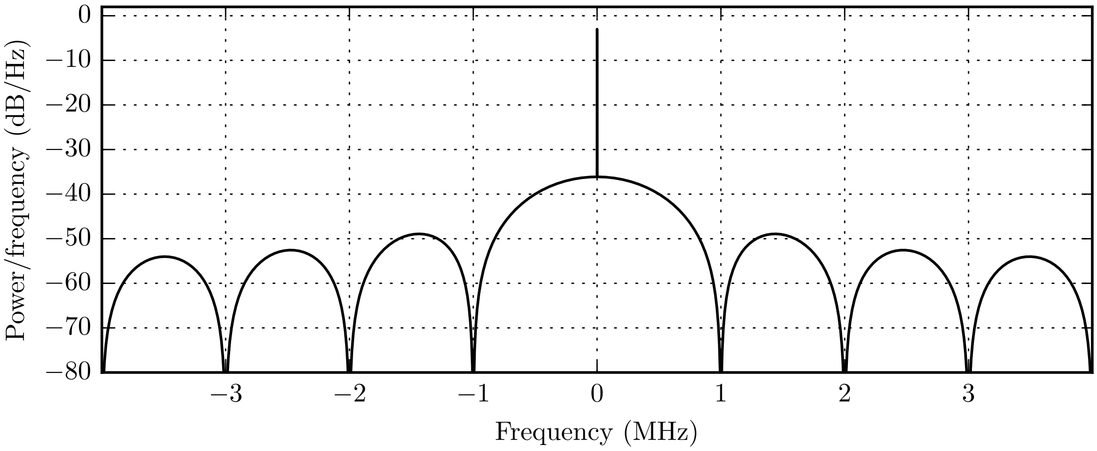


Figure 3.6: The periodogram of an OOK-modulated Gold code with 2047 chips and a chip rate of 1 MHz, normalised to unity power.

and that an unmodulated carrier with high spectral density is visible. The spectral density of the unmodulated carrier is 3 dB/Hz, i.e. half of the signal's total power. Note that, as can be predicted from Equations (3.23) and (3.33), the power spectral density of the unmodulated carrier is $10 \log_{10}(2047) = 33\,\mathrm{dB}$ greater than the highest spectral density of the wideband components.

3.3.3 Frequency shift keying (FSK)

Binary Frequency Shift Keying (BFSK) was not considered since it is spectrally less efficient than OOK and BPSK, i.e. it takes up more bandwidth for the same bit rate [29]. The use of FSK is very rarely seen in DSSS designs [36].

3.4 Signal detection

Up to this point we have assumed the presence of a positioning signal and focused on obtaining a good estimation of the signal's arrival time. There is, however, a problem that precedes the problem of arrival time estimation. How do we decide whether a valid positioning signal is present or not? The positioning signal may be short in duration while noise and interfering signals are continuously present in the signal being received. The power of the positioning signal at the receiver is unknown and can vary from a value that saturates the receiver's ADC when the transmitter is close to the receiver, to a value that is far below the noise floor for a distant transmitter. As discussed in Section 3.1.2, a spreading code will provide a distinct peak at the output of the correlator, but how is the "distinctness" of the peak quantified and when is the peak "distinct enough"? What is a good threshold for deciding whether a signal is present or not?

The ability to discern between the presence of a desired pattern and the presence of noise is a common problem in many fields of study. A means to quantify and understand this ability is through signal detection theory. Detection theory is an extensive topic and there is no shortage of books and papers in which it is being discussed. It is outside the scope of this dissertation to study it in depth and to apply it competently to a complex model. This section provides a quick glimpse at detection theory as applicable to a simplistic model of an arrival time detector. Refer to [37] and [38] for more information on this topic.

3.4.1 Introduction to signal detection theory

This subsection provides a brief overview of signal detection theory based on the description given by [37]. Suppose the presence of a positioning signal has to be decided based on one or more measurements. Signal detection can be considered as a statistical hypothesis testing problem where the detector has to choose one of two hypotheses based on the measurements [20, 37]:

- the null hypothesis, H_0, that no positioning signal is present and that the measurements are thus the result of noise and interference only, or
- the alternative hypothesis, H_1, that a positioning signal is present and that the measurements are the result of the combination of the positioning signal, noise, and interference.

Selecting H_1 when H_0 is true results in a *false positive*, also called a *false alarm*, or in statistics, a type I error. Selecting H_0 when H_1 is true results in a *false negative*, also called a *miss* or a type II error. The goal is to minimise the probability of a miss (P_m), which is equivalent to maximising the probability of a detection (P_d), while simultaneously minimising the probability of a false alarm (P_{fa}). An ideal detector would yield $P_d = 1$ and $P_{fa} = 0$. However, an inherent trade-off exists between P_d and P_{fa}. An increase in a detector's probability of detection implies an increase in its probability of false alarm as well. An important design parameter of the detector is the desired balance between P_d and P_{fa}. What constitutes a good balance depends on the application and what the implications are of acting on a false alarm. For example, a spurious detection could be an annoyance when it results in an incorrect position estimate, but it could be life threatening if a weapon is fired as the result of a detection.

The Neyman–Pearson lemma can be used to construct a decision rule that will make the optimal choice between the two hypotheses. The lemma provides a decision rule that will provide the best possible probability of detection under the condition that the probability of false alarm may not exceed a given maximum value. Suppose the measurements to be tested are given as a vector of N samples,

$$\boldsymbol{y} = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^T$$

and suppose a statistical signal model is available that describes the PDF of the measurements under both the null hypothesis and the alternative hypothesis. Let Λ be the ratio of the likelihood that the observations $\boldsymbol{y}$ will be observed if the signal is present to the likelihood that $\boldsymbol{y}$ will be observed if the signal is absent, thus:

$$\Lambda(\boldsymbol{y}) = \frac{p_{\boldsymbol{y}}(\boldsymbol{y}|H_1)}{p_{\boldsymbol{y}}(\boldsymbol{y}|H_0)}. \tag{3.34}$$

The Neyman–Pearson lemma states that the optimal decision rule that will maximise P_d subject to the condition that P_{fa} does not exceed a given tolerable value α, is the likelihood-ratio test

$$\Lambda(\boldsymbol{y}) \underset{H_0}{\overset{H_1}{\gtrless}} \eta. \tag{3.35}$$

This equation states that the alternative threshold H_1, representing a decision that the positioning signal is present, should be chosen if the likelihood ratio evaluated for the observed data $\boldsymbol{y}$ exceeds the threshold η, and that the null hypothesis H_0, representing a decision that the signal is not present, should be chosen if the ratio is below the threshold.

The threshold η should be such that P_{fa} does not exceed the maximum value α. It is evident from Equation (3.35) that a false detection will occur when $\Lambda(\boldsymbol{y})$ exceeds η while H_0 is in fact true. The threshold η that will ensure that $P_{fa} = \alpha$ is thus

$$P_{fa} = P(\Lambda > \eta|H_0) = \alpha \tag{3.36}$$

which can also be expressed as the integral of the PDF of Λ:

$$\int_{\eta}^{+\infty} p_{\Lambda}(\Lambda|H_0)\, d\Lambda = \alpha. \tag{3.37}$$

It is impractical to evaluate the PDFs and calculate the likelihood ratio for each outcome that needs to be tested. Instead, once statistical models are available that describe the PDFs of the random variable $\boldsymbol{y}$ as a parametric function of its outcome for both hypotheses, Equation (3.35) can be rearranged with all terms that explicitly include $\boldsymbol{y}$ isolated on the one side, combined into a function $\Upsilon(\boldsymbol{y})$, and all the constants can be moved to the other side. The function Υ is called a *sufficient statistic*. More generally, a sufficient statistic is a function of $\boldsymbol{y}$, $\Upsilon(\boldsymbol{y})$, that allows the likelihood ratio[1] to be expressed as a function of $\Upsilon(\boldsymbol{y})$ instead of $\boldsymbol{y}$. This allows a decision to be made based on $\Upsilon(\boldsymbol{y})$, which is a quantity that may be easier to compute from the signal. The decision rule in Equation (3.35) can be expressed in terms of the sufficient statistic:

$$\Upsilon(\boldsymbol{y}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \tag{3.38}$$

where γ is a threshold to the sufficient statistic such that the desired probability of false alarm is not exceeded, that is

$$\int_{\gamma}^{+\infty} p_{\Upsilon}(\Upsilon|H_0)\, d\Upsilon = \alpha. \tag{3.39}$$

[1] or a monotonically increasing function applied to the likelihood ratio, such as the log-likelihood ratio

A general description of signal detection theory was given above. An example of how detection theory can be applied to construct a detector based on a simple statistical signal model is described next. Again, the essence is provided here, but the reader is referred to [37] for more detail.

3.4.2 A detector for a signal in AWGN

A simple detector can be derived from the Neyman–Pearson lemma when the presence of a signal in zero-mean Gaussian noise needs to be determined. Suppose the received signal can be modelled as N complex samples that are the samples of a positioning signal with unknown amplitude and unknown phase corrupted by AWGN when the positioning signal is present, and samples consisting of solely noise when the signal is absent. Thus, when the positioning signal is present, then

$$\boldsymbol{y} = Ae^{j\theta}\boldsymbol{h} + \boldsymbol{w} \tag{3.40}$$

and when the signal is absent, then

$$\boldsymbol{y} = \boldsymbol{w} \tag{3.41}$$

where A and θ are the unknown amplitude scaling factor and the unknown phase offset of the positioning signal; $\boldsymbol{h}$ is a template of the expected positioning signal; and $\boldsymbol{w}$ is the noise modelled as a multivariate complex Gaussian distribution. Suppose $\boldsymbol{w}$ is a vector of N independent complex random variables with identical distribution, where the real and imaginary parts are both distributed as identical Gaussian distributions of zero mean and variance $\sigma_w^2/2$, thus:

$$w = w_I + jw_Q \tag{3.42}$$

$$w_I, w_Q \sim \mathcal{N}(\mathbf{0}_N, \sigma_w^2 \mathbf{1}_N/2) \tag{3.43}$$

where $\mathbf{0}_N$ is the N-dimensional null vector, and $\mathbf{1}_N$ the N-dimensional unit vector. Note that since $\boldsymbol{w}$ is zero-mean and Gaussian distributed, the expected value of the noise power is

$$E\{P_w\} = \|\boldsymbol{w}\|^2 = \sigma_w^2. \tag{3.44}$$

The output of a matched filter that cross-correlates $\boldsymbol{y} = \begin{bmatrix} y_1 & \dots & y_N \end{bmatrix}^T$ with the template $\boldsymbol{h} = \begin{bmatrix} h_1 & \dots & h_N \end{bmatrix}^T$ is

$$\text{match}(\boldsymbol{y}) := \boldsymbol{h}^H \boldsymbol{y} = \sum_{i=1}^{N} h_i^* y_i \tag{3.45}$$

where $\boldsymbol{h}^H$ denotes the conjugate transpose (Hermitian) of $\boldsymbol{h}$.

It can be shown that the magnitude of the output of the matched filter that correlates $\boldsymbol{y}$ with the template $\boldsymbol{h}$ is a sufficient statistic for an optimum detector based on the Neyman–Pearson lemma [37], thus

$$\Upsilon(\boldsymbol{y}) = \left|\boldsymbol{h}^H\boldsymbol{y}\right|. \tag{3.46}$$

Furthermore, based on the statistical model, the probability of false alarm is given by

$$P_{fa} = \int_{\gamma}^{+\infty} p_{\Upsilon}(\Upsilon|H_0)\, d\Upsilon = e^{\frac{-\gamma^2}{E_h\sigma_w^2}} \tag{3.47}$$

where $E_h := \boldsymbol{h}^H\boldsymbol{h}$ is the energy of the template signal. Equation (3.47) can be inverted to calculate the minimum threshold that would ensure that the probability of false alarm does not exceed the design value $P_{fa} = \alpha$:

$$\gamma = \sqrt{-E_h\sigma_w^2 \ln\alpha}. \tag{3.48}$$

Substitute Equations (3.46) and (3.48) into Equation (3.38) to obtain the decision rule in terms of α:

$$\left|\boldsymbol{h}^H\boldsymbol{y}\right| \underset{H_0}{\overset{H_1}{\gtrless}} \sqrt{-E_h\sigma_w^2 \ln\alpha}. \tag{3.49}$$

Note that the decision rule in Equation (3.49) is a function of the SNR, or more specifically, the signal-plus-noise-to-noise ratio. To see this more clearly, let χ be the ratio of the power of the matched filter output with the received signal as input to the power of the output when only noise is present at the input:

$$\chi := \frac{\left|\boldsymbol{h}^H\boldsymbol{y}\right|^2}{\left|\boldsymbol{h}^H\boldsymbol{w}\right|^2} = \frac{\left|\boldsymbol{h}^H\boldsymbol{y}\right|^2}{E_h\sigma_w^2}. \tag{3.50}$$

The decision rule in Equation (3.49) can then be rewritten in terms of χ:

$$\chi \underset{H_0}{\overset{H_1}{\gtrless}} -\ln\alpha. \tag{3.51}$$

In conclusion, the likelihood ratio test in Equation (3.35) seems difficult to compute, but it developed into the elegant equation in Equation (3.51). The equation shows that, for the simplistic model of a positioning signal with unknown amplitude and phase against a background of AWGN, the optimal detector for a given maximum false alarm rate α boils down to a comparison between the SNR of the received signal and a constant threshold, which is the negative of the natural logarithm of the desired false alarm rate.

3.5 Carrier detector

It was shown in Section 3.4.2 that the presence of a positioning signal can be decided based on the output of the correlator. If the positioning signals are short of duration and transmitted infrequently, then most of the incoming data will not contain any positioning signals. In this section we show that the unmodulated carrier that is visible for OOK modulation can be used to detect whether the carrier is present or not. Carrier detection relieves the receiver's computational burden since portions of the incoming signal in which the carrier is absent will most likely not contain a positioning signal and can be discarded prior to cross-correlation.

In this section, we first prove that the OOK-modulated positioning signal can be approximated as an unmodulated carrier for the sake of carrier detection. It is then shown that, if the carrier is present, the carrier frequency can be estimated from the DFT sample that exhibits the highest spectral density. Finally, a decision rule is presented for deciding whether the carrier is present.

3.5.1 Approximate positioning signal as a single tone

As shown in Section 3.3, the OOK-modulated positioning signal can be decomposed into two components: the wideband PSK-modulated code signal and a narrowband unmodulated carrier. From the Fourier transform of Equation (3.30) it follows that the energy spectrum density of the positioning signal is:

$$|X_{\text{ook}}(f)|^2 = |0.5\, X_{\text{psk}}(f) + 0.5\, X_{\text{carrier}}(f)|^2 \tag{3.52}$$

where $X_{\text{ook}}(f)$ is the Fourier transform of the OOK-modulated positioning signal, and $X_{\text{psk}}(f)$ and $X_{\text{carrier}}(f)$ the Fourier transforms of the wideband PSK code and the narrowband unmodulated carrier respectively. For brevity, we omit the arguments of the functions and assume that the max operator is evaluated for all f in the domain of X_{ook}. It follows that:

$$4\,|X_{\text{ook}}|^2 = |X_{\text{psk}}|^2 + 2\,\text{Re}(X^*_{\text{psk}} X_{\text{carrier}}) + |X_{\text{carrier}}|^2 \tag{3.53}$$

Furthermore, from Equation (3.21) and Equation (3.33) it follows that:

$$\begin{aligned} \max|X_{\text{psk}}| &= \sqrt{N_c} T_c \\ \max|X_{\text{carrier}}| &= N_c T_c \\ \therefore \max|X_{\text{carrier}}| &\gg \max|X_{\text{psk}}| \quad \text{if } N_c \gg 1, \end{aligned} \tag{3.54}$$

where T_c is the chip period and N_c the number of bits in the positioning code. Furthermore,

$$\begin{aligned} \max|X_{\text{carrier}}| &\gg \max\left|X^*_{\text{psk}}\right| \\ \max|X_{\text{carrier}}|\,|X_{\text{carrier}}| &\gg \max\left|X^*_{\text{psk}}\right|\,|X_{\text{carrier}}| \\ \max|X_{\text{carrier}}|^2 &\gg \max\left|X^*_{\text{psk}} X_{\text{carrier}}\right| \\ \max|X_{\text{carrier}}|^2 &\gg \max\left(\text{Re}(X^*_{\text{psk}} X_{\text{carrier}})\right) \end{aligned} \tag{3.55}$$

since

$$\mathrm{Re}(X_{\mathrm{psk}}^{*} X_{\mathrm{carrier}}) \leq \left| X_{\mathrm{psk}}^{*} X_{\mathrm{carrier}} \right| .$$

By using the results of Equations (3.54) and (3.55), and applying them to Equation (3.53), the following approximation is reached:

$$\max |X_{\mathrm{ook}}|^2 \approx \max |X_{\mathrm{carrier}}|^2 \tag{3.56}$$

if $N_c \gg 1$. Thus, the maximum spectral density within the OOK signal can be attributed to the maximum spectral density of the unmodulated carrier component. Furthermore, the contribution of X_{psk} to the spectrum is minuscule in comparison to the maximum spectral density. Consider the two terms in Equation (3.53) that contain X_{psk}. For all f it holds that:

$$|X_{\mathrm{psk}}(f)| \leq \max |X_{\mathrm{psk}}(f)| \ll \max |X_{\mathrm{carrier}}(f)| \tag{3.57}$$

and

$$\mathrm{Re}\left(X_{\mathrm{psk}}^{*}(f) X_{\mathrm{carrier}}(f) \right) \leq \left| X_{\mathrm{psk}}^{*}(f) X_{\mathrm{carrier}}(f) \right| \ll \max |X_{\mathrm{carrier}}(f)|^2 . \tag{3.58}$$

Without considering the details and without presenting a thorough mathematical analysis, the contribution of X_{psk} can be approximated as noise in comparison to the peak's magnitude based on Equations (3.56) to (3.58). For the sake of carrier detection, we can ignore X_{psk} and approximate the positioning signal as an unmodulated carrier and thus a single-tone signal.

3.5.2 ML carrier frequency estimation from the DFT

In [39] it is shown that the frequency of a single-tone signal with unknown parameters corrupted by AWGN can be estimated from complex-valued samples by calculating the Discrete-Time Fourier Transform (DTFT) and selecting the frequency component with the largest amplitude. More concretely, consider a received signal of the form

$$x[n] = A_0 e^{j(2\pi f_0 T_s n + \theta_0)} + w[n] \tag{3.59}$$

for $n = 0, \dots, N-1$ where the frequency f_0, the amplitude A_0, and the phase offset θ_0 of the signal are unknown. The variable T_s denotes the sampling period, and $w[n]$ is a zero-mean complex jointly-Gaussian random variable. Then, the MLE for the tone's frequency is [39]

$$\hat{f}_{\mathrm{ML}} = \arg\max_{f \in \Omega} |X(f)| \tag{3.60}$$

with $\Omega \subseteq [0, \infty)$ and where $X(f)$ is the DTFT:

$$X(f) = \sum_{n=0}^{N-1} x[n] e^{-j2\pi f T_s n}. \tag{3.61}$$

However, it is impractical to calculate $X(f)$. Instead, the coarse (approximate) estimate of $\hat{f}_{\mathrm{ML}}$ can be obtained from an N-point Discrete Fourier Transform (DFT) of the N time-domain samples, which is a finite-length sampled version of the DTFT:

$$X[k] = X\left(\frac{k}{NT_s}\right) \tag{3.62}$$

$$\hat{k}_{\mathrm{ML}} = \arg\max_{k} |X[k]| \tag{3.63}$$

$$\hat{f}_{\mathrm{ML}} \approx \frac{\hat{k}_{\mathrm{ML}}}{N} f_s \tag{3.64}$$

for $k = 0, 1, ..., N-1$ and where $f_s = 1/T_s$ denotes the sample rate. The Fast Fourier Transform (FFT) is commonly used to compute the DFT with a time complexity of $O(N \log N)$. Because of the periodicity of the DFT, $X[k-N] = X[k]$. Thus, if N is even and $\hat{k}_{\mathrm{ML}} > N/2$, or if N is odd and $\hat{k}_{\mathrm{ML}} \geq (N+1)/2$, then

$$\hat{f}_{\mathrm{ML}} \approx \left(\frac{\hat{k}_{\mathrm{ML}}}{N} - 1\right) f_s. \tag{3.65}$$

3.5.3 Single-tone signal detection

In the discussion above we assumed that the positioning signal is present and showed that, if the carrier is present, the most likely estimate of the carrier's frequency will be the frequency component of the power spectrum with maximum spectral density. However, the signal is not always present. Consequently, it is necessary to have a rule for deciding upon the presence of the carrier. Since the signal is approximated as a single-tone, most of the signal's power will be distributed at or within a narrow band of the carrier frequency when it is present. It can be expected that a simple detector will determine the presence of a single-tone signal by comparing the spectral density at the ML tone frequency to a threshold. This is indeed the case. The continuous spectral density cannot be computed efficiently, but as shown in [40], optimum detection performance can be achieved from the standard periodogram, which is the spectral density estimation given by the squared magnitude of the DFT.

Consider a simple signal model where the received signal is given by Equation (3.59) when the signal is present and by $x[n] = w[n]$ when the signal is absent. The periodogram of the signal is

$$S_x[k] := \frac{1}{N} |X[k]|^2, \quad 0 \leq k \leq N-1. \tag{3.66}$$

A decision rule based on the peak value of the periodogram is [40]:

$$S_x[\hat{k}_{\mathrm{ML}}] \underset{H_0}{\overset{H_1}{\gtrless}} \gamma. \tag{3.67}$$

The null hypothesis, H_0, representing a decision that the carrier is not present, is decided when the maximum magnitude is below the threshold γ, and the alternative hypothesis, representing a decision that the carrier is present, when the magnitude is above the threshold γ.

The threshold value can be decided based on a desired maximum probability of false alarm α. Based on the statistical signal model, the probability of false alarm, P_{fa}, is given by (adapted from [40]):

$$P_{fa} = 1 - \left(1 - e^{-\frac{\gamma}{\sigma_w^2}}\right)^N \tag{3.68}$$

where σ_w^2 represents the power of $w[n]$ (the noise power). Rearranging the equation and setting $P_{fa} = \alpha$ yields:

$$\gamma = \sigma_w^2 \beta \tag{3.69}$$

$$\beta := -\ln\left(1 - \sqrt[N]{1-\alpha}\right). \tag{3.70}$$

If α and N are constants, then β is a constant too. Substituting Equation (3.69) into Equation (3.67) yields

$$\frac{S_x[\hat{k}_{\mathrm{ML}}]}{\sigma_w^2} \underset{H_0}{\overset{H_1}{\gtrless}} \beta, \tag{3.71}$$

which shows that, similar to the correlation detector in Equation (3.51), the carrier detector essentially checks the SNR of the received signal against a minimum SNR β for a given maximum probability of false alarm.

3.6 Chapter summary

DSSS is commonly used in data communication to spread a signal over a bandwidth that is significantly greater than the minimum required for communicating the data. To achieve the higher bandwidth, each data symbol is effectively sliced into a predefined sequence of bits of much shorter duration. The sequence of bits, called a code, is also known at the receiver, and is synchronised to the code embedded in the received signal. The synchronisation can be performed with a discrete correlator that slides the template signal over the received signal one sample at a time. A code with good autocorrelation properties will not show strong correlation with time-shifted replicas of itself, causing a distinct sharp correlation peak at the output of the correlator only when the received signal aligns with the template signal. Since the position of the correlation peak marks a specific moment in time, it can be used to estimate the arrival time of the positioning signal.

The resolution of the arrival time estimate is limited by the step size of the correlator since the template may not line up perfectly with the code embedded in the incoming signal. The resolution can however be improved to be smaller than the step size by interpolating between the samples of the correlation peak.

PSK is generally used to modulate the binary code onto a carrier for transmission. With PSK modulation, the energy of a random code is spread over a wide bandwidth, without a distinct peak at any particular frequency. The carrier is suppressed, which is advantageous in terms of arrival time estimation accuracy, but makes carrier recovery challenging and computationally expensive. Carrier recovery is imperative if the code is long or if the frequency stability of the local oscillators is unsatisfactory, since a frequency mismatch may cause different sections of the code to become out of phase and may impede the correlation process. If the code is modulated using OOK, about half of the signal's energy is contained within a narrow band of spectral components at the carrier frequency, which simplifies carrier recovery.

For intermittent positioning signals it is necessary to have a decision rule for determining the presence of the signal. If a simplistic model is assumed, the decision rule can be as simple as comparing the SNR of the correlation peak to a set threshold. Similarly, for an OOK-modulated code, the presence of a carrier signal can be decided based on the SNR of the frequency component in the periodogram with the highest spectral density.

Chapter 4

Design

Errors are not in the art, but in the artificers.

— Isaac Newton

With the stage set by the first three chapters, we can now proceed to integrate the knowledge into a design. In this chapter, we present a design for multilateral TDOA radio positioning, using DSSS positioning signals, that makes provision for the shortcomings of inexpensive hardware. The structure, techniques and algorithms are described without assuming a specific hardware implementation.

The chapter is structured as follows. In Section 4.1, the premises of the design are stated in terms of the imperfections that should be accounted for and the extent to which errors due to the imperfections will be tolerated. The top-level system architecture and an overview of the system are then given in Section 4.2, after which the design of each functional unit is expanded upon in the subsequent sections: the transmitter in Section 4.3, the receiver in Section 4.4 and the positioning server in Section 4.5. Lastly, the chapter is concluded with a brief summary of the design in Section 4.6.

4.1 Design premises

The primary goal is that the design should be implementable using inexpensive COTS receiver hardware. As such, it should, by design, be resilient to the inaccuracies and shortcomings introduced by cheap hardware. The imperfections for which the design makes provision are delineated below.

Imperfection 1 (Unknown sample time):

It is assumed that the sampling device that is used at the receiver does not provide the time at which a sample is taken and that an unknown and time-varying delay occurs between the time that a sample is taken and the time it is received by the signal processor. Consequently, an accurate estimate of the time at which a signal arrives at the receiver is not available.

Imperfection 2 (No accurate common time base):

It is assumed that the receiver devices do not possess the ability to keep time accurately relative to a common time base. The precise time at which an event has occurred at one receiver can thus not be related to the clock of another receiver. It is, however, assumed that the real-time clocks of the receivers are coarsely synchronised to allow events at different receivers to be matched together.

Imperfection 3 (Unsynchronised sample rate and non-coherent phase):

It is assumed that the receivers' clocks are free-running and independent. GPS Disciplined Oscillators (GPSDOs) are used by similar TDOA tracking systems [1, 16] to maintain precise synchronization between the receivers and to provide a frequency stability of 1 ppb or better. However, a GPSDO adds cost and complexity to the design of the receiver, and inexpensive off-the-shelf hardware does not provide such capabilities without modifications to the circuitry.

The frequency stability that can be achieved with low-cost free-running clocks is limited. Even with a Temperature Compensated Crystal Oscillator (TCXO), a frequency tolerance of 1 ppm or worse can be expected. As a preliminary assumption, we assume that the frequency tolerance of the receiver will be up to 10 ppm. It is furthermore assumed that the clock frequency will change ("drift"), but that the change will be gradual. There will thus be unknown and time-varying differences between the sample rates of the receivers.

Imperfection 4 (Limited computational power):

It is assumed that the computational power of the receiver devices is restricted by the requirement that the devices should be cheap and by the assumption that receiver devices will be deployed without access to mains electricity. It is furthermore assumed that the sample rate is constrained by the capabilities of the inexpensive RF receiver and by the limited computational power that is available for signal processing.

Imperfection 5 (Limited stability of transmitter oscillator):

It is assumed that the transmitter's carrier frequency may change over time due to the limited stability of the transmitter's oscillator caused by factors such as ageing and changes in temperature. It is furthermore assumed that the oscillator exhibits good short-term stability to ensure that the drift of the transmitter's carrier frequency is negligible during a transmission.

An implementation for which some or all of the imperfections mentioned above do not apply would, of course, not perform any worse, but may even perform better and can still be used with the design outlined in this chapter.

Finally, it is assumed that the system will be used in an LOS environment and that the effects of multipath propagation can be ignored.

4.2 System architecture

Before each part of the system is described in detail, a high-level overview of the system is first presented. As depicted in Figure 4.1 and expanded in Figure 4.2, the system consists of three types of entities: transmitters, receivers, and a central positioning server. The transmitters are mobile units with unknown positions that periodically transmit a short positioning signal, e.g. small tags attached to wild animals. Receivers are base stations and have well-known positions that are fixed, probably at high sites for maximum LOS coverage. Every transmitter is not necessarily in reach of all of the receivers, but each transmitter should be in reach of at least three, preferably four, receivers.

A fourth role also exists as a special case of a transmitter: a beacon. Except for a few small changes such as a larger battery, the hardware of a beacon is identical to that of a normal transmitter. The difference is that a beacon has a known and fixed position. A beacon is identical to a transmitter from the perspective of the receiver: both a mobile unit's and a beacon's TOA should be estimated by the receiver. They are, however, treated differently by

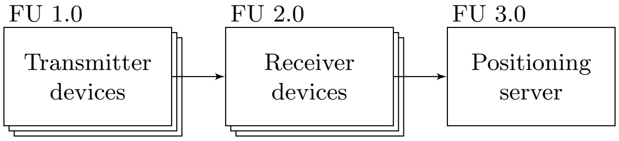


Figure 4.1: Top-level system architecture showing the different types of entities that are involved.

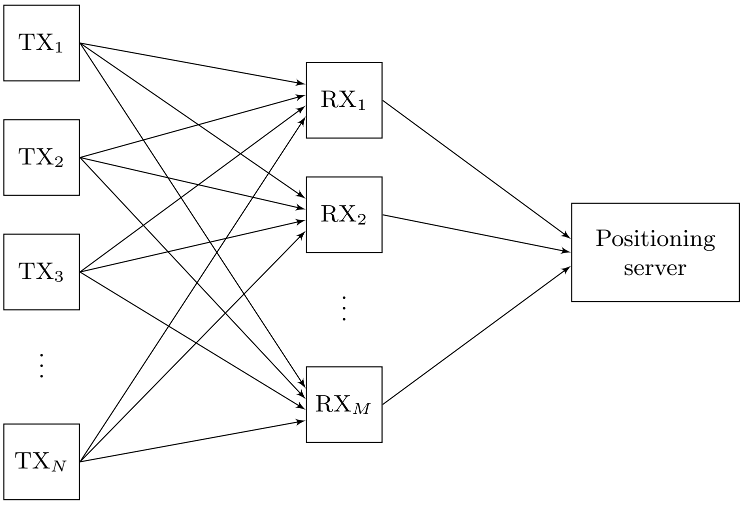


Figure 4.2: Top-level system architecture expanded to show the interaction between multiple transmitters and receivers.

the positioning server. The beacon or beacons assist in estimating the difference in arrival time without accurate estimates of the sample time, with unsynchronised real-time clocks, and with differences in the receivers' sample rates (Imperfections 1 to 3).

The receivers continuously search for the presence of positioning codes within the incoming signal. When a receiver detects a positioning signal, it records information for matching detections from different receivers, for calculating the TDOA and for identifying the transmitter, as well as supplementary information for estimating the quality of the detection. The receivers communicate the detection information to the positioning server. The positioning server is a central server that integrates the detection information into a data store, matches detection events from different receivers that can be attributed to the same positioning signal, synchronises to the beacon transmissions and estimates the positions of the transmitters. We will refer to the processing that is performed on the receiver as *signal processing* and the processing that is performed on the server as *detection processing*.

The role of each of the entities is expanded upon in the subsequent sections of this chapter.

4.3 Transmitter

Even though the transmitter is a rather simple device, and even though we are mostly concerned with the design of the receiver and the positioning server in this dissertation, it is still necessary to have a glimpse at the design of the transmitter.

The transmitter can be dissected into six essential functional units, as manifested in the functional system architecture block diagram in Figure 4.3. Each of the blocks is described in more detail below.

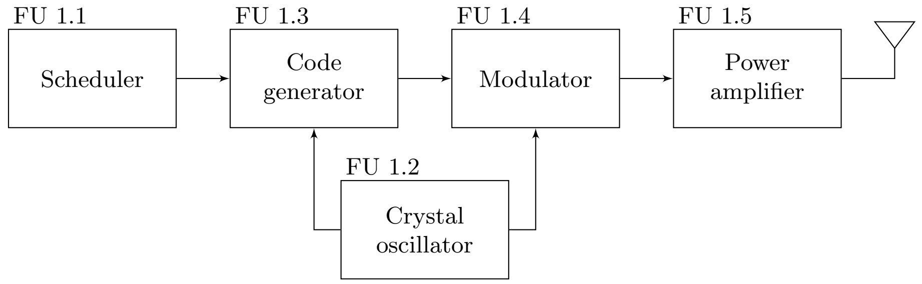


Figure 4.3: Block diagram showcasing the functional units of the transmitter.

Scheduler The scheduler determines when and how often the positioning code should be transmitted. To save energy, the device can enter a low-power sleep mode between transmissions. The scheduler can be implemented using a microcontroller.

Code generator The code generator produces a binary sequence with good auto-correlation properties. The code can be stored in memory, or in the case of Gold codes, generated on the fly using a linear-feedback shift register. A microcontroller can be charged with the task of generating the code. Even though any of the codes mentioned in Section 3.1.2 can be used, we chose a family of Gold codes since it is well-known, already used in similar applications such as [1] and GPS [11], and it provides a set of codes with small cross-correlations within the set in case different codes should be used for identifying different transmitters.

Modulator Different schemes for modulating the DSSS positioning code were examined in Section 3.3. We chose to use OOK. In comparison with PSK, a positioning signal that is modulated with OOK sacrifices about half of the wideband spectral components' energy to a narrowband unmodulated carrier. However, OOK modulation presents a few advantages:

- *It retains a narrowband signal that can be used for carrier recovery.* As illustrated in Section 3.3.1, to retain at least half of the correlation peak power of a positioning signal with a duration of 2 ms, the frequency offset should be no worse than 222 Hz. If a 433 MHz carrier is used, that amounts to a frequency stability of about 0.5 ppm, which is an unrealistic goal for a small transmitter device that will be used outdoors without any temperature compensation and for which size, cost and weight are significant concerns.

  When a long random code is modulated using BPSK, the carrier power is spread over a wide band of frequencies, which makes carrier recovery problematic — especially if the spectral density of the wideband signal is below the noise floor. An OOK-modulated random code, on the other hand, retains about half of the signal's power within a narrow band of frequencies, as demonstrated in Section 3.3.2.
- *It provides a means to sieve the incoming signal* at the receiver and consequently a means to only correlate sections that may contain positioning signals. This helps to reduce the computational requirements of the receiver and to reduce false positives. This is expanded upon in Section 4.4.2.
- *It provides a fingerprint for transmitter identification.* Since the carrier can be recovered, each transmitter can transmit at a slightly different carrier frequency. The frequency of the carrier can then be estimated and used to identify the transmitter.
- *It is easy to implement.* The implementation can be as simple as generating an unmodulated carrier signal and toggling the supply voltage of the power amplifier.

Crystal oscillator A crystal oscillator is required to provide sufficient short-term stability of the carrier frequency while the code is being transmitted. A crystal oscillator, potentially the same oscillator as used for generating the carrier, is also required by the code generator to ensure that the chip rate remains stable.

Power amplifier (PA) The last stage of the transmitter's circuit amplifies the signal and drives the transmitting antenna. Filters are required to attenuate harmonics, limit the signal's bandwidth, and ensure that the applicable regulatory specifications are met.

The most important characteristic of the transmitter device is low energy consumption. The energy consumption has an impact on the size, weight and endurance of the tag. Furthermore, it is advantageous in reducing the size of the transmitter device if as much of the functionality as possible is incorporated into a single Integrated Circuit (IC).

4.4 Receiver

The purpose of the receiver device is to detect the presence of a positioning signal, to estimate the arrival time of the signal, and to pass the detection information on to the positioning server. The receiver can be divided into three functional units, as illustrated in Figure 4.4: an RF receiver, signal processor and communication device. There are myriads of different ways in which each of these blocks can be implemented. For example, the RF receiver could be a COTS Software-Defined Radio (SDR) device consisting of an antenna, amplifier, mixer, and ADC. The signal processor could be a computer, a DSP chip, or an FPGA. The communication device could be a GPRS module for uploading the data to a server via the Internet, or it could be an RF transceiver for relaying the data to another base station.

The essence of the receiver device lies in the signal processing algorithms. It is assumed that the incoming signal is sampled as complex values and that signal processing is performed digitally on a discrete-time discrete-amplitude signal. The goal is to implement the signal processing algorithms in software, but to keep the door open for alternative implementations. The algorithms presented in this chapter are, insofar as possible, described abstractly without assuming a software implementation.

The signal processor is divided into the following four top-level modules: a data slicer, carrier recoverer, Sample-of-Arrival (SOA) estimator and a detection reporter, as depicted in Figure 4.5. Each of the modules is enlarged upon next.

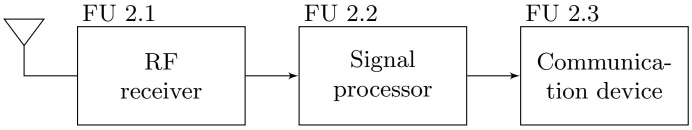


Figure 4.4: Block diagram showcasing the top-level functional units of the receiver.

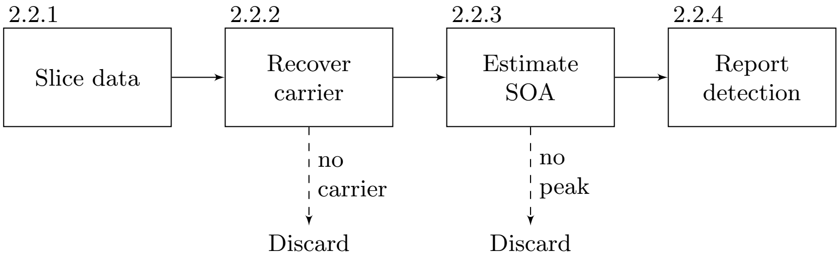


Figure 4.5: Flow diagram depicting the top-level steps involved in processing the signal.

4.4.1 Slicer

The RF receiver produces a continuous stream of time-domain samples which have to be processed in real-time, i.e. without falling behind even if the input is continuing for an unlimited period of time. Some of the signal processing operations (e.g. convolution) are slow when applied to a continuous stream of data while faster algorithms are available that can only be applied to discrete blocks of data. Furthermore, if a software processor is used, it is easier and, when using Single Instruction, Multiple Data (SIMD) instructions, faster to work with discrete blocks of data. With discrete blocks of data, it is also possible to process multiple blocks in parallel if necessary.

Some operations operate on present as well as past samples. To be able to simulate a sliding window, and thus to ensure that there are no discontinuities in the output of a continuous operation that uses past samples when it is applied to subsequent blocks of data individually, it is necessary that the blocks overlap. For example, to be able to continuously apply a Finite Impulse Response (FIR) filter of order N, each block should repeat the last N samples of the previous block.

The slicer divides the data into fixed-length blocks. Each block overlaps with the previous block by an amount of samples we call the *history length*. Concretely, if the blocks are of length L_B, and the history length is L_H, then block $b \geq 0$ consists of the samples

$$x_b[n] = \{x[b(L_B - L_H)], \ldots, x[b(L_B - L_H) + L_B - 1]\}, 0 \leq n \leq L_B - 1 \tag{4.1}$$

where $x[n]$ is defined for $0 \leq n < \infty$ and represents all the complex-valued samples since the receiver has started sampling.

4.4.2 Carrier frequency recovery

The key benefit of using OOK modulation is that a distinct narrowband carrier spectral line exists which can be utilised for carrier recovery.

In Section 3.3 it was shown that the spectrum of an OOK modulated spread spectrum code can be decomposed into two parts, namely a wideband spread spectrum signal and a narrowband unmodulated carrier. An example of the power spectral density of an OOK-modulated Gold code was displayed in Figure 3.6, from which it is clear that a distinct peak is visible at the carrier frequency. It was shown that half of the signal's total energy is contained in the narrowband peak at the carrier frequency. It is this distinct narrowband carrier that enables carrier synchronisation.

Traditional closed-loop carrier recovery methods such as a Costas loop or a Phase-Locked Loop (PLL) are challenging to apply as the positioning signal is short in duration. Instead, we employ an *ad-hoc* method for carrier frequency recovery based on the maximum frequency

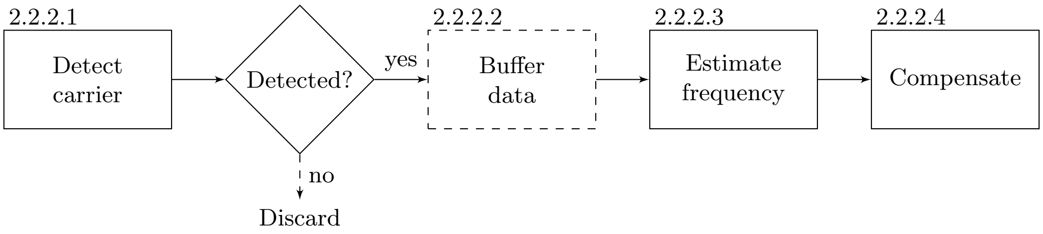


Figure 4.6: Flow diagram depicting the steps involved in recovering the carrier frequency.

component of the power spectrum of the block of data.

The carrier recovery algorithm can be decomposed into three steps, as depicted in Figure 4.6:

1. *detect carrier*: detect the presence of a carrier and discard the block of data if the carrier is absent;
2. *estimate carrier frequency*: obtain a fine-grained estimate of the carrier frequency offset;
3. *compensate*: compensate for the frequency offset by shifting the carrier to DC.

An optional fourth step, *buffering the data*, can be added between the carrier detector and the interpolator for asynchronous processing.

The phase of the carrier is not compensated for, only the frequency. The importance of frequency compensation for successful correlation was illustrated in Figure 3.5. The phase offset will, however, be carried over to the output of the correlator since complex-valued samples are being used. Only the magnitude of the correlation output is currently being used, but in the future the phase information can be recovered from the correlation output if necessary.

Carrier detector

A strategy for detecting the presence of a carrier from the DFT of the positioning signal when an OOK-modulated code is being used was presented in Section 3.5. In summary, the positioning signal is approximated as a single-tone signal against a background of AWGN, allowing the presence of the carrier to be decided with a comparison of the SNR to a set threshold. The decision rule from Equation (3.71) that is used to decide whether a block contains a carrier signal is:

$$\frac{\left|X_b[\hat{k}_{\mathrm{ML}}]\right|^2}{\sigma_w^2} \underset{H_0}{\overset{H_1}{\gtrless}} \beta. \tag{4.2}$$

where σ_w^2 is the noise power, $X_b[k]$ the DFT of $x_b[n]$, $0 \leq k \leq L_B - 1$, and $\hat{k}_{\mathrm{ML}}$ the index of the DFT sample with maximum spectral density:

$$\hat{k}_{\mathrm{ML}} = \underset{k \in K}{\arg\max} \left|X_b[k]\right|^2 . \tag{4.3}$$

To reduce the probability of false alarm, the domain of k over which the maximum magnitude is being calculated is limited to the frequency range over which the carrier frequency is to be expected, thus $K = [k_{\text{min}}, k_{\text{max}}]$, $0 \leq k_{\text{min}} \leq k_{\text{max}} \leq L_B - 1$.

Since the noise power of the received signal is not known, we estimate it from the block of data. The noise power is approximated as the mean power of the received signal when the spectral power at the carrier frequency is excluded. Let P_{mean} be the mean power,

$$P_{\text{mean}} := \frac{1}{L_B} \sum_{n=0}^{L_B-1} |x_b[n]|^2 = \frac{1}{L_B^2} \sum_{n=0}^{L_B-1} |X_b[n]|^2 \tag{4.4}$$

and let $\hat{P}_{\text{carrier}}$ be the estimated power spectral density at the carrier frequency,

$$\hat{P}_{\text{carrier}} := S_x[\hat{k}_{\text{ML}}] = \frac{1}{L_B} \left| X_b[\hat{k}_{\text{ML}}] \right|^2. \tag{4.5}$$

The noise power is then estimated as

$$\hat{P}_{\text{noise}} := \frac{1}{L_B - 1} \left(L_B P_{\text{mean}} - 2\hat{P}_{\text{carrier}} \right) \approx \sigma_w^2. \tag{4.6}$$

The carrier power is subtracted twice since it is expected that the wideband frequency components in the positioning signal's spectrum will contain approximately the same amount of energy as the narrowband unmodulated carrier.

Moreover, the estimated carrier SNR is

$$\begin{aligned} \hat{\chi}_c &:= \frac{\hat{P}_{\text{carrier}}}{\hat{P}_{\text{noise}}} \\ &= \frac{\left| X_b[\hat{k}_{\text{ML}}] \right|^2}{\hat{X}_{\text{noise}}^2} \end{aligned} \tag{4.7}$$

with

$$\hat{X}_{\text{noise}}^2 := L_B \hat{P}_{\text{noise}} = \frac{1}{L_B - 1} \left[\left(\sum_{k=0}^{L_B-1} |X_b[k]|^2 \right) - 2 \left| X_b[\hat{k}_{\text{ML}}] \right|^2 \right]. \tag{4.8}$$

Finally, substituting Equation (4.7) into Equation (4.2) yields the decision rule:

$$\frac{\left| X_b[\hat{k}_{\text{ML}}] \right|^2}{\hat{X}_{\text{noise}}^2} \underset{H_0}{\overset{H_1}{\gtrless}} \beta. \tag{4.9}$$

Equation (4.9) shows that the null hypothesis, H_0, is decided when the estimated SNR is below the minimum SNR β. This represents a decision that the carrier is not present. The alternative hypothesis, H_1, which represents a decision that the carrier is present, is decided when the minimum SNR value is exceeded. The minimum SNR that would ensure that the (theoretical) probability of false alarm does not exceed α can be calculated from Equation (3.70):

$$\beta = -\ln\left(1 - \sqrt[q]{1-\alpha}\right) \tag{4.10}$$

where

$$q := k_{\max} - k_{\min} + 1. \tag{4.11}$$

Since the carrier detector is only a prefilter, a high probability of detection is more important than a low probability of false alarm. The threshold can be calculated based on a relatively generous probability of false alarm in order to increase the probability of detection. The threshold should, on the other hand, not be too low, since the assumption of AWGN may not be valid in the presence of interference.

Buffer

As the first step in the signal processor, the carrier detector should be fast; it has to continuously process the incoming signal, which is sampled at a high sample rate, in real-time without lagging behind. If the positioning signals are transmitted infrequently (e.g. a few tags transmitting once every second), most of the incoming blocks of data will not contain any positioning signals since the positioning signal is short-lived. The majority of blocks can thus be eliminated by the carrier detector. To ensure that subsequent signal processing steps do not block and slow down the carrier detector, blocks of data for which a carrier is being detected can be stored in a queue. The carrier detector can continue, while the rest of the signal processing can be performed asynchronously on the reduced number of blocks. This reduces the computational requirements of the receiver if positioning signal transmissions are infrequent since subsequent signal processing after carrier detection is subject to a relaxed real-time constraint.

Carrier frequency offset estimator

The frequency of the carrier is already known after carrier detection; the index of the maximum-magnitude component of the DFT, $\hat{k}_{\mathrm{ML}}$, represents the carrier frequency (Equation (4.3)). However, the DFT estimates the spectral density only at a limited number of discrete frequencies, which introduces a quantisation error in the frequency estimation if the carrier frequency is not equal to one of the discrete frequency components of the DFT. The index $\hat{k}_{\mathrm{ML}}$ obtained from the DFT only represents a coarse approximation of the carrier frequency. As shown in Section 3.3.1, a frequency error results in a reduction in the power of the correlation peak. It is thus advantageous to obtain a more accurate estimate of the carrier frequency. One approach for improving the accuracy is to interpolate between the peak of the DFT and the samples on either side of the peak.

Interpolation is performed by fitting a model of the expected shape of the peak to multiple samples near the peak of the DFT. From Equation (3.32) it follows that the magnitude of the Fourier transform of the unmodulated carrier is:

$$|X_{\text{carrier}}(f)| = T_d \left|\operatorname{sinc}\left(T_d f\right)\right|. \tag{4.12}$$

Consider the DFT pair [41]

$$x_n = \frac{1}{W} \operatorname{rect}\left(\frac{n}{W}\right) \longleftrightarrow X_k = \underset{W}{\operatorname{asinc}}\left(\frac{k}{N}\right) \tag{4.13}$$

where N is the length of the DFT. The function asinc is the aliased sinc function, also called the Dirichlet function, which is the DFT-equivalent of the sinc function and which is defined as

$$\underset{W}{\operatorname{asinc}}(x) = \begin{cases} \frac{\sin(\pi W x)}{W \sin(\pi x)} & x \neq 0 \\ 1 & x = 0. \end{cases} \tag{4.14}$$

Using the transform pair in Equation (4.13), it can be shown that the magnitude of the DFT of the unmodulated carrier, i.e. the DFT equivalent of Equation (4.12), is:

$$|X_{\text{carrier}}[k]| = T_d \left|\underset{L_T}{\operatorname{asinc}}\left(\frac{k+\delta}{L_B}\right)\right|. \tag{4.15}$$

Note that the shape of the carrier peak is dictated by the asinc function.

Let

$$D(k) := \underset{L_T}{\operatorname{asinc}}\left(\frac{k}{L_B}\right). \tag{4.16}$$

The value of δ in Equation (4.15) can be estimated by fitting the main lobe of $D(k)$ to $X_c[k]$. Let l be the number of samples at either side of $\hat{k}_{\text{ML}}$ that should be used for fitting $D(k)$. Let $\boldsymbol{z} := \begin{bmatrix} A & \delta \end{bmatrix}^T$ be the unknown parameters, where A is the amplitude of the subsample peak and δ is the offset between the subsample peak and the coarse peak at index $\hat{k}_{\text{ML}}$. The model function is

$$f(x, \boldsymbol{z}) = |A \cdot D(x - \delta)|.$$

The model function can be fit to $X_c[k]$ using NLLS:

$$\hat{\boldsymbol{z}}_{\text{LS}} = \begin{bmatrix} \hat{A}_{\text{LS}} \\ \hat{\delta}_{\text{LS}} \end{bmatrix} = \arg\min_{\boldsymbol{z}} \|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{z})\|^2 \tag{4.17}$$

with

$$\boldsymbol{x} := \begin{bmatrix} -l, & \ldots, & l \end{bmatrix}^T, \tag{4.18}$$

$$\boldsymbol{y} := \begin{bmatrix} \left|X[\hat{k}_{\text{ML}} - l]\right|, & \ldots, & \left|X[\hat{k}_{\text{ML}} + l]\right| \end{bmatrix}^T, \tag{4.19}$$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{z}) := \begin{bmatrix} f(-l, \boldsymbol{z}) & \ldots & f(l, \boldsymbol{z}) \end{bmatrix}^T, \tag{4.20}$$

and the domain of $\boldsymbol{z}$ such that $0 < A \leq A_{\text{max}}$ and $-0.5 \leq \sigma \leq 0.5$. The estimated fractional index of the peak is then $\hat{k}_{\text{LS}} = \hat{k}_{\text{ML}} + \hat{\delta}_{\text{LS}}$. Equation (4.17) is non-linear and has to be solved iteratively. Initial values for the unknown parameters are:

$$\boldsymbol{z}_0 = \begin{bmatrix} \left|X[\hat{k}_{\text{ML}}]\right| \\ 0 \end{bmatrix}.$$

Even though a good NLLS estimation algorithm should converge quickly, a faster and simpler alternative is to approximate the shape of the peak as a parabola and to estimate the offset using parabolic interpolation. The estimate of the offset using Equation (3.4) is

$$\hat{\delta}_Q = \frac{|X_1| - |X_{-1}|}{4\,|X_0| - 2\,|X_{-1}| - 2\,|X_1|} \tag{4.21}$$

where $X_{-1} = X[\hat{k}_{\text{ML}} - 1]$, $X_0 = X[\hat{k}_{\text{ML}}]$ and $X_1 = X[\hat{k}_{\text{ML}} + 1]$. The estimated fractional index of the peak is then $\hat{k}_Q = \hat{k}_{\text{ML}} + \hat{\delta}_Q$. Refer to [42] for an analysis of the performance of parabolic interpolation when used for single-tone frequency estimation from a DFT. In [42] it is shown that parabolic interpolation improves the estimation accuracy significantly. For a length-16384 DFT, the estimator tracks the CRLB up to a SNR of about 60 dB.

Carrier frequency offset compensator

Once the carrier has been detected and the carrier frequency offset estimated, the signal can be synchronised to the carrier by applying a frequency shift such that the carrier frequency is shifted to 0 Hz. If the carrier frequency is such that the number of DFT indices by which the signal should be shifted is an integer, the shift can easily be performed in linear time in the frequency domain using a circular shift:

$$X'[k] \equiv X[(k + \hat{k}_{\text{ML}}) \bmod N] \qquad 0 \leq k \leq N - 1. \tag{4.22}$$

Compensating for the carrier frequency offset is, however, more complicated when the frequency shift is not a multiple of the DFT sample interval. This is the case when the fine-grained carrier frequency is estimated through interpolation. A solution is to shift the frequency in the time domain by means of one of the basic properties of the Fourier transform, the frequency-shift property [43]:

$$x'[n] = x[n]e^{j2\pi nk/N} \qquad 0 \leq n \leq N - 1 \tag{4.23}$$

where k is a real value of DFT samples by which the signal's frequency spectrum should be shifted, for example, $k = \hat{k}_{\text{LS}}$ or $k = \hat{k}_Q$.

It is, however, advantageous to perform the frequency shift in the frequency domain. The DFT is already being computed for carrier recovery, and the next module, the SOA estimator, also requires the DFT of the signal. Performing the frequency shift in the time domain would require an additional DFT calculation. Each additional transformation between the signal's time domain representation and its frequency domain representation increases the computational requirements of the signal processor.

A new method was devised to avoid the additional DFT calculation while compensating for a frequency offset at a finer granularity than the DFT sample interval. The signal processing step after carrier recovery involves cross-correlating the received signal with a template signal. The cross-correlation equation can be rewritten to apply the frequency shift to the template instead of the received signal. It follows from the equation of the cross-correlator, Equation (3.3), and the frequency-shift property, Equation (4.23), that:

$$\begin{aligned} y[n] &= \sum_{l=0}^{N-1} h^*[l] \cdot x'[n+l] \\ &= \sum_{l=0}^{N-1} h^*[l] \cdot x[n+l] e^{j2\pi(n+l)k/N} \\ &= e^{j2\pi nk/N} \left(\sum_{l=0}^{N-1} \left(h[l] e^{-j2\pi lk/N} \right)^* \cdot x[n+l] \right) \end{aligned} \tag{4.24}$$

where $y[n]$ is the correlator output and $h[n]$ represents the template signal zero-padded to a length of N. Only the magnitude of the correlation output is used during signal processing:

$$|y[n]| = \left| \sum_{l=0}^{N-1} \left(h[l] e^{-j2\pi lk/N} \right)^* \cdot x[n+l] \right|. \tag{4.25}$$

The same correlation output can thus be obtained by shifting the template signal instead of the received signal itself. To compensate for a frequency offset in the received signal, the frequency of the template can be shifted by the negative of the frequency offset.

Frequency-shifted templates are precomputed for a set of evenly spaced frequency shifts between -0.5 and 0.5 DFT samples. The carrier frequency is compensated for to the nearest DFT sample by applying a circular shift to the DFT of the received signal (Equation (4.22)). The residual frequency offset will then be between -0.5 and 0.5 DFT samples, which is compensated for by correlating the signal with the precomputed template that has a frequency shift closest to the negative of the residual frequency offset.

4.4.3 SOA estimator

A desirable capability in a TDOA system is the ability to estimate the TOA at each receiver for every positioning signal that is being transmitted. With this information, calculating the TDOA values is as simple as subtracting the TOA values from each other. However, due to Imperfection 1, the exact time at which the samples were taken is unknown, and due to Imper-

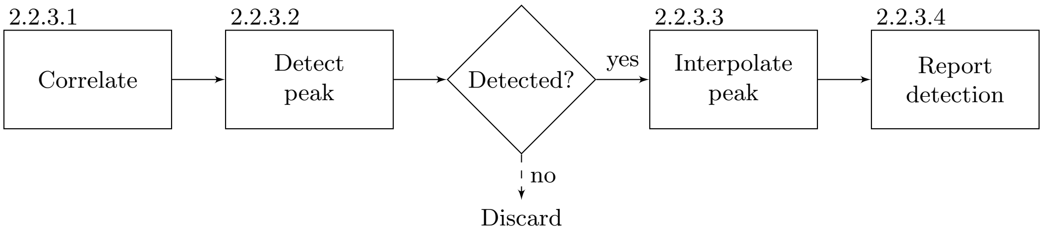


Figure 4.7: Flow diagram depicting the steps involved in estimating the SOA.

fection 2, a sufficiently accurate common time base is unavailable so that the time reported by one receiver cannot be accurately compared to the time reported by another receiver. Calculating the TDOA from TOA values is thus not viable. Instead, the TDOA is calculated from the differences in the Sample-of-Arrival (SOA) values of beacon detections and mobile unit detections. We define the SOA value as the index of a sample; the index is the number of samples the receiver has captured since it has started capturing data. The SOA value represents the index of the sample at which the template lines up with the incoming signal. Depending on the implementation, it could be defined as the sample at which the match with the template starts or the sample at which the match ends, as long as the convention is consistent. Since a SOA value is relative to the receiver's start time, it is only meaningful relative to other SOA values from the same receiver.

The purpose of the SOA estimator is to detect the presence of the positioning signal and, if present, to estimate the SOA. The process is divided into four steps, as depicted in Figure 4.7:

1. *cross-correlate* the DSSS signal with a template,
2. *detect* peaks in the correlation output that represent positioning signals,
3. *interpolate*, for each peak, between the samples close to the peak to obtain a subsample estimate of the SOA, and
4. *report* the detection to a central server so that detections from different receivers can be aggregated and the position estimated from the detection information.

Correlator

To detect the presence of the positioning signal, it is necessary to continuously match a template of the expected signal to the incoming signal by shifting the incoming signal past the template one time step at a time. The match can easily be performed using discrete cross-correlation. Let $h[n]$ be a sequence of length L_T, representing the samples of the PSK-modulated template, and let $x[n]$ be the samples of the incoming signal synchronised to the carrier. The output of the correlator is

$$y[n] = \sum_{l=0}^{L_T-1} h^*[l] \cdot x[n+l], \tag{4.26}$$

where $y[n]$ represents the correspondence between the template and a positioning signal that starts at sample n.

A continuous stream of samples is assumed in Equation (4.26), but the data is being processed in fixed-length blocks of length L_B. Equation (4.26) can be rewritten to apply cross-correlation to each block individually:

$$y_b[n] = \sum_{l=0}^{L-1} h^*[l] \cdot x_b[n+l] \qquad \text{for } 0 \leq n \leq L_B - L_T \tag{4.27}$$

where $x_b[n]$ represents the n-th sample in block b. If the history length is L_H, then

$$x_b[n] = x[b\,(L_B - L_H) + n] \qquad \text{for } 0 \leq n \leq L_B - 1 \tag{4.28}$$

$$y_b[n] = y[b\,(L_B - L_H) + n] \qquad \text{for } 0 \leq n \leq L_B - L_T. \tag{4.29}$$

Each $y_b[n]$ sequence is thus a segment of $y[n]$. To ensure that the cross-correlation is calculated for a template starting at each time step, thus ensuring that there are no gaps when the cross-correlation outputs of the individual blocks are stitched together to form $y[n]$, it is necessary that:

$$b(L_B - L_H) + (L_B - L_T) + 1 \geq (b+1)(L_B - L_H) \tag{4.30}$$

$$\therefore L_H \geq L_T - 1. \tag{4.31}$$

If $L_H > L_T - 1$, then $y_b[n]$ and $y_{b+1}[n]$ will have an overlap of $L_H - L_T + 1$ samples.

The time complexity of ordinary time-domain cross-correlation is $O(L_B L_T)$, which is too slow for real-time operation when the template is a few thousand samples long. Fortunately, the matched filter can be sped up by performing the cross-correlation in the frequency domain where convolution becomes a linear-time multiplication operation instead of a quadratic time operation. Equation (4.27) is equivalent to the discrete convolution of $x_b[n]$ and $h^*[-n]$:

$$y_b[n] = x_b[n] * h^*[-n]. \tag{4.32}$$

From the convolution theorem it follows that

$$y_b[n] = \text{DTFT}^{-1}\big\{\text{DTFT}\{x_b\} \;\cdot\; \text{DTFT}\{h_r^*\}\big\} \tag{4.33}$$

where $h_r[n] = h[-n]$. It is, however, impractical to calculate the DTFT. If $x_b[n] = 0$ for n outside the interval $[0, L_B]$ and $h_b[n] = 0$ for n outside the interval $[0, L_T]$, then the DFT-equivalent of the convolution theorem is

$$x_b[n] *_{L_B} h_r^*[n] = \text{DFT}^{-1}\big\{\text{DFT}\{x_b\} \;\cdot\; \text{DFT}\{h_r^*\}\big\} \tag{4.34}$$

for $0 \leq n \leq L_B - 1$, where $x *_N y$ denotes the N-point circular convolution of x and y. Since

$h[n] = 0$ for $n > L_T - 1$, the first $L_B - L_T + 1$ values of the circular convolution are equivalent to linear convolution:

$$x_b[n] *_N h_r^*[n] = x_b[n] * h_r^*[n] = y_b[n] \qquad \text{for } 0 \leq n \leq N - L. \tag{4.35}$$

It can be shown that, if $G(f)$ is the Fourier transform of $g(t)$, then the Fourier transform of $g^*(-t)$ is $G^*(f)$. Thus,

$$\textit{DFT}\{h_r^*\} = \textit{DFT}\{h\}^*. \tag{4.36}$$

It follows from Equations (4.34) to (4.36) that Equation (4.27) can be computed with DFTs by means of the following formula:

$$y_b[n] = \textit{DFT}^{-1}\{X_b \cdot H^*\} \qquad \text{for } 0 \leq n \leq L_B - L_T, \tag{4.37}$$

where X_b and H are the DFTs of x_b and h respectively, $x_b[n]$ is a sequence of length L_B, and $h[n]$ is a sequence of length L_T zero-padded to length L_B. To ensure that each sample of $y[b]$ is contained in at least one correlator output block, it is necessary that $L_H \geq L_T - 1$. This method of calculating a discrete convolution of a very long sequence from the DFT of short overlapping segments is generally referred to as the overlap–save method [43].

Computing the product of X_b and H^* takes only linear time, but computing $y_b[n]$ using Equation (4.37) also necessitates the computation of two DFTs and one inverse DFT. The DFTs can be calculated with a complexity of $O(L_B \log L_B)$ when the FFT algorithm is used. The time complexity for computing Equation (4.27) is thus $O(L_B \log L_B)$, compared to the $O(L_B L_T)$ time complexity of ordinary cross-correlation. A shorter block length will reduce the time complexity for processing an individual block, but will increase the total number of blocks and will make the size of the overlaps (history length) more significant in comparison to the block length. There is a trade-off that needs to be considered between the linearithmic time complexity that scales faster than the length of the blocks and the total number of samples that are discarded due to the overlapping segments.

Assume the speed of the correlation algorithm is mostly determined by the number of complex multiplications. To calculate an FFT, approximately $L_B \log_2 L_B$ complex multiplications are required. If H^* is precomputed, then the number of complex multiplications for calculating $y_b[n]$ using Equation (4.37) is that of one FFT of length L_B, as well as L_B complex multiplications in the frequency domain, and one inverse FFT of length L_B. If $L_H = L_T - 1$, then the average number of complex multiplications per output sample is

$$\frac{2L_B \log_2 L_B + L_B}{L_B - L_T + 1}.$$

The number of complex multiplications per output sample for ordinary time-domain correlation is L_T.

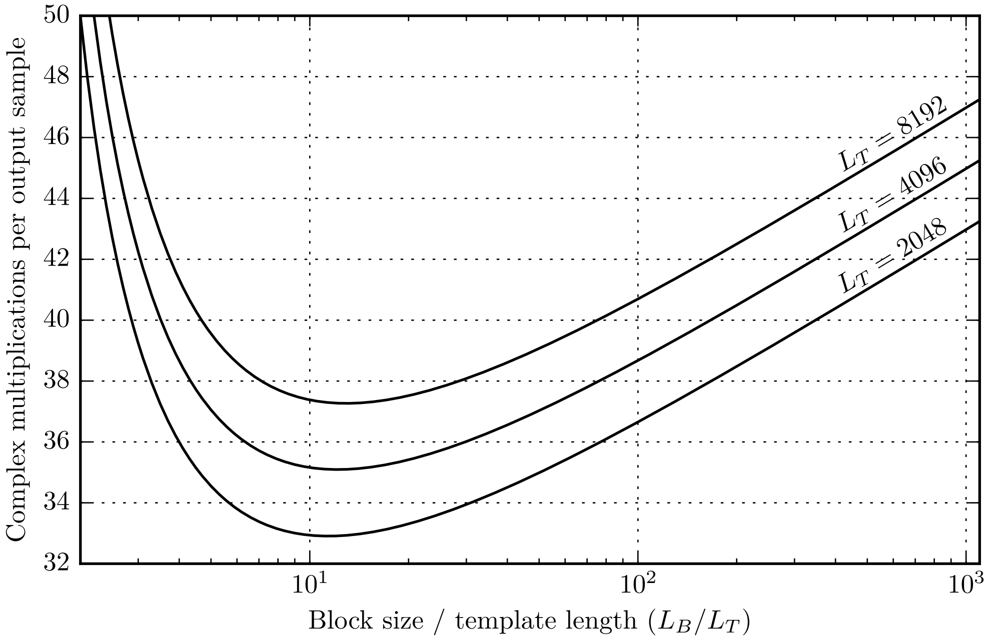


Figure 4.8: Number of complex multiplications per correlator output sample for different block lengths and different template lengths when using the overlap–save method.

The number of complex multiplications per correlator output sample as a function of block length is displayed in Figure 4.8 for when the overlap–save method is used with three different template lengths: 2048, 4096 and 8192. The block length is scaled as a multiple of the template length. It is evident from this graph that the optimal block length is approximately $L_B = 12L_T$. However, this observation is made with the assumption that the correlator is applied to all the blocks. Since blocks are prefiltered based on the presence of a carrier, it is preferable to use shorter blocks even though the shorter block length exhibits slightly lower performance per sample. Furthermore, to optimise the efficiency of the FFT algorithm, it is preferable that the block length is a power of two. Consequently, we chose the block length to be

$$L_B = 2^{\lceil \log_2(3L_H) \rceil}.$$

Peak detector

The presence of a positioning signal in a block of data is decided based on the output of the correlator. The output of the correlator represents the resemblance between a template and the incoming signal at different code phases. Since a spreading code is used, the correlator should yield a distinct peak at the sample where the incoming signal matches up best with the template. The presence or absence of the distinct peak can be used to determine whether a positioning signal is present in a block of data.

It is not sufficient to check the peak against a constant threshold since the positioning signal's power and the noise power may vary. Instead, a decision rule is constructed based on signal detection theory (see Section 3.4.1). The received signal is approximated as a scaled and phase-shifted replica of the template signal against a background of AWGN when the positioning signal is present, and as only white Gaussian noise when it is absent. The simplistic model in Equations (3.40) and (3.41) is assumed. Assuming this model, the optimal decision rule that maximises the probability of detection for a given probability of false alarm entails a comparison between the SNR of the correlation peak and a minimum threshold (Equation (3.51)).

For simplicity, it is assumed that a block of data will contain no more than one positioning signal. It is assumed that if a positioning signal is present in a block of data, then the position of the positioning signal within the block of data will be the sample at which cross-correlation with the template signal yields the maximum magnitude within the block—that is, the correlation peak. Let p be the index of block b's correlation peak:

$$p := \arg\max_{n \in S} |y_b[n]| \,. \tag{4.38}$$

The range over which the maximum is evaluated is such that the ranges of subsequent blocks do not overlap:

$$S := [o, L_B - L_T - o] \tag{4.39}$$

$$o := \left\lfloor \frac{L_H - L_T + 1}{2} \right\rfloor . \tag{4.40}$$

This is to ensure that the same correlation peak value does not trigger a detection in more than one block. The overlap is split almost equally between the two edges of the block's correlation output to ensure that the extra correlation values can be used for interpolation when the peak is located at either edge of the correlation output.

Let $\boldsymbol{x}$ be a column vector consisting of the samples in the block that yield the correlation peak when being correlated with the template:

$$\boldsymbol{x} = \Big[x_b[p], \quad \ldots, \quad x_b[p + L_T - 1]\Big]^T . \tag{4.41}$$

The decision rule is then (from Equation (3.51))

$$\chi \underset{H_0}{\overset{H_1}{\gtrless}} \beta \tag{4.42}$$

where H_1 represents the hypothesis that the positioning signal is present and H_0 that it is absent, β is the decision threshold, and χ is an SNR. More specifically, χ is the ratio of the correlation peak power to the template-correlated noise power. From Equation (3.50):

$$\chi := \frac{P_{\text{peak}}}{P_{\text{corr-noise}}} \tag{4.43}$$

$$P_{\text{peak}} := \left|\boldsymbol{h}^H \boldsymbol{x}\right|^2 = |y_b[p]|^2 \tag{4.44}$$

$$P_{\text{corr-noise}} := \left|\boldsymbol{h}^H \boldsymbol{w}\right|^2 = E_h \sigma_w^2. \tag{4.45}$$

All the values that are required for calculating χ are known except for the noise power σ_w^2. We estimate the noise power as the mean power of the block of data when the power of the correlation peak is excluded:

$$\hat{P}_{\text{corr-noise}} := \frac{E_h E_b - P_{\text{peak}}}{L_B} \tag{4.46}$$

$$E_b := \sum_{n=0}^{L_B-1} \left|x_b[n]\right|^2. \tag{4.47}$$

Finally, for a given maximum probability of false alarm α, the decision threshold β, which is the minimum SNR for a correlation peak to be considered as a valid positioning signal, can be set to $\beta = -\ln \alpha$. The probability of false alarm does not have to be set to a conservative value. A reasonable level of false positives can be tolerated since most false positives will only trigger one receiver. A detection is only valid when similar detections are recorded by at least two other receivers.

Interpolator

If the verdict of the detector is that the block contains a positioning signal, then the index of the sample within the correlator's output that has the largest magnitude represents the code phase where the template lines up best with the positioning signal. If a positioning signal is detected in block b, if p is the index of the correlation peak (Equation (4.38)), and if o is the minimum value that p can take on (Equations (4.39) and (4.40)), then the integer-valued SOA relative to the receiver's first sample is

$$s_{\text{d}} = (L_B - L_H)\, b + p - o. \tag{4.48}$$

The discrete time steps of the correlator introduce a quantisation error in the arrival time. For a detection with an integer-valued SOA of s_{d}, the signal may have arrived at any time between $s_{\text{d}} - 0.5$ and $s_{\text{d}} + 0.5$. For a sample rate of 2 MS/s, the quantisation error in arrival time can be up to 1 µs, which corresponds to a positioning error of about 300 m with a DOP of one.

As discussed in Section 3.2, a better arrival time estimate can be obtained by interpolating between the samples of the correlation peak. Multiple values near the correlation peak can be used to fit a model of the peak. The model's peak then represents the estimated position where the positioning signal lines up best with the template.

A comparison of different interpolation techniques is presented in Section 6.2.2. In summary, we selected Gaussian interpolation since it is easy to implement, fast to execute and yielded some of the best results amongst the interpolation techniques that we have tested.

Using Gaussian interpolation (Equation (3.5)), the position of the model's peak relative to the integer-valued correlation peak index is:

$$\hat{\delta} = \frac{\ln|y_1| - \ln|y_{-1}|}{4\ln|y_0| - 2\ln|y_{-1}| - 2\ln|y_1|} \tag{4.49}$$

where $y_{-1} = y_b[p-1]$, $y_0 = y_b[p]$, and $y_1 = y_b[p+1]$. The estimated SOA of the positioning signal is then

$$\hat{s}_q = s_\mathrm{d} + \hat{\delta}. \tag{4.50}$$

To ensure that Gaussian interpolation can be performed when the correlation peak is located at the edges of the correlator output, it is necessary that the correlator output for subsequent blocks overlap by at least two samples. Thus, o in Equation (4.40) should be at least one, and $L_H \geq L_T + 1$.

Note that only the magnitude of the correlation samples is currently being used. In future study, the I/Q phase of the correlation peak can be reported together with the SOA and, provided that the SOA estimate is sufficiently accurate, used to obtain sub-wavelength accuracy.

Detection reporter

When a positioning signal is detected, the receiver records an information set that describes the detection. This includes a coarse timestamp for matching detections from different receivers, the SOA, information that can be used for identifying the transmitter such as the carrier frequency, supplementary information for estimating the quality of the detection such as the signal-to-noise ratio, and the receiver's unique identifier.

4.5 Positioning server

A central server is used to estimate the position of the mobile units at different points in time by aggregating and processing the detection information it receives from the receivers. If the positions of the mobile units need to be tracked in real time, the server would continuously receive new detection information from the receivers and calculate new position estimates. If real-time tracking is not a requirement, the detection information can be accumulated and processed in bulk. For simplicity, we processed the data in bulk for the proof-of-concept system, but the algorithms can easily be adjusted for use on a continuous stream of detection information. As illustrated in Figure 4.9, the positioning server is divided into five modules:

1. an *aggregator* that collects and combines the detection information from the receivers into a single data store,

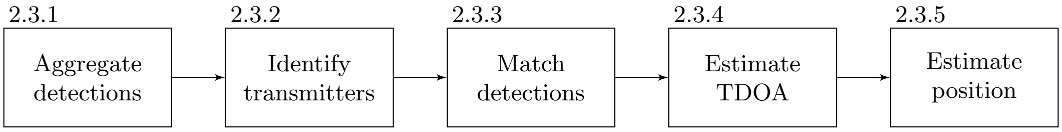


Figure 4.9: Flow diagram depicting the steps involved in estimating position from the detection information reported by the receivers.

2. an *identifier* that identifies, for each detection, the transmitter that the positioning signal originated from,
3. a *matchmaker* that matches detections from different receivers that can be attributed to the same positioning signal transmission,
4. a *TDOA estimator* that uses beacon transmissions to synchronise SOA values from different receivers and to estimate TDOA values of mobile unit detections relative to beacon detections, and
5. a *position estimator* that uses the estimated TDOA values and the positions of the receivers to estimate the positions of the mobile units.

Each of the modules is expanded upon below.

4.5.1 Aggregator

The aggregator combines the detection information from all the receivers into a central data store. The implementation could be as simple as merging several text files for data that is being processed offline, or a database to which new detections are continuously being added.

4.5.2 Identifier

To be able to discern detections from different transmitters, a means is required for identifying the transmitter from the detection information. We follow a simple strategy for the prototype system; each transmitter is set to transmit at a unique carrier frequency close to the receiver's centre frequency. The carrier frequency estimated during carrier recovery is then used to identify the transmitter. For each detection, the perceived carrier frequency is compared to the nominal carrier frequencies of the transmitters, and the identifier of the transmitter with nominal frequency closest to the perceived frequency is selected.

The spacing between the carrier frequencies of the transmitters should be wide enough to account for fluctuations in the carrier frequency due to the Doppler effect and due to the limited stability of the transmitter's crystal oscillator, as well as differences in the perceived carrier frequency due to the unsynchronised receiver clocks. Due to the wide bandwidth occupied by the positioning signal and the limited sample rate of the receiver, identification based on carrier frequency does not scale well to hundreds or thousands of transmitters. This can be replaced or combined with one or more alternative identification strategies, such as using a unique CDMA code for each transmitter, using a message signal of which the bits represent the ID and for which each bit is modulated with the code sequence, relying on a predictable transmission schedule, or transmitting a second code after a variable time interval that is unique for each transmitter.

A secondary responsibility that is assigned to the identifier is to remove duplicate detections. A portion of the positioning signal may be contained in the block prior to or following the block that contains the full positioning signal. The autocorrelation of the positioning code features secondary peaks that are, for a code with good autocorrelation properties, much weaker than the principal peak. However, with a strong positioning signal and without the principal peak in the same block to compare the secondary peak against, the energy contained in the secondary peak may be strong enough to trigger a spurious detection. A simple detection removal strategy is followed to eliminate the secondary detections. All detections are removed for which another detection exists that has the same receiver ID and the same transmitter ID, but with a higher correlation peak amplitude, and of which the SOA differs by no more than L_T, where L_T is the length of the template signal.

4.5.3 Matchmaker

To successfully determine the position of a transmitter, the transmitter's positioning signal should be detected by multiple receivers. Each receiver detects and reports the presence of positioning signals independently. The purpose of the matchmaker is to match detections reported by different receivers that can be attributed to the same transmission.

Detections from different receivers cannot be matched based on their SOAs, since SOA values are only meaningful relative to other SOA values from the same receiver. Instead, a coarse timestamp is reported with each detection. Detections from different receivers with the same transmitter ID and with timestamps that do not differ by more than a set threshold are grouped together. The threshold should be less than the minimum transmission period of the transmitters to be able to discern between different transmissions from the same transmitter. To use this matching technique, the offset between the time reported by the real-time clocks of the different receivers should not differ by more than a quarter of the transmission interval of the transmitters. For example, if the transmission intervals of the transmitters are ten seconds, then the real-time clocks should be synchronised to within less than 2.5 s, which can easily be performed with NTP if the receivers are connected to the Internet.

4.5.4 TDOA estimator

Our purpose is to estimate the position of a transmitter based on the difference in the time at which the positioning signal arrives at multiple receivers. A set of TDOA values $\{\Delta t^{(i,j)}\}$ is required, where each TDOA value $t^{(i,j)}$ is per definition the difference in the time at which the same positioning signal arrived at receivers i and j respectively:

$$\Delta t^{(i,j)} := t^{(i)} - t^{(j)}. \tag{4.51}$$

A superscript is used to denote which receiver or receivers a variable relates to. The receivers do not provide the TOA values directly, but they can be calculated from the SOA values if the exact time at which each of the receivers started sampling is known, and if every receiver sampled at a known and constant sample rate. If receiver i started sampling at time $t_0^{(i)}$ and if it sampled at a sample rate of $f_s^{(i)}$ samples per second, then sample $s^{(i)}$ was taken at time

$$t^{(i)} = t_0^{(i)} + \frac{s^{(i)}}{f_s^{(i)}}. \tag{4.52}$$

Unfortunately, the time at which the first sample was taken is unknown.

TDOA positioning uses the difference in TOA to eliminate the time of transmission from the positioning equation (Equation (2.9)). The term $t_0^{(i)}$ in Equation (4.52) can be eliminated in a similar fashion by calculating the difference in the time at which two different transmissions arrived at the same receiver. This is where one or more beacon transmitters with fixed and known positions play a role. If $s_m^{(i)}$ is the SOA at receiver i for a transmission from mobile unit m, and if $s_b^{(i)}$ is the SOA at the same receiver for a transmission from beacon b, then the difference in the time at which the receiver detected these signals is:

$$t_m^{(i)} - t_b^{(i)} = \left(t_0 + \frac{s_m^{(i)}}{f_s^{(i)}}\right) - \left(t_0 + \frac{s_b^{(i)}}{f_s^{(i)}}\right) = \frac{s_m^{(i)} - s_b^{(i)}}{f_s^{(i)}}. \tag{4.53}$$

Consider the difference in the time difference at which the positioning signals from mobile unit m and beacon b are detected by receivers i and j respectively:

$$\left(t_m^{(i)} - t_b^{(i)}\right) - \left(t_m^{(j)} - t_b^{(j)}\right) = \frac{s_m^{(i)} - s_b^{(i)}}{f_s^{(i)}} - \frac{s_m^{(j)} - s_b^{(j)}}{f_s^{(j)}}.$$

Rearranging the terms on the left yields:

$$\left(t_m^{(i)} - t_m^{(j)}\right) - \left(t_b^{(i)} - t_b^{(j)}\right) = \frac{s_m^{(i)} - s_b^{(i)}}{f_s^{(i)}} - \frac{s_m^{(j)} - s_b^{(j)}}{f_s^{(j)}}.$$

Substituting the TDOA value with a single variable (Equation (4.51)) and rearranging once again provides an equation for calculating the TDOA:

$$\Delta t_m^{(i,j)} = \frac{s_m^{(i)} - s_b^{(i)}}{f_s^{(i)}} - \frac{s_m^{(j)} - s_b^{(j)}}{f_s^{(j)}} + t_b^{(i)} - t_b^{(j)} \tag{4.54}$$

The value of $t_b^{(i)} - t_b^{(j)}$ is a known constant. It can be computed with Equation (2.10) since the position of the beacon and the receivers are known. Furthermore, the SOA values are reported by the receivers and are thus known. Thus, Equation (4.54) would have sufficed for calculating TDOA values if the receivers' clocks were synchronised to ensure a known and constant sample rate.

Sample rate error analysis

The clocks at the receivers are independent and unsynchronised (Imperfection 3) and may exhibit a frequency error between 1 ppm and 10 ppm. If the nominal sample rate is f_s, then the actual sample rate of receiver i is

$$f_s^{(i)} = f_s\left(1 + \epsilon^{(i)}\right) \tag{4.55}$$

where $\epsilon^{(i)}$ is the sample rate error of receiver i, $\left|\epsilon^{(i)}\right| \leq 10^{-5}$.

For concise notation, set $\Delta s_{m,b}^{(i)} := s_m^{(i)} - s_b^{(i)}$ and let $\Delta\tau_{m,b}^{(i,j)}$ be the difference in the time difference at which a signal from mobile unit m and beacon b arrive at receivers i and j respectively, as calculated from the SOA values, thus

$$\Delta\tau_{m,b}^{(i,j)} := \frac{\Delta s_{m,b}^{(i)}}{f_s^{(i)}} - \frac{\Delta s_{m,b}^{(j)}}{f_s^{(j)}}. \tag{4.56}$$

Equation (4.54) can be rewritten as

$$\Delta t_m^{(i,j)} = \Delta\tau_{m,b}^{(i,j)} + t_b^{(i)} - t_b^{(j)}. \tag{4.57}$$

The purpose now is to obtain a good estimate of $\Delta\tau_{m,b}^{(i,j)}$ in order to estimate the TDOA. To analyse the impact of the sample rate error, substitute Equation (4.55) into Equation (4.56) and expand:

$$\begin{aligned}
\Delta\tau^{(i,j)} &= \frac{\Delta s^{(i)}}{f_s\left(1 + \epsilon^{(i)}\right)} - \frac{\Delta s^{(j)}}{f_s\left(1 + \epsilon^{(j)}\right)} \\
&= \frac{\Delta s^{(i)}\left(1 + \epsilon^{(j)}\right) - \Delta s^{(j)}\left(1 + \epsilon^{(i)}\right)}{f_s\left(1 + \epsilon^{(i)}\right)\left(1 + \epsilon^{(j)}\right)} \\
&= \frac{1}{1 + \epsilon^{(i)} + \epsilon^{(j)} + \epsilon^{(i)}\epsilon^{(j)}}\left(\frac{\Delta s^{(i)} - \Delta s^{(j)}}{f_s} + \frac{\Delta s^{(i)}\epsilon^{(j)} - \Delta s^{(j)}\epsilon^{(i)}}{f_s}\right)
\end{aligned} \tag{4.58}$$

where the subscript "r, b" has been omitted for simplicity, and where $\breve{\tau}^{(i,j)}$ is the value of $\Delta\tau^{(i,j)}$ when using the nominal sample rate, which is a known value. Set

$$\lambda^{(i,j)} := \frac{1}{1 + \epsilon^{(i)} + \epsilon^{(j)} + \epsilon^{(i)}\epsilon^{(j)}} \tag{4.59}$$

$$e^{(i,j)} := \frac{\Delta s^{(i)}\epsilon^{(j)} - \Delta s^{(j)}\epsilon^{(i)}}{f_s}. \tag{4.60}$$

Equation (4.58) can now be written in terms of the error coefficient $\lambda^{(i,j)}$ and error term $e^{(i,j)}$:

$$\Delta\tau^{(i,j)} = \lambda^{(i,j)} \left(\breve{\tau}^{(i,j)} + e^{(i,j)}\right) \tag{4.61}$$

If $\left|\epsilon^{(i)}\right| \leq \epsilon_{\max}$ and $\left|\epsilon^{(j)}\right| \leq \epsilon_{\max}$, then

$$\frac{1}{1 + \epsilon_\lambda} \leq \lambda^{(i,j)} \leq \frac{1}{1 - \epsilon_\lambda} \tag{4.62}$$

where $\epsilon_\lambda = 2\epsilon_{\max} + \epsilon_{\max}^2$. It can be shown that, if $\epsilon_\lambda < 1$, then

$$|1 - \lambda| \leq \frac{\epsilon_\lambda}{1 - \epsilon_\lambda}. \tag{4.63}$$

Thus, with $\epsilon_{\max} \leq 10^{-5}$, the error in the value of $\Delta\tau^{(i,j)}$ due to λ will be less than 20 ppm. If the distances between the two receivers, the beacon, and the transmitter under consideration are no more than 20 km, then the error due to λ will be less than 0.4 m. This back-of-the-envelope calculation is an upper bound, and in practice the error contribution would be even more negligible. It is thus safe to assume that $\lambda \approx 1$. Equation (4.61) can then be approximated as:

$$\Delta\tau^{(i,j)} \approx \breve{\tau}^{(i,j)} + e^{(i,j)}. \tag{4.64}$$

Let $\tau_{m,b}^{(i)}$ be the difference in the time at which the transmission from mobile unit m and the transmission from beacon b are detected by receiver i:

$$\tau_{m,b}^{(i)} := t_m^{(i)} - t_b^{(i)}. \tag{4.65}$$

and similarly, let $\tau_{m,b}^{(j)}$ be the difference at receiver j. Omitting the subscript, substituting Equation (4.53) into Equation (4.65), and substituting that result into Equation (4.60) for both $\tau^{(i)}$ and $\tau^{(j)}$ yields:

$$e^{(i,j)} = \tau^{(i)}\epsilon^{(j)} - \tau^{(j)}\epsilon^{(i)}. \tag{4.66}$$

It is evident from Equation (4.66) that the error is a function of the difference in the time at which the positioning signal from the mobile unit and the signal from the beacon are being detected. Since the beacon has a fixed position, any transmission from the same beacon would produce the same TDOA value. The error can thus be reduced by selecting the beacon detection with the SOA that is closest to the mobile unit's SOA. This is not a surprising result and could have been foreseen through intuition. The longer the time that passes between the beacon detection and the mobile unit detection, the more samples are being taken and the longer is the time over which the sample rate error is being integrated. However, simply taking the nearest beacon detection is not sufficient. Even when the beacon and the mobile unit detections are within one second of each other, the error being introduced in the TDOA estimate due to a relatively small clock offset of 1 ppm at only one of the receivers would be 1 µs. This TDOA error is equivalent to a positioning error of roughly 300 m when the DOP is one. The error can be reduced by decreasing the time between subsequent beacon transmissions, but transmitting too frequently is impractical. Instead, we devised a method for reducing the difference between the mobile unit SOA and the beacon SOA by interpolating between two beacon detections.

Interpolate between two beacon detections

A fast and easy technique for reducing the time between the mobile unit detection and the beacon detection is to estimate the SOA of a virtual beacon transmission that is detected at the same time as the mobile unit detection. This virtual detection can be obtained by interpolating between two detections from the same beacon.

Let the two beacon detections with SOAs closest to the SOA of the mobile unit detection be denoted by the subscripts b_1 and b_2 respectively. The sample rate of receivers i and j can then be estimated as

$$\hat{f_s}^{(k)} = \frac{s_{b_2}^{(k)} - s_{b_1}^{(k)}}{\Delta t_b} \qquad \text{for } k = i, j \tag{4.67}$$

where Δt_b is the time elapsed between the two beacon transmissions.

Consider a hypothetical beacon transmission of which the SOA at receiver i is equal to the SOA of the mobile unit, thus

$$\begin{aligned} \hat{s}_b^{(i)} &= s_m^{(i)} \\ \hat{t}_b^{(i)} &= t_m^{(i)}. \end{aligned}$$

The estimated SOA of the hypothetical beacon transmission at receiver j is then

$$\begin{aligned} \hat{s}_b^{(j)} &= s_{b_1}^{(j)} + \hat{f_s}^{(j)} \left(\hat{t}_b^{(i)} - t_{b_1}^{(i)}\right) \\ &= s_{b_1}^{(j)} + \hat{f_s}^{(j)} \left(\frac{s_m^{(i)} - s_{b_1}^{(i)}}{\hat{f_s}^{(i)}}\right) \\ &= s_{b_1}^{(j)} + \left(s_{b_2}^{(j)} - s_{b_1}^{(j)}\right) \underbrace{\frac{s_m^{(i)} - s_{b_1}^{(i)}}{s_{b_2}^{(i)} - s_{b_1}^{(i)}}}_{w_b^{(i)}:=} \\ &= w_b^{(i)} s_{b_2}^{(j)} + (1 - w_b^{(i)}) s_{b_1}^{(j)}. \end{aligned} \tag{4.68}$$

The estimated value of $\Delta\tau^{(i,j)}$ is

$$\Delta\hat{\tau}^{(i,j)} = \frac{w_b^{(i)} s_{b_2}^{(j)} + (1 - w_b^{(i)}) s_{b_1}^{(j)} - s_m^{(j)}}{f_s} \tag{4.69}$$

where f_s is the nominal sample rate, and the estimated error is

$$\hat{e}^{(i,j)} = -\hat{\tau}^{(j)} \epsilon^{(i)}. \tag{4.70}$$

If the distance between the beacon, the mobile unit and the two receivers is no more than 20 km, then

$$\left|\hat{\tau}^{(j)}\right| \leq \frac{20\,000\,\mathrm{m}}{c}$$

and if $\left|\epsilon^{(j)}\right| \leq 10^{-5}$, then

$$\left|e^{(i,j)}\right| \leq \frac{0.2\,\mathrm{m}}{c}.$$

Thus, if the DOP is one, then the positioning error due to $\left|e^{(i,j)}\right|$ will be 0.2 m at most, which is negligible.

For the derivation above, we assumed that the short-term stability of the sampling clock is good enough to approximate the clock error as a constant offset. The time interval between subsequent beacon transmissions should be short enough to ensure that changes in the receiver's clock frequency are negligible during the time that passes between the beacon detection and the mobile unit detection. For the error analysis, we also assumed precise SOA measurements.

Fitting a model to synchronise the SOA values

The error due to the limited accuracy of the SOA measurements of the beacon detections can be reduced by extending the TDOA estimation technique discussed above to a least squares problem over multiple beacon detections. Changes in the clock error can be estimated with an interpolation technique that incorporates a more complicated model of the sample rate error.

Suppose the sample rate of receiver i can be modelled as a linear function of the sample index,

$$f_s^{(i)}(s) = f_s \left(1 + \epsilon_0^{(i)} + \epsilon_1^{(i)} s\right) \tag{4.71}$$

where f_s represents the nominal sample rate, $\epsilon_0^{(i)}$ the clock offset, and $\epsilon_1^{(i)}$ the clock drift. The SOA at one receiver can then be related to the SOA values of another receiver using a quadratic equation:

$$\hat{s}^{(j\to i)}(s) = a_0 + a_1 s + a_2 s^2 \tag{4.72}$$

where $\hat{s}^{(j\to i)}(s)$ is a model function that transforms an SOA value from receiver j to a value that can be compared to the SOA values of receiver i. The coefficients a_0, a_1 and a_2 are different for different pairs of receivers i and j.

Given a set of SOAs of beacon transmissions that were detected at both receiver i and receiver j, the coefficients of the model can be estimated by means of an LS fit. Let

$$\boldsymbol{a} := \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix}^T,$$

and let

$$\begin{aligned} \boldsymbol{s}_b^{(i)} &:= \begin{bmatrix} s_{b(1)}^{(i)} & \dots & s_{b(N)}^{(i)} \end{bmatrix}^T \text{ and} \\ \boldsymbol{s}_b^{(j)} &:= \begin{bmatrix} s_{b(1)}^{(j)} & \dots & s_{b(N)}^{(j)} \end{bmatrix}^T \end{aligned}$$

be the beacon detections from receiver i and j respectively, where $s_{b(z)}^{(i)}$ and $s_{b(z)}^{(j)}$ are a pair of detections that originated from the same beacon transmission, for $1 \leq z \leq N$.

Let $\boldsymbol{s}_b^{(j@i)}$ be the SOAs of beacon detections at receiver j as if receiver j were at the same position as receiver i, thus adapted for the difference in distance between the beacon and the respective receivers:

$$\boldsymbol{s}_b^{(j@i)} := \begin{bmatrix} s_{b(1)}^{(j)} + \Delta s_{b(1)}^{(i,j)} & \dots & s_{b(N)}^{(j)} + \Delta s_{b(N)}^{(i,j)} \end{bmatrix}^T \tag{4.73}$$

where $\Delta s_{b(z)}^{(i,j)}$ is the difference in SOA due to the difference in distance between the beacon and receiver i and receiver j respectively. This is a known value since the positions of the receivers and the beacons are known, so:

$$\Delta s_{b(z)}^{(i,j)} := \frac{f_s}{c} \left(\left\| \boldsymbol{x}_i - \boldsymbol{x}_{b(z)} \right\| - \left\| \boldsymbol{x}_j - \boldsymbol{x}_{b(z)} \right\| \right) \tag{4.74}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the positions of receivers i and j respectively, and $\boldsymbol{x}_{b(z)}$ is the position of the beacon where beacon transmission z originated from. Note that the beacon transmissions may originate from multiple beacons at different positions.

The coefficients of the model in Equation (4.72) can be estimated from the LS fit of $\boldsymbol{s}_b^{(i)}$ and $\boldsymbol{s}_b^{(j@i)}$ to the model function:

$$f(s, \boldsymbol{a}) = a_0 + a_1 s + a_2 s^2 \tag{4.75}$$

$$\boldsymbol{a}_{\text{LS}} := \arg\min_{\boldsymbol{a}} \sum_{z=1}^{N} \left(s_{b(z)}^{(i)} - f\left(s_{b(z)}^{(j@i)}, \boldsymbol{a} \right) \right)^2. \tag{4.76}$$

Finally, the model in Equation (4.72) can be used with the LS coefficients to estimate the TDOA between receivers i and j for a detection from mobile unit m:

$$\Delta \hat{t}_m^{(i,j)} = \frac{1}{f_s} \left(s_m^{(i)} - \hat{s}^{(j \to i)}(s_m^{(j)}) \right). \tag{4.77}$$

To limit the impact of changes in the sample rate and differences between the actual sample rate and the model, it is best to fit a new model for each mobile unit detection and only use beacon detections that are close to the SOA of the mobile unit detection. Alternatively, a piecewise model can be constructed with the coefficients recalculated over different segments of the beacon detections.

Furthermore, note that model functions can be formed from other model functions through composition

$$\hat{s}^{(k \to i)}(s) = \hat{s}^{(j \to i)}(\hat{s}^{(k \to j)}(s)). \tag{4.78}$$

The model is thus not limited to include only the beacon transmissions that were detected at receivers i and j, but can be composited from detections at other receivers. This composition should be used sparingly since SOA estimation errors will accumulate.

4.5.5 Position estimator

Once the TDOA estimates are known, the information we are interested in can finally be calculated, namely the positions of the mobile units. A multitude of algorithms exists for estimating position from TDOA estimates. We employed the Levenberg–Marquardt algorithm for the proof-of-concept implementation. Refer to Sections 2.2.3 and 2.2.4 for information about position estimation algorithms.

Additional information such as the DOP at the position estimate, mean SNR of the detections that were involved, and confidence ellipse can be calculated and reported with the position estimate. Furthermore, an extra processing step can be added after the position estimator to remove outliers and even out estimation errors by calculating the average of multiple position estimates or by applying path-based filtering and smoothing algorithms.

4.6 Chapter summary

In summary, the system consists of transmitters in the form of mobile units (of which the position should be estimated) and beacons (of which the position is known), receivers, and a central positioning server. The transmitters periodically transmit an OOK-modulated positioning code. The receivers continuously search for positioning signals from the transmitters.

At each receiver, the RF signal is down-converted, sampled and split into fixed-length blocks of complex samples. For each block, the presence of the positioning signal's carrier is determined based on the SNR of the frequency component of the block's DFT that yields the maximum spectral density within the window of expected frequencies. If the carrier is deemed absent, the block is discarded. If present, the carrier frequency is estimated, and the signal within the block of data is synchronised to the carrier frequency. The block of data is then cross-correlated with a template signal to determine the presence of the positioning code. If the SNR of the correlation peak exceeds a set threshold, the positioning signal is considered present. If present, a better estimate of the position where the positioning signal lines up best with the template is obtained through interpolation of the correlation peak. This position, which represents the time at which the positioning signal arrived at the receiver, is expressed in terms of a real-valued sample index, called the Sample-of-Arrival (SOA). The receiver reports the SOA together with additional information about the detection to a positioning server.

The positioning server collects the detections from all the receivers and saves them in a data store. For each detection, the transmitter the positioning signal originated from is identified from the carrier frequency that was reported by the receiver. Detections that are reported by different receivers but that can be attributed to the same transmission are then matched together. Next, for each group of detections from a mobile unit transmission, beacon transmissions that were detected within a short time frame from the mobile unit transmission are used to create a model that relates the SOA values of different receivers to each other. The model and the SOA values of the mobile unit that were reported by the different receivers are then used to estimate the TDOA values. Lastly, the TDOA values and the positions of the receivers are used to estimate the position of the mobile unit.

The next chapter describes a proof-of-concept implementation of the design discussed in this chapter. This implementation was developed to test the accuracy and feasibility of the design.

Chapter 5

Implementation with RTL-SDR and Raspberry Pi

This chapter describes the implementation of the design in the previous chapter using inexpensive hardware for a proof-of-concept positioning system. The chapter starts with a brief discussion in Section 5.1 of the hardware and software implementation for transmitting a positioning code at a high baud rate. Section 5.2 details the hardware that is used for signal processing at the receivers. Motivations are given for the choice of an SDR as radio front-end, an RTL-SDR as SDR platform, and a Raspberry Pi as the signal processor. Furthermore, the characteristics of a preamplifier, designed to improve the sensitivity and selectivity of the receivers, are outlined. A description of the architectural design and implementation of the signal processing software on the receivers and the detection processing software on the server is given in Section 5.3. The implementation of a prototype positioning system that was constructed to test the performance and feasibility of the design as a whole is discussed in Section 5.4, as well as the values of the parameters that were used for tests and experiments. Section 5.5 concludes the chapter with a summary of the implementation.

5.1 Transmitter

5.1.1 Tag device

Since the design and implementation of the transmitter are not the focus of this dissertation, existing hardware was employed as far as possible. Instead of developing a tailor-made transmitter device for TDOA positioning, we utilised existing general-purpose tag devices that were designed and developed by the industry partner, *Wireless Wildlife*. These devices are being used by the industry partner for various tracking and remote sensing applications, most involving the use of the device as a tag that is attached to livestock or wildlife. The device sports

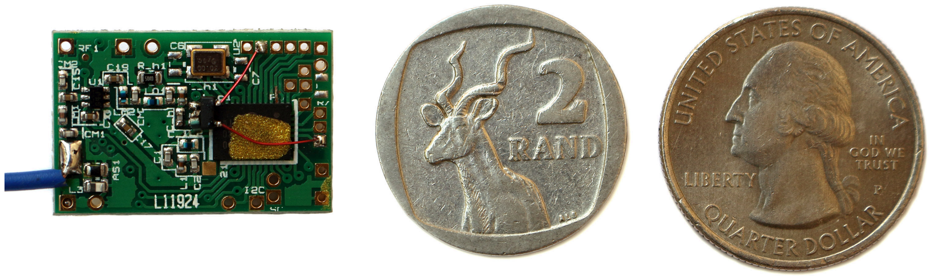


Figure 5.1: Transmitter board next to a two rand and a quarter dollar coin.

a versatile design that can be adapted for different applications. Devices can be customised with a range of sensors and modules, such as an activity sensor, pressure sensor, temperature sensor, GPS module, GSM module, and flash memory for data logging. At the core of the device is the Silicon Labs Si1000 chip: a low-power 8-bit 8051-compatible microcontroller (MCU) with an integrated sub-1 GHz RF transceiver [44].

A photograph of the transmitter board is displayed in Figure 5.1. The board measures 25 × 16 mm, but the size can be reduced when support for the additional sensors and modules that are not required for TDOA positioning is removed.

5.1.2 OOK at high baud rate

The tag device had to be modified before it could be used for TDOA positioning. Even though the device features a transceiver with support for OOK, FSK and GFSK, the transceiver's maximum data rate is only 256 kbit/s for FSK modulation and 40 kbit/s for OOK modulation [44]. These low data rates do not supply a signal with a sufficiently wide bandwidth. As a makeshift solution, we reworked the devices to toggle the supply voltage of the Power Amplifier (PA) from a digital output of the MCU. The positioning code is transmitted by configuring the transceiver to transmit an unmodulated carrier wave. The carrier wave is then modulated by rapidly switching the state of the MCU's digital output, which in effect switches the PA's supply voltage between a connected state (when the code bit is one) and a disconnected state (when the code bit is zero).

The switching is performed through a transistor to limit the current being sourced from the MCU output pin. A photograph of the modification applied to the board to enable OOK modulation at a high chip rate is displayed in Figure 5.2. The pull-up inductor of the PA, which was connected to the power supply, was lifted and connected to the drain of a P-channel power MOSFET soldered in mid-air. The gate of the MOSFET is connected to the MCU output pin and the source of the MOSFET to the power supply.

A fifth-order Chebyshev low-pass filter and harmonic termination circuit follows the power amplifier. Harmonics resulting from the switching of the power amplifier are thus attenuated

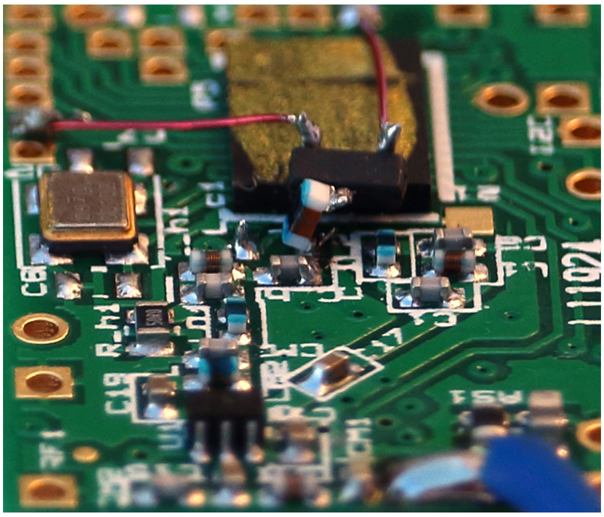


Figure 5.2: Close-up of the MOSFET soldered onto the transmitter board for toggling the power supply of the PA. This modification enables OOK modulation at a high baud rate.

without any modifications to the transmitter board.

5.1.3 Software

The positioning code, a Gold code with a length of 2047 bits, is stored in the code memory of the MCU. The same code sequence is stored on all the transmitters, while each transmitter is set to transmit at a unique carrier frequency around 434 MHz, each a few kilohertz apart from the other. The code is transmitted at a chip rate of 1 MHz by latching the stored code bit-by-bit onto the output pin that modulates the carrier wave. With a system clock of 25 MHz, there are exactly 25 CPU clock cycles between two bits. The software that outputs the code was implemented in assembly language to ensure exact and deterministic timing between the code bits. The transmitter can be set to retransmit the code at a given periodic interval. The MCU and transceiver chip is set to a low-power sleep mode between subsequent transmissions. In the sleep mode, the MCU consumes very little power; the current consumption is only about 0.7 µA. An integrated clock allows the MCU to wake up after a set period of time.

5.1.4 Energy consumption

The transceiver consumes about 90 mA during transmission, but the transmission time is only 2 ms long. Thus, a single transmission consumes about 180 µAh, which amounts to 324 µJ of energy when a 1.8 V battery is being used. With Example 1.1 as baseline, the energy consumption is about three orders of magnitude smaller than the energy that is consumed by a GPS-enabled tag.

5.1.5 Accurate chip rate

Another modification that was applied to the transmitter board is the addition of an external crystal oscillator for the MCU. Since the code modulation is performed with the MCU, the accuracy of the chip rate is determined by the accuracy of the MCU's system clock. An external oscillator was already present for the RF transceiver module, but the MCU is independent from the transceiver. Without an external crystal oscillator, the best precision that can be obtained with the MCU's precision internal oscillator is 20 000 ppm. This results in a chip rate that varies significantly from one transmitter to another. A chip rate mismatch between the transmitted signal and a template widens the correlation peak and introduces an uncertainty as to where the signals line up best. An external crystal oscillator with a frequency error of 20 ppm was connected to the MCU to ensure an accurate chip rate.

5.2 Receiver hardware

5.2.1 RF receiver

A host of generally available, low-cost and easy-to-use RF receiver, transmitter and transceiver devices exist for digital data communication. These devices, however, cannot be used as the RF receiver in a TDOA positioning system since they solve a different problem. Whereas these devices are concerned with data communication and have implementation details such as demodulation, preamble detection and packet handling encapsulated behind an abstracted interface, the transfer of data is not the goal when estimating position. The precise time at which the signal arrives is the information we are interested in — information that is not provided by these devices because it is not relevant to data communication. Since arrival time estimation is a specialised and relatively rare problem outside the realm of GPS and radar, RF receivers with arrival time estimation capabilities are not available as a commercial product on the market. The design and development of special-purpose hardware are costly and time-consuming, and may not be economically viable for the niche market of wildlife tracking.

Software defined radios

Instead of developing special-purpose hardware, we use a general-purpose Software-Defined Radio (SDR) receiver and hand over the specialised signal processing to a general-purpose processor. This allows mass-produced, low-cost COTS hardware to be used, while special-purpose signal processing is developed in software, which is easy and cheap to develop in comparison with hardware. It has the additional benefit that the signal processing algorithms can be experimented with and modified in an easy and cost-effective manner. No hardware reconfiguration is required for a different arrival time estimation technique or a new positioning signal, but an upgrade is as simple as replacing the software.

One of the best known SDRs is the Universal Software Radio Peripheral (USRP), designed and sold by Ettus Research. It features an open-source hardware design with well-defined characteristics and open performance data, datasheets that are freely available, open-source drivers and support for several software frameworks. However, even with its reputation as a low-cost SDR platform, the cheapest model, the USRP B200, costs $686. For a four receiver positioning system, about $2744 would thus have to be spent on the SDRs, which renders the positioning system economically infeasible for many wildlife tracking applications.

RTL-SDR

Instead of working with a costly SDR platform, we employ inexpensive digital TV tuners that are repurposed as SDRs. It was discovered that generally available DVB-T USB dongles with the Realtek RTL2832U chip have an undocumented mode in which the raw I/Q samples can be accessed directly, which effectively turns the TV tuner into a cheap wideband SDR [45]. This device is commonly referred to as an *RTL-SDR*. There is a wide variety of models from different vendors available on the market. All the RTL-SDR dongles contain the Realtek RTL2832U chip as a controller, for data acquisition, and as USB interface, but the tuner IC being used, the tolerances on auxiliary components such as crystal oscillators, and the overall build quality vary widely from one model to another. An unbranded RTL-SDR dongle can cost as little as $10, whereas models with a TCXO and better component tolerances are sold for about $20.

The resolution of the I/Q samples produced by the RTL2832U chip is 8 bits [45]. The maximum theoretical sample rate is 3.2 MS/s. However, when the host does not consume the data fast enough, the samples are dropped by the chip. A safe maximum sample rate that does not cause samples to be dropped is reported to be between 2.4 MS/s [45] and 2.56 MS/s [46]. The frequency range of the RTL-SDR depends on the tuner that is used for the particular model. The Rafael Micro R820T, which is found in many if not most RTL-SDR variants on the market today, can be tuned to a frequency from 24 MHz to 1766 MHz. Refer to [45–47] for more information about the typical specifications of RTL-SDR dongles.

Most user-level SDR software for the RTL-SDR uses the device through an open-source software library known as librtlsdr [46]. Since there is no public datasheet of the RTL2832U chip available, this unofficial library is based on an analysis of and reverse engineering of the commands being sent between the official Realtek drivers and the Realtek chip.

A $20 RTL-SDR is more than thirty times cheaper than the cheapest USRP model. It is, however, unfair to compare the two since they play in different leagues. The USRP is a professional-grade SDR and vastly outperforms the RTL-SDR in every aspect. For example, it has a higher sample rate, higher sample resolution, wider frequency range and better frequency stability. The characteristics of this carefully and professionally designed SDR are well documented and support is readily available. In contrast, the RTL2832U chip was not originally intended to be used as an SDR and information about its internals is unavailable; no official

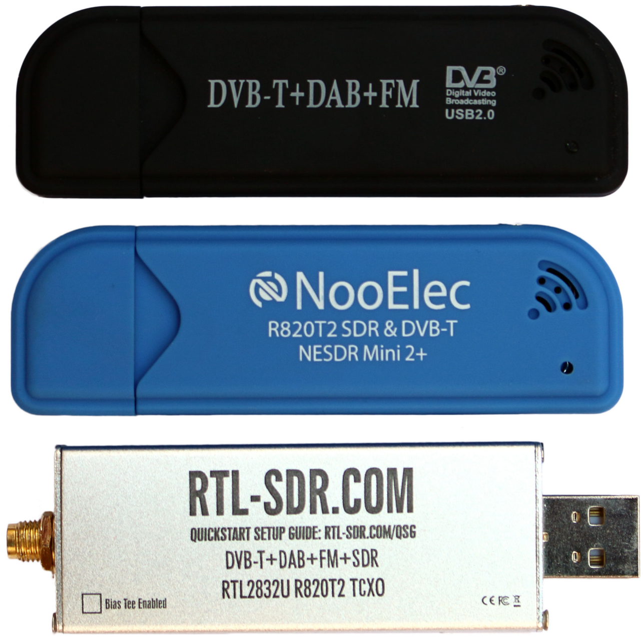


Figure 5.3: Three different RTL-SDR variants: an unbranded variant, a variant sold by NooElec and another variant sold by RTL-SDR.com.

schematics or publicly available datasheets exist. However, the RTL-SDR is incomparable in terms of its low cost and the value it provides at its pricing level.

Three different kinds of RTL-SDR devices are displayed in Figure 5.3. For the tests in this dissertation, we used version two of the variant sold by RTL-SDR.com [48], which was purchased at a cost of $20 per unit. This model features a Rafael Micro R820T2 tuner and a TCXO with an initial frequency error of 2 ppm and a temperature drift of 1 ppm [49].

5.2.2 Signal processor

A signal processor is required to process the raw I/Q samples from the SDR in real-time and estimate the arrival time of positioning signals. Real-time digital signal processing with a deterministic time delay can be performed with a DSP chip or an FPGA, but that would require a custom-developed processor and RF receiver board. On the other end of the spectrum is a general-purpose PC, which is versatile, allows for fast development and can easily be interfaced with the SDR device, but it is costly and its power consumption is prohibitive. The receiver stations will be deployed outdoors on masts without access to mains electricity. The size, weight, and power consumption of the signal processor should be taken into account in addition to the cost.

With the rapid growth in the processing power provided by general-purpose Single-Board Computers (SBCs) in recent years, SBCs have become powerful enough for many real-time signal processing applications. An SBC is not nearly as powerful as a PC, but it provides the advantage of low cost, small footprint and relatively low power consumption. An SBC is not as power efficient and its processing delay not as deterministic as a dedicated DSP chip, but it is unmatched in its versatility, usability, and the rapid development it enables. It also allows the installation of an advanced operation system such as Linux. The host of libraries, tools, drivers and software available on Linux can be leveraged to simplify and speed up development. It provides flexibility in the choice of programming language and effortless transition between development and field tests since the same software can be executed on a development PC as well as an SBC. Complex devices can be attached effortlessly; adding internet connectivity to the receiver in an outdoor environment is as simple as attaching a mobile broadband modem and, if not already supported by the OS, installing the drivers.

We employed the Raspberry Pi for the prototype positioning system. The Raspberry Pi is a popular low-cost, credit card-sized, mass-produced Single-Board Computer (SBC). The single-core CPU of the first generation Raspberry Pi is not powerful enough to compute the FFT of the signal in real-time. It can, however, perform carrier detection in real-time when the FFT calculation is offloaded to the GPU. The second and third generation of the Raspberry Pi devices provide vastly better performance. Whereas the Raspberry Pi 1 has a single-core CPU without NEON support, the Raspberry Pi 2 and 3 have four cores with NEON support. NEON is a general-purpose SIMD instruction set that can be used to accelerate multimedia and signal processing algorithms. With this acceleration, a single core of the quad-core CPUs found in the Raspberry Pi 2 and 3 can compute the FFT of the incoming signal in real-time.

The Raspberry Pi 2 and the Raspberry Pi 3 are both sold for $35 per unit. Since they cost the same and the Raspberry Pi 3 provides slightly better performance, we chose to use the Raspberry Pi 3 as the signal processor for the prototype positioning system.

5.2.3 Preamplifier

SDRs are usually designed to be tuned over a wide range of frequencies. SDRs such as the RTL-SDR amplify the signal from the antenna over a wide range of frequencies, with most of the filtering being performed after the mixing stage. Strong out-of-band signals such as GSM signals can generate intermodulation products that can cause interference and can even saturate the SDR's ADC. A preamplifier was designed to improve the selectivity of the receivers by suppressing out-of-band signals. The low-noise amplifier also provides gain at a low noise figure to increase the sensitivity of the receiver. Since the design of the preamplifier does not fall within the scope of this dissertation, it was designed by a third party with prior experience with amplifier design at the NWU's Unit for Space Research.

The preamplifier with its enclosure removed is displayed in Figure 5.4, shown connected to an RTL-SDR. The circuit of the preamplifier is simple, consisting of only one transistor and a few passive components. It is designed with a gain of about 20 dB at 434 MHz to amplify the

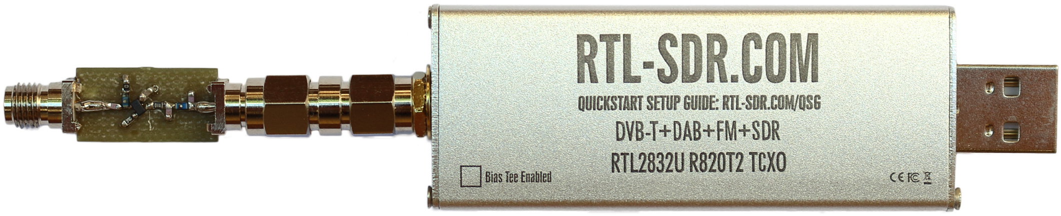


Figure 5.4: Preamplifier without its enclosure, shown connected to an RTL-SDR.

positioning signals and a gain of less than −10 dB at 900 MHz to suppress GSM signals. The noise figure is about 0.5 dB at 434 MHz. The amplifier is powered by USB bus power with a bias tee and can be fit in line with the antenna.

The custom-built preamplifier does not adhere to the requirement of COTS receiver hardware, but it is optional and can be replaced with an alternative off-the-shelf solution such as the HAB-FPA434 from Uputronics.

5.3 Signal and detection processing software

The hardware of the prototype positioning system is simple; it consists of an RTL-SDR and a Raspberry Pi. The essence of the design that was described in Chapter 4 and of the positioning system itself lies within the software.

The receiver signal processing software and the server position estimation software were implemented and experimented with in Python. Python is a popular high-level, dynamic programming language with a free and open source reference implementation, CPython. It is an extensible and portable general-purpose language. The multitude of libraries available for Python enables it to be used for various purposes. Extensive mathematical libraries are available that enable Python to be used as a scientific scripting language. Python's focus on code readability, conciseness, as well as its interpreted nature allows for easy experimentation within a few lines of code. In comparison to a numerical programming language such as Matlab, Python provides the benefits of a generic language; scientific code can make use of the myriads of practices, tools and libraries available for general software development. The scientific code can, for example, easily interface with database software, and it can be easily extended with a graphical user interface and transformed into a product. Another advantage of the use of Python instead of proprietary software such as Matlab is that it does not impose any entry barriers in terms of cost. It is free and open source, even for commercial use.

The libraries we used are a numerical library, *Numpy* [50], scientific library, *Scipy* [51], and data visualisation library, *matplotlib* — three libraries commonly used together for scientific programming in Python. Even though both Numpy and Scipy provide routines for calculating the FFT, we imported *pyFFTW* [52] as an optional library to speed up FFT computations.

5.3.1 Software architecture

The Python implementation of the signal processing and position estimation algorithms, codenamed *Thrifty*, is divided into multiple Python modules, each with a Python Application Programming Interface (API) and a Command-Line Interface (CLI). The different modules can be chained together through Python software, e.g. connecting the APIs of the individual modules together and potentially interacting with a database, or simply by running the individual modules in the command-line and connecting them as a pipeline. The CLI of each module reads delimiter-separated text files, configuration files and command-line arguments as input and writes text files as output. External software and extension scripts for analysing, visualising or experimenting with the data can easily import the simple data format. Furthermore, different algorithms can be experimented with without difficulty; any module can be replaced with another version that uses a different algorithm or that executes the existing module with different parameters, without having to recompute the output of the preceding module.

The division of the modules follows the format of the steps in Chapter 4. Consider Figure 5.5, which depicts the data flow of a pipeline that connects the different modules together. The module `carrier_detect.py` takes the raw 8-bit, I/Q-interleaved samples from the RTL-SDR, splits it into fixed-length blocks of data (Section 4.4.1), and checks for the presence of a carrier (Section 3.5). When a carrier is detected in a block of data, a space-delimited line of text is given as output, containing the block identifier, a timestamp, and the samples in the block encoded with base64[1]. The input data can be provided as a binary file with the raw samples or as a stream of bytes from the standard input stream, which can be real-time data from the RTL-SDR being streamed by the `rtl_sdr` command-line tool. Likewise, the output data can be written to a file which, by convention, has the `.card` file extension (short for *car*rier *d*etection), or it can be streamed directly to the next module by connecting the standard output stream of the carrier detection module to the standard input stream of the next module (typically referred to as *piping*). The carrier detection module also takes a configuration file as input, *detector.cfg* by default, which contains settings such as the block length, history length, and detection threshold. Note that the modular design follows the Unix philosophy of software development.

The next module, `detect.py`, takes the carrier detections as input and, for each carrier detection, compensates for the frequency offset of the carrier (Section 4.4.2), determines whether

[1] When the carrier detection is written to a file, the binary sample data from the SDR is encoded with base64, an encoding for translating binary data into printable ASCII characters. This allows each carrier detection to be stored in a text file, one per line, without the problems involved with delimiting or detecting boundaries of binary data. Faster solutions exist, but the simplicity and versatility of a text file is beneficial for working with the data during laboratory experiments.

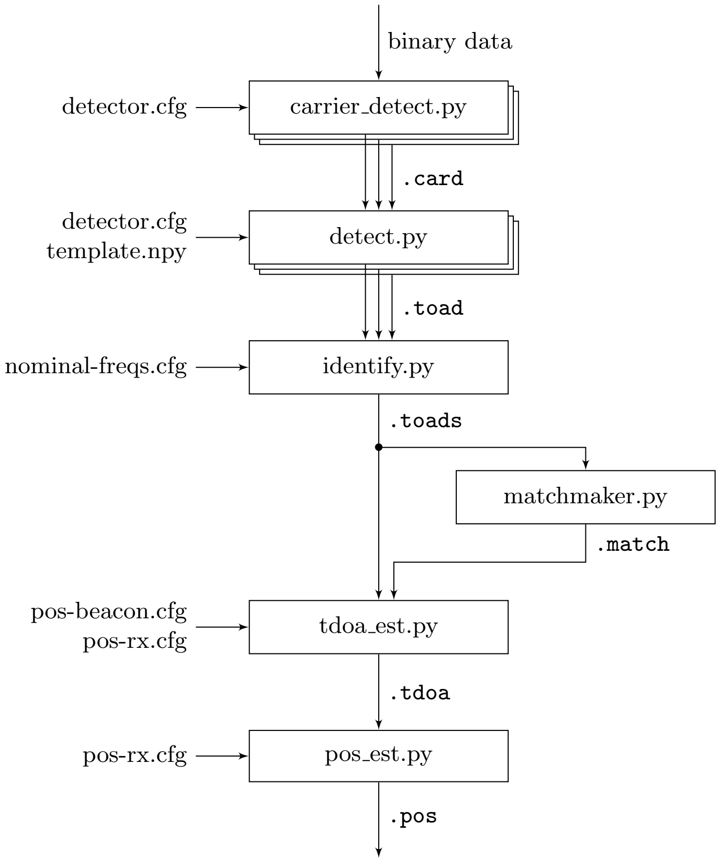


Figure 5.5: The flow of data through a pipeline of software modules connected together through their CLI interfaces, from the raw SDR samples up to position estimates.

the positioning signal is present, and estimates the SOA if it is present (Section 4.4.3). If a positioning signal is present, a line of space-delimited values describing the detection is provided as output. Information such as the receiver identifier, coarse block timestamp, estimated SOA, estimated carrier frequency, and the estimated signal-to-noise ratio of the correlation peak and the carrier are contained in the output. The output is conventionally written to a `.toad` file (short for *TOA d*etection).

The `identify.py` module takes one or more `.toad` files as input, usually one per receiver, as well as a configuration file with the nominal frequencies of the transmitters. It merges the detections, identifies the transmitter IDs of each detection based on the carrier frequency, and removes duplicate detections (Section 4.5.2). The detections are sorted by their timestamps and then written to a `.toads` file, which has the same format and information as a `.toad` file except that the transmitter ID is being added.

Next, the `.toads` file is passed to the `matchmaker.py` module. This module matches detections from different receivers that can be attributed to the same transmission based on the coarse timestamps reported by the receivers (Section 4.5.3). For each group of detections that matches, a space-separated list of the detection indices in the group — that is, the line numbers in the .toads file that belong together — are given as output. This is conventionally written to a `.match` file.

The `.match` file is then passed, together with the `.toads` file, to the `tdoa_est.py` module. The `tdoa_est.py` module estimates the TDOA values of mobile unit transmissions by relating the SOA values of different receivers to each other by means of a model built from beacon detections (Section 4.5.4). For each group of detections from the same mobile unit transmission, the TDOA values are estimated for all combinations of receivers that detected the transmission. The difference in arrival time (TDOA) of a mobile unit transmission at two receivers is estimated by first collecting the beacon transmissions that were detected by both of these receivers within a limited time frame from the mobile unit detection. The beacon detection data is then used to fit a quadratic model for relating SOA values from one receiver to another. Using this model and the SOA values of the mobile unit detection, the TDOA value is estimated. For each TDOA estimation, the TDOA value, the IDs of the two receivers that are involved, the coarse timestamp of the detection group, the transmitter ID of the mobile unit, the estimated quality of the mobile unit detection, and the estimated quality of the model are outputted as a row of space-delimited values. In addition to the `.match` and the `.toads` data files, the module also takes two configuration files as input: one containing the coordinates of the receivers (`pos-rx.cfg`) and the other the coordinates of the beacons (`pos-beacon.cfg`). These coordinates are used to calculate the distances between the receivers and the beacons in order to compensate for the difference in the time at which beacon transmissions arrive at different receivers.

Finally, the output of `tdoa_est.py`, which is conventionally written to a `.tdoa` file, is given as input to the `pos_est.py` module, which, for each set of TDOA estimates from the same detection group, estimates the position of the mobile unit (Section 4.5.5) and outputs the mobile unit ID, the coarse timestamp of the detection group, the coordinate of the estimated position, and the DOP at the estimated position as a row of space-delimited values. The output represents “raw” position estimates and can, if necessary, be passed to another module which improves the position estimates by averaging over multiple estimates or by applying path-based filtering and smoothing algorithms. The unfiltered “raw” position estimates are used for the results in this dissertation.

5.3.2 Tools

In addition to the main modules that are described above, various utility modules and scripts were also coded. For example, `chip_rate_search.py` estimates the chip rate of a positioning signal by searching for the chip rate that maximises the amplitude of the correlation peak for a given block of captured data. The fine-tuned chip rate estimate is used by another utility, `template_generate.py`, to generate an ideal template signal from a Gold code sampled at the given sample rate. The utility `template_extract.py` extracts a template from captured data by using a base template, usually the generated template, to extract the positioning signal with the best SNR. A pilot experiment showed that SOA estimation with a captured template provides a TDOA estimate with a standard deviation that is between 6 % and 15 % lower than the same measure when an ideal template signal is being used.

Various analysis tools were coded to inspect and visualise data at different levels of granularity and different stages of the software pipeline. For example, `detect_analysis.py` applies the signal processing algorithms for detecting a positioning signal and plots the discrete signals at different stages of signal processing. It provides figures such as those in Section 6.1, which proved to be useful for troubleshooting and verification. Another module, `toads_analysis.py`, takes a `.toads` file as input and provides an aggregate view of the detection data with metrics such as the mean carrier and correlation SNR of detections from different transmitters at the respective receivers. It can also be used to create figures for visualising different aspects of the detection data, such as the carrier frequency or SNR over time, the number of detections that were detected for each transmitter at each of the receivers over time, and the distribution of the SOA offsets.

The modules, especially the main modules, are tested individually with unit tests to verify that the implementation is correct and to ensure that they do not regress when changes are applied.

5.3.3 Fast carrier detection

The Python implementation of the receiver signal processing algorithms is fast enough to perform detections in real-time on a PC but falls behind on older or less capable hardware such as a netbook or Raspberry Pi. The Python implementation allowed for easy experimentation and changes during initial pilot tests. The slow speed was not a problem since powerful hardware was used for signal processing. A PC was used as a receiver station, or the raw samples were captured to a file on less capable hardware and processed offline on faster hardware. However, it became impractical to keep using this setup for subsequent tests, and a faster solution was required that could be executed on a netbook or a Raspberry Pi.

The most time-critical module is `carrier_detect.py`. Carrier detection needs to be performed on every block of data without falling behind, whereas subsequent signal processing modules are only applied to the subset of blocks that have made it through the carrier detection filter. The time-critical carrier detection module was reimplemented in C to enable real-time carrier detection on less powerful hardware.

The C carrier detection software, code-named *fastcard*, uses librtlsdr [46] to interact with the RTL-SDR and to stream samples from the RTL-SDR without the overhead of a pipe. The samples are added to a circular buffer with the producer, which interacts with the RTL-SDR, and the consumer, which processes the data, running on separate threads. This ensures that a temporary delay caused by the consumer does not block the producer thread and result in samples being dropped by the RTL-SDR. The consumer divides the data into fixed-length blocks and calculates the FFT for each block of data using FFTW [53], a free and fast implementation of the FFT algorithm with optimisations for various SIMD instruction sets. Offloading the FFT calculation to the Raspberry Pi's GPU is also supported, which is useful for carrier detection on a Raspberry Pi 1. The presence of a carrier signal is determined based on the peak magnitude of the FFT within the expected window of frequencies and the noise power estimated from the block of data. An optional lightweight library with mathematical routines optimised with SIMD instructions, the *Vector-Optimized Library of Kernels* (VOLK) [54], is used to speed up some of the mathematical operations. The C carrier detection software produces a `.card` file and is thus a drop-in replacement for *carrier_detect.py.*

Replacing only the carrier detector with a fast C implementation allows for real-time carrier detection while the rest of the signal processing, which is subject to a relaxed real-time constraint, is carried out in a high-level language for easy experimentation. With this configuration, signal processing can be performed in real-time on a Raspberry Pi 3 if there are less than 20 detections per second. A throughput of 20 detections per second is equivalent to 20 transmitters each transmitting once a second if no duplicate detections are produced in blocks of data prior to or following the primary detection. This low throughput is sufficient for small-scale tests, but impractical for a production environment. To obtain a higher throughput, faster hardware should be used or the detection software should be replaced with a faster implementation. The latter approach was followed, as described next.

5.3.4 Fast detector

Once a stable and feasible design had been established using the Python implementation, the detection software was reimplemented in C++ to demonstrate the feasibility of real-time signal processing on a Raspberry Pi 3. The C++ implementation, code-named *fastdet*, performs end-to-end signal processing from reading the raw samples from the RTL-SDR up to writing the detection information to a `.toad` file. It is a drop-in replacement for both *carrier_detect.py* and *detect.py*. It links to *fastcard* to reuse the data input and carrier detection functionality that were implemented in C. The C++ signal processing software is fast enough for real-time signal processing in a production environment, as detailed in Section 6.4.1.

5.4 Prototype positioning system

During development, measurements were taken with the RTL-SDR connected to laptops, and algorithms were experimented with on captured data. After the pilot tests, a prototype positioning system with four receiver stations was constructed to test the performance and feasibility of a system with inexpensive receiver hardware as a whole.

5.4.1 Overview

Each receiver is composed of three inexpensive and generally available mass-produced devices: a Raspberry Pi 3, an RTL-SDR dongle and a USB wireless broadband (3G/HSPA+) modem. A photograph of the hardware for four receiver stations is displayed in Figure 5.6. Each receiver station is also equipped with a preamplifier, a 5 dBi omni-directional antenna, a 65 Wh battery and a 20 W solar panel. A photograph of one of the receiver stations is shown in Figure 5.7.

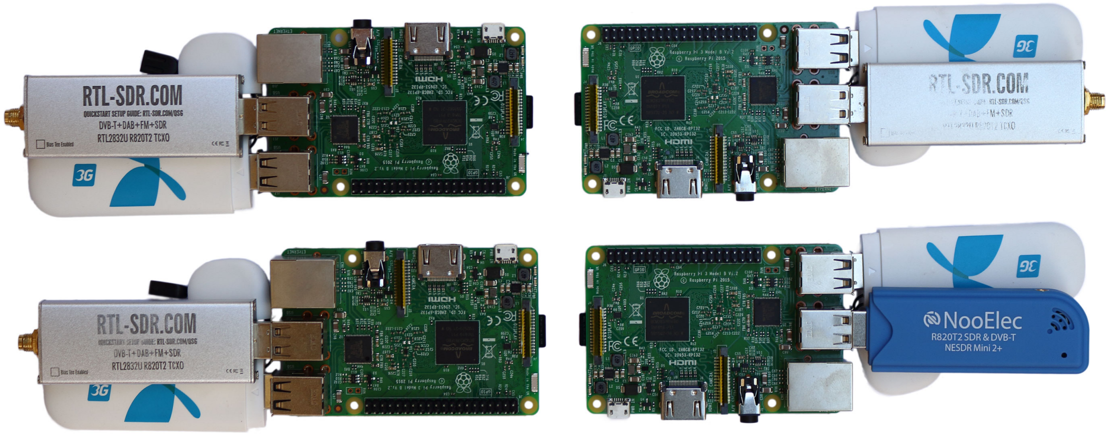


Figure 5.6: Off-the-shelf hardware for four receiver stations. Each receiver consists of a Raspberry Pi 3, an RTL-SDR and a wireless broadband modem.

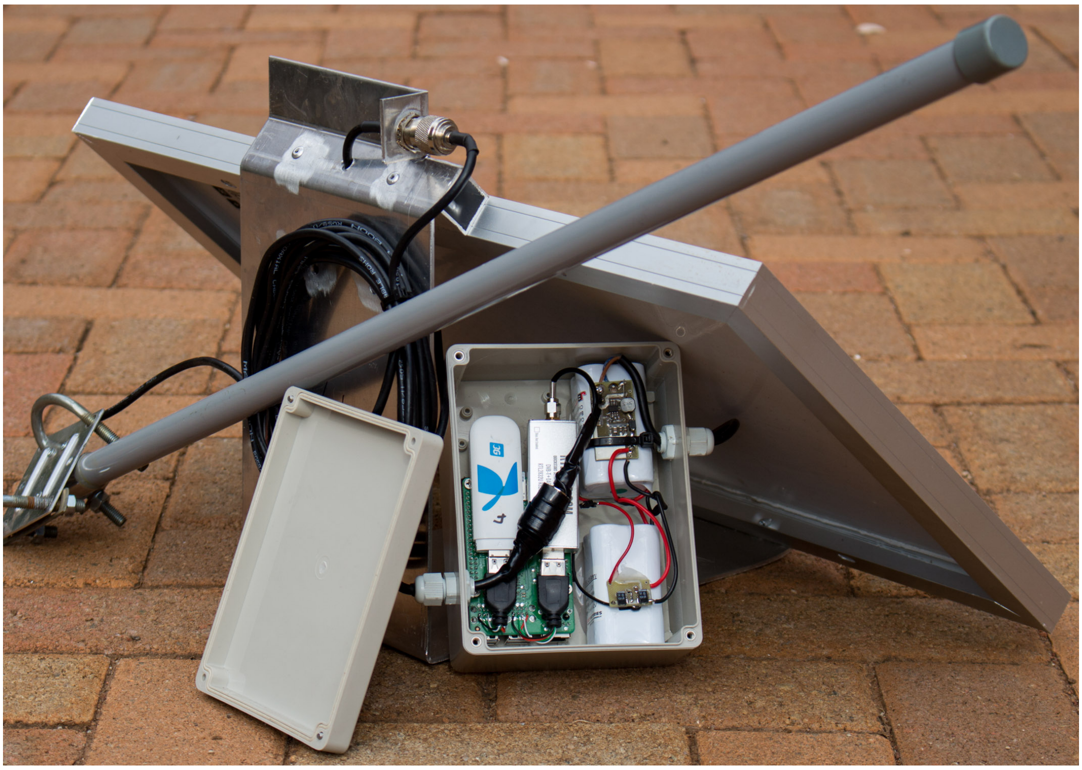


Figure 5.7: Photograph of a receiver station with the lid of the enclosure removed to display the Raspberry Pi, RTL-SDR, mobile broadband modem and power supply that are mounted inside the enclosure. A solar panel is located behind the enclosure.

The cost of the core hardware modules — the Raspberry Pi and an SD card, the RTL-SDR and the mobile broadband modem — amounts to approximately $100 per receiver station. A positioning system with four receiver stations can thus be put together for less than $800 when a maximum of $100 is spent on each receiver station for auxiliary hardware such as batteries, solar panels, antennae and enclosures.

5.4.2 System software

Debian Jessie is used as operating system on the Raspberry Pi in the form of the Raspbian Jessie Lite image. The system is configured with services for synchronising time, managing the execution of the detection software, uploading detections to a central server, and for remote management.

Time synchronisation

To allow detections at different receivers to be matched, a coarse timestamp relative to a common time base is reported together with each detection. The Raspberry Pi has no hardware clock, but NTP servers on the Internet are commonly used to set the system clock after a restart. It is furthermore not sufficient to leave the clock free-running after an initial synchronisation.

The system clocks will drift apart if they are not continuously being disciplined.

NTP is used for coarse clock synchronisation at the receivers given that the receivers are already connected to the Internet. However, the high and unpredictable network delay over a mobile internet connection degrades the accuracy that can be attained with NTP. It was found that, when the time is synchronised with NTP via a mobile broadband modem, the system clocks of two Raspberry Pi devices can differ by up to 0.5 s. This difference is detrimental to detection matching since it causes an ambiguity when the transmitters transmit every 0.8 s. To resolve the ambiguity, a beacon transmitter was programmed to transmit two closely spaced positioning signals every 60 s. The positioning server can use this pair of detections, which represents a unique event every 60 s, to rectify timestamp offsets of up to 30 s.

In the future, a real-time clock (RTC) module can be attached to the Raspberry Pi to maintain reasonable clock synchronisation with intermittent internet connectivity or a GPS module can be attached for more accurate time keeping.

Detection service

A *systemd* service is installed that starts the detection software on startup after waiting for NTP time synchronisation, and that restarts the detection software automatically when the software fails.

Uploading detections

For the prototype system, a simple approach is followed for uploading the detections to a central server. A cron job runs *rsync* every ten minutes to upload the `.toad` file to the server. The *rsync* utility will securely and reliably transfer only the parts of the file that have changed, which is any new detections that have not been transferred yet. This ensures that all the detections will be transferred eventually even with unreliable internet connectivity. Even though this setup is acceptable for the prototype system, a more sophisticated approach with a database or data streaming service would be better suited for a production system.

Remote management

The mobile broadband modems connect to the Internet via a restricted Access Point Name (APN) without a public IP address and with all incoming ports blocked. To enable remote access to the receiver stations, a reverse SSH tunnel is configured on each receiver. On start-up or whenever internet connectivity is re-established, the receiver establishes an SSH connection to a server with a public IP address and configures reverse port forwarding. Connections to the server on the configured port will be forwarded to the receiver via the SSH connection.

Table 5.1: Parameters that were used during the test of the prototype positioning system.

Parameter	Value
Transmitter	
Code type	Gold code
Code length	2047 bits
Chip rate	1 MHz
Nominal carrier frequency	433.83 MHz + i × 2.5 kHz
Transmission period	0.8 s
Receiver: tuner	
Centre frequency	433.83 MHz
Sample rate	2.4 MS/s
Receiver: signal processing software	
Block length (L_B)	16384 samples
Template length (L_T)	5205 samples
History length (L_H)	5210 samples
Detections processing software	
Matchmaker window	0.2 s
Beacon window for TDOA estimations	8 s

5.4.3 Summary of parameters

The parameters that were used for testing the prototype positioning system are outlined in Table 5.1. Unless specified otherwise, these are also the parameters that were used for the other experiments and pilot tests that are described in Chapter 6.

5.5 Chapter summary

In this chapter, a proof-of-concept implementation was presented for the design in Chapter 4.

An existing tag device that is used for various animal tracking and remote sensing applications was modified to transmit a wideband signal for TDOA positioning. The spreading code is modulated onto a carrier with OOK. A high baud rate is achieved by toggling the power amplifier's supply voltage from an output pin of an MCU. An external oscillator was connected to the MCU to ensure an accurate chip rate. It is approximated that the transmitter consumes about three orders of magnitude less energy for a position estimate than what would have been consumed if GPS were used to determine the position.

Two generally available, low-cost, general-purpose devices are used to receive and process positioning signals; a $20 RTL-SDR is used as RF receiver and a $35 Raspberry Pi 3 is used as signal processor. Furthermore, a low-noise amplifier is used to improve the sensitivity and selectivity of the receiver.

The essence of the positioning system lies in the software. The software enables specialised TDOA positioning on general-purpose hardware. What is sacrificed in hardware to enable a simple and low-cost receiver device, such as clock synchronisation, is compensated for in software.

The signal and detection processing software was implemented and experimented with in Python. The software features a modular design. The modules can be used and connected together from the command-line with each module reading its input from and writing its output to delimiter-separated text files. A fast implementation of the signal processing software was implemented in C/C++, enabling the software to be used on a Raspberry Pi for real-time signal processing.

Four prototype receiver stations were constructed. Each receiver station consists of a Raspberry Pi 3, RTL-SDR dongle, low-noise amplifier, wireless broadband modem, omni-directional antenna, batteries and a solar panel. The receiver station can be constructed for less than $200, amounting to $800 for a four-receiver positioning system.

The next chapter describes experiments that were conducted with the implementation that was discussed in this chapter.

Chapter 6

Results

Experiments that were conducted to test the design and the proof-of-concept implementation are discussed in this chapter. First, the signal processing algorithms are verified visually with a sanity check in Section 6.1. Simulations and sensitivity analyses that were performed to assess and compare different estimation methods with each other are presented in Section 6.2. After an outline of the methodology in Section 6.2.1, correlation peak interpolation methods are compared in Section 6.2.2, carrier peak interpolation methods in Section 6.2.3, and TDOA estimation techniques and parameters in Section 6.2.4. A pilot field test that was conducted to determine the accuracy and precision of the design and its implementation in real-world conditions is then described in Section 6.3. Tests that were conducted with the prototype receiver stations are discussed in Section 6.4, including the speed of the signal processing software on a Raspberry Pi, the power consumption of the stations and an integration test. Finally, the strategy that was followed to validate and verify the design and implementation is pointed out in Section 6.5.

6.1 Sanity check

In this section, a block of data that was captured during a field test is depicted at different stages of signal processing to qualitatively assess the performance of the signal processing algorithms and verify their correctness. The block of data was captured during a test in a rural area with the transmitter and the receiver about 2 km apart. The field test is described in more detail in Section 6.3.

Figure 6.1 shows the estimated spectral density of the signal in the form of a periodogram calculated from the FFT of the block of samples. A distinct narrowband peak with high spectral density can be observed near the centre, representing the carrier frequency.

The narrowband carrier peak within the FFT is shown in Figure 6.2. Figure 6.2a displays the carrier peak before carrier recovery, with a sinc-like function fit to the maximum-magnitude

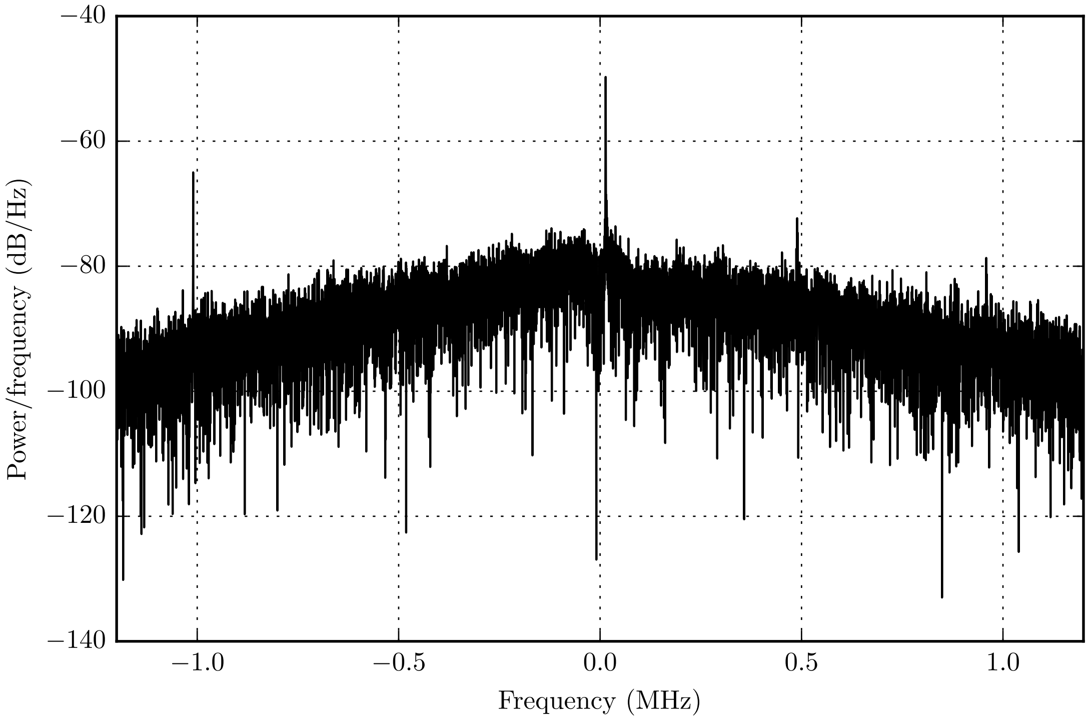
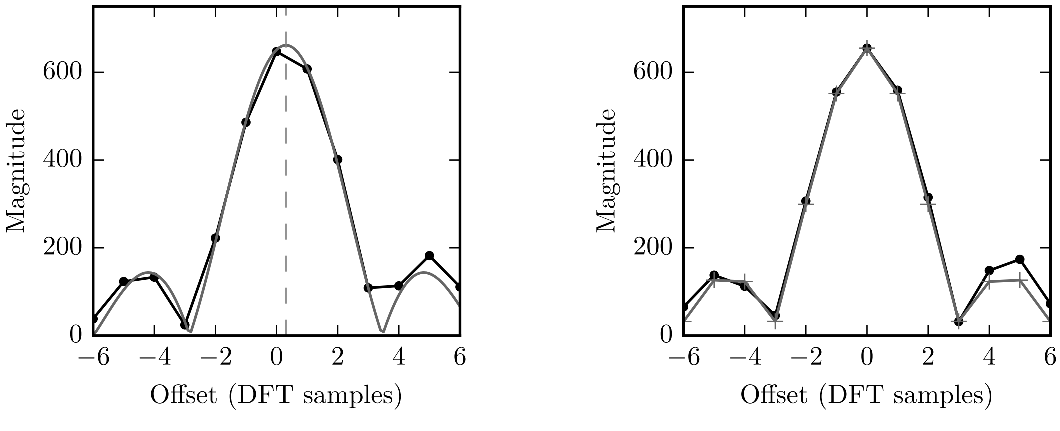


Figure 6.1: Estimate of the power spectral density (PSD) of the received signal before carrier frequency recovery.



(a) Before carrier recovery

(b) After carrier recovery

Figure 6.2: Peak samples of the FFT (black) with a sinc-like interpolation function superimposed (grey).

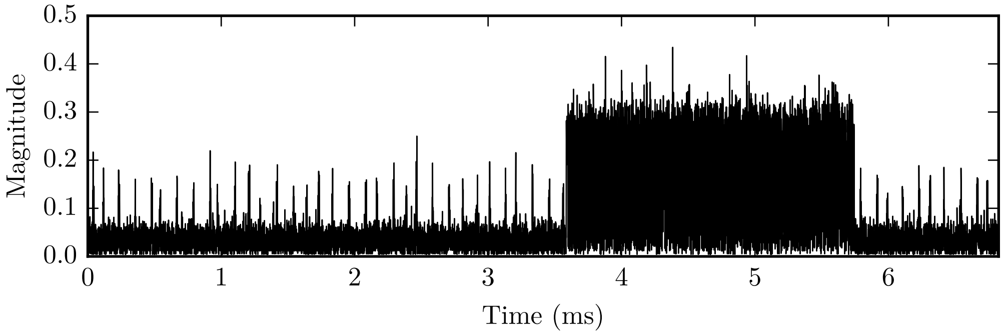


Figure 6.3: Magnitude of the time-domain samples after carrier frequency recovery.

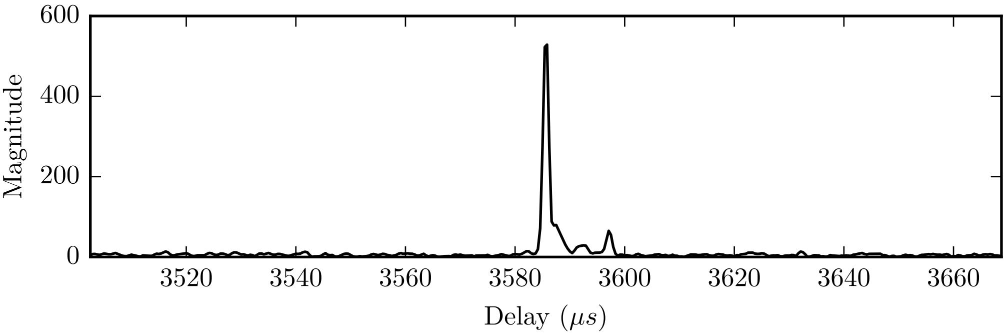


Figure 6.4: Extract from the correlator output showcasing the distinct correlation peak and a few multipath components.

sample and three samples on either side (Equation (4.17)). The carrier frequency is estimated from the position of the interpolation curve's peak. Figure 6.2b displays the carrier peak after compensating for the estimated carrier frequency, as well as the sinc-like interpolation function sampled at the same frequencies. The carrier peak follows the shape of the interpolation function very well, illustrating that the sinc-like function provides an appropriate model of the carrier peak. This also signifies that an accurate estimate of the carrier frequency was obtained from the interpolation peak, which verifies that the implementation of the carrier frequency estimation and carrier recovery algorithms is correct.

The magnitude of the time-domain samples after carrier frequency offset compensation is displayed in Figure 6.3. A positioning signal with a duration of about 2 ms is clearly visible above the noise level. Subsequent references to *received signal* in this section refer to the samples after carrier frequency recovery.

Figure 6.4 shows an extract from the cross-correlation of the block of data with a template of the expected positioning signal. Note the distinct peak at the delay where the template matches up with the received signal. Additional peaks with smaller magnitudes are also visible to the right of the primary peak, which can most likely be attributed to multipath propagation.

The correlation peak with a Gaussian curve fit to the maximum-magnitude sample and two

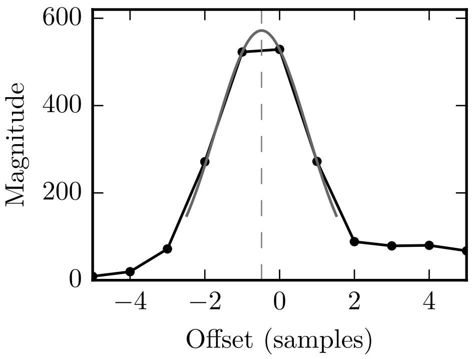


Figure 6.5: Gaussian curve fit to three samples surrounding the correlation peak.

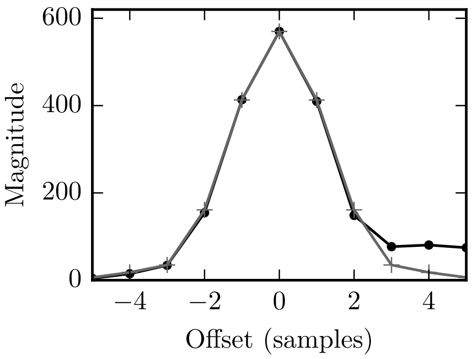
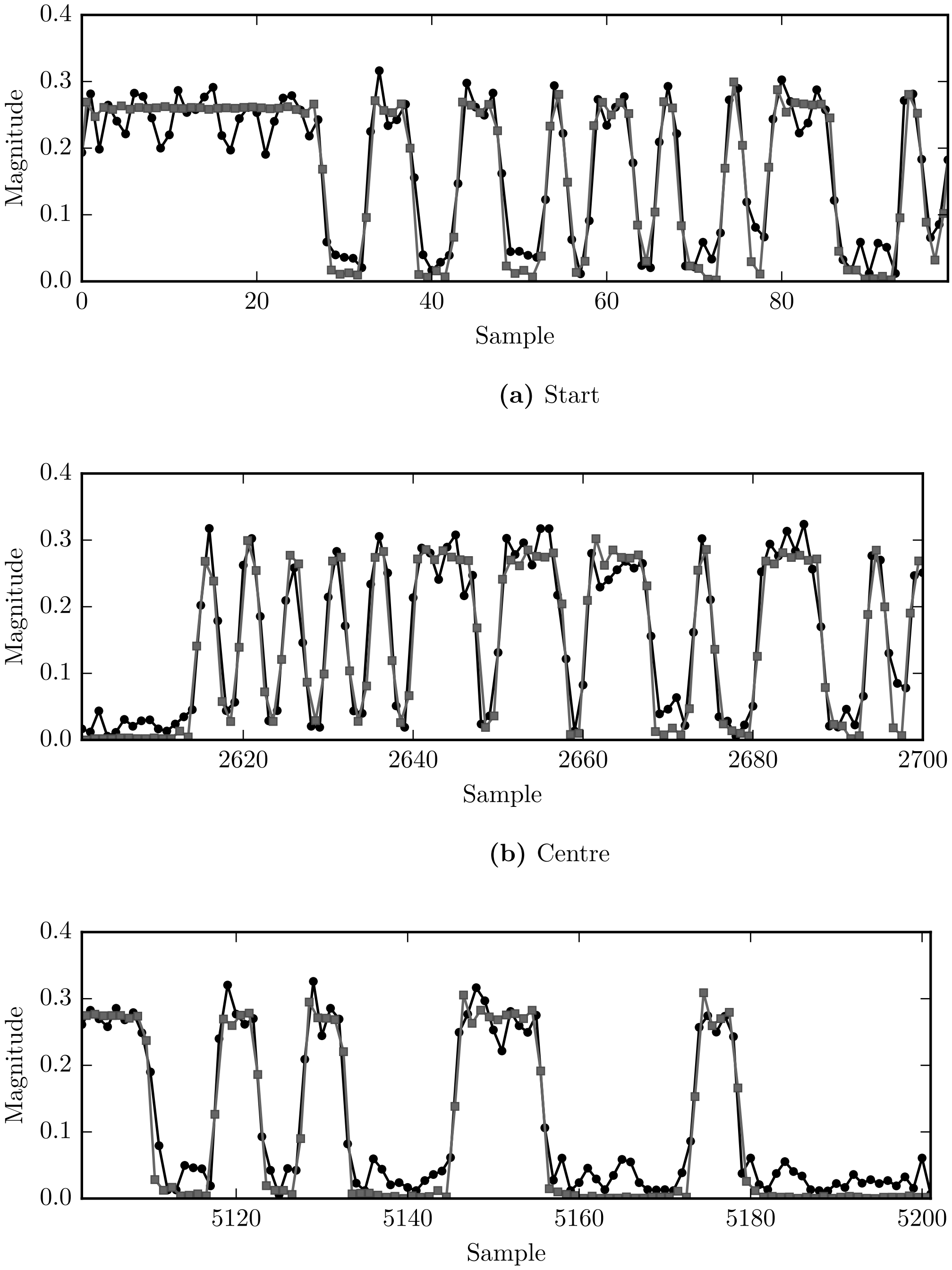


Figure 6.6: Comparison between the cross-correlation peak, time-shifted to align with the interpolation peak, and the template's autocorrelation peak.

adjacent samples is shown in Figure 6.5. The blunt peak showcases the limited resolution of the discrete-time samples caused by the fact that the signal delay is not an exact multiple of the sampling period and thus not synchronous to the sampling phase. A large approximation error will be introduced if the delay of the maximum-magnitude cross-correlation sample is taken as the arrival time. A better estimate is obtained through interpolation. The subsample arrival time is estimated from the position of the extremum of the Gaussian curve, which in this case is located about half a sample to the left of the peak sample.

Figure 6.6 displays the cross-correlation peak of the received signal resampled such that the sampling phase is synchronised with the estimated code phase of the positioning signal. The synchronisation is performed by shifting the signal a fractional number of samples using the time-shifting property of the Fourier transform. The FFT of the received signal is multiplied by a linear phase that is equivalent to a time shift equal to the subsample offset between the maximum-magnitude cross-correlation sample and the interpolation peak. The time-shifted signal is then the inverse FFT of the product. The autocorrelation peak of the template is also displayed in Figure 6.6, scaled in magnitude such that the maximum-magnitude samples of the autocorrelation and the cross-correlation signals are aligned. The synchronised cross-correlation peak is an almost perfect replica of the autocorrelation peak, signifying that the template is a good representation of the positioning signal and that the Gaussian interpolation peak provides a good estimate of the subsample delay.

Figure 6.7 shows three snippets of the received positioning signal with the template signal superimposed at the estimated arrival time. The template signal is translated vertically to transform it from a bipolar representation of the code to a unipolar representation, and scaled such that the energy of the template signal and the received code signal are equal. A snippet at the start (Figure 6.7a), centre (Figure 6.7b), and end (Figure 6.7c) of the code is displayed. It is evident that the template aligns very well with the received signal throughout the entire length of the positioning signal, showcasing that the chip rate of the template matches the chip



(a) Start

(b) Centre

(c) End

Figure 6.7: Template signal (grey) superimposed on the received positioning signal (black) at the estimated arrival time.

rate of the received signal, and evincing the accuracy of the arrival time estimation technique.

6.2 Assessment and comparison of estimation methods

It is evident from the design in Chapter 4 that several parameter estimation problems are involved from when the samples are taken right up to the final position estimate. For example, the discrete nature of digital signal processing requires interpolation techniques to mitigate the quantisation error introduced by the limited number of discrete-time samples. The maximum-magnitude peak of the DFT is interpolated to estimate the carrier frequency. Similarly, interpolation is applied to the samples of the cross-correlation peak to obtain a subsample estimate of the arrival time. Interpolation is also involved in the calibration of arrival time estimates from different receivers.

The invention or choice of estimation methods is key to accurate position estimation. The methods being employed have profound implications on the computational requirements of the detector and the accuracy of the final position estimate. As such, different methods were assessed and weighed up against each other.

6.2.1 Methodology

In this section, different estimation methods are evaluated on the basis of results from theoretical simulations and sensitivity analyses on sets of test data that were captured empirically. A sensitivity analysis is performed by replacing a single algorithm within the SDR or TDOA estimation software with an alternative method while the rest of the software and parameters remain intact. The software is then executed on the test data from carrier detection up to TDOA estimation, and the variance of the end result, the TDOA estimates, is calculated over time intervals in which the receivers and transmitters are all stationary. The values of the variance yielded by different methods for the same test data provide an indication of how the methods influence the precision of the eventual position estimate.

Road test data

Two sets of test data are used for the sensitivity analyses. *Road test data* refers to data that was captured during the experiment detailed in Section 6.3. During this experiment, two receivers were positioned about 9 km apart in a rural area, a beacon was placed at a fixed position in the middle between the two receivers, and a mobile transmitter was moved from the one receiver to the other, stopping for a few minutes at fixed positions along the way. For the sensitivity analysis, the variance of the TDOA estimates is calculated at each stop and combined to give rise to a mean variance. The results provide an estimate of the mean positioning precision in real-world conditions under different SNRs.

Laboratory test data

With the *road test data*, the variance is calculated over short periods of time. Another set of data, the *laboratory test data*, tests the accuracy and precision under excellent SNR conditions over a longer period of time to establish a lower bound on the expected positioning error. This test data was captured with two transmitters and two receivers collocated. To ensure that the ADC of the SDR does not saturate, quarter-wave whip antennas were connected to the receivers without any preamplifiers, the gain of the tuner was set to its lowest value (0 dB), and the transmission power of the transmitters was set to the lowest setting (1 dBm). The two SDRs were connected to the same PC, but their clocks were running independently. Each transmitter transmitted a positioning signal every 0.8 seconds. Data was captured over a period of 100 minutes, amounting to about 15 000 detections per receiver and about 7500 TDOA estimates.

Since the expected TDOA of the collocated transmitters is known to be zero, the biasing error of the estimator can be determined in addition to its variability. The standard deviation of the TDOA estimates is used as a measure of precision, i.e. a measure of the variability of the estimator. The mean of the TDOA estimates is used as a measure of accuracy, i.e. a measure of the bias of the estimator. For some of the tests, the Root-Mean-Square Error (RMSE) is calculated as a single measure that encompasses both the variability and the bias of the estimator.

Runtime

The resulting estimation error is not the only factor that should be considered when different methods are compared to each other. A method may be highly accurate, but too slow to adhere to the real-time constraints of the receiver. To compare the computational cost of the methods, the total runtime of the signal processing software was recorded during the sensitivity analysis tests. The runtime is just a ballpark figure of the computational cost. The algorithms have not been optimised, and the runtime represents the performance of naive implementations of the methods. The sensitivity analysis tests were carried out with the Python implementation of the signal processing and detection processing software. The software was executed on a single core of an Intel Core i7-3630QM CPU at a clock speed of 2.4 GHz.

Default methods for sensitivity analysis

Unless specified otherwise, the default algorithms that were used for the sensitivity analysis tests are:

- a least squares fit of a sinc-like function for carrier interpolation (Equation (4.17)),
- carrier frequency compensation using the frequency-shift property (Equation (4.23)),

- Gaussian interpolation to estimate the subsample arrival time from the correlation peak (Equation (4.49)), and
- a second-order polynomial fit to the SOAs of beacon transmissions that were detected within 8 seconds of the mobile unit detection to relate SOA values from different receivers to each other (Equation (4.75)).

6.2.2 Correlation peak interpolation

Six interpolation methods were evaluated in terms of how well they estimate the arrival time of a positioning signal from the correlator output. The methods are:

- *None*: do not perform any interpolation and simply use the delay of the maximum-magnitude correlation output as the arrival time estimate,
- *Parabolic*: fit a parabola to the three samples surrounding the correlation peak (Equation (3.4)),
- *Gaussian*: fit a Gaussian curve to the three peak samples (Equation (3.5)),
- *Cosine*: fit a cosine function to the three peak samples (Equation (3.6)),
- *Autocorr*: an improvised method by which the amplitude and delay of a scaled and time-shifted replica of the cross-correlation peak is fit to the template's autocorrelation peak using NLLS (Equation (3.13)),
- *Maximise*: use numerical optimisation to iteratively search for the delay that yields the maximum correlation peak amplitude when the time delay is compensated for with the time-shifting property of the Fourier transform (Equation (3.8)).

The methods were evaluated by means of a simulation and with a sensitivity analysis evaluated on two sets of test data.

Time-shift simulation

A simulation was performed to investigate and compare the theoretical performance of correlation interpolation methods under different SNR conditions. A template of a length-2047 Gold code with a chip rate of about 1 MHz sampled at 2.4 MS/s was used for the simulation. This template was extracted from the samples of a real positioning signal transmission that was captured with an RTL-SDR and is the same template that is being used for the detector's correlator. A simulated received signal is formed by transforming the template signal from a bipolar representation of the code to a unipolar representation, adding a time delay of between -0.5 and 0.5 samples, and adding zero-mean complex Gaussian noise to attain the desired

SNR. The non-integer time delay is performed with the time-shifting property of the Fourier transform; the FFT of the signal is calculated, it is multiplied with a linear phase, and then the inverse FFT of the product is calculated. The simulated received signal is then correlated with the original unshifted and uncorrupted template. The index of the maximum-magnitude output of the correlator is obtained and the subsample time delay is estimated with each of the interpolation methods. The actual time delay is subtracted from the estimated time delay to calculate the estimation error. The simulation is repeated 5000 times with the same SNR, but each time with a new random delay, distributed uniformly between -0.5 and 0.5 samples. The resulting estimation errors are collected and the RMSE at the given SNR is calculated for each interpolation method. This process is repeated for different SNRs, and the results are summarised in a graph of RMSE over SNR for each of the interpolation methods.

In addition to the RMSE of each of the interpolation methods, the Cramér–Rao Lower Bound (CRLB) is calculated at each SNR to compare the results to the theoretical lower bound. It is shown in [18] that if $s(t)$ is an OFDM signal consisting of complex-valued symbols S_n modulated onto N subcarriers with a subcarrier spacing of f_{sc}, represented by the equation

$$s(t) = \frac{1}{\sqrt{N}} \sum_{n=-\left\lfloor \frac{N-1}{2} \right\rfloor}^{\left\lfloor \frac{N-1}{2} \right\rfloor} S_n e^{j2\pi n f_{sc} t}, \tag{6.1}$$

then the CRLB of a time estimator that estimates the delay τ of $s(t-\tau)$ from N time-domain samples $\boldsymbol{m}$ is:

$$\mathrm{Var}[\hat{\tau}(\boldsymbol{m})] \geq \frac{\sigma^2}{8\pi^2 \sum_{n=-\left\lfloor \frac{N-1}{2} \right\rfloor}^{\left\lfloor \frac{N-1}{2} \right\rfloor} n^2 \left|S[n]\right|^2} \tag{6.2}$$

if the sample rate is $f_s = N f_{sc}$. This equation can be repurposed to calculate the CRLB for a discrete-time signal from its DFT by noting that Equation (6.1) represents the inverse DFT when the sample period is T_s and $f_{sc} = \frac{1}{T_s}$.

If $h[n]$ are the samples of the template, $0 \leq n \leq N-1$, and H is the DFT of h,

$$H[k] = \sum_{n=0}^{N-1} h[n] e^{-j2\pi kn/N}, \qquad -\left\lfloor \frac{N-1}{2} \right\rfloor \leq k \leq \left\lfloor \frac{N-1}{2} \right\rfloor \tag{6.3}$$

then the lower bound on the variance of an estimator that estimates the delay from the samples of a time-delayed AWGN-corrupted replica of the template is:

$$\mathrm{Var}[\hat{\tau}(\boldsymbol{x})] \geq \frac{\sigma^2 N^3 T_s}{8\pi^2 \sum_{n=-\left\lfloor \frac{N-1}{2} \right\rfloor}^{\left\lfloor \frac{N-1}{2} \right\rfloor} n^2 \left|H[n]\right|^2} \tag{6.4}$$

where the vector $\boldsymbol{x}$ represents the samples and where σ^2 is the noise power.

The results of the simulation are displayed in Figure 6.8. The root-mean-square of the estimation errors is shown as a function of the SNR for each of the interpolation methods. The

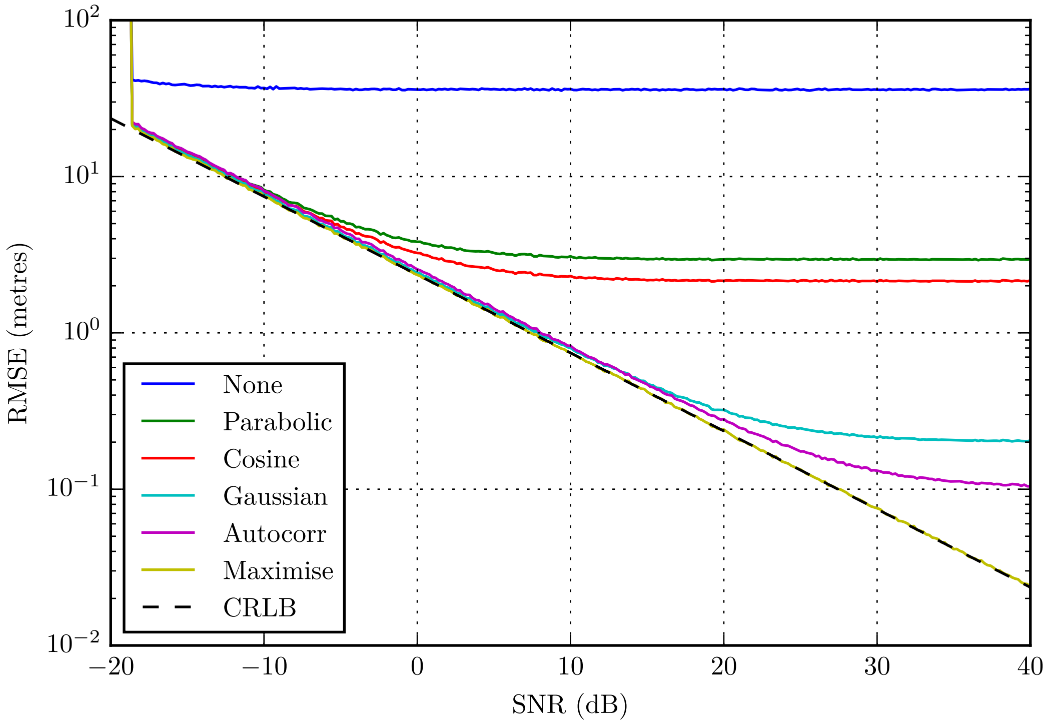


Figure 6.8: Comparison of RMSE of simulated time delay estimates at different SNRs for various correlation peak interpolation methods.

estimation error is expressed in metres, representing the distance light would travel during the time interval of the time-delay estimation error.

The interpolation methods perform equally poor at very low SNRs below −10 dB. When the SNR drops to below −19 dB, the correlation peak at the expected delay is overshadowed by noise of greater magnitude, and delay estimation fails completely. Each of the interpolation methods tracks the CRLB at very low SNR and performs increasingly better as the SNR is increased. The decrease in estimation error with an increase in SNR is not unbounded, however. The improvement in estimation error slows down and reaches an asymptote, revealing the minimum estimation error, i.e. maximum resolution, of the interpolation method.

As can be expected, the worst estimation error is obtained when no interpolation method is employed. Without interpolation, a constant RMSE of about 36 m is observed. This corresponds to the expected standard deviation of a delay, δ, with a uniform distribution between -0.5 and 0.5 samples, $\delta \sim \mathcal{U}(-0.5, 0.5)$:

$$\begin{aligned} \sigma_{X\sim\mathcal{U}(a,b)} &= \frac{1}{\sqrt{12}}(b-a) \\ \therefore \sigma_\delta &= \frac{1}{\sqrt{12}} \text{ samples} \\ \sigma_\tau c &= \sigma_\delta \times \frac{c}{f_s} = 36\,\text{m}. \end{aligned}$$

Parabolic interpolation performs the worst amongst the interpolation methods, exhibiting a lower bound at an RMSE of 3 m. Cosine interpolation performs marginally better, reaching its limit at 2.1 m. Gaussian interpolation performs vastly better, with an RMSE of about 0.2 m at 40 dB. The *autocorr* estimation algorithm performs slightly better than the Gaussian interpolation method with almost half the RMSE at 40 dB.

The *maximise* algorithm tracks the CRLB, which seems peculiar at first sight. A closer investigation reveals that this is to be expected. The *maximise* algorithm uses the same technique to shift the signal in time as is being used to create the simulated received signal from the template. The algorithm essentially searches for the time delay that would reverse the received signal to its original form, the template signal. It will not track the CRLB when a different technique is used to simulate the time delay, or when the received signal is not directly based upon the template.

Sensitivity analysis using test data

The simulation provides a good indication of how the interpolation methods weigh up against each other and what the lower bounds of their estimation errors are in different SNR conditions, but a simulation can never account for all the factors that could have an impact on the resulting estimation error. To determine and compare the accuracy of the interpolation methods in real-world conditions, a sensitivity analysis was performed using field test data. For each of the methods, the interpolation method of the signal processing software was replaced, and the signal and detection processing software was executed on the two sets of test data described in Section 6.2.1.

Table 6.1 shows the estimated precision and accuracy of the TDOA estimates resulting from each of the interpolation methods, as well as the runtime of the modified detection processing software. The precision and accuracy are estimated by means of the standard deviation and the mean error of the TDOA estimates. These measures are expressed in metres instead of

Table 6.1: Resulting estimated precision (standard deviation) and accuracy (mean error) of the TDOA estimates and runtime of signal processing software when the signal and detection processing software is executed with different correlation peak interpolation methods, evaluated on two sets of captured test data.

	Laboratory test data			Road test data	
Method	Mean (m)	Std. dev. (m)	Runtime (s)	Std. dev. (m)	Runtime (s)
None	−1.16	51.66	251	53.86	82
Parabolic	0.40	5.44	252	5.15	83
Cosine	0.44	4.15	253	4.36	82
Gaussian	0.52	0.90	253	3.49	80
Autocorr	0.54	0.11	433	3.45	126
Maximise	0.55	0.13	467	3.68	140

seconds to provide an indication of the positioning error than can be expected when the DOP is one. For example, a standard deviation of 3 m indicates that, most of the time, the difference in arrival time of a positioning signal between two receivers is estimated to within 10 ns from the mean, which is equivalent to 0.024 samples.

As with the simulations, the estimation error is large without any correlation peak interpolation. The TDOA estimation error without interpolation is almost 50 % greater than the corresponding delay estimation error in the simulation. This can be attributed to the fact that, whereas the error in the simulation represents the error of a single arrival time estimate, the TDOA estimate is the combination of multiple arrival time estimates. The errors of at least four arrival time estimates contribute to the total TDOA estimation error, namely one estimate at each receiver for the mobile unit's transmission and at least one estimate at each receiver for the beacon transmission.

The ranking order of the interpolation methods in terms of their performance is the same as the order that was observed from the simulation, except for the *maximise* algorithm that now ranks between cosine interpolation and Gaussian interpolation. Gaussian interpolation demonstrates the best precision of the three three-point interpolation methods. It exhibits a standard deviation of 0.9 m for the *laboratory test data* and 3.49 m for the *road test data*. The *autocorr* method outperforms all the other interpolation methods, especially under good SNR conditions. For the *laboratory data*, a standard deviation of 11 cm is attained, equating to a resolution of smaller than a thousandth of a sample. The *Gaussian* method and the *autocorr* method perform equally well on the *road test data*.

A biasing error is visible from the results of the *laboratory test data* for all the interpolation methods, with only slight variations from one method to the other. The biasing error can be attributed to the non-coherent phase of the receiver clocks. The phase of the correlation peak can be used in the future to compensate for the error and to improve the position estimates to a sub-wavelength accuracy.

As can be expected, there is little difference in the computational time between the three-point interpolation methods, i.e. parabolic, Gaussian and cosine interpolation. The iterative methods, *autocorr* and *maximise*, are significantly slower than the closed-form methods. However, with the *autocorr* method the subsample SOA estimation does not necessarily have to be performed on the receiver. A few samples near the correlation peak can be recorded with the rest of the detection information. The task of estimating the subsample SOA can then be offloaded to the server that aggregates and processes the detections from all the receivers.

Even though the *autocorr* method provides the smallest estimation error, it is significantly slower than *Gaussian interpolation* with the current implementation and only outperforms Gaussian interpolation under good SNR conditions. Gaussian interpolation was selected as the correlation interpolation method of choice for the prototype positioning system since it is fast, simple to implement and sufficiently accurate.

6.2.3 Carrier peak interpolation

With a block length of 16384 samples and a sample rate of 2.4 MHz, the DFT of the block of samples represents the spectral densities at discrete frequencies that are about 146 Hz apart. When the frequency of the maximum-magnitude sample of the DFT is taken as the carrier frequency offset, the maximum quantisation error will be only 73 Hz. For a positioning signal with a duration of $T_D = 2\,\text{ms}$, the maximum reduction in the signal's power due to the quantisation error is estimated as (Equation (3.29)):

$$20 \log_{10} (\operatorname{sinc} (T_D \, \Delta f)) = -0.31\,\text{dB}, \tag{6.5}$$

which is minuscule. Stated differently, the end of the positioning signal will be at most 53° out of phase with the start of the signal. It is thus expected that the quantisation error of the carrier frequency estimate will not have a significant impact on the TDOA estimate.

This subsection investigates if and to what extent subsample interpolation of the carrier peak or the lack thereof affects the accuracy of the consequential arrival time estimate when the DFT length is 16384 and the sample rate is 2.4 MS/s. Different interpolation methods are first compared with each other using a simulation. The effect of DFT interpolation on the TDOA estimate is then assessed with a sensitivity analysis.

The interpolation methods that are being evaluated are:

- *None*: estimate the carrier frequency offset from the integer-valued index of the DFT sample with the largest magnitude,
- *Parabolic*: fit a parabola to the three DFT samples surrounding the carrier peak (Equation (4.21)),
- *Gaussian*: like parabolic interpolation, but fit a Gaussian curve (Equation (3.5)),
- *Cosine*: like parabolic interpolation, but fit a cosine curve (Equation (3.6)),
- *Sinc*: use NLLS to fit a sinc-like function to a few samples surrounding the peak (Equation (4.17)).

Frequency-shift simulation

The frequency estimation errors of the carrier interpolation methods were evaluated with a simulation by applying each of the methods to a frequency-shifted template signal corrupted by complex-valued AWGN. The simulation was conducted in the same manner as the correlation peak interpolation simulation, except that the template is zero-padded to a length of 16384 samples, that the received signal is simulated by shifting the template in frequency by a random frequency with a uniform distribution between −0.5 and 0.5 DFT samples, and that, instead of

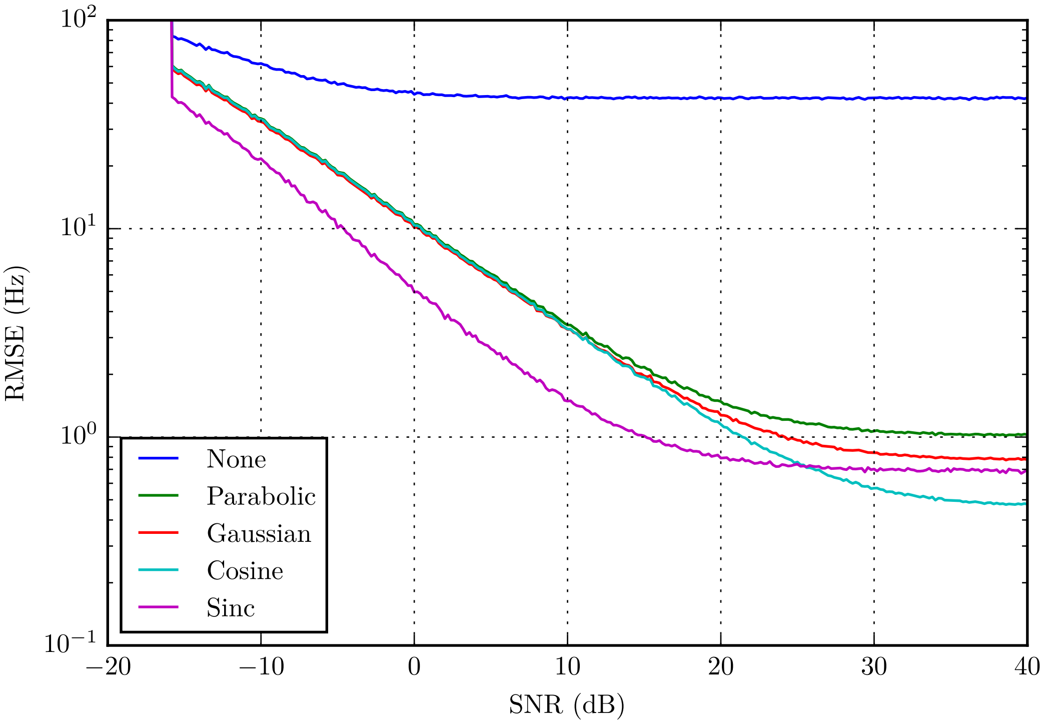


Figure 6.9: RMS of the frequency estimation error at different SNRs for various carrier frequency offset interpolation methods, resulting from a simulation by which a template signal is shifted in frequency by between −0.5 and 0.5 DFT samples.

estimating the time delay, the interpolation methods are used to estimate the carrier frequency offset from the samples surrounding the maximum-magnitude DFT peak. The simulation is executed over a range of SNR values, and for each SNR value, the RMSE is computed from 5000 estimates, each with a random frequency offset.

The simulation results are depicted in Figure 6.9. The RMS estimation error without any subsample interpolation is about 46 Hz, which corresponds to the uniform distribution. The three-point interpolation methods perform equally well under low SNR conditions. At high SNRs, parabolic interpolation reaches its minimum estimation error at about 1 Hz, Gaussian interpolation at about 0.8 Hz, and cosine interpolation at approximately 0.5 Hz. Sinc-like interpolation performs better than the three-point methods at low SNRs but reaches its minimum estimation error at about 0.7 Hz.

In summary, if the carrier frequency estimation error has a significant impact on the TDOA estimation, it is expected that, amongst the methods that were evaluated, sinc-like interpolation will perform the best when the SNR is below approximately 25 dB, and cosine interpolation slightly better when the SNR is above 25 dB.

Table 6.2: Estimated precision and accuracy of TDOA estimates resulting from various carrier frequency interpolation methods when evaluated on the test data.

	Laboratory test data			Road test data	
Method	Mean (m)	Std. dev. (m)	Runtime (s)	Std. dev. (m)	Runtime (s)
None	0.58	0.90	195	3.58	64
Parabolic	0.52	0.90	195	3.49	64
Gaussian	0.52	0.90	195	3.49	64
Cosine	0.52	0.90	194	3.49	64
Sinc	0.52	0.90	250	3.49	82

Sensitivity analysis using test data

The simulation demonstrates the carrier frequency estimation error of the interpolation methods, but we are primarily interested in the error of the resulting arrival time estimate. Unlike correlation peak interpolation where the interpolation technique affects the arrival time estimate directly, the carrier offset affects the arrival time estimate only indirectly.

Table 6.2 displays the resulting standard deviation and mean error of the TDOA estimates of the two sets of test data for each interpolation method. Note that the estimated accuracy and precision of the TDOA estimates are almost identical regardless of which interpolation method is used and regardless of whether interpolation is applied or not. In comparison with no subsample interpolation, there is a negligible improvement of 6 cm in the mean error of the *laboratory test data* and a small improvement of about 0.1 m in the standard deviation of the *road test data* when an interpolation method is applied. No differences are visible between the accuracy and precision of the different interpolation methods.

Together with the carrier frequency offset estimation technique, it is also necessary to consider the technique that is used to compensate for the frequency offset. Three techniques for frequency offset compensation were outlined in Section 4.4.2:

- *integer*: to ignore the subsample carrier frequency offset and shift the DFT by an integer number of samples (Equation (4.22)),
- *time-domain*: to compensate for the frequency offset in the time-domain using the frequency-shift property of the Fourier transform (Equation (4.23)), and
- *precompute*: to use a newly devised technique involving a set of precomputed frequency-shifted correlation templates (Equation (4.24)).

Like the interpolation method, the frequency offset compensation technique can have an impact on the accuracy of the TDOA estimate, but more important is the technique's impact on the runtime of the signal processing software.

Table 6.3 exhibits the results of a sensitivity analysis of the compensation techniques on the

Table 6.3: Estimated precision and accuracy of TDOA estimates and runtime of signal processing software for different frequency shift algorithms.

	Laboratory test data		Road test data	
Method	RMSE (m)	Runtime (s)	Std. dev. (m)	Runtime (s)
Integer	1.07	89	3.58	28
Precompute	1.04	91	3.49	29
Time-domain	1.04	194	3.49	64

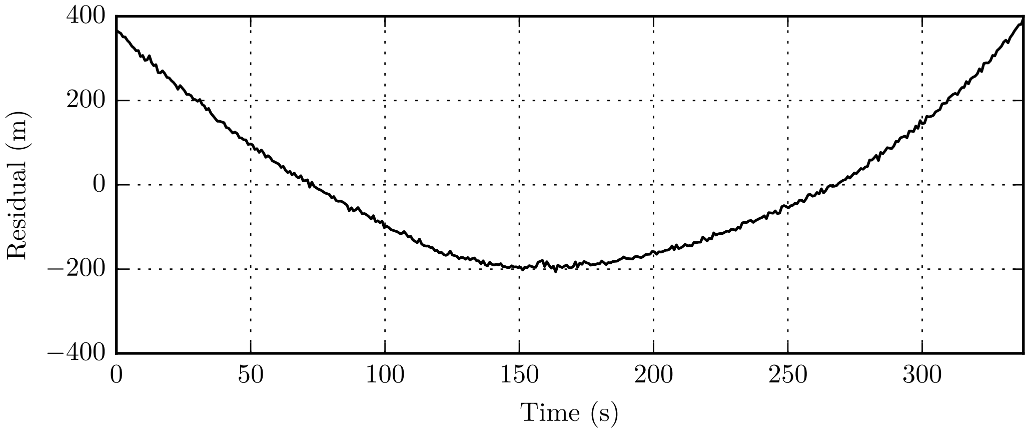
resulting error and standard deviation of the two sets of test data's TDOA estimates, as well as the runtime of the signal processing software. Parabolic interpolation is used to estimate the subsample carrier frequency. A set of 11 templates is used for the *precompute* method.

Compensating for the frequency offset in the time-domain has substantial implications on the runtime of the signal processing software, mostly due to an additional DFT computation. For both the *laboratory* and the *road test data*, the runtime of the software is more than twice as long when the *time-domain* technique is being used instead of the *integer* shift technique. The *precompute* technique exhibits the same improvement in the error and variance of the TDOA estimates as the *time-domain* technique, but it does so without undermining the runtime performance of the software.

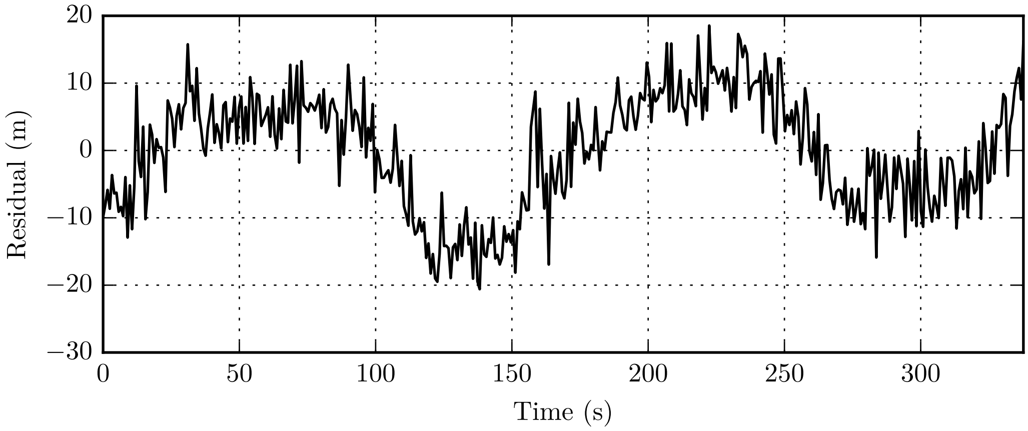
In summary, with a DFT of length 16 384 and a sample rate of 2.4 MS/s, the effect of the DFT quantisation error on the TDOA estimate is unsubstantial. Subsample carrier frequency estimation and compensation are not worth the extra computation time of frequency compensation in the time-domain or the added code complexity of a set of precomputed templates. The quantisation error may, however, be significant for different parameter values, e.g. if the length of the DFT is shorter, or when the phase information is incorporated to obtain sub-wavelength accuracy. In those cases, the *precompute* technique will be valuable to mitigate the quantisation error without additional computational complexity.

6.2.4 TDOA estimation

For a given transmission from a transmitter, each receiver reports the TOA as an SOA value, which is relative to the first sample of that receiver. The clocks of the receivers are free-running and independent, resulting in variations in the sample rates of the receivers. To calculate the TDOA, it is necessary to relate the SOA values of one receiver to another. As detailed in Section 4.5.4, beacon transmissions are used for this purpose. To mitigate the effect of changes in the receiver's sample rate that occur between the beacon detection and the mobile unit detection, we interpolate between the SOA values of multiple beacon detections. In this subsection, the assumed model of the clock error and the performance of the TDOA estimation techniques that were derived in Section 4.5.4 are analysed and validated.



(a) Residuals of a linear function fit to the scatter plot, showcasing the large error even after estimating and compensating for a constant clock offset.



(b) Residuals of a quadratic function fit to the scatter plot, showcasing the error after estimating and compensating for a linear clock drift in addition to a constant clock offset.

Figure 6.10: Residuals of a linear and quadratic function fit to a scatter plot of the SOA values of beacon transmissions at one receiver against the corresponding SOA values at another receiver for an extract of the road test data.

Behaviour of receiver clocks

We first investigate the behaviour of the sample clocks of two receivers relative to each other. The behaviour is difficult to analyse directly, so we analyse it indirectly by investigating the relationship between the SOA values of beacon transmissions at one receiver and the corresponding SOA values at another receiver.

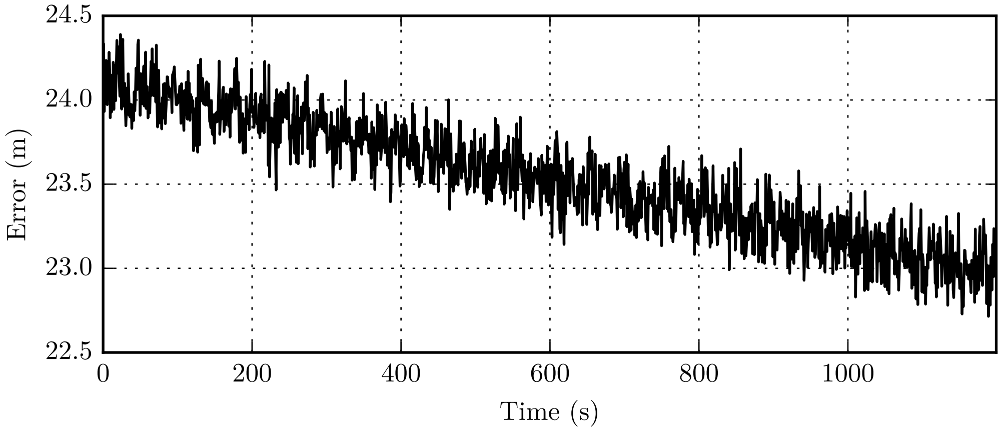
Figure 6.10a displays the residuals after a linear function has been fit through a scatter plot of the SOA values at one receiver against the corresponding SOA values at another receiver for an extract of the beacon detections from the *road test data*. The residuals represent the errors after estimating and compensating for a constant offset between the sample rates of the two receivers. The residuals would have been zero if a linear function modelled the relationship perfectly. Even though the residuals are in terms of samples, they are expressed in metres as the distance light would travel in air. It is evident from Figure 6.10a that a linear function is not a good model for relating the SOA values. A clock drift is clearly visible that results in large residuals when a linear relationship is assumed.

Figure 6.10b is similar to Figure 6.10a, except that the residuals are those of a quadratic function that is fit to the data instead of a linear function. The residuals represent the errors after estimating and compensating for a constant offset and a linear drift between the sample rates. The relatively small magnitude of the residuals proves that the difference in SOA values can primarily be attributed to a constant offset and linear drift between the sample rates of the two receivers, which substantiates the use of a second-order model function in Section 4.5.4. Even though the magnitudes of the residuals are much smaller than in the linear model, a time-varying change in the sample rate that can be attributed to neither a constant offset nor a linear drift is still visible. For this extract of data the residuals in Figure 6.10b resemble the shape of a cosine function, but this is not always the case. The residual errors validate the observation in Section 4.5.4 that the beacon detections that are used for relating SOA values should be within a limited time frame from the mobile unit detection to minimise the residual error.

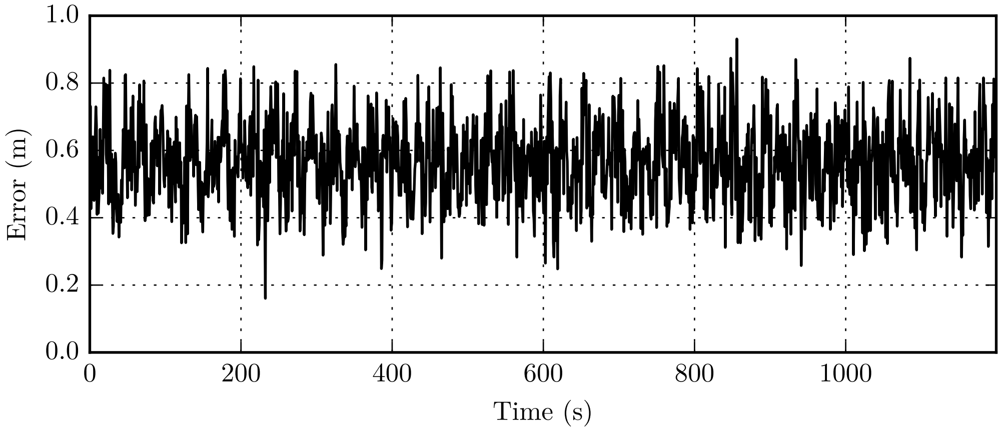
Interpolation between beacon detections

A naive approach for calibrating the SOA values of a mobile unit detection is to do so relative to the closest beacon detection. Figure 6.11a shows the error of the TDOA estimate for an extract of the *laboratory test data* when one of the transmitters is used as a beacon and the other as a mobile unit, and when for each mobile unit transmission, only one beacon transmission, the closest beacon transmission, is used for relating SOA values (Equation (4.54)). With a transmission period of 0.8 s, the maximum time interval between a mobile unit transmission and the beacon transmission is 0.4 s. It is evident from Figure 6.11a that, as predicted in Section 4.5.4, the divergence of the receiver clocks is significant enough in this short period of time to exhibit a large TDOA estimation error — about 23 m in this case. Also visible from the figure is that the error changes gradually over time, which can be attributed to the gradual change in the time interval between the beacon transmission and the mobile unit transmission due to independent clocks at the transmitters.

In Section 4.5.4 it was shown that the error can be reduced by interpolating between the SOA values of more than one beacon transmission. Figure 6.11b displays the error of the TDOA estimates when the SOA values of the mobile unit detections are calibrated using linear interpolation between the two closest beacon detections (Equation (4.68)). Note that the interpolation method cuts down the estimation error significantly.



(a) TDOA estimation errors when, for each mobile unit transmission, the beacon transmission that was detected closest to the mobile unit transmission is used for relating the SOA values from different receivers.



(b) TDOA estimation errors when interpolating between the two closest beacon transmissions.

Figure 6.11: Comparison of the TDOA estimation errors for an extract of the laboratory test data when only one beacon detection is used to relate SOA values, in contrast with the errors when interpolating between two beacon detections.

Table 6.4: Standard deviation and mean error of TDOA estimates that are calculated from captured test data using different methods for calibrating SOA values with beacon detections.

	Laboratory test data		Road test data
Method	Mean (m)	Std. dev. (m)	Std. dev. (m)
Nearest	24.78	4.59	9.23
Linear	0.55	0.12	4.88
LS fit: linear	0.70	0.49	3.40
LS fit: quadratic	0.54	0.11	3.45
LS fit: cubic	0.54	0.11	3.49

Another method is to use least squares to fit a model for calibrating SOA values to multiple beacon transmissions that are chronologically close to the mobile unit detection (Equation (4.75)), and to then evaluate the model at the SOA of the mobile unit (Equation (4.77)). Note that this is also a form of interpolation. Using more than two beacon detections for interpolation helps to average out estimation errors.

Table 6.4 displays a comparison of the TDOA estimation methods in terms of the resulting mean error and standard deviation of the TDOA estimates when the detections from the *laboratory* and the *road test data* are used. *Nearest* refers to the use of a single beacon transmission, the one closest to the mobile unit transmission, for relating SOA values (Figure 6.11a). *Linear* refers to linear interpolation between the two closest beacon transmissions (Equation (4.68)). *LS fit* refers to an LS fit of a polynomial model function to the SOA values of beacon transmissions that were detected within 10 seconds from the mobile unit transmission (Equation (4.75)). The LS fit is conducted with a linear, quadratic, and cubic model function. The *autocorr* interpolation method was used to estimate the SOA values of the detections from the test data.

As expected, the use of only a single beacon detection results in large estimation errors. The estimated accuracy and precision of the TDOA estimates are significantly better when interpolating between two beacon transmissions. With the *road test data*, even better results are obtained, since SOA estimation errors of the beacon detections are evened out with an LS fit over multiple detections. Due to the high SNR and low estimation error, using more beacon detections does not produce significantly better results with the *laboratory test data*, but rather increases the error if the fit function does not model the clock error appropriately, as observable from the results of the LS fit of a linear function. With the *road test data*, the degree of the polynomial does not influence the standard deviation of the TDOA estimates significantly.

Window size

The size of the window over which beacon detections are taken is a parameter that needs to be balanced. The larger the window, the more observations there are to average out the

Table 6.5: Standard deviation of the TDOA estimates for different window sizes when a quadratic model is fit to the SOA values of the beacon transmissions. The window size is the maximum time difference between a mobile unit transmission and the beacon transmissions that will be used to build a model for relating SOA values.

Window size (s)	Laboratory test data (m)	Road test data (m)
2	0.11	4.55
4	0.11	3.87
8	0.11	3.53
16	0.13	3.34
32	0.30	3.36
64	1.47	4.98

measurement errors and the smaller the resulting estimation error is. However, the larger the window, the larger the impact will be of clock errors that are not being compensated for by the model function.

The resulting standard deviation of the TDOA estimates of the two sets of test data is given in Table 6.5 for different window sizes when a quadratic model is fit to the SOA values of the beacon transmissions. The results confirm the observation that was made in the previous paragraph and in Section 4.5.4. Of the six window sizes given in Table 6.5, the best results are obtained from the *road test data* when the window size is 16 seconds. For the *laboratory test data*, for which the SOA estimation errors are small, a larger window does not help to reduce the estimation error, but will only increase the error when the window is too large.

6.3 Pilot field test

6.3.1 Methodology

A pilot field test, which we refer to as the *road test*, was conducted on a relatively straight road in a rural area just outside of the city Potchefstroom. Two receivers were placed about 9 km apart. Each receiver station consisted of a yagi antenna placed about 1.5 m above the ground, a low-noise amplifier, an RTL-SDR, and a laptop. A beacon transmitter was strapped about 3 m above the ground to a utility pole next to the road in the middle between the two receivers. A mobile transmitter was mounted on top of a motor vehicle. The vehicle was driven from the one receiver to the other, stopping for a few minutes at fixed positions along the way. The C implementation of the carrier detection software was used on the laptops to stream the raw samples from the SDR, to divide them into fixed-length blocks, and to save blocks of data in which a carrier is detected to disk for offline processing and analysis.

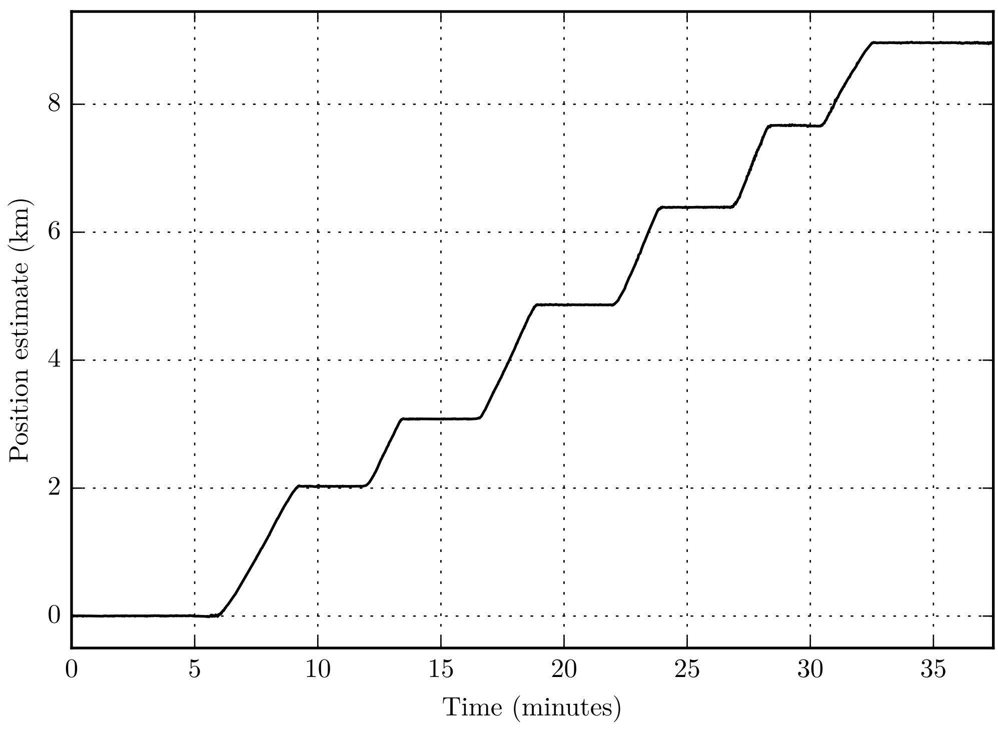


Figure 6.12: Position of the mobile transmitter over time during the pilot test, estimated using TDOA positioning from the data that was captured.

6.3.2 Position estimates

One-dimensional position estimates

Figure 6.12 displays the position of the mobile transmitter over time, as estimated from the data that was captured on the receivers, with the use of the signal and detection processing software. From this figure, it is clearly visible when and where the vehicle was stationary and when it was in motion. With TDOA positioning the position can only be estimated in one dimension when there are only two receivers. The position estimates are expressed as one-dimensional coordinates relative to one of the receivers.

Table 6.6 shows, for each stop, the number of position estimates at that stop, the mean position of the mobile transmitter, and the standard deviation of the position estimates. The mean standard deviation is 1.75 m, calculated as the standard deviation of the residuals at all the stops, each residual being the position after the mean position at that stop has been subtracted.

Table 6.6: The number of position estimations, the mean position, and the standard deviation of the position estimates of the mobile transmitter at the stops between the two receivers.

		Position	
Stop #	Count	Mean (m)	Std. dev. (m)
1	181	0	1.87
2	160	2027	1.37
3	201	3079	1.46
4	206	4864	1.78
5	184	6387	2.02
6	81	7668	1.94
7	232	8960	1.76
Total	1245	—	1.75

Precision of two-dimensional estimates

It follows from Equation (2.10) that, for one-dimensional positioning estimation, the relationship between the one-dimensional position x and the TDOA value $t_{i,j}$ is

$$|x - x_i| - |x - x_j| = c \cdot t_{i,j} \tag{6.6}$$

where x_i and x_j are the positions of the receivers. Let $r_{i,j}$ be the TDOA distance, i.e. $r_{i,j} = c \cdot t_{i,j}$. If $x_i \leq x \leq x_j$, then

$$x = \frac{1}{2} r_{i,j} + \frac{1}{2}(x_i + x_j). \tag{6.7}$$

Since the positions of the receivers are fixed, the position estimate is equal to half the TDOA distance plus a constant. The derivative of the position with respect to the TDOA distance estimate is:

$$\frac{dx}{dr_{i,j}} = \frac{1}{2}. \tag{6.8}$$

Thus, with two receivers, the DOP of the one-dimensional position estimate is 0.5 at any position between the two receivers. The positioning error is thus half the TDOA estimation error. For example, if the standard deviation of the TDOA distance estimates is 3.5 m (11.7 ns), the standard deviation of the one-dimensional position estimates will be 1.75 m. It is thus expected that, based on the results in Table 6.6, the standard deviation of the errors of two-dimensional position estimates when three or more receivers are present will be about 3.5 m at a position with a DOP of one.

Error relative to GPS

A mobile phone was used to log the GPS coordinates of the motor vehicle. A comparison between the position estimate from the TDOA system and the estimate calculated from the

Table 6.7: Comparison between the mean position estimates of the TDOA system for each stop and the position estimates calculated from coordinates that were recorded with a GPS device.

	Mean position		
Stop #	TDOA (m)	GPS (m)	Difference (m)
1	−0.04	0.23	−0.27
2	2027.00	2025.15	1.85
3	3078.86	3073.72	5.14
4	4863.62	4865.94	−2.31
5	6387.24	6393.11	−5.87
6	7667.73	7666.81	0.92
7	8959.58	8961.59	−2.01
Mean			−0.36
RMSE			3.26

GPS coordinates is displayed in Table 6.7 for each of the stops. To transform the GPS coordinates to a one-dimensional position estimate, a virtual TDOA distance estimate was calculated from the distances between the GPS coordinates recorded at the stop and the receivers' coordinates. The haversine formula was used to calculate the distance between a pair of coordinates. The difference between the position estimate from the TDOA system and the estimate calculated from the GPS coordinates reflects the mean error of the TDOA system relative to GPS. This is, however, only a rough comparison and does not reflect the actual error of the TDOA system since the GPS coordinates are inaccurate, perhaps even less accurate than the TDOA estimates. The estimated accuracy that was reported on the screen of the mobile phone is 3 m. Furthermore, the GPS coordinates were not captured at the position of the tag, but between 1 m and 4 m away from it. Moreover, we do not have an accurate estimate of the positions of the receivers and the beacon, which introduces a biasing error in the position estimates.

Despite the inaccuracies described above, the position estimates correspond very well. The mean difference is −0.36 m and the RMSE is 3.26 m.

6.3.3 Carrier frequency

The carrier frequency offsets of the two transmitters that were observed at each of the receivers, as estimated from the carrier peak in the DFT of the blocks of data, are displayed in Figure 6.13. From this figure it can be observed that the carrier frequencies of the transmitters were spaced about 3.5 kHz apart. Also visible is the difference between the estimates from the two receivers, which can be attributed to the unsynchronised receiver clocks.

The frequency offset of the mobile unit varies between 13.8 kHz and 14.9 kHz at receiver 1 and has a standard deviation of approximately 220 Hz. This variation can be attributed to the influence of temperature on the operating frequency of the transmitter's crystal oscillator.

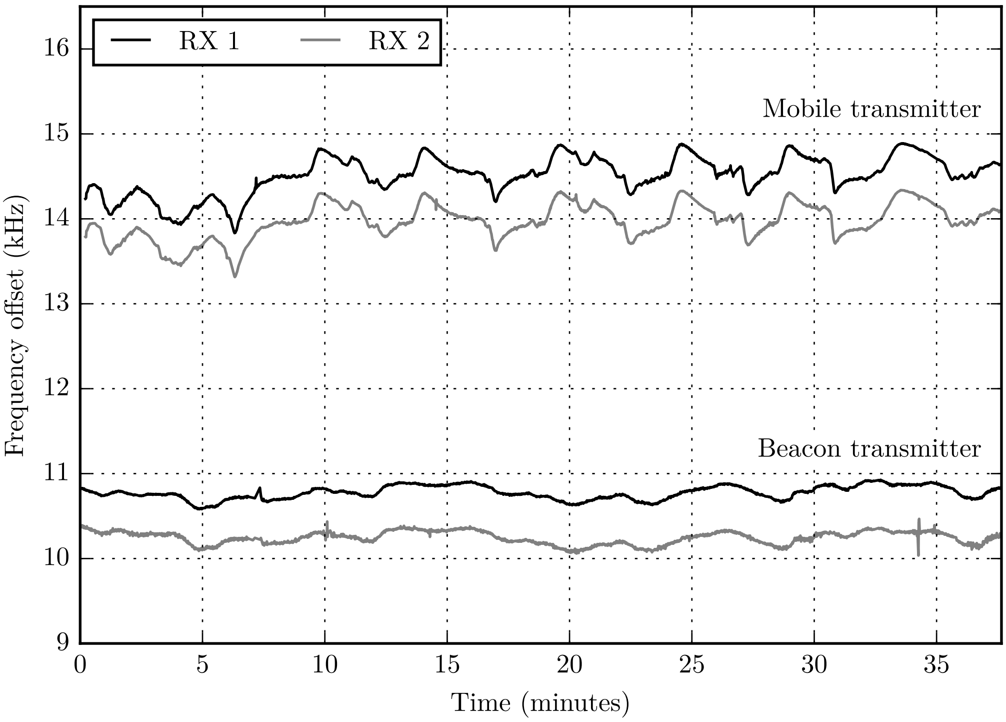


Figure 6.13: Carrier frequency offsets of the transmitters that were observed at each of the receivers, estimated from the DFTs of the data blocks.

During the test, the circuit board was exposed to the sun without any enclosure. This caused the components, and more specifically the crystal oscillator, to heat up in the sun while the transmitter was stationary and to cool down in the wind when the vehicle was in motion. Doppler shift also affects the carrier frequency that is being observed by the receivers, but in this case, the Doppler shift is small in comparison with the total variation of more than 1 kHz. When a transmitter is attached to an animal, the temperature will not change as quickly as it did during this test, but the transmitter will still be exposed to a broad range of temperatures, which will affect the carrier frequency. With the large variation in the carrier frequency, position estimation would have failed for most of the transmissions from the mobile unit if the frequency offset had not been estimated and compensated for. Carrier recovery is thus important when transmitters without high-precision temperature compensated oscillators are used.

6.4 Prototype receiver station

6.4.1 Software performance

If cheap RTL-SDRs can be used but the signal processing algorithms that are required are too slow to perform on inexpensive off-the-shelf signal processing hardware, the goal of inexpensive receiver stations that consist of off-the-shelf hardware will be jeopardised. With a sample rate of 2.4 MS/s, block length of 16384 samples, and history length of 4920 samples, the signal pro-

Table 6.8: Results from a performance test of the single-threaded signal processing software on a Raspberry Pi 3.

Software	Runtime (s)	Throughput (blocks/s)	Real-time
Carrier detection: Python	893.6	33.7	0.16x
Carrier detection: C	84.2	357.4	1.71x
Signal processing: Python	1454.0	20.7	0.10x
Signal processing: C / C++	190.1	158.3	0.76x

cessing software should be capable of processing at least 209 blocks per second to keep up with the stream of data from the SDR, i.e. for real-time processing. To validate the practicability of using a low-cost SBC for real-time signal processing, a performance test was conducted on a Raspberry Pi 3. An RTL-SDR was used to capture 34 500 blocks worth of data. This data was used as input to the signal processing software, and the runtime of the software was measured. The carrier detection threshold and correlation threshold were set to zero to ensure that all the processing steps are performed for each block of data, representing a worst-case scenario. The performance of both the Python and the C/C++ implementation was tested, first carrier detection only, and later the full stack of signal processing from raw samples to SOA estimation. For this test, the subsample carrier offset was estimated with parabolic interpolation, but the carrier frequency offset was only compensated for to the closest DFT bin.

The results of the test are displayed in Table 6.8. The runtime, the throughput in blocks per second, and a multiplier of the real-time throughput are given for the Python and the C/C++ implementation of the carrier detection and of the full signal processing software.

As discussed in Section 5.3.3, carrier detection is the most time-critical module. The modules after carrier detection are subject to a relaxed real-time constraint when only a subset of the blocks contain positioning signals, whereas carrier detection needs to be performed on every block without falling behind. With a throughput of 33.7 blocks/s, the Python implementation of the carrier detector achieves only 16 % of the throughput that is required for real-time processing. The C implementation *(fastcard)*, on the other hand, is about ten times faster than the Python implementation and can process the data at a throughput that is 71 % faster than the speed at which the data is being generated.

The throughput of the Python implementation for all stages of signal processing (*detect.py*) measures 20.7 blocks/s, which is ten times slower than real-time processing. The software can be used for real-time processing if the C implementation of the carrier detection software is used as a prefilter that is running on a different thread and if a carrier is detected in less than approximately 20 blocks/s.

End-to-end signal processing is performed at a speed of 158.3 blocks/s with the C/C++ implementation *(fastdet)*. This throughput is about 7.6 times faster than the throughput of the Python implementation and only 24 % slower than the rate at which the data is being generated. Note that this throughput is sufficient for real-time processing in most applications since a carrier will not be present in every block of data. Consequently, not all the signal processing steps will be applied to every block of data. The C/C++ implementation is fast enough for about 118 detections per second while carrier detection is performed in real-time, which is equivalent to a maximum of about 118 transmitters each transmitting once a second (assuming no collisions and no duplicate detections).

All the signal processing steps are currently performed on a single thread, while there are four cores available on the Raspberry Pi 3. The software can easily be enhanced to distribute signal processing across more than one core and harness the unused processing power. For example, blocks of data can be distributed across four threads, each thread performing signal processing in parallel. With parallel processing, the runtime of the software can be reduced to support end-to-end signal processing, i.e. detections in all of the blocks, in real-time.

6.4.2 Power consumption

The power consumption of the Raspberry Pi 3 with the RTL-SDR and the mobile broadband modem connected and with the signal processing software running, is about 3.7 W when the broadband modem is idle and approximately 4.7 W when data is being uploaded to the Internet.

6.4.3 Integration test

An integration test was performed to test the prototype receiver stations described in Section 5.4 as a whole. That is, to test that all the components work together when using solar-powered receiver stations that detect positioning signals in real-time using an RTL-SDR and a Raspberry Pi and that upload the detections to a central server through a mobile broadband modem.

Similar to the road test, two receivers were placed about 8.4 km apart just outside of the city Potchefstroom. A beacon transmitter was placed at a fixed position between the two receivers, two times closer to the one receiver than to the other receiver. A mobile transmitter was mounted on top of a vehicle, and the vehicle was driven from the one receiver to the other, stopping at fixed positions along the way.

The standard deviation of the one-dimensional position estimates is 4.62 m, and the RMSE is 4.12 m relative to position estimates calculated from inaccurate GPS measurements.

The standard deviation is more than twice the corresponding figure for the road test. There are several explanations for this discrepancy, including lower SNRs, strong interference, strong out-of-band signals, an omnidirectional antenna with lower gain and lower frequency selectivity, and extremely noisy beacon detections due to bad placement.

Nonetheless, the purpose of the integration test was not to characterise the accuracy and precision of the system. The results of the *road test* have already established a baseline for the performance that can be expected under favourable conditions. The accuracy and precision that were observed from the integration test are satisfactory, and the results from the integration test do prove the synergism between the constituent parts.

6.5 Verification and validation

6.5.1 Definition

A wide variety of definitions exists for *verification* and *validation* [55]. Sometimes the distinction between these two terms is unclear and sometimes the terms are even used interchangeable. In academic literature, the terms verification and validation are commonly used in the context of simulation models, where *verification* refers to

> "the process of determining that a computational model accurately represents the underlying mathematical model and its solution" [56]

and *validation* to

> "the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model". [56]

These definitions, however, do not suit the nature of the study in this dissertation where we have presented and tested not a simulation model but a design. Broad definitions that include product development are given by ISO 9000 [57], which defines *verification* as

> "confirmation, through the provision of objective evidence, that specified requirements have been fulfilled",

and *validation* as

> "confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled".

In [58], validation is summarised by the question "Are you building the right thing?" and verification as the question "Are you building it right?".

For the purpose of this study, the definitions from [56] are adapted and applied to an implementation that represents an underlying design, instead of a simulation that represents an underlying mathematical model. The *verification* process is used to confirm that the implementation is an accurate representation of the design, i.e. that the design is implemented correctly ("built right"). The *validation* process is used to confirm that the design assumptions and the models used within the design represent the real-world conditions accurately, that the design addresses its intended use ("building the right thing"), and that the research questions have been answered.

6.5.2 Methodology

Even though the design presented in Chapter 4 is based on mathematical equations and theoretical analyses, a pragmatic approach was followed for the validation and verification of the design. The design was verified and validated by building a proof-of-concept implementation and testing it empirically. No simulation can model all of the complexities and intricacies of an actual implementation and its behaviour in real-world conditions. The most reliable way to check the feasibility of a design is to implement the design and to test the actual implementation.

The results that were presented in this chapter serve simultaneously as verification of the implementation and validation of the design. The results show how individual aspects of the design and implementation were tested, as well as the testing of the design and implementation as a whole. The manner in which the results validate and verify the design and implementation, as well as additional verification and validation strategies that were followed, are elaborated upon next.

6.5.3 Verification

The various facets of the verification strategy that was followed are described below.

Best practices for software development The software was developed using best practices for software development, such as modularity, using static code analysis tools and writing unit tests. Unit tests verify the correctness of individual parts of the software and ensure that unanticipated regression errors are not introduced when changes are made.

Verification tools Software tools were developed for visualising and analysing data at different stages of signal and detection processing. These tools aid in visual verification of the algorithms and troubleshooting of the design and implementation. For example, Section 6.1 showcases a visual verification of the correctness of the signal processing algorithms and their implementation.

Pilot tests Pilot tests were conducted with the proof-of-concept implementation and the precision and accuracy of the final output were estimated using statistical measures. The results from these end-to-end empirical tests verify every aspect of the design and implementation since any significant mistake within the system would have had an adverse effect on the results.

For example, Section 6.2 shows how simulations and sensitivity analyses were used to verify the impact of individual components and parameters on the final output. Section 6.3 demonstrates how a pilot field test was used to verify the correct implementation of all aspects of a proof-of-concept hardware implementation with RTL-SDRs in real-world conditions. The test verifies the hardware and software implementation of the transmitter, the SDR, the signal processing software, and the detection processing software. The test shows that the combination of these components delivers precise positioning over a distance of 9 km. Section 6.4.1 verifies that the software is fast enough for real-time processing on an SBC. Section 6.4.3 verifies the end-to-end functionality of an implementation with prototype receiver stations under unfavourable conditions.

Comparison to related work The implementation and results of the empirical tests were compared to related work. The precision of the proof-of-concept positioning system with unsynchronised receivers presented in this dissertation is similar or better than the precision reported for related systems with synchronised receivers. A comparison is outlined in Section 7.3.

6.5.4 Validation

The results that were presented in this chapter show how the proof-of-concept implementation was used to validate the design. For instance, Section 6.2.2 validates the importance of correlation peak interpolation for accurate positioning. It also validates that Gaussian interpolation is a good choice for the prototype positioning system, providing a balance between speed and accuracy. Section 6.2.3 validates that carrier interpolation can help to improve the precision of the arrival time estimate, even though the improvement is negligible for the parameter values that we have chosen. The performance improvement when compensating for the subsample carrier frequency offset using precomputed templates instead of shifting the signal in the time-domain is also validated in Section 6.2.3.

Section 6.2.4 validates numerous aspects of the design and analysis of the TDOA estimator that were discussed in Section 4.5.4. In Section 6.2.4 it is validated that a quadratic function provides a good model for the difference in SOA between two receivers and that the time interval over which the model is fit should be limited to restrict residual errors. Furthermore, Section 6.2.4 validates that the divergence of the receiver clocks between the mobile unit transmission and the nearest beacon transmission is significant enough to be severely detrimental to the accuracy of the arrival time estimate. It is also shown that interpolating between the two nearest beacon transmissions is an effective way to mitigate this divergence. Moreover, it is validated that an LS fit of a quadratic model results in better estimates than linear interpolation since measurement errors are evened out. All in all, the results in Section 6.2.4 are a testimony to the fact that precise synchronisation can be obtained with unsynchronised clocks when the methods that we have presented in this dissertation are used.

The pilot test presented in Section 6.3 validates the design as a whole. The test validates that the design and the software based on the design enables cheap, off-the-shelf, unsynchronised SDRs to be used for precise and accurate position estimation over a range of 9 km. The results show that the design successfully alleviates the impact of errors caused by the shortcomings of the low-cost SDRs. It is also shown that carrier frequency offset is a valid concern and that OOK modulation proves to be beneficial for estimating and compensating for the offset.

Section 6.4 validates that the design makes provision for inexpensive, off-the-shelf, general-purpose signal processing hardware. Section 6.4.1 shows that the design is fast enough for real-time signal processing on a Raspberry Pi 3. Section 6.4.3 validates that it is possible and feasible to use the receiver stations that we have constructed for accurate TDOA positioning.

The next chapter describes, as the final step of validation, how the work we have presented in this dissertation answers the research questions that were set out in the first chapter.

Chapter 7

Conclusion

This chapter wraps up the work that was presented in this dissertation. First, a summary of the work is given in Section 7.1, after which the key findings of the study are outlined in Section 7.2. In Section 7.3, the work is brought into perspective with related work by contrasting the properties of the positioning system presented in this dissertation to the properties of two related systems. Some of the unique contributions that were made in this study are emphasised in Section 7.4. Recommendations for future work are reflected upon in Section 7.5. Finally, remarks regarding alternative applications are made in Section 7.6.

7.1 Overview of work

The feasibility of an inexpensive hyperbolic positioning system for tracking wildlife using off-the-shelf receiver hardware was investigated and proved in this dissertation.

Chapter 1 served to substantiate the significance of the research and to formulate the research problem and objectives. It was demonstrated that two significant constraints in wildlife tracking systems are the energy consumption of the tags attached to the animals and the cost of the positioning system. It was shown that a simple solution for tracking the position of an animal is to attach a GPS-enabled tag, but that a network-based positioning system can reduce the power consumption of the tag by a few orders of magnitude. Existing network-based positioning systems were investigated, but it was found that they use precisely synchronised receiver stations that are built from custom-designed electronics or expensive off-the-shelf hardware. Furthermore, these receiver stations are not being sold commercially. An innovative approach was followed in this study with the focus on a simple and easily reproducible receiver design that uses low-cost off-the-shelf hardware with unsynchronised, inaccurate clocks. The research problem that was addressed in this dissertation is to investigate whether such a design would be feasible for tracking animals, how it can be implemented and what the estimated accuracy is that can be expected from it.

Chapters 2 and 3 set the foundation for the design with summaries and analyses of various principles and aspects pertaining to TDOA positioning, and with derivations of equations that were used in the design. It was shown in Chapter 2 that network-based Time Difference of Arrival (TDOA) positioning, also called hyperbolic positioning, is a technique whereby a network of receivers at known locations is used to calculate the position of a transmitter from the difference in the time at which a signal from the transmitter arrives at three or more receivers. It was demonstrated in Chapter 2 that the positioning error is a function of the TDOA measurement errors and the mobile unit – base station geometry. The latter, the DOP, is the same regardless of the base station equipment or measurement techniques being used, and the accuracy of the TDOA measurements is thus the primary concern when a system is being designed. It was also shown that the accuracy of the TDOA measurements depend on the SNR and bandwidth of the positioning signal and that spread spectrum techniques can provide suitable signals with high energy and wide bandwidth.

Arrival time estimation using Direct Sequence Spread Spectrum (DSSS) signals was discussed in Chapter 3. Interpolation methods were presented for improving the resolution of the arrival time estimate to a resolution that is smaller than the sample period. It was shown that BPSK modulation is generally used for DSSS, but that BPSK does not allow for fast carrier recovery of a short-lived positioning signal. The adverse consequences of a carrier frequency offset were illustrated and it was shown that the frequency offset can be recovered easily when OOK modulation is used to modulate the positioning signal, since about half of the signal's energy is then concentrated within a narrow band at the carrier frequency. Furthermore, detection theory was used to derive a simple decision rule for deciding upon the presence of a valid positioning signal as well as a rule for detecting the presence of a carrier.

Chapter 4 details a novel design of a TDOA positioning system. The design makes provision for receiver devices that have time-varying differences in their sample rates due to unsynchronised clocks, that have limited computational power, and that cannot give accurate estimates of the time at which samples were taken. The design also makes provision for transmitters with limited frequency stability. The positioning code is modulated with OOK since it is easy to implement, it enables simple and fast carrier recovery, it reduces the computational requirements of the receiver by discarding blocks of data for which no carrier is detected, and it allows the receiver to distinguish between detections from different transmitters based on the carrier frequency offset. Periodic transmissions from one or more transmitters with fixed and known positions, called *beacons*, are used to synchronise and relate arrival time estimates from different receivers to one another.

The design was verified and validated by building a proof-of-concept implementation and performing empirical tests. In Chapter 5, a simple and versatile implementation of the design was outlined that is easy and cheap to reproduce. Instead of developing tailor-made hardware for the receivers, the complexity of the design was moved into the software to allow for the use of inexpensive and readily available general-purpose hardware. An RTL-SDR, which is a cheap

$20 SDR, is used as an RF receiver, and a $35 Raspberry Pi 3 is used for signal processing. The signal and detection processing software was implemented and experimented with in Python. A fast implementation of the signal processing software was implemented in C/C++ for real-time signal processing on a Raspberry Pi 3. Existing RF transceiver devices were adapted to be used as transmitter devices in a TDOA positioning system. It is estimated that the tag device consumes at least three orders of magnitude less energy per position estimate with TDOA positioning than it would consume if GPS had been used.

Tests that were conducted with the proof-of-concept implementation to confirm the feasibility and validity of the design and implementation for position estimation are described in Chapter 6. Various aspects of the design were tested and different techniques and parameters were compared through simulations and sensitivity analyses using test data that was captured during a field test as well as test data with excellent SNR that was captured in a laboratory environment. A pilot field test was conducted with two RTL-SDR receivers spaced 9 km apart. The standard deviation of the TDOA estimates was about 11.5 ns, which is equivalent to a precision of 3.5 m for two-dimensional position estimates. The throughput of the signal processing software was tested on a Raspberry Pi 3, which showed that the C/C++ implementation is fast enough for real-time processing with up to 118 detections per second. An integration test was performed with prototype receiver stations, each receiver station consisting of an RTL-SDR, Raspberry Pi 3, mobile broadband modem, solar panel and battery pack, which proved that TDOA positioning can be performed with receiver stations that are constructed from low-cost general-purpose off-the-shelf hardware modules.

7.2 Key findings

The key findings from this dissertation are:

- It is feasible to perform TDOA positioning with receiver stations that have independent clocks and that are constructed from inexpensive general-purpose off-the-shelf hardware.
- Excellent accuracy and precision can be obtained despite the constraints imposed by a simple design with low-cost hardware. A two-dimensional positioning precision of up to approximately 3.5 m can be achieved with receivers that are positioned 9 km apart. A precision of 11 cm can be achieved in a laboratory environment under good SNR without considering the carrier's phase.
- Arrival time estimates from receivers with unsynchronised, inaccurate clocks can be calibrated in software after signal processing. Fitting beacon transmissions to a model of the clock drift is an effective method for synchronising arrival time estimates.
- The use of OOK for modulating DSSS positioning signals is beneficial for reducing the computational requirements of the receiver and for allowing the use of transmitters with

low-cost clocks.

- Subsample interpolation of the correlation peak is crucial for precise positioning. The choice of interpolation method has a substantial impact on the precision. The best precision was obtained with an improvised method involving an LS fit to the autocorrelation peak, and slightly worse precision was obtained with Gaussian interpolation.

7.3 Comparison to related work

As mentioned in Chapter 1, there are two other TDOA positioning systems for wildlife tracking that we are aware of; the one was developed by a group at Cornell University, which we will refer to as *Cornell*, the other by The Hebrew University of Jerusalem in conjunction with Tel–Aviv University, referred to as *ATLAS*. A comparison between the properties of those two systems and the one presented in this dissertation is outlined in Table 7.1 and expanded upon below. The comparison is based on the limited amount of information provided in [1, 12–16]. Consequently, estimations are used and assumptions are made where the source material does not provide enough information.

Receiver clock synchronisation The greatest aspect that differentiates our design from the other two is that it works with unsynchronised receiver clocks that are independent. The other two systems both rely on GPSDOs at the receivers for precise clock synchronisation and clock stability. Precise synchronisation simplifies position estimation but adds to the cost and complexity of the hardware. Our system, on the other hand, makes use of beacon detections to estimate and compensate for clock drifts in software after data acquisition. ATLAS also makes use of beacons to compensate for clock offsets between the receivers, but they use the beacons differently and still rely on a GPSDO for clock stability.

Off-the-shelf hardware An advantage of our system over the other two is that the receiver stations are constructed from low-cost off-the-shelf hardware modules. Cornell uses a purpose-built receiver circuit board with a custom RF receiver circuit, a pricey DSP chip and a GPS module. It is difficult to estimate the cost of their receiver stations with limited information about the design, but it is estimated that both the unit cost and the development cost will be substantial due to the custom design and low production volumes. ATLAS uses general-purpose off-the-shelf hardware products, but the products are expensive. The total cost of the products that they use for the RF module, i.e. an *Ettus Research USRP N200* radio, *WBX* RF daughterboard and supposedly a GPSDO kit from Ettus Research, amounts to approximately $2800. Their RF module is thus about three orders of magnitude more expensive that the RTL-SDR that is being used as RF module in our system. Furthermore, they use a PC for signal processing, which adds at least $300 to the cost of the receiver station. Our system, on the other hand, uses an inexpensive $35 SBC.

Table 7.1: Comparison between the properties of the positioning system presented in this dissertation *(NWU)* and the properties of two related systems, one developed by The Hebrew University of Jerusalem *(ATLAS)* and another by a group at Cornell University *(Cornell)*.

Property	Cornell	ATLAS	NWU
Receiver hardware			
Off-the-shelf modules:	No	Yes	Yes
RF module:	Purpose-built	USRP	RTL-SDR
DSP module:	DSP chip	PC	SBC
Requires GPSDO:	Yes	Yes	No
Uses beacons:	No	Yes	Yes
Sample rate (MS/s):	2.8125	8.33	2.40
Power consumption (W):	16	>60[1]	4
Est. RF module cost ($):	Unknown[2]	2800[3]	20
Est. DSP module cost ($):	130[4]	300[5]	35
Software			
DSP software:	Embedded C	C & Java	C & C++
Localisation software:	Matlab	Java & Matlab	Python
Interpolation method:	Unknown	Parabolic	Gaussian
Released software:	No	No	Yes
Transmitters			
Code type:	Gold	Gold	Gold
Code length (chips):	2047	8191	2047
Chip rate (MHz):	1	1	1
Signal duration (ms):	4	8	2
Centre freq. (MHz):	140	434	434
Modulation:	PSK	FSK	OOK
Statistics			
Std. dev. reported (m):[6]	9	3	3.5
Mean error reported (m):[6]	19[7]	5 – 15	1 – 6[7]
References:	[1, 12–14]	[15, 16]	Dissertation

[1] Estimated lower bound on the power consumption of a PC and a USRP N200 with WBX daughterboard.

[2] It is estimated that both the unit cost and development cost will be substantial due to the custom design.

[3] Estimated as the cost of an Ettus Research USRP N200, WBX daughterboard, and GPSDO kit.

[4] Estimated cost of a Texas Instruments TMS320C6416 DSP chip.

[5] Estimated as the cost of a PC.

[6] Only a rough comparison since each system was tested differently and in a different environment.

[7] Error calculated relative to inaccurate GPS measurements.

Power consumption Our system's receiver station has the lowest power consumption of the three systems. With a power consumption of 4 W, the receiver stations can be powered from batteries and solar panels without difficulty. It is assumed that the ATLAS receiver stations cannot be powered from batteries but require mains electricity due to the use of PCs. It is estimated that the power consumption of their receiver stations are at least 60 W. Cornell's receiver stations are powered from batteries and have a power consumption of 16 W.

Positioning signal All three systems use Gold codes at a chip rate of 1 MHz. The code length in Cornell's and our system are the same, but Cornell's signal duration is two times longer since they transmit the code twice; they first transmit a common code for acquisition and synchronisation, then a unique code to identify the transmitter. ATLAS uses a code that is four times longer, resulting in a signal duration that is four times longer than the duration in our system. Each of the three systems uses a different modulation technique. Cornell uses PSK, ATLAS uses FSK and we use OOK. Our system gets along with the lowest receiver sample rate, a sample rate of 2.4 MS/s. Cornell uses a sample rate of 2.8125 MS/s and ATLAS uses a sample rate of 8.33 MS/s.

Precision and accuracy The estimated precision and accuracy of our system are similar to or better than those of the other two systems. The standard deviation that was reported for two-dimensional position estimates of a transmitter at a fixed location is approximately 9 m for Cornell's system, 3 m for ATLAS and 3.5 m for our system. A mean positioning error of between 5 m and 15 m is reported for ATLAS, taken relative to a location determined by a surveyor. The mean positioning errors relative to inaccurate GPS measurements are reported to be about 19 m for Cornell's system. We observed, also relative to inaccurate GPS measurements, a mean error of between 1 m and 6 m for our system. These figures provide only a rough comparison and should not be compared directly since they present statistics of tests that were conducted differently, in different environments, with different receiver configurations and with different SNRs.

7.4 Contributions

Some of the unique contributions made in this study are emphasised in this section.

Low-cost unsynchronised receivers In this dissertation, a novel approach for accurate TDOA positioning was presented that, using the design that was devised, does not require the receivers to be synchronised and can be implemented with low-cost general-purpose off-the-shelf hardware modules. Receiver stations can be constructed for an estimated cost of $100 per receiver, excluding the cost of auxiliary components such as batteries and antennae. To the best of the author's knowledge, no other TDOA positioning system exists in the same price range.

Software The software forms the essence of the system that transforms the general-purpose hardware into a TDOA positioning system. The author contributed fast signal processing software for real-time signal processing on a Raspberry Pi, software for detection processing that calculates position estimates from the detections reported by the receivers, and various utility modules for data analysis and troubleshooting. The software consists of about 5000 lines of Python code, 2100 lines of C code and 600 lines of C++ code. The complete source code of the proof-of-concept implementation has been released as open-source software under a GNU GPL license to allow collaboration with research groups working in similar directions, to facilitate further experimentation, to enable reproducibility of the results and to leverage future research efforts. The source code is available at `https://github.com/swkrueger/Thrifty`. This software enables an accurate, simple, inexpensive and easily reproducible TDOA positioning system to be constructed with Raspberry Pi 3 and RTL-SDR devices. The authors are not aware of the existence of other free or open-source TDOA positioning software.

Literature study This dissertation presents literature studies and theoretical analyses that provide a fundamental understanding of all the principles that pertain to the design of a TDOA positioning system, as well as an understanding of how these principles interconnect to give rise to a positioning system. Topics that were covered include positioning principles, DOP, estimation theory, the CRLB, DSSS, spreading codes, subsample interpolation, modulation schemes and signal detection theory. We are unaware of other literature that provides a holistic view of TDOA positioning from analyses of the fundamentals, to a design that incorporates those fundamentals, to a practical implementation of the design, to results that show how the implementation was tested.

Techniques Various techniques for improving the precision and for reducing the computational requirements were analysed, compared and conceived in this dissertation. For example, this study compared different subsample interpolation techniques in terms of their estimation error and computational requirements to select the best technique for precise positioning on limited hardware. Furthermore, an improvised interpolation technique was presented that exhibits excellent precision, especially under good SNR. Different modulation techniques were analysed and the advantage of using OOK to modulate the positioning code, namely that it enables simple and fast carrier recovery, was demonstrated. A technique was devised for compensating for a subsample frequency offset in the frequency domain without slowing down the signal processor. A technique was presented for relating the TOA from unsynchronised receivers to one another and different parameter values were analysed and compared.

Publications In addition to this dissertation, two peer-reviewed conference papers were delivered [59, 60].

7.5 Recommendations for future work

The accuracy and precision of the prototype positioning system have exceeded our expectations. We did not expect to attain a precision of 11 cm, which is less than a thousandth of a sample, under excellent SNR and a precision of 3.5 m, which is less than a 35th of a sample, in real-world conditions, without clock synchronisation and with low-cost TV tuners repurposed as SDRs that have many disadvantages and imperfections in comparison with high-end SDR platforms.

The work presented in this dissertation lays the foundation for further study. Despite the success of the proof-of-concept system, many opportunities for further tests and enhancements exist. Recommendations for future work are presented below.

Field test We have evaluated the performance of the proof-of-concept system based on the standard deviation and mean error of the TDOA estimates of two receiver stations. A next step would be to characterise the precision and accuracy of the position estimates by performing a field test in a game reserve or similar environment with at least three receiver stations positioned at well-defined locations at least 5 km apart and with transmitters placed at fixed and well-defined locations. A follow-up test to characterise the system with tags attached to wild animals is suggested.

Characterise configurations and parameters There are myriads of variables in the positioning system, each having an impact on the performance of the system. To understand the performance of the system under difference conditions, it will be necessary to test it with different parameter values and different configurations. Variables that can be investigated include receiver placement, antenna selection, code length, the number of code repetitions, beacon placement, the number of beacons and the beacon transmission period. The system can be characterised in different environments to study the effect of severe interference and multipath propagation.

Furthermore, the system can be tested in a different frequency band. The 433.04–434.79 MHz frequency range is an ISM band in South Africa shared among many different users, with the result that interfering signals are commonly observable [61]. An alternative wideband channel that may be used is 148–152 MHz, a quiet frequency range designated for wildlife tracking in national game parks. The same receiver hardware can be used since, with an SDR, changing to another frequency band is as simple as changing a software setting.

Phase information The carrier phase is available from the I/Q samples and can be used to obtain sub-wavelength precision. The carrier phase is especially useful when the resolution of the arrival time estimate is smaller than a wavelength, but can also be used for improved precision when the resolution is larger than a wavelength and additional information is available to resolve ambiguities, such as many detections from a stationary mobile unit, or detections from many receiver stations.

Interpolation method The performance of five correlation peak interpolation methods were assessed in this dissertation, but there is an abundance of other interpolation methods in literature that can be evaluated as well.

Software refinements There are many refinements that can be made to the signal and detection processing software. For example, signal processing can be distributed across multiple cores for faster signal processing. Furthermore, the noise power is currently estimated separately for each block of data, which results in inaccurate estimates of the noise power when a strong signal is present. Instead, the noise can be computed from a moving average over multiple blocks of data.

Alternative transmitter device Existing general-purpose tags were modified for the pilot tests to enable them to transmit a positioning signal. A purpose-fit circuit board can be designed for TDOA tracking that is simpler, smaller and lighter. Moreover, alternative RF and MCU modules that have smaller footprints and that have support for OOK at a high baud rate can be investigated.

Alternative signal processor Alternative off-the-shelf modules for the receiver device can be investigated. For example, an Android phone with an RTL-SDR connected via USB On-The-Go can be used for makeshift receiver stations. The smartphone has a battery, a powerful CPU and GPU for signal processing, GSM connectivity for reporting detections to a central server, and GPS for determining the position of the receiver. The smartphone does not have to be expensive. At the time of writing, a cheap Android phone with USB On-The-Go could be purchased for about $70.

Alternative RF receiver General purpose, off-the-shelf hardware is the fastest and most cost effective solution when only a few receivers are required. If many receivers have to be manufactured for a large receiver array or for sale as commercial products, the higher production volumes will make it economically viable to design a custom-built RF receiver with low per-unit cost that is tailored towards arrival time estimation to provide better characteristics than that provided by general-purpose hardware.

7.6 Concluding remarks

The TDOA positioning system that was presented in this study is not limited to wildlife tracking as the only application. It can be used for any application where objects need to be tracked with simple, small, low-power, long-lasting, inexpensive tags over a large but limited geographical area. Since the tags are small and inexpensive, almost anything can be tracked. Examples of other applications include asset tracking, livestock monitoring, tracking people or objects on a farm, construction site or plant, or fine-grained tracking of contestants at a racing competition such as a marathon.

Moreover, the techniques that were presented in this dissertation apply to a more general problem than just position estimation. The same techniques can be applied to other applications where data from multiple synchronised receivers is being cross-correlated, e.g. for long baseline phased array receivers. Synchronising the receivers with traditional methods adds to the cost and complexity of the hardware; using a common clock is hard to achieve and expensive with large arrays or over long distances, and using a reference clock such as a GPSDO adds to the cost of the receivers. Instead, inexpensive receivers with unsynchronised clocks can be used together with one or more beacon transmitters. Periodic synchronisation pulses from a beacon transmitter can then be used to synchronise the data in software after data acquisition by estimating and compensating for clock offset and clock drift. Example applications include passive radar and aperture arrays for radio astronomy.

Bibliography

[1] R. B. MacCurdy, R. M. Gabrielson, and K. A. Cortopassi, "Automated wildlife radio tracking," in *Handbook of Position Location*, S. A. Zekavat and R. M. Buehrer, Eds., Hoboken, NJ, USA: John Wiley & Sons, Inc., Sep. 2011, pp. 1129–1167.

[2] V. Zeimpekis, G. M. Giaglis, and G. Lekakos, "A taxonomy of indoor and outdoor positioning techniques for mobile location services," *ACM SIGecom Exchanges*, vol. 3, no. 4, pp. 19–27, Dec. 2002.

[3] S. Wang, J. Min, and B. K. Yi, "Location based services for mobiles: technologies and standards," in *IEEE international conference on communication (ICC)*, 2008, pp. 35–38.

[4] C. James, "Multilateration: radar's replacement?" *Avionics magazine*, vol. 31, no. 4, p. 30, 2007.

[5] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems.* Artech house, 2013.

[6] S. Maddio, A. Cidronali, and G. Manes, "Smart antennas for direction-of-arrival indoor positioning applications," in *Handbook of Position Location*, S. A. Zekavat and R. M. Buehrer, Eds., Hoboken, NJ, USA: John Wiley & Sons, Inc., Sep. 2011, p. 321.

[7] L. D. Mech and S. M. Barber, "A critique of wildlife radio-tracking and its use in national parks," *Biological Resources Management Division, US National Park Service, Fort Collins, CO Technical Report*, 2002.

[8] J. A. Cordier, "An investigation into the design, development, production and support of a wildlife tracking system based on GSM/GPS technologies," Master's thesis, North-West University, 2006.

[9] *ORG1411: Nano Hornet GPS antenna module*, ORG1411, Rev. 2.0, OriginGPS, Oct. 2014.

[10] *Si4010-C2: crystal-less SoC RF transmitter*, Si4010-C2, Rev. 1.0, Silicon Laboratories, Inc., 2010.

[11] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance*, 2nd edition. Lincoln, MA: Ganga-Jamuna Press, 2011.

[12] R. B. MacCurdy, R. M. Gabrielson, E. Spaulding, A. Purgue, K. A. Cortopassi, and K. Fristrup, “Real-time, automatic animal tracking using direct sequence spread spectrum,” in *Wireless Technology, 2008. EuWiT 2008. European Conference on*, IEEE, 2008, pp. 53–56.

[13] ——, “Automatic animal tracking using matched filters and time difference of arrival.,” *Journal of Communications*, vol. 4, no. 7, 2009.

[14] T. Piersma, R. B. MacCurdy, R. M. Gabrielson, J. Cluderay, A. Dekinga, E. L. Spaulding, T. Oudman, J. Onrust, J. A. van Gils, D. W. Winkler, and A. I. Bijleveld, “Fijnmazige positiebepaling van individuen in groepen: de principes en drie toepassingen van toa-tracking,” *Limosa*, vol. 87, no. 2, pp. 156–167, 2014.

[15] O. Kishon, “The ATLAS wildlife localization system: system design and implementation,” Master’s thesis, The Blavatnik School of Computer Science, Tel Aviv University, Israel, 2015.

[16] A. W. Weiser, Y. Orchan, R. Nathan, M. Charter, A. J. Weiss, and S. Toledo, “Characterizing the accuracy of a self-synchronized reverse-GPS wildlife localization system,” in *Proceedings of the 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Vienna, Austria, Apr. 2016.

[17] J. A. Van Allen, “Basic principles of celestial navigation,” *American Journal of Physics*, vol. 72, no. 11, pp. 1418–1424, 2004.

[18] S. Sand, A. Dammann, and C. Mensing, *Positioning in Wireless Communications Systems.* Wiley, 2014.

[19] A. Bensky, *Wireless positioning technologies and applications.* Artech House, 2007.

[20] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software.* New York, NY, USA: John Wiley & Sons, 2002.

[21] H. C. So, Y. T. Chan, and F. K. W. Chan, “Closed-form formulae for time-difference-of-arrival estimation,” *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2614–2620, Jun. 2008.

[22] J. Smith and J. Abel, “Closed-form least-squares source location estimation from range-difference measurements,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 12, pp. 1661–1669, Dec. 1987.

[23] B. T. Fang, “Simple solutions for hyperbolic and related position fixes,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 5, pp. 748–753, Sep. 1990.

[24] G. A. Mizusawa, “Performance of hyperbolic position location techniques for code division multiple access,” PhD thesis, Virginia Polytechnic Institute and State University, 1996.

[25] H. C. So, “Source Localization: Algorithms and Analysis,” in *Handbook of Position Location*, S. A. Zekavat and R. M. Buehrer, Eds., Hoboken, NJ, USA: John Wiley & Sons, Inc., Sep. 2011, pp. 25–66.

[26] G. Shen, R. Zetik, and R. S. Thoma, "Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment," in *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, Mar. 2008, pp. 71–78.

[27] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*, 2nd ed. Artech House, 2006.

[28] G. Hein, J. Avila-Rodriguez, and S. Wallner, "The DaVinci Galileo code and others," *Inside GNSS*, vol. 1, no. 6, pp. 62–74, 2006.

[29] L. Frenzel, *Principles of Electronic Communication Systems*, 3rd ed. New York, NY, USA: McGraw-Hill, Inc., 2008.

[30] R. Dixon, *Spread spectrum systems with commercial applications*, 3rd ed. New York: Wiley, 1994, ISBN: 978-0-471-59342-3.

[31] J. Jedwab and S. Lloyd, "A note on the nonexistence of barker sequences," *Designs, Codes and Cryptography*, vol. 2, no. 1, pp. 93–97, 1992.

[32] L. Svilainis, "Review of high resolution time of flight estimation techniques for ultrasonic signals," in *2013 International Conference NDT, Telford, UK, (Sep. 8-12, 2013)*, 2012, pp. 1–12.

[33] T. Wiens and S. Bradley, "A comparison of time delay estimation methods for periodic signals," 2012. [Online]. Available: `http://www.nutaksas.com/papers/wiens_dsp_delay.pdf` (visited on 06/16/2016).

[34] L. Svilainis, K. Lukoseviciute, V. Dumbrava, and A. Chaziachmetovas, "Subsample interpolation bias error in time of flight estimation by direct correlation in digital domain," *Measurement*, vol. 46, no. 10, pp. 3950–3958, 2013.

[35] L. Zhang and X. Wu, "On the application of cross correlation function to subsample discrete time delay estimation," *Digital Signal Processing*, vol. 16, no. 6, pp. 682–694, Nov. 2006.

[36] E. McCune, *Practical digital wireless signals.* Cambridge University Press, 2010.

[37] M. A. Richards, *Fundamentals of radar signal processing*, 2nd ed. New York: McGraw-Hill Education, 2014.

[38] S. M. Kay, *Fundamentals of statistical signal processing, vol. II: detection theory.* Upper Saddle River, N.J: Prentice-Hall PTR, 1998.

[39] D. Rife and R. Boorstyn, "Single tone parameter estimation from discrete-time observations," *IEEE Transactions on information theory*, vol. 20, no. 5, pp. 591–598, 1974.

[40] H. C. So, Y. T. Chan, Q. Ma, and P. C. Ching, "Comparison of various periodograms for sinusoid detection and frequency estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, pp. 945–952, Jul. 1999.

[41] J. O. Smith, *Spectral audio signal processing.* W3K Publishing, 2011. [Online]. Available: `http://ccrma.stanford.edu/~jos/sasp/`.

[42] Y. Liao, “Phase and frequency estimation: high-accuracy and low-complexity techniques,” PhD thesis, Worcester Polytechnic Institute, 2011.

[43] S. K. Mitra, *Digital signal processing: a computer-based approach*, 4th ed. New York, NY: McGraw-Hill, 2011, International edition.

[44] *Si1000/1/2/3/4/5 datasheet*, Rev. 1.3, Silicon Laboratories, Inc., 2013.

[45] *About RTL-SDR.* [Online]. Available: `http://www.rtl-sdr.com/about-rtl-sdr/` (visited on 09/13/2016).

[46] S. Markgraf, D. Stolnikov, and Hoernchen, *Rtl-sdr*, Open Source Mobile Communications (osmocom), Dec. 20, 2014. [Online]. Available: `http://sdr.osmocom.org/trac/wiki/rtl-sdr` (visited on 09/13/2016).

[47] *RTL-SDR and GNU Radio with Realtek RTL2832U, E4000 and R820T*, Jun. 30, 2016. [Online]. Available: `http://superkuh.com/rtlsdr.html` (visited on 09/13/2016).

[48] *New RTL-SDR dongles with metal case available in our store*, Jan. 14, 2016. [Online]. Available: `http://www.rtl-sdr.com/new-rtl-sdr-dongles-back-in-stock-in-amazon/` (visited on 09/13/2016).

[49] *New products: $20 RTL-SDR with 1 ppm TXCO, SMA F connector and R820T2 now available in our store*, Aug. 15, 2015. [Online]. Available: `http://www.rtl-sdr.com/new-products-rtl-sdr-with-1ppm-tcxo-sma-f-connector-r820t2-bias-tee-improved-tolerances-direct-sampling-break-out-pads-now-available-in-our-store/` (visited on 09/13/2016).

[50] *Numpy*, version 1.8.2, Aug. 9, 2014. [Online]. Available: `http://www.numpy.org/` (visited on 09/24/2016).

[51] *Scipy*, version 0.14.0, May 3, 2014. [Online]. Available: `http://www.scipy.org/` (visited on 09/24/2016).

[52] H. Gomersall, *pyFFTW*, version 0.9.2, Sep. 20, 2013. [Online]. Available: `https://pyfftw.github.io/pyFFTW/` (visited on 09/24/2016).

[53] M. Frigo and S. G. Johnson, *Fastest Fourier Transform in the West (FFTW)*, version 3.3.4, Mar. 16, 2014. [Online]. Available: `http://www.fftw.org/` (visited on 09/24/2016).

[54] *Vector-Optimized Library of Kernels (VOLK)*, version 1.3, Jul. 2, 2016. [Online]. Available: `http://libvolk.org/` (visited on 09/24/2016).

[55] P. G. Maropoulos and D. Ceglarek, “Design verification and validation in product lifecycle,” *CIRP Annals-Manufacturing Technology*, vol. 59, no. 2, pp. 740–759, 2010.

[56] American Society of Mechanical Engineers (ASME), *Guide for verification and validation in computational solid mechanics*, Mar. 29, 2006.

[57] ISO 9000:2015(E), *Quality Management Systems: Fundamentals and Vocabulary*, Sep. 2015.

[58] B. Boehm, *Software Engineering Economics.* Prentice-Hall, 1981.

[59] S. W. Krüger and A. S. J. Helberg, “An arrival-time detection technique for multilateration-based automatic wildlife tracking,” in *Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2014*, Nelson Mandela Bay, South Africa, Sep. 2014, pp. 477–478.

[60] ——, “Fundamentals of and considerations for the design of a multilateral one-way TDOA positioning system using DSSS,” in *Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2015*, Hermanus, South Africa, Sep. 2015, pp. 293–298.

[61] Independent Communications Authority of South Africa (ICASA), *Radio frequency spectrum regulations 2015*, Government Gazette No. 38641:3 (Notice 279), vol. 597, South Africa, Mar. 30, 2015. [Online]. Available: `http://www.gpwonline.co.za/Gazettes/Gazettes/38641_30-3_Icasa.pdf`.