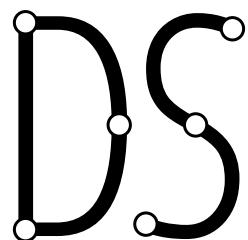


BACHELOR THESIS

**Localization and Angle-of-Arrival
in Bluetooth Low Energy**

Valentin Horst

Kiel, Germany 2021



KIEL UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
DISTRIBUTED SYSTEMS GROUP

Supervisor 1: Prof. Dr. Olaf Landsiedel

© Valentin Horst, 2021.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Kiel, 30. September 2021

Abstract

The Angle of Arrival (AoA) technique can be used to detect which direction a Bluetooth Low Energy (BLE) signal is coming from. There has been a higher focus on AoA using Switch Antenna Arrays (SAA) in recent years, possibly allowing for efficient indoor positioning. This thesis tested a widely-used indoor positioning system with a SAA using two AoA implementations, one pre-installed approximation based algorithm and one algorithm we implemented based on the mathematical background. We found the approximation based algorithm to perform well in indoor and outdoor environments. The mathematically based algorithm produced less accurate results. Comparisons to other research shows additional data processing does not yield better results than the approximation based algorithms. The most accurate angle estimation can be achieved by neural networks.

Contents

1	Introduction	1
2	Background	3
2.1	Bluetooth Low Energy	3
2.2	Indoor Positioning Techniques	4
2.3	Antenna Arrays	4
2.4	Constant Tone Extension	5
2.5	I/Q Sampling	6
2.6	AoA Estimation	7
2.7	Error Sources	9
2.8	Accuracy Improvements	10
3	Related Works	11
4	Implementation	13
4.1	System Design	13
4.2	Valid I/Q Samples	13
4.3	Basic AoA Implementation	14
4.4	TI AoA Implementation	14
4.5	Sampling Configuration	15
4.6	Sample Control	16
5	Results	19
5.1	Methodology	19
5.1.1	Software	19
5.1.2	Hardware	19
5.1.3	Node Roles	20
5.1.4	General Setup	21
5.1.5	Data Collection	22
5.2	Empirical Measurements	23
5.2.1	Indoor Daytime	24
5.2.2	Indoor Nighttime	24
5.2.3	Outdoors	27
5.3	Discussion	29
5.3.1	General Observations	29
5.3.2	Angular Accuracy	29
5.3.3	Movement	29
5.3.4	Data Processing	30
5.3.5	Neural Networks	30
6	Conclusion	33
6.1	Summary and Results	33

Contents

6.2 Future Work	33
List of Figures	35
List of Tables	37
Bibliography	39

Chapter 1

Introduction

Positioning systems offer a wide range of applications and have always been a focus of research. The problem of accurate positioning dates back millennia, when people used stars in the night sky for navigation. Today, the area of outdoor positioning has been partially solved by using Global Navigation Satellite Systems (GNSS) [1]. One of these systems, the Global Positioning System (GPS) has found widespread use. However, satellite based positioning systems require a direct line of sight. Thus, they suffer from accuracy loss in indoor environments. This led to the need for other solutions for indoor positioning. Unlike GPS for outdoor positioning, there is currently no standard system for indoor positioning [2]. The Internet of Things (IoT) becoming more popular increased the interest and demand in research for indoor positioning. The release of Bluetooth 5.1 in 2019 which included a new feature for direction finding was meant to fill the gap for an indoor positioning standard [3]. Prior Bluetooth location services relied only on signal strength to determine distances, [4]. Now, devices could give the direction to another device, allowing for improved accuracy.

Special interest was given to Bluetooth Low Energy (BLE) and Angle of Arrival (AoA) technology. BLE offers low cost, a low power consumption and widespread availability. AoA uses a Switch Antenna Array (SAA) to derive the angle of an incoming signal from phase differences. It offers the possibility of higher accuracy in comparison to other indoor positioning techniques such as signal strength based approaches.

Industry sectors with an interest in such a technology include hospitals, malls or manufacturing [1]. For example, hospitals could use it to track assets and find equipment faster [5]. Time could be of the essence in such environments. Another use could be wander management in elder care. Malls and airports could create more accurate direction finding apps to navigate more easily in them [6].

There are already companies offering BLE AoA based indoor localization products. They mainly target the enterprise market. For example, to track machinery or tools. However, one such company claiming to be a leading market figure has only 2600 employments of their system [7]. There are no apps on smartphones using BLE AoA since both Android and iOS don't fully support the BLE 5.1 direction finding as of writing of this thesis.

Texas Instruments (TI), one of the biggest technology cooperations in the USA, offers many Microcontroller Units (MCU) for different signal processing applications. Some MCUs offer an experimental setup for indoor positioning using BLE and AoA with accompanying software. Their development of these so-called development kits

1. Introduction

is still ongoing and in an early stage. Only one simple algorithm is implemented. They offer quick access for indoor positioning research and are therefore often used in this field of research [1][3][8].

This thesis used BLE based AoA with one such development kit to do the following:

1. Give a general overview of the technology and theory behind AOA using BLE.
2. Implement an AoA algorithm in Python which can easily be extended.
3. Test the development kits and the algorithms on their accuracy in different locations and compare them to other research results.

This thesis consists of 6 chapters. After the introduction, information on the background is given in Chapter 2. This includes basics in antenna technology, BLE and AoA calculations. This is followed by an overview of related works in Chapter 3. In Chapter 4 the implementation of our algorithm is described and compared to another algorithm by TI. The algorithms are tested in three different environments and discussed in Chapter 5. All measurement data taken and code produced in this thesis can be found on the GitLab server of the institute ¹. Finally, we give a conclusion to the thesis in Chapter 6.

¹<https://git.informatik.uni-kiel.de/stu116004/ble-aoa>

Chapter 2

Background

This chapter provides an overview of the technologies used throughout this thesis and lays the theoretical foundations for the implementation presented in Chapter 4. First, we give a brief introduction to BLE. Section 2.2 describes different indoor positioning techniques. In Section 2.3 the special antenna requirements for AoA are explained. Section 2.4 introduces the Constant Tone Extension (CTE) for BLE and Section 2.5 describes the I/Q sampling. The mathematics behind AoA estimation are explained in Section 2.6. Finally, potential sources of errors and methods to compensate them are described in Section 2.7 and 2.8.

2.1 Bluetooth Low Energy

BLE is a wireless technology independent of classic Bluetooth. BLE offers many advantages over other wireless technologies, making it especially viable to be used for indoor positioning. It offers sub-meter accuracy, a low power consumption and low cost, beating comparable technologies like Wi-Fi and Ultrawide Band [9].

The Bluetooth standard was originally proposed by engineers working for Ericsson (Lund, Sweden) in 1989 [1]. The aim was to 'unite' communication protocols. The name Bluetooth comes from a medieval Scandinavian king who united Denmark. The first specification, Bluetooth 1.0, was released in 1999. In 2004 Bluetooth was already installed on 250 million devices. In 2010 Bluetooth 4.1 was launched, introducing BLE technology. Finally, in 2019 Bluetooth V5.1 included support for direction finding.

BLE communication mainly works by sending small packets at regular intervals. Just as classic Bluetooth, BLE operates in the 2.4 GHz band. In contrast to Bluetooth, which uses 79 channels, BLE uses only 40 channels. Each channel's center frequency is given by $2402 + k \cdot 2$ MHz (for $k = 0, \dots, 39$) [10]. So every channel's center frequency is separated by 2 MHz. Adaptive frequency hopping is used to lower the effects of interference. Each consecutive packet is sent using a different channel following a pseudo random hopping sequence. Depending on environmental conditions, BLE can connect devices up to 10 m away from each other.

2.2 Indoor Positioning Techniques

In this section, the four main techniques for indoor positioning are described.

Received Signal Strength (RSS) uses a signal's power to determine the distance between a receiver and a transmitter. Wireless signals lose their power as they propagate. If the original signal strength is known, the distance to the sender can be estimated by the receiver from the received signal's power. There are clear benefits to this technique. It uses cheap hardware. Its implementation is also the easiest of all four techniques. But this method's accuracy is rather poor [1]. Experiments using RSS for indoor positioning observed an error of over half a meter [2]

Time of Arrival (ToA) sends a signal which time-stamped. From the signal's speed and the elapsed time, a distance can be calculated. This technique requires a precise synchronization between the transmitter and receiver. To achieve high accuracy, atomic clocks would be needed, making this technique expensive. Because of this, using ToA in indoor positioning is considered challenging [11].

By themselves, RSS and ToA provide only distance information. Given at least three transmitters whose positions are known in advance, the receiver's position can be calculated by trilateration. The following technique produce position information directly.

Time Difference of Arrival (TDoA) uses three stationary transmitters and one receiver. Only the transmitters need to be synchronized. Knowing the signal's speed and given the three time-stamped signals, three distances can be calculated. A benefit of this technique is, that no synchronization with the receiver is needed. Disadvantages are the use of more hardware and still having to synchronize the transmitters. This technique can give an estimation of the receiver's position, given the three distances and knowing the transmitters positions.

Angle of Arrival uses a single transmitter and a single receiver. The receiver is equipped with an antenna array. Small differences in the signal received by different antennas of the array can be used to estimate the angle of the receiver in relation to the transmitter. This technique uses no synchronization, which is a benefit in comparison to time based techniques. As seen in Figure 2.1, the location of the receiver can be calculated by triangulation. In contrast to all other techniques discussed above, only two transmitters are required.

2.3 Antenna Arrays

The AoA estimation technique requires special hardware in the form of so-called antenna arrays. Antenna arrays consist of two or more antennas positioned in a certain geometry. For example, the antennas of a Uniform Linear Arrays (ULA) are arranged in a straight line with a uniform distance to each other. The advantages

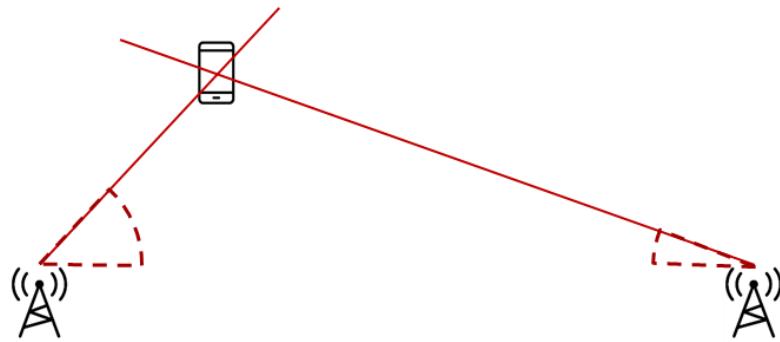


Figure 2.1: Given a setup with two transmitters, the position of a receiver (here represented by a smartphone) can be calculated by triangulation.

of using an ULA are its simplicity and low cost. There are other, more complex antenna arrays shown in Figure 2.2. They require more expensive hardware and more complex signal processing, but on the other hand they allow for more than one angle to be calculated. Furthermore, antenna arrays can be categorized as Switch Anntena Arrays (SAA). This means that only one antenna is receiving the signal at a time. In this thesis, a SAA with a ULA layout is used.

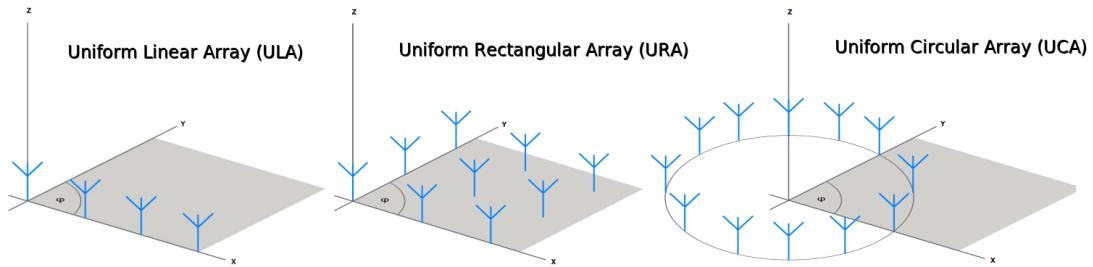


Figure 2.2: Three different antenna arrays designs. The blue symbols are the individual antennas with Cartesian axis showing exactly how they are spaced to each other. Taken and modified from [4].

2.4 Constant Tone Extension

The BLE Link Layer is responsible for advertising, scanning and maintaining connections. It has one packet format used for two packet types, advertising channel packets and data channel packets. The format is shown in Figure 2.3. Advertising channel packets can broadcast data if no full connection is needed, and they can discover and connect to peripherals. Data channel packets are used to send data in a full connection.

The Constant Tone Extension is an optional field in BLE Link Layer packets [12]. This field consists of the data used for AoA estimation. The PDU field on the other hand usually carries the actual payload. The CTE field only consists of 1 bits. This way all phase shifts of the signal are removed which would normally result from the

2. Background

non-uniform data structure. This causes the BLE radio to produce a single tone sine wave while transmitting this field.

The sampling structure of the CTE is specified in the Bluetooth 5.1 Specification [12]. The structure is shown in Figure 2.4. The duration of the CTE can be set between 16 and 160 μs . The first 4 μs are the Guard Period, the next 8 μs are the Reference Period. The signal from the remaining time of a CTE is used for the AoA estimation. It consists of alternating time slots, switch slots and sample slots. They can be 1 or 2 μs long. The switch slot is used for switching the antenna. The sampling is done in the both slots. BLE's physical layer offers several transmission modes. The mode BLE was released with is called 1M Phy or uncoded PHY and operates at a bit rate of 1 Mbit/s. Other modes do not support the CTE [1].



Figure 2.3: The BLE's Link Layer packet format using the standard transmission mode. Grey fields are optional. Taken and modified from [13]

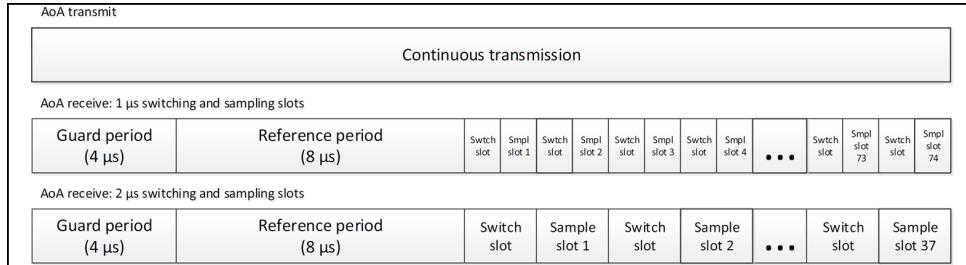


Figure 2.4: The sampling structure of the CTE as specified in [12]. Two sampling rates are available, 1 and 2 μs . In both cases, the Guard Period and Reference Period are 4 μs and 8 μs respectively. Figure taken from [12].

2.5 I/Q Sampling

To process a radio signal digitally, it needs to be sampled. A radio signal is a physical wave which can be received by an antenna. The radio signal is continuous whereas, a computer can only process discrete values. Therefore, the value of the signal is recorded at discrete points in time. This concept is referred to as sampling. Normal sampling does not provide information on a signal's overall amplitude and its current phase [14]. The phase of a wave is its position in its cycle, and the amplitude is its maximum magnitude. Thus, a more advanced version of regular sampling is used, namely I/Q sampling. In I/Q sampling, the real valued signal is extended to a complex valued signal whose real part is referred to as the I or "in phase" component and the imaginary part is referred to as the Q or "out of phase" component. The out of phase component is the same as the in phase component, but

shifted by 90 degrees. When an I/Q sample is viewed as a complex number in polar coordinates, the magnitude of the complex number is the signal's amplitude, and the argument of the complex number is the phase of the signal. This relationship is depicted in Figure 2.5.

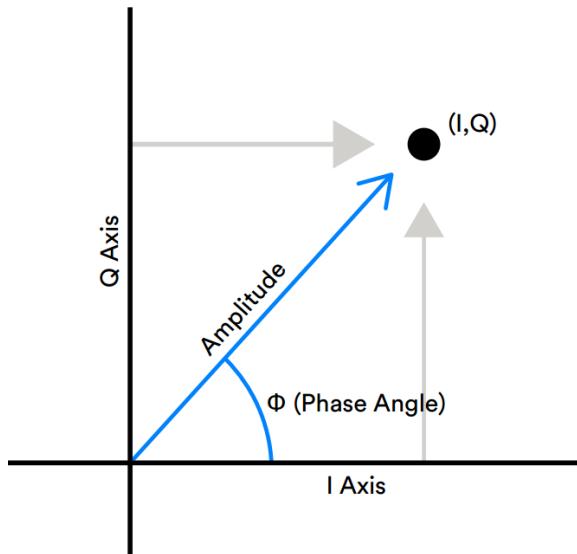


Figure 2.5: An I/Q sample marked as a black dot shown using Cartesian coordinates. The in phase component marks the horizontal axis and the out of phase component marks the vertical axis. The corresponding polar coordinates are shown in blue. Taken and modified from [3].

2.6 AoA Estimation

In this section, we will derive a formula for the estimation of the angle a radio signal arrived at when it was received by an antenna array. For simplicity, we consider the radio signal to be transmitted in the absence of barriers or other impeding factors in this section. An overview over potential error sources that can affect the AoA estimation in real world environments and how they can be accounted for will be given in the last two sections of this chapter.

Under the assumed ideal conditions, a transmitted signal travels in every direction at the speed of light. Thus, the wavefront of the signal can be viewed as a sphere expanding from the signal's source. As the sphere's radius grows, its surface area increases quadratically. Since the signal's power is distributed over this surface, the signal strength decreases. This effect is exploited by the RSS positioning technique explained in section 2.2. The AoA estimation is based on a different physical property of the wave, namely its phase.

The phase of a periodic signal indicates the signal's position in its cycle and is measured as an angle between 0 and 360 degrees or 2π radians. The wavefront is the set of all points where the signal has the same phase. Imagine an antenna at a

2. Background

fixed distance from the source. As the wavefront expands, the phase measured at this antenna will cycle through all possible values in a linear fashion. When the wavefront expanded by a full wavelength λ , the phase as measured at the antenna's location is back at its initial value.

Now imagine that we have a second antenna at a distance d from the first antenna. Suppose that the distance d is smaller than the wavelength λ . Under this assumption, the wavefront arrives at both antennas within the same cycle. Given the difference $\Delta\Phi$ of the phases measured at the two antennas, the distance r the wave must travel from one antenna to the next can be calculated as follows.

$$r = \frac{\Delta\Phi}{2\pi} \cdot \lambda \quad (2.1)$$

As can be seen in Figure 2.6, a right-angled triangle is formed, provided that the antennas are far enough away from the source for the curvature of the wavefront to be neglectable. The triangle's hypotenuse is the distance between the antennas d and the travel distance r is the side adjacent to the angle of interest. Thus, the angle Θ can be calculated as follows.

$$\Theta = \arcsin\left(\frac{r}{d}\right) \quad (2.2)$$

$$= \arcsin\left(\frac{\lambda\Delta\Phi}{2\pi d}\right) \quad (2.3)$$

The question that remains is how we can obtain the phase difference. In the previous section, the concept of I/Q sampling was introduced. We have seen that the argument of the I/Q sample when viewed as a complex number is the phase of the signal. Given one I/Q sample for each antenna as a complex number

$$x_1 = r_{I1} + j r_{Q1} \quad (2.4)$$

$$x_2 = r_{I2} + j r_{Q2} \quad (2.5)$$

we could compute the argument of x_1 and x_2 and subtract the resulting values.

$$\Delta\Phi = \arg(x_1) - \arg(x_2) \quad (2.6)$$

However, in the literature [8] it is common to compute the difference as follows.

$$\Delta\Phi = \arg(Z) \quad (2.7)$$

$$Z = x_1 \cdot x_2^* \quad (2.8)$$

That both methods are equivalent can easily be demonstrated when x_1 and x_2 are written in polar coordinates.

$$x_1 = a_1 e^{j\Phi_1} \quad (2.9)$$

$$x_2 = a_2 e^{j\Phi_2} \quad (2.10)$$

$$Z = x_1 \cdot x_2^* \quad (2.11)$$

$$= a_1 e^{j\Phi_1} \cdot a_2 e^{-j\Phi_2} \quad (2.12)$$

$$= a_1 a_2 e^{j(\Phi_1 - \Phi_2)} \quad (2.13)$$

$$\Delta\Phi = \arg(x_1) - \arg(x_2) \quad (2.14)$$

$$= \Phi_1 - \Phi_2 = \arg(Z) \quad (2.15)$$

2.7 Error Sources

In the previous section we have worked under the assumption that the signal is transmitted under perfect conditions. In reality this condition is not satisfied. This section lists possible sources of error and points out methods for how these errors can be accounted for.

The major source of interference noise stems from multipath propagation [1]. A single signal wave can reflect on surfaces, for example, windows, mirrors and tiles. This leads to both the reflected signals and the direct signal arriving at the receiver using multiple paths. It can result in constructive or destructive interference. Its effect on the received signal can be reduced using more complex algorithms.

Another error source comes from the analog-to-digital conversion done by an analog-to-digital converter (ADC). This form of noise is instrument-dependent [1]. More precise hardware leads to less prevalence of this error type. The downside to this is the higher cost of higher precision hardware.

The sampling rate at which the samples are taken does drift due to the imperfection of the hardware [15]. A later sample on the same antenna could be sampled a bit later or earlier than the sampling rate would suggest. The error does accumulate over the course of a measurement.

The switching of the antennas employed by SAAs also results in a measurement error. [1]. Antennas require a settling time. The resulting error decreases with time. This error is mostly dealt with by discarding a certain number of samples taken during a single snapshot.

The frequency band used by BLE is part of an internationally reserved portion of the radio spectrum. [4]. These frequency bands are referred to as the ISM radio bands and are intended for use in industrial, scientific and medical applications. No special license is required to operate in an ISM band. As a consequence, BLE has to share the frequency band with a variety of other widely used applications such as Wi-Fi. The overlapping frequency usage of BLE and Wi-Fi is shown in Figure

2.6. There is a different error from interference with Wi-Fi for every one of the 37 BLE channels used for data transmission [8]. Its effect can be lowered by adding a compensation for every channel.

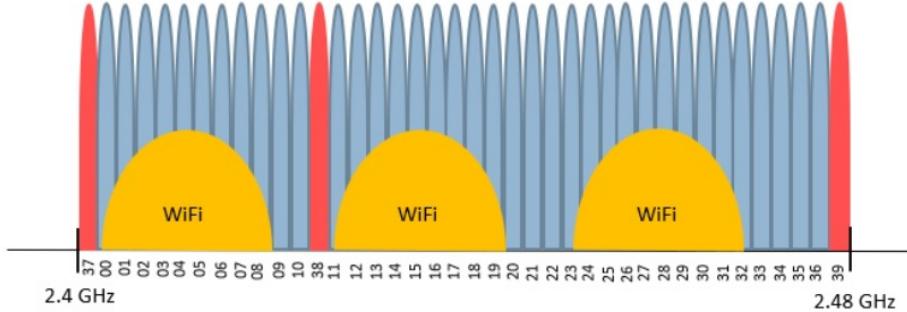


Figure 2.6: Both Wi-Fi and Bluetooth use the same ISM band. Blue are BLE's 37 data transmission channels. The three red channels are used for advertisements. Taken from [8].

2.8 Accuracy Improvements

The formula presented in Section 2.6 for AoA estimation does not compensate for any error. Angle estimation can be improved for better accuracy.

One option is to run data processing on the I/Q samples, phase differences or estimated angles. Another option is to use other algorithms for the AoA estimation. Two algorithms stand out for their accuracy and widespread use, the MUSIC algorithm and conventional beamforming. Due to the scope of this thesis, they won't be further discussed. They can be read about in more detail in the work of Atteya and Amer [1].

One processing framework was proposed by Hajiakhondi-Meybodi and Mohammadi [8]. They filter and compensate in all three major steps of the AoA estimation. First, a smoothing filter is run on the raw I/Q samples. This is mainly done to remove the impact of the multipath interference. In a second step, another filter is used on the phase differences to remove impact of the antenna switching. The resulting angle is lastly compensated for Wi-Fi interference and the use of different channels by applying a compensation vector.

Neural networks can also be applied for AoA estimation. Neural networks are 'learning algorithms'. They are trained on large data sets. A trained neural network recognizes patterns in the data. Errors in the BLE signal are such patterns, which the neural network could exclude from the resulting angle estimation. Further information on neural networks can be read in the work of Goodfellow et al. [16].

Chapter 3

Related Works

In this chapter, we want to give an overview over research with close proximity to our work. Most prior research using BLE for angle based localization was done since 2019, the year the Bluetooth 5.1 positioning feature was introduced.

A paper from Lund University [1] includes several tests using TI's development kits for BLE AoA. They found the setup with two receivers to not yield better accuracy. This is one reason why the more simple setup with one receiver and one transmitter was chosen for this thesis. Their tests in near ideal environments with reduced multipath signals resulted in angles close to the real alignment. A typical corridor turned out to be a more challenging environment. They conclude the technology to have room for improvement.

A group from Montreal [8] has implemented several filter methods to improve the accuracy of AoA estimations using the same hardware as we used. They claim their method to have significant potential in achieving more accurate angle estimations.

A paper from Cominelli et al. [17] titled 'Dead on Arrival: An Empirical Study of The Bluetooth 5.1 Positioning System', argue less optimistically about Bluetooth based positioning achieving commercial usability. One of their main concerns is the lack of security in the BLE packet structure, allowing attackers to mislead the system. The packet structure only allows checking correctness in the CRC part, as seen in Section 2.4. The CTE only follows after the CRC and is not checked for correctness. There is also no other check for interference. This makes BLE AoA indoor localization less viable for applications with high security requirements. Their tests showed BLE positioning within a few centimeters to be difficult.

Sambu and Won [18] used the same hardware as we did for BLE AoA estimations. They conducted tests with the pre-installed algorithm by TI in indoor and outdoor environments and achieved very accurate results overall. Their outdoor measurements had a standard deviation of 0.3 degrees and their indoor measurements showed a standard deviation of 1.1 degrees.

3. Related Works

Chapter 4

Implementation

In this chapter, the implementation of two alternative AoA estimation algorithms are presented. Section 4.1 gives a general overview of the system used for the angle estimations, to clarify at which steps the two different AoA algorithm are used. A common preprocessing step is described in Section 4.2. Both algorithms are explained in more detail in Section 4.3 and 4.4. Lastly, the settings the hardware was configured with are discussed in Section 4.5 and 4.6.

4.1 System Design

This section is meant to give an overview of the general system design to acquire AoA estimations. Further details on the utilized software and hardware are given in Section 5.1. Generally speaking, our system setup is composed of three main components. There is one transmitter and one receiver device. The receiver is connected to a laptop and is equipped with an antenna array.

The transmitter is sending BLE packets with the CTE enabled. A Python script running on the laptop commands the receiver to establish a connection to the transmitting device. The receiver board switches between its antennas and samples the signal. The MCU connected to the antenna array calculates the AoA estimations internally using a pre-installed algorithm. The applied algorithm is explained in Section 4.4. The angle estimations are sent to the laptop, where they are written into a log file.

Using an alternative Python script, the receiver can be instructed to send the raw I/Q samples to the laptop instead. The collected I/Q samples are written into a Comma Separated Values (CSV) file. This file is processed on the laptop using a basic AoA algorithm as explained in 4.3.

4.2 Valid I/Q Samples

Both algorithms have in common that they first need to decide which I/Q samples to use for AoA estimation. Recall the sampling structure as shown in Figure 2.4. We consider an ULA consisting of $N_e = 3$ antennas with a distance of $d_0 = 3.5\text{cm}$. Our hardware records the I/Q samples at a rate of 4 MHz. One CTE is set to $160\ \mu\text{s}$. However, the first $4\ \mu\text{s}$ belong to the Guard Period in which no samples are collected. Therefore, for one BLE packet $(160\mu\text{s} - 4\mu\text{s}) \cdot 4\text{MHz} = 624$ samples are collected. The first 32 samples of a packet are discarded as they are from the Reference Period, in which no switching is done. Afterwards, the antennas take

turns sampling. Each antenna is active for $4\mu s$. Therefore, each antenna samples $4\mu s \cdot 4MHz = 16$ I/Q samples before switching to the next antenna in the array. These 16 samples are referred to as a snapshot. The last antenna switches back to the first. As explained in Section 2.7 the first 8 samples of a snapshot are discarded because they belong to the switching slot. We conducted several small tests to verify that using only the 8 samples from the sample slots to remove the impact from the switching error performs best.

4.3 Basic AoA Implementation

$\Theta_{average}$ is calculated as the mean of a list of Θ . One Θ is calculated for each pair of antennas and I/Q sample pair as shown in Formula 2.2. The distance between antenna a and b is calculated as $d = |b - a| \cdot d_0$. The wavelength $\lambda = \frac{1}{f}$ is calculated from the BLE channel's center frequency f , as given in Section 2.1. Since the phase of an I/Q sample is within the range of 180 to -180 degrees, we need further logic for to get the result from the right quadrant. This is done by the `atan2` function. Its process is shown in Table 4.1.

The `arcsine` function takes values from -1 to 1. The basic AoA algorithm does only guarantee this requirement if $\Delta\Phi \leq \frac{2\pi \cdot d}{\lambda}$ is the case. We observed that the I/Q samples from the second and third antenna often led to a wrong values for the phase differences. Thus, only the remaining two column pairs are used for the calculations. The same approach has been used by the implementation of TI. Nevertheless, measurement errors can cause phase differences to take on invalid values. The `arcsine` function in Python returns `nan` in this case. Thus, these values must be discarded.

4.4 TI AoA Implementation

The implementation used by the software provided by TI deviates from the basic version [15]. The handling of the samples follows the same structure as the basic AoA implementation. The phase differences are calculated using an approximation of the `arctan` function. This is done for better performance. Since the sampling frequency itself fluctuates, a compensation has been included. This is done by comparing the change of phase from samples of the first antenna in a pair with the same antennas samples from its previous snapshot. This is also supposed to lower the impact of the

Table 4.1: I/Q require additional consideration if the phase is calculated from Cartesian coordinates. These depend on the value of the I component and Q component.

	I > 0	I < 0
Q >= 0	phase = arctan(Q/I)	phase = 180 + arctan(Q/I)
Q < 0	phase = arctan(Q/I)	phase = -180 + arctan(Q/I)

error from the ADC. The compensated phase differences are not converted to angles by the use of trigonometry. Instead, the averaged compensated phase differences are moved by an offset value and multiplied by a gain value. The result is then compensated for the frequency error by a factor depending on the BLE channel. The code contains a lookup table for ULAs containing these correction factors. This is just a minor change, with the highest value being 3 degrees. This process turns the phase difference into the estimated angle for the transmitter without the need to know the distance between the antennas or the wavelength of the signal. Another benefit is the increased performance. The method of acquiring the offsets and gains is shown in Figure 4.1. Measurement done in a lab were modified to fit the expected angles. The lab conditions are not further described. It can be assumed, that background noise and errors from the hardware itself get accounted for this way. This method is not expected to have an impact on errors from external sources. The resulting angle does actually not consider the pair consisting of the second and third antenna. It's included in the all calculations, but then omitted in the last step.

4.5 Sampling Configuration

The sampling configuration was set in the Python scripts provided in TI's software. The documentation can be viewed at [15]. There are eight basic CTE settings which can be modified. The following settings were used in every measurement. No measurement was done using any other configuration. Since many of them were found to don't work correctly, they are explained in more detail in this section.

1. BLE supports one and two microseconds for the switching slot duration for switching the antennas. Using $1 \mu s$ with the used hardware is deprecated. Switching slots were therefore set to $2 \mu s$.
2. The sample rate sets the frequency at which the CTE should be sampled. The hardware supports 1 - 4 MHz. A higher resolution leads to a better averaging in the angle estimation, as well as better results using smoothing algorithms on the raw data. All settings were tested with 1 - 3 MHz resulting in unusable data with no resemblance to two waves for the I/Q data. Therefore, only the highest possible setting of 4 Mhz was chosen.
3. The sample size can be either 8 or 16 bit, with 8 being used for the measurements.
4. The sample control setting is discussed separately in the next section.
5. Lastly, one setting enables the sampling itself. The pattern length sets the amount of antennas to be used, and the antenna pattern setting chooses the antennas in the antenna array to be used. Sampling was enabled, and the antennas 0, 1 and 2 were used, so all antennas of the left antenna array.

4.6 Sample Control

The sampling control in the sampling configuration is used to set which antenna is used, as well as setting the raw data filter. Only antenna array 1 was used. The other option would be to use antenna array 2. Even though the antenna board was designed with the intention of using both antenna arrays simultaneously, it is only possible to use a single antenna array at a time. TI cites avoiding instabilities as their reason for removing the option of using both antennas in recent versions of their software [19]. It would still be possible to use both antennas by implementing the array switching and logic for the processing yourself. This possibility wasn't further followed because of the complexity in its implementation, and TI themselves citing having problems with this feature's implementation.

The filter option is described to either filter out the samples taken during the switching and save the remaining samples taken during the sample slots, or to save every sample taken during the CTE of a BLE packet [15]. One pair, consisting of a sample slot and switching slot, lasts $4 \mu s$ and takes 16 samples using a sampling frequency of 4 MHz. Using no filter results in 624 I/Q samples for every BLE packet as expected. Turning the filter on leads to 90 samples being saved for every BLE packet. If every sampling slot was filtered out, 312 samples would be expected. Having to remove the Reference Period in the angle estimation calculations leads to a too small number of samples and a higher error in the estimation using only 90 samples. Without knowing which I/Q sample was sampled by which antenna makes processing these samples impossible. Thus, no filtering was used in the measurements. Both the AoA implementations remove the 8 samples taken during the switching slots instead of using a filter.

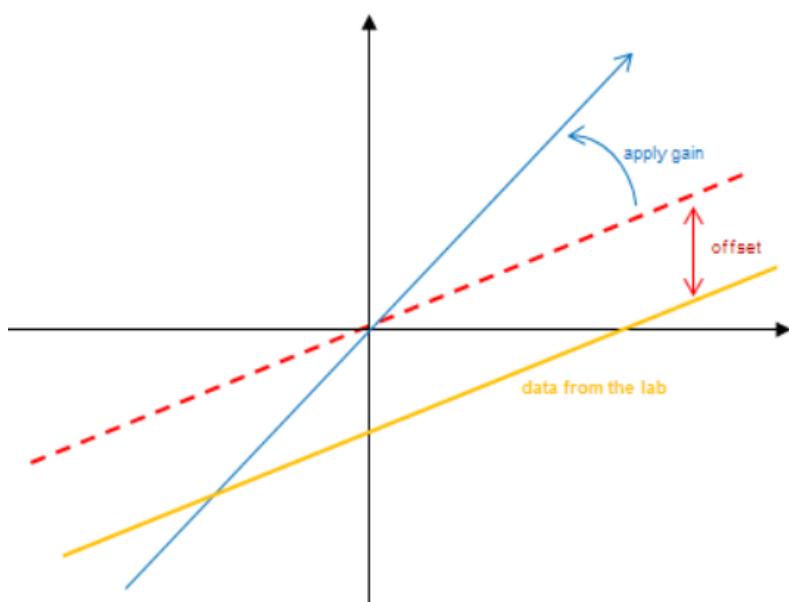


Figure 4.1: Method for finding the values needed to substitute the \arcsin function part of the calculation. The x axis shows the actual angle, and the y axis shows the phase difference for the yellow line and estimated angle for the blue line. Lab measurements were moved by an offset and multiplied by a gain to match the expected result. Taken from [15].

4. Implementation

Chapter 5

Results

In the following chapter, the implementations we presented in Chapter 4 are evaluated. First, we give an introduction in Section 5.1 to the hardware and software setup used. Afterwards, we show the evaluation with the empirical measurements in different environments in Section 5.2 and discuss the results in Section 5.3.

5.1 Methodology

5.1.1 Software

The software used with the hardware was TI's SimpleLink™ CC13x2 and CC26x2 software development kit (SDK). The SDK is a software package designed for TI MCUs for sub 1 GHz and 2.4 GHz application development. The version 5.10.00.02 of the SDK was used. Even though new versions were regularly released, no update was done during this thesis' work to avoid further complications. An important aspect to mention is the instability of the software due to its early experimental stage. Measurements were at times crashing in more than 90% of cases. This happened at various stages of a measurement process. The initial connection through advertisement packets didn't complete, or crashed at random points in the AoA calculation. The hardware's firmware was updated at least once a week, with crashes occurring less often with further updates.

5.1.2 Hardware

Two products from Ti listed below are used.

1. The wireless microcontroller used is the SimpleLink CC2652R LaunchPad supporting Bluetooth, Zigbee and Thread [20]. Hereinafter, it will be referred to as "LaunchPad". TI also provides software support with their SDK [21].
2. The second piece of hardware used is the BOOSTXL-AOA [22]. It is shown in Figure 5.1. The board contains 2 ULAs with 3 dipole antennas each and can be connected to the LaunchPad. Hereinafter, it will be referred to as "Antenna board".

The antennas of a single ULA are placed 3.5 cm apart. A single antenna can receive signals from 360 degrees. The way the AoA calculation works, there is no way to tell if the signal originated from the front or from behind. The Antenna boards employ metal grounding next to the antennas to "attenuate" the signal from behind the ULA [1]. This gives the Antenna board a way to differentiate the front and back

signals using RSS. The front sides have a higher RSS. This could be employed, but has currently been disabled in the SDK, since only one ULA can be used at a time.

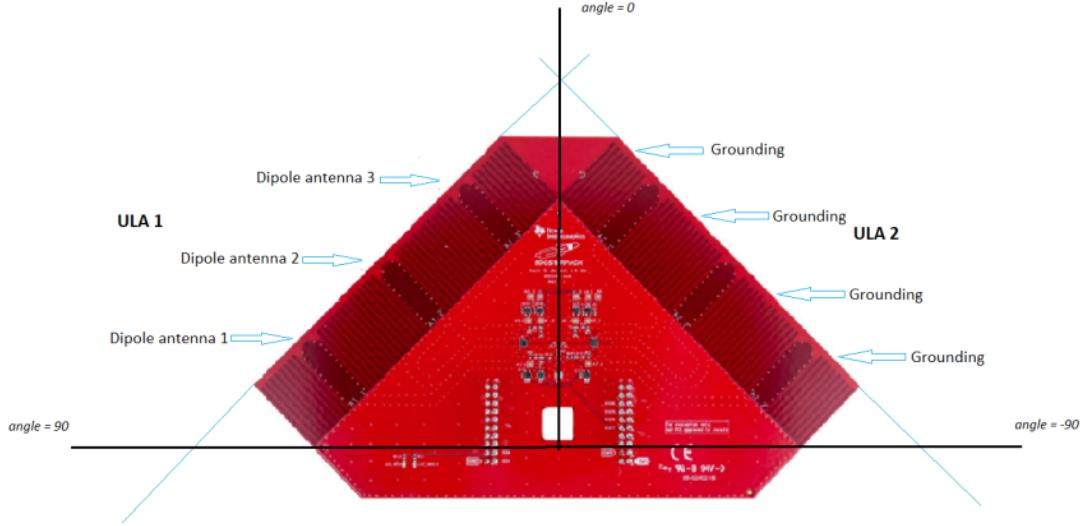


Figure 5.1: The Antenna board with 90, 0 and -90 degrees marked as oriented by the SDK. The angles are shifted by 45 degrees in comparison to the orientation of the first ULA. Also shown is the antenna position on the first ULA and grounding on the second. Each is also present on the other ULA. Taken from [1].

5.1.3 Node Roles

TI's SDK contains three different software examples which can be flashed onto the LaunchPads. The examples demonstrate how to build a Real Time Locating System (RTLS) Network. An RTLS Network uses several Nodes to not only get one angle, but an estimation for a location, as described in Section 2.2. In such a network, the nodes can be assigned different roles. To explain the used setup, we give an overview of the roles a node can have in an RTLS.

1. **The Central** is implemented in the `rtls_master` example. It establishes a connection with the Peripheral and does the I/Q sampling.
2. **The Peripheral** is implemented in the `rtls_slave` example. It sends data packets with CTE. The Peripheral is the device to be located.
3. **The Passive** is implemented in the `rtls_passive` example. It can join the connection between the Central and Peripheral. It follows the connection and also collects I/Q samples. The Passive is optional.
4. **The Node Manager** is a Laptop using a Python script to interact with the other nodes. It is used to run the measurements and log different data.

5.1.4 General Setup

Topology

TI offers a variety of setups and operation modes [13]. A connection mode and a connectionless mode are available. Furthermore, the setup can use the Passive or run without one. The Passive and Central can both give the angle estimation, or either one of them alone.

The setup used consists of one LaunchPad mounted with an antenna board as the Central and another LaunchPad as the Peripheral, illustrated in Figure 5.2. To actually connect the LaunchPad to the antenna board, it was necessary to re-solder capacitor C51 on the LaunchPad [23]. Only the Central is connected to the Node Manager, a laptop. The Central functions as the locator, giving the raw I/Q data or angle estimations. The Peripheral is the node sending the CTE signal and whose angle in relation to the Central is to be estimated.

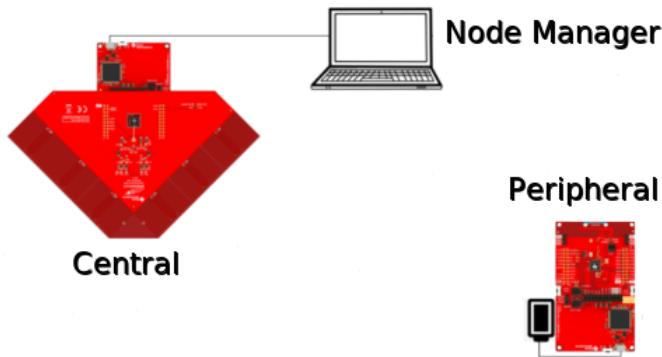


Figure 5.2: The general hardware setup used for all measurements. A RTLS Central node is connected to a Laptop. The RTLS Peripheral sends the signal to be located

Manual positioning

The hardware was placed on empty cardboard boxes measuring $25 \times 45 \times 50\text{ cm}$ as seen in Figure 5.2. One box for the transmitter and one box the receiver. This was done to allow for a more even signal propagation. Both were placed at the boxes' edges, facing each other. The laptop that was connected to the receiver was operated from a table or standing from 1 m to the left of the receiver facing the transmitter. Measurements were done from -90 to 90 degrees in steps of 30 degrees.

The coordinate system used by TI differs from the coordinate system assumed by the basic AoA algorithm. As shown in Figure 5.1 the TI algorithm gives the AoA from the middle between both antenna arrays. In the coordinate system of the basic AoA algorithm, the 0 degree line is perpendicular to the three antennas. Thus, it is shifted by 45 degrees anti-clockwise with respect to the TI coordinate system. To place the ULA precisely, stretched yarn and a triangular ruler were carefully used.

5. Results

The point of transmission from the transmitter was chosen as the middle of the 2 cm long antenna on the MCU. For the basic algorithm, the reference point for the left antenna array was chosen as the center of the middle antenna. The reference point for the TI algorithm was chosen as the middle of the antenna board. The placement error is assumed to be around 2 degrees. Due to the hardware being lightweight, all cables needed to be fixed in position by using tape for every angle. Sudden gusts of wind are easily able to move the hardware, requiring measurements to be discarded. Weights couldn't be used, since they could negatively impact the signal's transmission. These factors make it difficult to create large amounts of measurement data.



Figure 5.3: The antenna board positioned on a box before a measurement is taken.

5.1.5 Data Collection

The LaunchPad's RF Core does the I/Q sampling from the tone send by the Peripheral. The RF Core is the transceiver used by the LaunchPads. The AoA estimation is done by the AoA Driver, which is part of the software run on the MCUs. TI offers several Python scripts in the SDK for interacting with the nodes. Three different modes can be enabled inside the scripts to extract the data at different points of the calculation [15].

1. AOA_MODE_RAW allows the logging of the unmodified I/Q samples.
2. AOA_MODE_PAIR_ANGLE logs the phase differences between all three possible antenna pairs using the TI algorithm.
3. AOA_MODE_ANGLE logs the angle estimations by TI's algorithm at the end of its calculation.

Only one of these data sets can be logged at a time given TI's script. The angle mode was used to log the final angles calculated by the TI algorithm. The raw mode was used to get the I/Q samples for the basic algorithm. For this purpose, the I/Q data was written into a CSV file like shown in Figure 5.4. The CSV file contains additional information like the packet number, signal strength and channel used.

	A	B	C	D	E	F	G
1	pkt	sample_idx	rssi	ant_array	channel	i	q
2	0	0	-45	1	22	-33	-80
3	0	1	-45	1	22	-73	40
4	0	2	-45	1	22	47	72
5	0	3	-45	1	22	70	-49
6	0	4	-45	1	22	-50	-71
7	0	5	-45	1	22	-67	53

Figure 5.4: Raw I/Q data exported into a CSV file from a measurement.

5.2 Empirical Measurements

The basic algorithm and TI algorithm were tested in the following three environments:

1. indoors at the evening,
2. indoors late at night and
3. outdoors.

The environments were chosen to test the impact of different error sources on the AoA estimation for both algorithms. Why these particular environments were chosen is explained in more detail in the following three sections.

Two plot types were used for visualization: box plots and error plots. In case of the error plots, the error is given as absolute deviation from the actual angle.

$$\text{error} = |\Theta_{\text{average}} - \Theta_{\text{actual}}| \quad (5.1)$$

Θ_{actual} is the actual angle from the antenna board to the transmitter and Θ_{average} is the average of all angles calculated for all measurement done for one actual angle. This error is shown as error bars on the ideal value in the error plots.

The box plots show a box for the 25th percentile or First Quartile (Q1) and the 75th percentile or Third Quartile (Q3) for all measurements of an actual angle. 75% of all angles are below the upper part of the box and 25% of all angles are below the lower part of the box. A line marks the median of all values inside the box. Extending from the boxes are whiskers marking the furthest value within 1.5 times the interquartile range (IQR). The IQR is calculated as $IQR = Q3 - Q1$. Any remaining measurements are outliers and are marked as circles.

5.2.1 Indoor Daytime

The indoor measurements were done deliberately in an environment with expected interference from Wi-Fi and multipath signals. This was done since the expected applications for BLE AoA for example in hospitals or airports would also have to deal with many error sources of this kind. The environment chosen was a living room, with one side having three floor to ceiling windows. The measurements were done in the afternoon with a download running on the laptop to create Wi-Fi interference. The setup was placed 1.5 m away, parallel to the windows. The errors for both algorithms are shown in Figure 5.5 as an error plot. We can generally see that the TI implementation offers better result across all angles. As mentioned in Section 4.3, there is no guarantee for the AoA estimation of the basic algorithm to return a valid value if the phase differences get too large. That is the reason for the missing data on -90 degrees for the basic algorithm. All values for the edges show a high error. The estimations show an increasing error the nearer the angle is to -90 or 90 degrees. Both algorithms perform well from -30 to 30 degrees. The TI algorithm shows a high precision as seen in Figure 5.6. Exceptions are 90 and -90 degrees, which have a lower precision. The basic algorithm on the other hand shows a lower precision and drift to 0 degrees over all angles. Excluding 90 and -90 degrees, the basic algorithm shows a standard deviation of 8.8 degrees and the TI algorithm 2.0 degrees.

5.2.2 Indoor Nighttime

The environment used was the same living room at the same position as before. The Wi-Fi was turned off and the measurements were started at 3 am. This way, Wi-Fi interference was meant to be reduced. As seen in Figure 5.7, all angle estimation of the basic algorithm show a higher error than the basic algorithm's estimation again. The edge cases at 90 and -90 degrees don't show the far lower accuracy as in the daytime measurement, seen in Figure 5.8. The basic algorithm shows lower precision and an overall drift to 0 degrees over all measured angles in comparison to the TI algorithm. Excluding 90 and -90 degrees, the basic algorithm shows a standard deviation of 8.4 degrees and the TI algorithm 1.8 degrees.

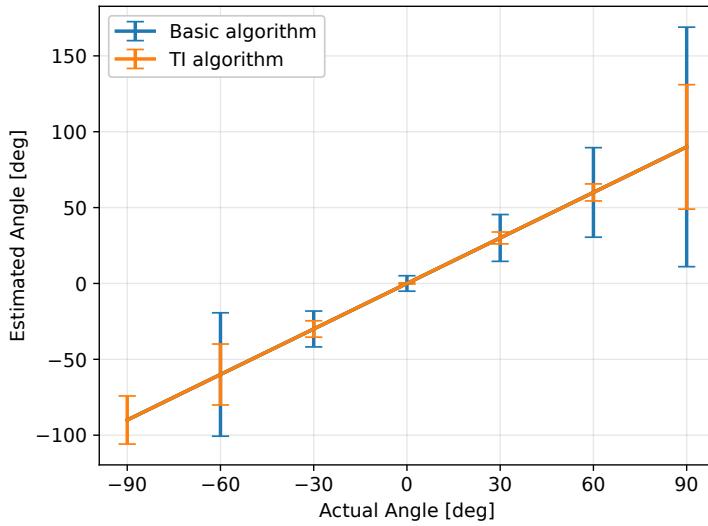


Figure 5.5: The error plot for the daytime measurements.

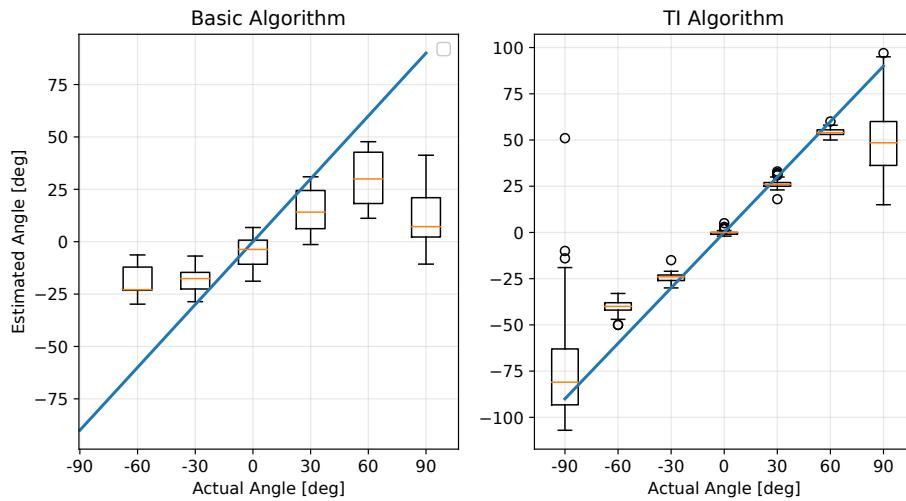


Figure 5.6: Boxplots for the basic algorithm (left) and the TI algorithm (right) for measurements done in an indoor environment during daytime.

5. Results

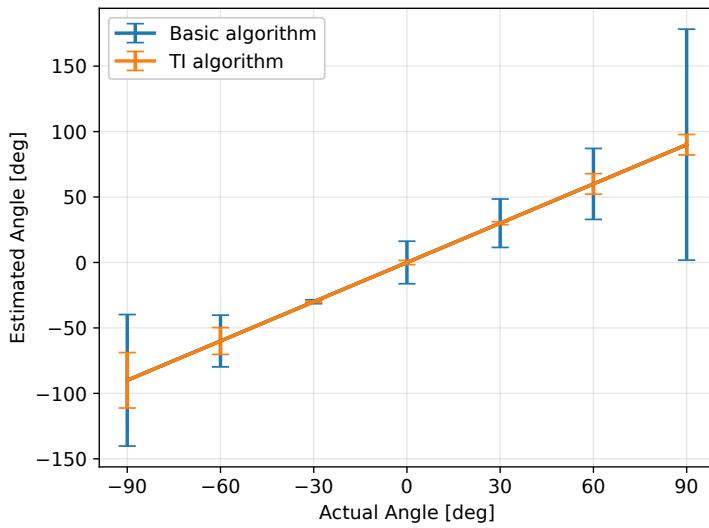


Figure 5.7: The error plot for the nighttime measurements.

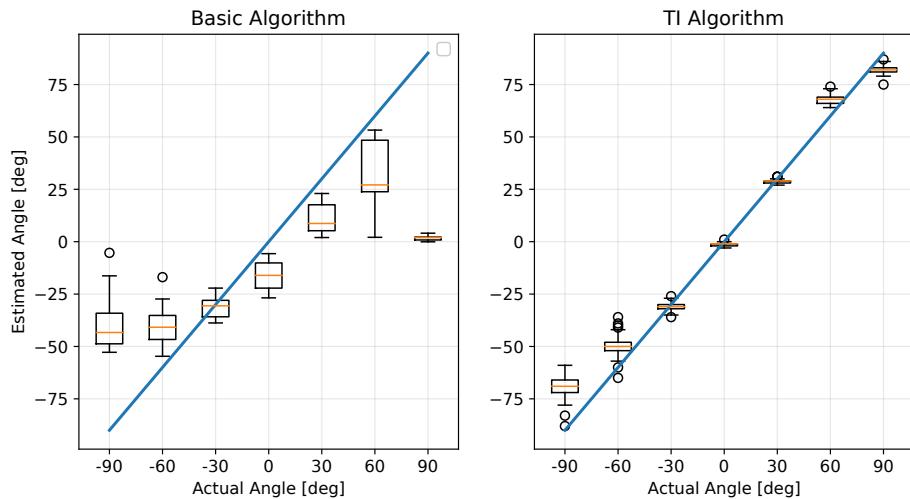


Figure 5.8: Boxplots for the basic algorithm (left) and the TI algorithm (right) for measurements done in an indoor environment during nighttime.

5.2.3 Outdoors

Normally, an anechoic chamber is used to test an antenna array with minimal external error sources [1]. The outdoor environment was chosen as the next best environment available for this purpose. No Wi-Fi signal was detected and no reflective surface was in sight. As seen in Figure 5.9, all angle estimation of the basic algorithm show a higher error than the TI algorithm's estimation again. However, the errors are lower compared to the indoor measurements overall. The TI algorithm shows a higher precision again as seen in Figure 5.10. The basic algorithm's boxes in Figure 5.10 are overall more than double as high and show a drift to 0 degrees again. The drift is less prevalent in comparison to the indoor measurements and only strongly affects -90, -60 and 90 degrees. Excluding 90 and -90 degrees, the basic algorithm shows a standard deviation of 4.8 degrees and the TI algorithm 1.6 degrees.

5. Results

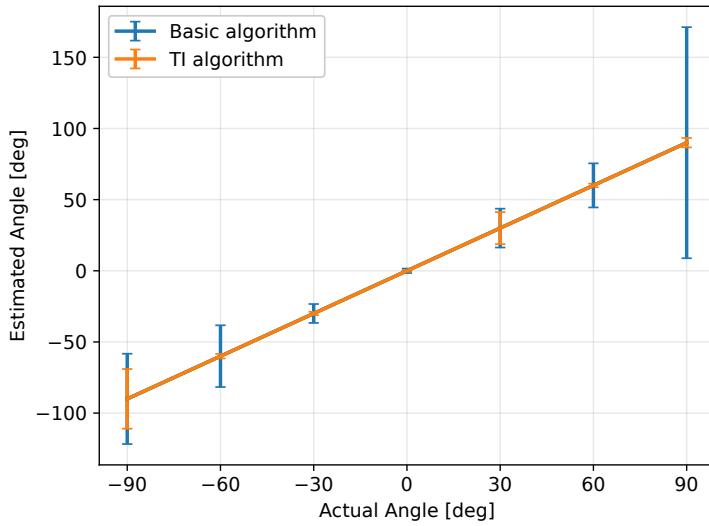


Figure 5.9: The error plot for the outdoor measurements.

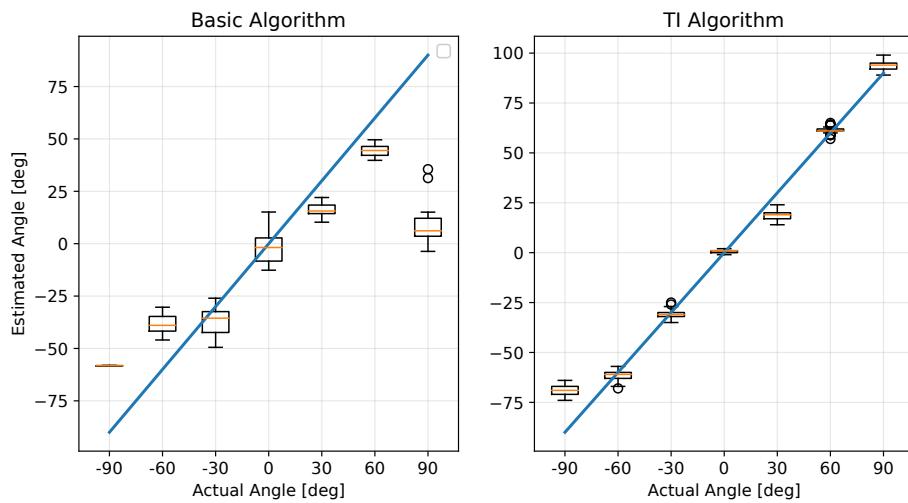


Figure 5.10: Boxplots for the basic algorithm (left) and the TI algorithm (right) for measurements done in an outdoor environment.

5.3 Discussion

5.3.1 General Observations

The overall better accuracy for both algorithms at nighttime compared to daytime is mostly likely due to the reduced Wi-Fi interference at night. Another accuracy improvement is observed in the outdoor measurements compared to the indoor measurements. The most likely explanation for this improvement is the reduced multipath error. In total, the indoor environment performs worst at day, better at night and the outdoor environment performed best. Additionally, improvements in the standard deviation can be observed in the same order. Our findings suggest the same impact of the multipath error and Wi-Fi interference on the measurements' precision in these environments. We suspect the difference in accuracy of the basic algorithm and the TI algorithm to be the result of the missing sampling drift compensation in the basic algorithm and the non-linearity, as discussed in Section 5.3.2. We suspect the sampling drift to be the main contributor to the error in our estimations using the basic algorithm. We also suspect the sampling drift to be responsible for the drift to 0 degrees observed in the basic algorithm estimations.

5.3.2 Angular Accuracy

All measurements show a high error near the edges, where $\Theta > 60$ degrees or $\Theta < -60$ degrees. Cominelli et al. [17] and Sambu and Won [18] used the same hardware and also observed this behavior. They argue the phase difference to be almost random when the signal source is near to the axis of the antenna array. The reason for this isn't stated. We suspect the antenna array's grounding to be at fault. In case of the basic algorithm, the error at the edges is influenced by the fact that the arcsine function is non-linear. The slope of the arcsine function increases near the edges. Therefore, phase differences need to be more precise the further the values approach the end of the arcsine's domain. This increases the error's effect on the angle estimation. Consequently, implementations using the direct mathematical background are unsuited for a wider angle range.

5.3.3 Movement

Our tests and most tests done by researchers haven't looked into AoA estimation of moving object. However, most commercial applications would use AoA for constantly moving objects, as explained in Chapter 1. The SDK offers an application giving live feedback using the TI algorithm. We observed, that the AoA estimations using the live feedback stabilizes in a few seconds after changing the location of the transmitter. This is the result of the TI algorithm being optimized for performance. Neural Networks and MUSIC also offer fast computation, making them viable for live feedback applications [24]. We expect the data processing framework designed by Hajiakhondi-Meybodi et al. [8] and similar approaches to be less viable for live feedback due to having several additional computational steps involved.

5.3.4 Data Processing

Hajiakhondi-Meybodi et al. [8] designed a processing framework build on their own implementation of the basic algorithm for AoA. They apply further processing on the I/Q samples, the phase differences and the angle estimations. A test of their processing method can be seen in Figure 5.11. They haven't given information on the test environment and how they calculated estimation errors. Comparing their results to our indoor measurements, a similar error for the basic AoA algorithm can be observed. The processing method seems to reduce the error to about half the error of their implementation of the basic algorithm. In contrast, the TI algorithm produced much lower errors than that in our findings. Therefore, we conclude the data processing method to perform worse than the TI algorithm. Whether the conclusion is true remains to be tested in additional experiments.

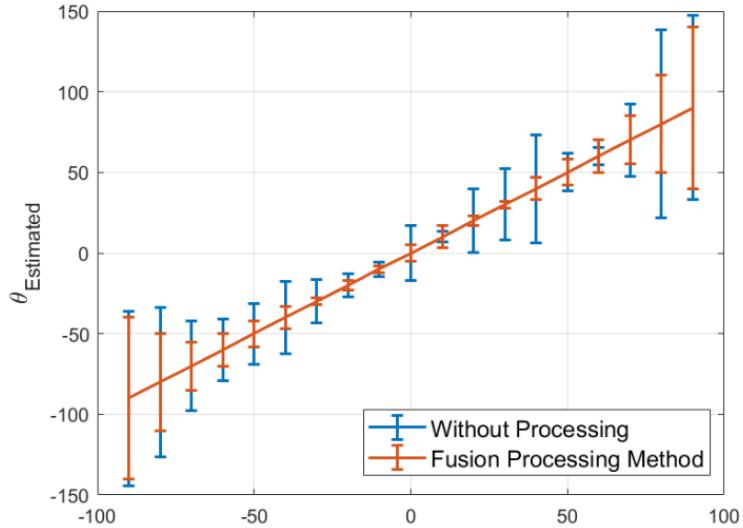


Figure 5.11: Results of a test done by Hajiakhondi-Meybodi et al [8]. Shown in blue are angle estimations errors of a basic AoA algorithm. Shown in orange are angle estimation errors with additional data processing.

5.3.5 Neural Networks

HajiAkhondi-Meybodi et al. [24] designed a BLE AoA neural network framework. They tested the framework with one experimental testbed. They divided a $5 \times 5m$ square room into 81 square zones. The transmitter was put on a movable object sending the BLE packets. Four antenna arrays and cameras were positioned at the corners of the room. The antenna arrays gathered I/Q samples while the camera system was used to obtain the transmitter's actual position in the room. They gathered 76,545 training points for training the neural network. Another 10,206 were used to then test the neural network. The test's results for a rectangular movement of the transmitter are shown in Figure 5.12. They show a clear difference in accuracy for the neural network and MUSIC algorithm, with the neural network being more accurate. HajiAkhondi-Meybodi et al. [24] state an accuracy of 87% for the neural network. The framework discussed in Section 5.3.4 is stated to be only

59% accurate. Of all methods for better accuracy stated in Subsection 2.8, neural networks can be considered the most effective method, based on the test result from HajiAkhondi-Meybodi et al. [24].

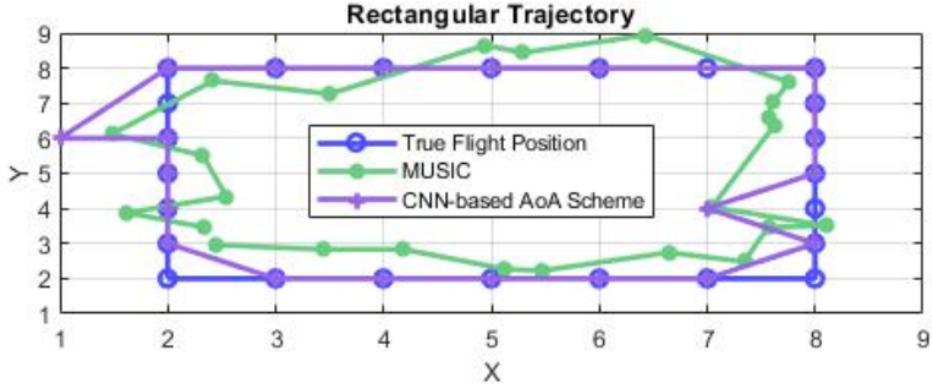


Figure 5.12: Results of a test done by HajiAkhondi-Meybodi et al [24]. The room was divided into 9×9 squares. The transmitter moved in a rectangular trajectory. The positioning from a camera system is shown in blue. A positioning estimation using MUSIC is shown in green. The violet line marks the estimations from a neural network.

5. Results

Chapter 6

Conclusion

In this final chapter, we conclude the thesis by summarizing our work and results in Section 6.1. Furthermore, an outlook is given on possible improvements for BLE AoA technology in Section 6.2.

6.1 Summary and Results

The goal of the thesis was to implement an AoA algorithm in Python that allows to implement improvements more easily than the highly optimized C code of the TI algorithm. In Chapter 2 we familiarized ourselves with BLE and the theory behind the AoA estimation. Based on theoretical background, we implemented a basic algorithm for AoA estimation ourselves and comprehended the algorithm by TI. We used development kits by TI to collect empirical measurements in Chapter 5 and used the collected data to compare the algorithms. We also compared the results to other research in BLE AoA. Despite our algorithm adhering to the underlying mathematical theory, it delivered poor results. The TI algorithm on the other hand performed considerably better and even outperforms methods using more involved data processing methods. However, a neural network based approach currently shows the best results. In summary, we achieved our goal of implementing a Python based AoA algorithm. However, given its poor results, we would recommend against continuing work on our basic algorithm. Instead, it seems to be more promising to use an approximative approach like TI's algorithm or neural networks to achieve more accurate AoA estimations.

6.2 Future Work

BLE AoA is still in its infancy and offers many opportunities for further research. Due to the findings of this thesis, we don't suggest further work on data processing or algorithms similar to the basic implementation. Instead, the focus should lie on approximation based algorithms, MUSIC or neural networks. Some questions can be further studied for these three algorithmic approaches. An increase of transmitters would require tests on their effect on each other's AoA estimation. How could they be scheduled for optimal estimations? Most environments for commercial use like hospitals, malls and airports would have a wide variety of objects obstructing the line of sight. Their effect on the signal could be studied, as well as ways to deal with an increase of obstructions.

Another area of interest will be BLE AoA integration into smartphones. We predict for both major smartphone systems to support the BLE 5.1 positioning feature in the

6. Conclusion

next few years. It's unlikely for antenna arrays to be integrated directly into smartphones. Still, it would require new BLE 5.1 compatible hardware to be installed, potentially slowing a widespread adaption. An obvious opportunity would be the development of AoA based apps. All current findings would have to be retested on phones, since they offer additional challenges. The signal would be distorted by traveling through the outer parts of a smartphone. Another error source could be a user's hand holding the smartphone.

Neural Networks for BLE AoA can be further analyzed in many aspects. How well does a trained and precise Neural Network estimate the AoA for other environments? Can it be trained to work on specific room types like hospital rooms without the need to be retrained? Could the training be sped up in certain environments, allowing for a faster installation of a Neural Network? Can enough training data be acquired without an expensive and fully automated system as used by HajiAkhondi-Meybodi et al. [24]?

List of Figures

2.1	AoA can be used to get the position of a receiver by triangulation.	5
2.2	Three different antenna array designs that can be used in AoA measurements.	5
2.3	The packet format used by BLE's link Layer.	6
2.4	The sampling structure of the CTE.	6
2.5	A single I/Q sample shown as Cartesian and polar coordinates.	7
2.6	The frequency used by both Wi-Fi and BLE.	10
4.1	Method for finding the values needed to substitute the arcsine function part of the calculation.	17
5.1	The Antenna board with 90, 0 and -90 degrees marked as oriented by the SDK.	20
5.2	The general hardware setup used for all measurements.	21
5.3	The antenna board positioned on a box before a measurement is taken. .	22
5.4	Example CSV file ouput of a measurement.	23
5.5	The error plot for the daytime measurements.	25
5.6	Boxplots for the TI algorithm and the Basic algorithm for measurements done in an indoor environment during daytime.	25
5.7	The error plot for the nighttime measurements.	26
5.8	Boxplots for the TI algorithm and the Basic algorithm for measurements done in an indoor environment during nighttime.	26
5.9	The error plot for the outdoor measurements.	28
5.10	Boxplots for the TI algorithm and the Basic algorithm for measurements done in an outdoor environment.	28
5.11	The positioning estimations using an additional processing method on the basic algorithm	30
5.12	The positioning estimations for a trained neural network	31

List of Figures

List of Tables

4.1 I/Q samples require additional consideration if the phase is calculated from Cartesian coordinates.	14
---	----

List of Tables

Bibliography

- [1] M. M. Amer and K. G. M. Atteya, “Indoor positioning bluetooth angle of arrival”, 2020.
- [2] S. Sadowski and P. Spachos, “Rssi-based indoor localization with the internet of things”, *IEEE Access*, vol. 6, pp. 30 149–30 161, 2018.
- [3] *Bluetooth direction finding. a technical overview*. [Online]. Available: <https://www.bluetooth.com/bluetooth-resources/bluetooth-direction-finding/> (visited on 04/09/2021).
- [4] *Enhancing bluetooth location services with direction finding*. [Online]. Available: https://www.bluetooth.com/blog/new-direction-finding-feature/?utm_campaign=location-services&utm_source=internal&utm_medium=blog&utm_content=how-bluetooth-rtls-solutions-can-improve-patient-care-and-minimize-the-spread-of-infection (visited on 04/09/2021).
- [5] *How bluetooth rtls solutions can improve patient care and minimize the spread of infection*. [Online]. Available: <https://www.bluetooth.com/blog/how-bluetooth-rtls-solutions-can-improve-patient-care-and-minimize-the-spread-of-infection/> (visited on 04/09/2021).
- [6] *New bluetooth direction finding feature will enhance location services solutions*. [Online]. Available: <https://www.bluetooth.com/blog/new-direction-finding-feature/> (visited on 04/09/2021).
- [7] *Quuppa, world's leading real-time location system (rtls) for indoor tracking*. [Online]. Available: https://www.quuppa.com/?gclid=EAIAIQobChMIP-DA0deP8wIVhaOyCh0ecQfKEAAYAyAAEgIVM_D_BwE.
- [8] Z. Hajiakhondi-Meybodi, M. Salimibeni, K. N. Plataniotis, and A. Mohammadi, “Bluetooth low energy-based angle of arrival estimation via switch antenna array for indoor localization”, in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, IEEE, 2020, pp. 1–6.
- [9] F. Zafari, A. Gkelias, and K. K. Leung, “A survey of indoor localization systems and technologies”, *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019. DOI: [10.1109/COMST.2019.2911558](https://doi.org/10.1109/COMST.2019.2911558).
- [10] L. Yao, *Bluetooth direction finding*, Bachelor Thesis, Sep. 2018.
- [11] F. Zafari, A. Gkelias, and K. K. Leung, “A survey of indoor localization systems and technologies”, *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019. DOI: [10.1109/COMST.2019.2911558](https://doi.org/10.1109/COMST.2019.2911558).
- [12] *BLE 5.1 Core Specification*. [Online]. Available: https://vtssociety.org/wp-content/uploads/2019/07/Core_v5.1.pdf (visited on 01/09/2021).

- [13] *Academy - bluetooth low energy angle of arrival (aoa)*. [Online]. Available: https://dev.ti.com/tirex/content/simplelink_academy_cc13x2_26x2sdk_5_10_00_00/modules/rtls_toolbox_ble5/ble_aoa/ble_aoa.html#install-the-software (visited on 01/08/2021).
- [14] *Iq sampling*. [Online]. Available: <https://pysdr.org/content/sampling.html>.
- [15] *RTLS Toolbox - Angle of Arrival*. [Online]. Available: https://software-dl.ti.com/simplelink/esd/simplelink_cc13x2_26x2_sdk/3.30.00.03/exports/docs/ble5stack/ble_user_guide/html/ble-stack-5.x-guide/localization-index-cc13x2_26x2.html#angle-of-arrival (visited on 01/08/2021).
- [16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [17] M. Cominelli, P. Patras, and F. Gringoli, “Dead on arrival: An empirical study of the bluetooth 5.1 positioning system”, in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, 2019, pp. 13–20.
- [18] P. Sambu and M. Won, “An experimental study on direction finding of bluetooth 5.1: Indoor vs outdoor”, *arXiv preprint arXiv:2103.04121*, 2021.
- [19] *Forum answer on why not both antennas of the launchpads can be used*. [Online]. Available: <https://e2e.ti.com/support/wireless-connectivity/bluetooth-group/bluetooth/f/bluetooth-forum/879770/ccs-cc2652r-aoa-use-both-array> (visited on 01/09/2021).
- [20] *SimpleLink™ multi-standard CC26x2R wireless MCU LaunchPad™ development kit*. [Online]. Available: <https://www.ti.com/tool/LAUNCHXL-CC26X2R1#description> (visited on 02/07/2021).
- [21] *SimpleLink™ CC13x2 and CC26x2 software development kit (SDK)*. [Online]. Available: <https://www.ti.com/tool/SIMPLELINK-CC13X2-26X2-SDK> (visited on 02/07/2021).
- [22] *SimpleLink™ Angle of Arrival BoosterPack*. [Online]. Available: <https://www.ti.com/tool/BOOSTXL-AOA> (visited on 02/07/2021).
- [23] *Meet the angle of arrival boosterpack*. [Online]. Available: https://dev.ti.com/tirex/explore/node?devtools=BOOSTXL-AOA&node=AHckEvhg0Y%203xs5rlangU2w_FUz-xrs__LATEST.
- [24] A. El Zooghby, C. Christodoulou, and M. Georgopoulos, “Antenna array signal processing with neural networks for direction of arrival estimation”, in *IEEE Antennas and Propagation Society International Symposium 1997. Digest*, IEEE, vol. 4, 1997, pp. 2274–2277.