UPPSALA
UNIVERSITET

UPTEC F 19027

Examensarbete 30 hp
Juni 2019

Estimation of Orientation in a Dual-Tag Ultra Wideband Indoor Positioning System

Oscar Johansson
Lucas Wassénius

UPPSALA
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Estimation of Orientation in a Dual-Tag Ultra Wideband Indoor Positioning System

Oscar Johansson, Lucas Wassénius

In this report the feasibility of using a dual-tag setup in an indoor positioning system was investigated. The reason for the dual-tag setup was to be able to estimate both position and orientation. The system was designed using UWB-technology, with an time of flight trilateration algorithm to calculate the position. The orientation was then estimated from the relative position between the two tags. The system was tested both with stationary tags, but also with the tags moving along two paths. These tests were conducted for different separation distance between the tags, namely 20 cm, 30 cm and 40 cm. The result was that the mean position error for stationary tags was less than 8 cm for all separations and the mean orientation error was less than 3° for all separations. For the moving tag tests a decrease of the error in orientation of about 30 % could be observed for a separation of 30 and 40 cm compared to 20 cm. However this difference is small in absolute values so more tests are needed to draw any conclusion about whether 30 and 40 cm tag separation performs better than 20 cm tag separation. The performance of the system could also be increased further by optimizing the anchor placement as well as the calibration of the antenna delays of the UWB-modules.

Handledare: Pedro de Oro
Ämnesgranskare: Uwe Zimmermann
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F 19027
Tryckt av: Uppsala

Sammanfattning

I det här projektet implementerades ett positioneringssystem för inomhusbruk. Detta system bestod av tre ankarnoder och två taggar, som var de noder i systemet vars position skulle bestämmas. Positionen av en tag bestämdes genom att mäta tiden det tog att skicka ett meddelande till varje ankare. Från dessa tider beräknades distansen till varje ankare som i sin tur användes i en trilatereringsalgoritm för att bestämma positionen. Målet med projektet var att undersöka möjligheten att utnyttja den relativa positionen mellan två av dessa taggar för att uppskatta riktningen på ett objekt, i detta projekt en robot. Tre tester gjordes på systemet. Ett där roboten placerades stillatående under mätningarna och två tester där roboten följde en förprogrammerad väg medan mätningarna utfördes. Dessa tester repeterades tre gånger med olika avstånd mellan taggarna för att undersöka om detta påverkade resultaten. Avstånden var 20 cm, 30 cm och 40 cm. Systemet hade en samplingsfrekvens av ungefär 4 Hz och all processering av data som behövdes för positionering utfördes i realtid.

Resultatet av det stationära testet blev mindre än 8 cm medelfel i position och mindre än 3° medelfel i riktning för alla tre avstånd mellan taggarna. I testen där roboten följde en förprogrammerad väg observerades en minskning av felet i riktning för 30 cm och 40 cm jämfört med 20 cm avstånd mellan taggarna. Denna minskning var ungefär 30 %. Även om det låter som en stor minskning är det fortfarande små vinklar. Detta betyder att det inte går att veta säkert om minskningen beror på förändringen i avstånd mellan taggarna eller om det är variationer i mätningarna. För att säkerställa resultatet skulle fler mätningar behöva utföras.

Det största problemet med systemet är att den uppmätta riktningen av roboten varierar mycket mer när den rör sig i y-led än i x-led. Detta beror på att standardavvikelsen i x-led är större än i y-led. Anledningen till detta är antagligen ankarplaceringen under försöken. I x-led var distansen mellan ankarna mindre än i y-led på grund av begränsningar i rummets storlek där försöken utfördes. Detta verifierades med simulationer där en försöksuppställning med lika lång distans mellan ankarna i både x-led och y-led användes. Denna simulation jämfördes med en simulation av den försöksuppställning som användes vid de utförda testerna. Då kunde det observeras att vid den tidigare simulationen minskade problemet nämnvärt jämfört med den senare. Detta tyder på att detta problem kan lösas med bättre ankarplacering.

Ett annat problem med systemet var svårigheten att kalibrera modulerna. Detta ledde till ett konstant fel i position på ungefär 5-8 cm som syntes i det stationära testet. Om en bättre metod för att kalibrera noderna utvecklades skulle detta fel kunna reduceras ytterligare.

Överlag så fungerar systemet bra, bortsett från dessa två problem. Felen bedöms vara tillräckligt små för att systemet ska kunna användas i till exempel ett navigationssystem för en enklare autonom robot som inte behöver navigera med hög noggrannhet. Om båda de nämnda problemen löstes verkar det som att det skulle resultera i ett system med relativt hög noggrannhet i både position och riktning jämfört med existerande positioneringssystem.

Contents

1 Introduction

Both indoor positioning systems and autonomous robots have recently gained a rising interest. Indoor positioning systems has grown together with the focus on internet of things and the epoch of information. The restriction of an indoor environment makes the global positioning system (GPS) lack in accuracy. Giving a reason for new technologies to emerge. Alongside these technologies, the frontier of machine learning has accelerated the development of autonomous robots and driving. Autonomous robots need accurate sensor information such as position and orientation to be able to navigate safely. To manage this several sensors are often used, resulting in large systems. For indoor positioning systems, safe navigation may instead be achievable by using accurate positioning sensors in a dual-tag setup. A dual-tag setup means that it is possible to position two reference points of the object. Using the relative position of these points an orientation of the object can be found. Using a single sensor type to find both position and orientation will reduce the system complexity compared to the use of sensor fusion. Developing this system does not only take accuracy into consideration, but also cost and scalability. Today, there exist both academical and commercial systems for indoor positioning for different applications. Such as analyzing the movement of people inside malls and train stations. Another application include locating expensive equipment within hospitals and factories. Only a handful of systems focus on the application of navigating autonomous indoor robots. This application has stricter requirements on accuracy and sampling rate, which is the focus of this project. The project will evaluate the possibilities to implement a system for said application with the focus on acquiring both position and orientation.

In this project Oscar Johansson had the main responsibility for the implementation of the tags and the robot used in the tests. He also had the responsibility for running the simulations of the system. While Lucas Wassénius had the main responsibility for implementing the anchors, doing the calibration of the system and the data analysis. The work for these parts has been done by the person responsible for that part. As for the rest of the project, close collaboration was carried out to make sure the different parts could be united into one system.

2 Background

2.1 Aim

The aim of the project is to implement an indoor positioning system for a robot that uses the relative position between two tags located on the same robot to determine the orientation of said robot. The positioning technology and algorithms will be chosen based on background research of existing academic and commercial indoor positioning systems. Lastly, the implemented system should be evaluated with fitting tests to be able to compare its performance to the already existing systems.

A successfully implemented system should be able to achieve a similar or better performance than the existing systems while having a cost that is also in parity with the existing systems. The performance will be based on the system's accuracy and precision in both position and orientation.

2.2 Research Question

A summary of the thesis aim can be expressed as the following research question:

Is it feasible to utilize the relative position between two tags on the same object to achieve a similar accuracy and precision as existing indoor positioning systems in terms of orientation and position?

2.3 Scope

These are the restrictions that are placed on the project. They are narrowing the scope of the project to make it reasonable within the given time frame. They are also assuming that we are investigating a system with the purpose of positioning an autonomous robot.

The system should be able to run and estimate position and orientation in real time.

The system will be tested in an indoor line of sight environment.

Tests with both stationary and moving tags will be performed to determine the accuracy and precision of the orientation and positioning.

The system does not have to support positioning of multiple tags simultaneously.

3 Theory

In this section the background and theory behind technologies used for indoor positioning will be explained.

3.1 Interpretation of Data

In positioning systems data needs to be gathered for the algorithms. This is most often done by wireless communication. Where there are multiple choices of data that can be measured from the messages. Some of the most commonly measured data will be explained in this chapter. These are received signal strength, angle of arrival, time difference of arrival and time of flight.

3.1.1 Received Signal Strength

Distance measurements can be gathered using RSSI (Received Signal Strength Indicator), which measures signal strength of a wide variety of signals[1]. The signal strength is decreasing with increasing distance. Therefore the signal strength can be converted to distance. Many receivers and transmitters have the function to measure signal strength directly. Using the manufacturer conversion table for the transmitter/receiver gain, the signal strength can be converted to distances. When converting the signal strength to distance, the RSSI method has several factors affecting the result. Losses in signal strength will effect the distance. Other factors affecting the signal strength is shadowing and multi-path. Where shadowing is the change of attenuation due to objects blocking the signal. Multi-path causes fluctuations in signal strength due to reflections on surfaces. A model[1] of the RSSI, considering the mentioned factors is shown in equation 1. Where d is the distance from transmitter to receiver and n is the path loss exponent which depends on the channel environment. Further, P_R is the received signal strength and P_t is the transmitted power. The transmitter and receiver gains are noted by G_t and G_r respectively. The parameters g and γ are the factors adjusting for fading based on Rayleigh and log-normal distribution[2].

$$P_R = \frac{G_t \cdot G_r}{4\pi} P_t \frac{g^2 \cdot \gamma}{d^n} \tag{1}$$

The difficulty of using RSSI to collect data for distances is the precision. For a 802.11 WLAN the limit is around 3 m median error[3]. It can be improved by using another interface for gathering data. But the main problems lies in estimating of filtering reflections and shadowing in a system. Therefore precision can be improved in an open room with low reflection and direct line of sight. Another factor affecting the precision of RSSI is the sample frequency. To get a better estimation of the position, several samples are used to create a normal distribution centered around the true position. Collecting data will take more time, but will improve the precision. The benefit of using RSSI to measure distances is that most smartphones already have integrated hardware and software for RSSI measurements.

3.1.2 Angle of Arrival

The angle of arrival (AoA)[12] can be used together with triangulation to estimate the position. A common way to find the angle of arrival is to use an antenna array, which consists of multiple antennas in a line and look at the difference in time when the signal arrives at different antennas. To do this an algorithm is used to estimate the angle from the signal data collected by the antenna array. Two classes of these estimation algorithms are beamforming and subspace-based methods.

Beamforming[13] utilizes weights to combine the signals from the different antennas. The weights make it possible to magnify the signals from certain directions while suppressing the signals from others. By changing the weights correctly, a sweep of all the angles can be executed and the signal strength at each angle can be measured. When the signal strength is plotted against the angle, peaks would appear at the angle of arrival. Some beamforming algorithms are Bartlett beamforming and minimum variance distortionless response. Bartlett beamforming compensates for the phase shift of the signals and uses this to magnify the signals from certain directions. This method works well when there's only one source signal, but for multiple signals the resolution becomes very low. Minimum variance distortionless response minimizes the signal from all directions except the desired direction. This method has a better resolution than Bartlett beamforming but it is still not optimal.

Subspace methods[13] are designed to extract the angle of arrival, or other parameters, from the signal. This means that they can not be used to receive signals from a certain direction, but only retrieve the parameters, for instance the angle of arrival. A common subspace method is multiple signal classification, MUSIC. With this method the goal is to find a vector q and a vector function $f(\theta)$ such that $q^H f(\theta) = 0$. Then the function $p(\theta) = \frac{1}{||q^H f(\theta)||^2}$ is plotted and the peaks will indicate the angle of arrival. Since $q^H f(\theta) = 0$ these peaks will be very sharp.

Aside from these two methods, the AoA can be determined using the properties of a camera[28]. For indoor positioning, this is commonly done in the visible light spectra. An important feature needed to find the AoA, is that the camera uses a biconvex lens. A biconvex lens does not refract light going through the center of the lens and is therefore able to calculate the angle of incidence without knowing the properties of the lens, see figure 1. Some additional properties of the camera are needed. One property is the distance between the lens and the imager. Other properties are the dimensions of the imager.

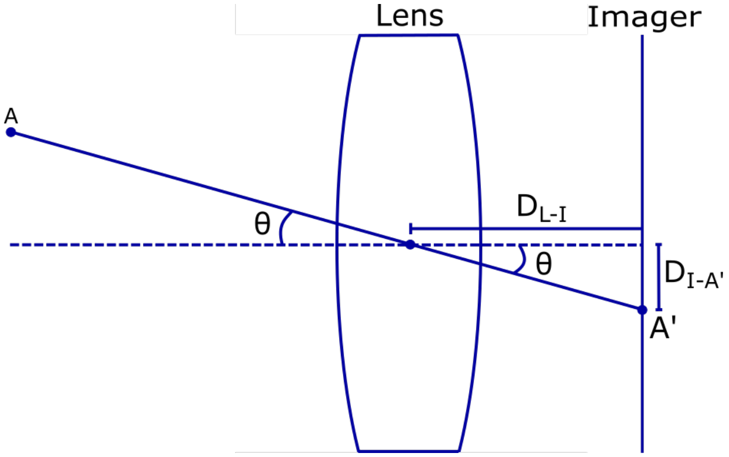


Figure 1: A figure displaying the parameters used to find the angle of arrival, θ, using a camera. D_{L-I} is the distance between the center of the lens and the center of the imager. $D_{I-A'}$ is the distance between the center of the imager to the point of the imager where the anchor-point is captured, A'.

Knowing the distances displayed in figure 1, the angle θ can be calculated using trigonometry, see equation 2. The accuracy of the angle will depend on the resolution of the camera since that determines how accurately $D_{I-A'}$ can be measured.

$$\theta = \tan^{-1}\left(\frac{D_{I-A'}}{D_{L-I}}\right) \tag{2}$$

3.1.3 Time Difference of Arrival

Time difference of arrival(TDOA)[14] measures the relative time of when the signals arrive at the different receivers of a system. This means that the receiver's clocks will need to be synchronized. The time difference Δt_{ij} between each pair of receivers with index i and j can be converted to a distance Δd_{ij} using the speed of the signal in the relevant medium. Assuming two receivers, B_1 and B_2 located as in figure 2, the solutions for the position P is given as a hyperbola that is also shown in the figure. To be able to determine the position two more beacon pairs are required. It is possible to use the same beacon in multiple pairs. This means that three receivers would be enough to determine the position. The speed of light is around 30 cm/nanosecond meaning that, if electromagnetic waves are captured, high frequency sensors are needed to find the time difference between the receivers. If other waves are used, such as sound waves, lower precision sensors can be used.

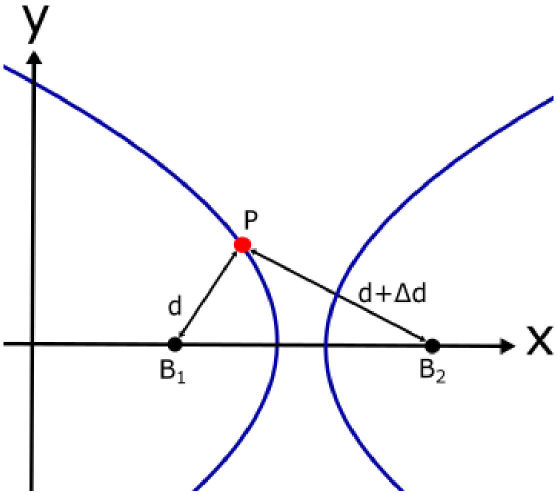


Figure 2: The figure shows the two receivers B_1 and B_2, with the estimated difference in distance Δd and the solutions for point P as a hyperbola.

3.1.4 Time of Flight

Time of flight (ToF)[25] is a method where the absolute time of the signal transmission is measured. Therefore, this method requires the receiver's clock and the sender's clock to be properly synchronized. When the time has been measured it is multiplied with the propagation speed of the signal in the medium, and the distance between the sender and receiver is obtained. Like TDOA, the hardware requirements of ToF depends on the propagation speed of the signal.

3.2 Position Calculation Algorithms

A position can be determined using collected data along with different algorithms. In this section some common algorithms, that are used in indoor positioning systems today, will be explained. The algorithms explained are trilateration, triangulation, fingerprinting, and dead reckoning.

3.2.1 Trilateration

To find a position in 3D space, four distances to known anchor-points can be measured and used to pinpoint the location. The algorithm used to do this is called trilateration. Trilateration is a popular algorithm for positioning systems. For instance, GPS systems today use trilateration to convert distance measurements of satellite positions to coordinates of the object.

The solution for trilateration is found with trigonometry in circles[4]. The concept and theory behind triangulation is easier to grasp in 2D-space. Assuming two anchor points, A_1 and A_2, and the distances, d_1 and d_2, between the anchors and the position of the object as in figure 3. A position, P, can be determined using 2D-trilateration. For an easier solution a 2D-plane is defined with the line between two anchor points on the x-axis. The anchor points are defined to be separated by Δ_x on the axis. One solution to find the position, P, is to use Pythagoras' theorem. Using the definitions in figure 3, the resulting equations for the coordinates of P are shown in equation 3 and 4

$$P_x = \frac{d_1^2 - d_2^2 + \Delta_x^2}{2\Delta_x} \tag{3}$$

$$P_y = \pm\sqrt{d_1^2 - P_x^2} \tag{4}$$

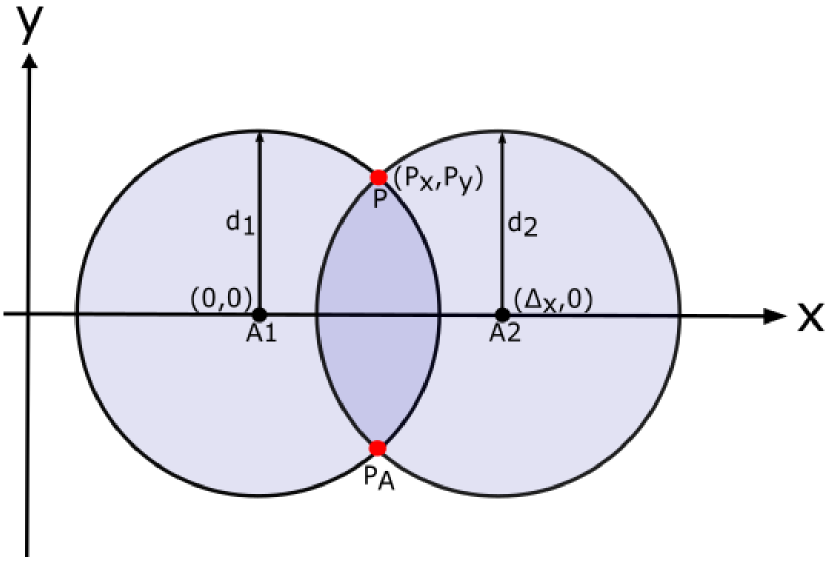


Figure 3: A figure illustrating the environment for the trilateration algorithm in 2D-space. Where the true solution is denoted by P and the ambiguous solution by P_A. The anchor-points are denoted by A_1 and A_2.

In equation 4 there exists two solutions for the P_y coordinate. Where one will be the true solution and the other an ambiguous solution. In figure 3 both the true solution and the ambiguous solution can be seen by the intersections of the circles. The ambiguous solution in 2D-space will always be a mirrored image around the axis between the two anchor-points. This solution can often be removed in navigation by some simple prediction algorithm. Predicting that the object would remain on the same side of the axis is a reasonable assumption unless the object is located close to the axis. Another way to remove the ambiguous solution is to add one additional anchor-point that is not located on the line between the two original anchor-points.

Extending this model to 3D-space with three anchor points in a plane will still give one ambiguous solution. The added definitions of the extended model are coordinates to locate the third anchor-point, $\Delta_{3,x}$ and $\Delta_{3,y}$, and the distance, d_3, from the new anchor-point to the position P, shown in figure 4. The solutions in 3D-space are found using Pythagoras theorem similarly to the solution in 2D-space.

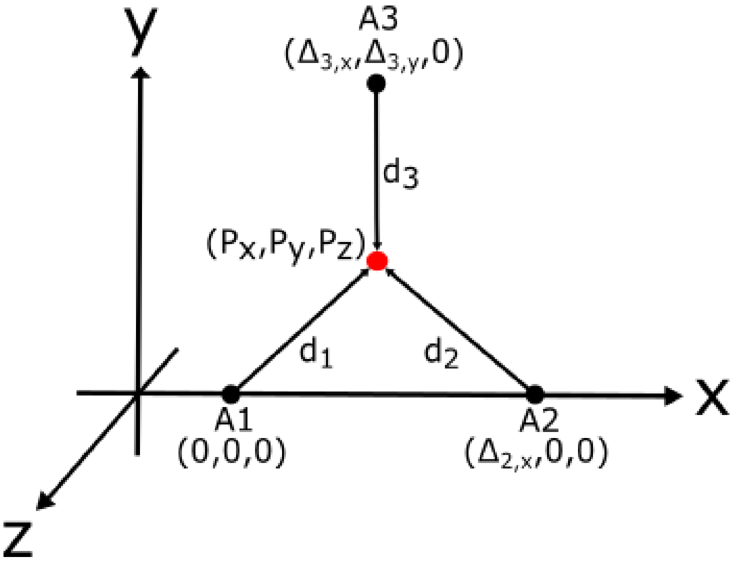


Figure 4: A figure illustrating the definitions used for the trilateration algorithm in 3D-space. Where the true solution is denoted by P and anchor-points for the trilateration are denoted by A_1, A_2 and A_3.

The equations to find the position, P, are shown in equation 5, 6 and 7. They are using the definitions from figure 4.

$$P_x = \frac{d_1^2 - d_2^2 + \Delta_{2,x}^2}{2\Delta_{2,x}} \tag{5}$$

$$P_y = \frac{d_1^2 - d_3^2 + \Delta_{3,x}^2 + \Delta_{3,y}^2 - 2\Delta_{3,x}P_x}{2\Delta_{3,y}} \tag{6}$$

$$P_z = \pm\sqrt{d_1^2 - P_x^2 - P_y^2} \tag{7}$$

In equation 7 there are two solutions for P_z. These solutions will be mirrored in a plane created by the three anchor-points. The true solution for the position can be found either by adding a fourth anchor point that is not located in the plane created by the other three anchor-points or by using some prediction algorithm like in the 2D-model. However, for an indoor positioning system the ambiguous solution can sometimes be discarded. If the plane created by the anchors is parallel to the roof, the component from ambiguous solution will be a solution mirrored vertically. Changes vertically for an indoor positioning systems can often be ignored unless the system need to consider height. In a special case where the three anchor points are located in the roof, the ambiguous solution will be located above the roof and can easily be identified and removed.

The precision of the trilateration algorithms depend on the measurements. A deviation in the measured distances will move the position within an error volume. The size of this volume depends on the precision of the measurement technique and the geometry of the anchor-points. A change in the location of the anchors can both increase and decrease the volume of inaccuracy. This concept is often called dilution of precision[5]. Generally moving the anchor-points closer to each other decreases the precision.

3.2.2 Triangulation

Instead of utilizing distances, like trilateration does, triangulation uses angles to position an object within an area. There are several triangulation algorithms. The most general algorithms are the geometric triangulation, geometric circle intersection, Newton-Raphson iterative method, and iterative search[6]. One of the advantages of using triangulation is that the orientation of the object can be determined in a client-based positioning system. While getting an orientation is an improvement, the geometry of triangulation is a complicated matter and can cause inconsistencies if not handled correctly. Ordering and geometry between the anchor-points need to be considered to find a position based on the angles in the system. To reliably find the position of an object with triangulation, three anchor-points are needed.

Iterative Search Method
The problem with using two anchor-points is that the orientation of the object cannot be decided. If the objects orientation is incorrect, the calculated position will also be incorrect. Three anchor points are used in the iterative search method. The method uses the three points to get three different positions based on every pair of two. If the orientation is correct then these three solutions will be equal in position. If the orientation is wrong the three points will be not be unanimous and will create a triangle. Iterating this calculation for different angles will give several solutions, resulting in a set of triangles. The perimeter of every triangle is compared. A larger perimeter is proportional to the error in position as the three calculations has a larger spread in positions. The triangle with the shortest perimeter is chosen as the best position.

One disadvantage with this solution is that two solutions exist for the orientation. Either the true solution or the mirrored solution shifted by 180°. This solution can be overseen by only iterating over $-90^\circ \leq \theta \leq 90^\circ$, assuming that the orientation is correct within the bound. The error of the method is inversely proportional to the number of iterations, though the computation time is directly proportional. Therefore a compensation between execution time and accuracy is done. Computational time can be reduced with adjustments to the algorithm. One example is to limit the angular span to reduce the number of iterations needed at the cost of decreasing the area in which the solution is predicted to be located.

Geometric Triangulation
The geometric triangulation algorithm uses all of the angles directly to compute a position and the orientation of the device relative to the anchor-points[7]. The algorithm is based on that the location of the anchor-points are known and the fixed angles between them can be determined. To understand the algorithm a plane of defined angles need to be examined. Figure 5 illustrates the problem and its definitions. Apart from the definitions in figure 5, nine parameters need to be known. Six of which is the Cartesian coordinates of each anchor point in the fixed plane, defined as $(A_{1,x}, A_{1,y}, A_{2,x}, A_{2,y}, A_{3,x}, A_{3,y})$ for anchor-points 1-3 respectively. The last three parameters are the angles between the anchor points and the objects, defined as $(\lambda_{A1}, \lambda_{A2}, \lambda_{A3})$ respectively.

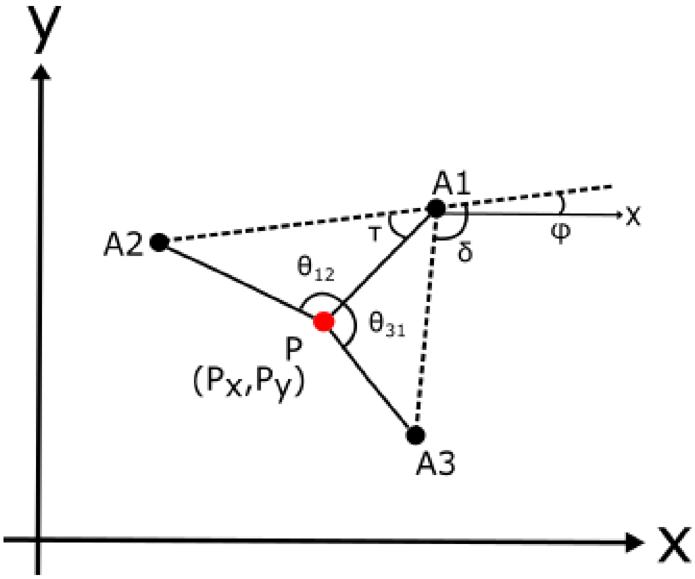


Figure 5: A figure to show the defined setup of the system to use the geometric triangulation algorithm in Cartesian coordinates.

The angles θ_{31} and θ_{12} in the figure is directly connected to the angles between the anchor-points and the object. These are computed by equation 8 and 9. Both of them need to be defined to be less than 180° for the algorithm to be stable. Therefore the labels need to be defined correctly. The labels in the system should increase counterclockwise. The angles in the system should then be checked so that θ_{31} and θ_{12} is less than 180°. If the condition is false, then labels can be rotated counterclockwise and the conditions checked again until they are fulfilled.

$$\theta_{31} = 360^\circ + \lambda_{A1} - \lambda_{A3} \tag{8}$$

$$\theta_{12} = \lambda_{A2} - \lambda_{A1} \tag{9}$$

~~For easier use in equations the difference between δ and θ_{31} is defined to be σ, see equation 10~~

$$\sigma = \delta - \theta_{31} \tag{10}$$

Using equation 8, 9, τ and the law of sines, equation 11 can be derived where a and b are the euclidean distances from anchor-point 2 to 1 and 3 to 1 respectively. Then τ is derived in equation 12. From these two equations the euclidean distance from the object to A_1 can be computed and defined as d, see equation 13.

$$p = \frac{a\sin(\theta_{12})}{b\cos(\theta_{31})} \tag{11}$$

$$\tau = \tan^{-1}\left(\frac{\sin(\theta_{12}) - p\sin(\sigma)}{p\sin(\sigma) - \sin(\theta_{12})}\right) \tag{12}$$

$$d = \frac{b\sin(\tau + \theta_{12})}{\sin(\theta_{12})} \tag{13}$$

With the definitions and derivations the position and orientation of the object can be found and derived as in equation 14, 15 and 16. The orientation of the object is labeled as P_θ.

$$P_x = A_{1,x} + d\cos(\phi + \tau) \tag{14}$$

$$P_y = A_{1,y} + d\sin(\phi + \tau) \tag{15}$$

$$P_\theta = \phi + \tau - \lambda_{A1} \tag{16}$$

This method can compute the position and orientation reliably when the object is within the triangle formed by the three anchor-points. When the object is on or outside of this triangle, the method can still give good results but do not work consistently. It is difficult to find a derivation of the area in which an algorithm works. The problems occur for different geometries of the triangles when the trigonometric functions approach zero.

Geometric Circle Intersection
Instead of using the angles directly to estimate the position, three unique circles can be found using three anchor-points. These circles can be found by multiple methods, some of them are described further in [6][8][9]. Each circle intersects a pair of anchor-points and the solution, in the way illustrated in figure 6. Then the solution can be found using the circles and trigonometry.

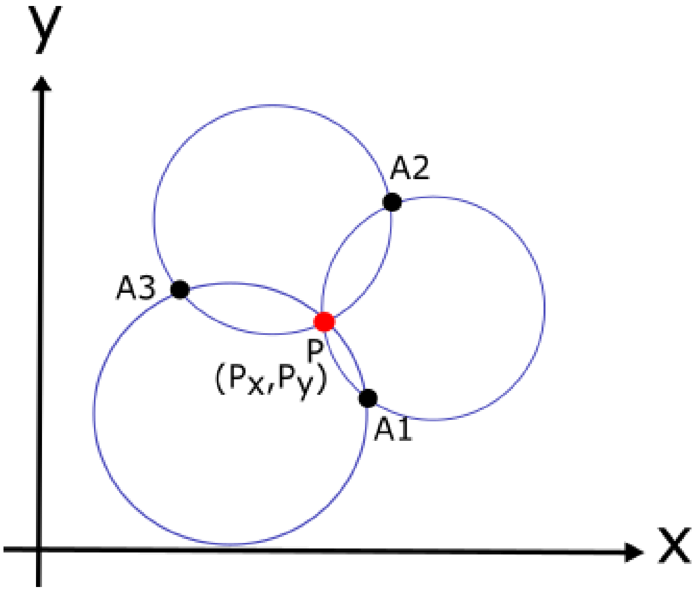


Figure 6: A figure displaying the method setup of geometric circle intersection.

Unfortunately there are not one unanimous solution. Some solutions of geometric circle intersection require proper ordering of the anchor-points, like in geometric triangulation. Others do not require any ordering[8]. One benefit of using the circle intersection method is that the method do not require true orientation of the object to find the position.

Newton-Raphson Iterative Search
If the angle between the object and three anchor points are known, a non-linear system of equations can be derived. This system of equations can be solved using Newton-Raphson Iterative search. Equation 17

shows the non-linear system to be solved, where P_x, P_y, P_θ is the coordinates and orientation of the object, λ_i is the angle between the object and anchor point i and lastly $A_{i,x}, A_{i,y}$ is the location of anchor point i in Cartesian coordinates. The non-linear system is the reason why it is difficult to determine some geometric solutions to the system. The Newton-Raphson method is a solution which has an iterative approach[6]. Instead of a geometric solution, the method extends the system with its Taylor series and then uses LU-decomposition and a prediction to solve the system. In equation 18, the first terms in the Taylor series are found and then it is assumed that the solution converges to create a linear system. This system is solved with LU-decomposition and the step is found for each coordinate. These are then added to the prediction to form a solution, see equation 19. If the difference between the prediction and the solution is within a tolerance, the solution is accepted. Otherwise the process it iterated with the solution as the new prediction until it is within the tolerance.

$$f_i(P_x, P_y, P_\theta) = \tan(\lambda_i + P_\theta) + \frac{A_{i,y} - P_y}{A_{i,x} - P_x} = 0; \qquad i = 1, 2, 3 \tag{17}$$

$$-f_i(P_x, P_y, P_\theta) = \frac{\partial f_i}{\partial P_x}\partial P_x + \frac{\partial f_i}{\partial P_y}\partial P_y + \frac{\partial f_i}{\partial P_\theta}\partial P_\theta \tag{18}$$

$$\begin{aligned} P_x^{new} &= P_x^{old} + \partial P_x \\ P_y^{new} &= P_y^{old} + \partial P_y \\ P_\theta^{new} &= P_\theta^{old} + \partial P_\theta \end{aligned} \tag{19}$$

A problem with the Newton-Raphson iterative method is that the solution can diverge and it is impossible for the system to know before the iteration starts. Depending on the prediction, the system will either converge or diverge. Though, when the solution diverges it can be interrupted and a new prediction can be made. Often another method is used together with the Newton-Raphson iterative method to make a good prediction of the position of the object.

All mentioned methods for triangulation has some benefits and some disadvantages. For example the geometric triangulation method and the circle intersection method both need to have proper ordering of anchor-points to get a result. The benefits with these methods are that there exist a space where a solution always will exist and that the orientation is not needed to compute the position. With the iterative methods computation time is a factor as the system of equations need to be solved in several iteration. The benefit is that they can often give a good result if the prediction is close to the true position. Other papers have compared these methods further[6][7]. Each of the methods has some disadvantages compared to the others which will either effect the timing or the accuracy[7]. Comparing the precision and accuracy, the Newton-Raphson method has the lowest standard deviation and mean error. In regard to computation time, the geometric triangulation method and circle intersection methods are the fastest[6].

3.2.3 Fingerprinting

Another method to estimate the position is to utilize fingerprinting together with RSSI measurements. The fingerprinting method consists of two phases, the offline phase and the online phase. In the offline phase, which is the training phase, data is collected from a set of reference points. In these points the RSSI from different access points are measured and stored as a radio map. This map is then used in the online phase to find the position for a device based on collected RSSI values which is compared to the radio map values.

To decide which position in the radio map is the best match for a set of measurements, different algorithms can be used. The next section will be explaining one of the most common algorithms, weighted k-nearest neighbors, in more detail.

Weighted k-nearest neighbors[10] is a deterministic method for estimating the position of a device using measured RSSI values from the access points and the stored radio map. Assuming the online measured RSSI values are collected over a period of time, R_p is the matrix in equation 20 with these values at point p.

$$R_p = \begin{bmatrix} \alpha_1^{(p)} & \alpha_2^{(p)} & \dots & \alpha_n^{(p)} \end{bmatrix} = \begin{bmatrix} rssi_{11}^{(p)} & rssi_{12}^{(p)} & rssi_{13}^{(p)} & \dots & rssi_{1n}^{(p)} \\ rssi_{21}^{(p)} & rssi_{22}^{(p)} & rssi_{23}^{(p)} & \dots & rssi_{2n}^{(p)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ rssi_{m1}^{(p)} & rssi_{m2}^{(p)} & rssi_{m3}^{(p)} & \dots & rssi_{mn}^{(p)} \end{bmatrix} \tag{20}$$

Where $rssi_{ij}^{(p)}$ is the measured RSSI from access point j at scanning time i. The mean and standard deviation are then calculated and written as in equation 21 and 22 where $\bar{\alpha}_j^{(p)}$ is the mean value and $\bar{s}_j^{(p)}$ is the standard deviation for the j:th column.

$$\bar{R}_p = \begin{bmatrix} \bar{\alpha}_1^{(p)} & \bar{\alpha}_2^{(p)} & \dots & \bar{\alpha}_n^{(p)} \end{bmatrix} \tag{21}$$

$$\bar{S}_p = \begin{bmatrix} \bar{s}_1^{(p)} & \bar{s}_2^{(p)} & \dots & \bar{s}_n^{(p)} \end{bmatrix} \tag{22}$$

Just like the online measured values, the radio map can be stored in a similar matrix. This matrix is shown in equation 23 with the measured RSSI values at each reference point q. The mean vector at reference point q is then written as $\bar{R}_q$ in equation 24.

$$R_q = \begin{bmatrix} \alpha_1^{'(q)} & \alpha_2^{'(q)} & \dots & \alpha_n^{'(q)} \end{bmatrix} = \begin{bmatrix} rssi_{11}^{(q)} & rssi_{12}^{(q)} & rssi_{13}^{(q)} & \dots & rssi_{1n}^{(q)} \\ rssi_{21}^{(q)} & rssi_{22}^{(q)} & rssi_{23}^{(q)} & \dots & rssi_{2n}^{(q)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ rssi_{c1}^{(q)} & rssi_{c2}^{(q)} & rssi_{c3}^{(q)} & \dots & rssi_{cn}^{(q)} \end{bmatrix} \tag{23}$$

$$\bar{R}_q = \begin{bmatrix} \bar{\alpha}_1^{'(q)} & \bar{\alpha}_2^{'(q)} & \dots & \bar{\alpha}_n^{'(q)} \end{bmatrix} \tag{24}$$

The first step to calculate the position of the device from these matrices, is to standardize the mean and the standard deviation of the measurements, $\bar{R}_p$ and $\bar{S}_p$. This is done by performing the calculations of equation 25 for each of the elements in $\bar{R}_p$ and $\bar{S}_p$. Here X_j is the j-th element of the $\bar{R}_p$ or $\bar{S}_p$ vector.

$$x_j = \frac{X_j - min(X)}{max(X) - min(X)} \tag{25}$$

If the standardized mean and standard deviation vectors are written as $\bar{R}'_p = \begin{bmatrix} \bar{r}_1^{'(p)} & \bar{r}_2^{'(p)} & \dots & \bar{r}_n^{'(p)} \end{bmatrix}$ and $\bar{S}'_p = \begin{bmatrix} \bar{s}_1^{'(p)} & \bar{s}_2^{'(p)} & \dots & \bar{s}_n^{'(p)} \end{bmatrix}$. Then the weight for each access point can be calculated as in equation 26.

$$w_j^{(p)} = 1 - \frac{r_j^{'(p)} + s_j^{'(p)}}{2} \tag{26}$$

The next step is to calculate the weighted euclidean distance to each of the reference points from the test point. This is done as in equation 27 where q is the reference point and p is the test point.

$$d_q = \frac{1}{n}\sqrt{\sum_{j=1}^{n} w_j^{(p)}(\bar{\alpha}_j^{(p)} - \bar{\alpha}_j^{(q)})^2} \tag{27}$$

The last step is to calculate the position from these distances. This is done with the calculation shown in equation 28. Y_q is the position of the q:th access point among the K nearest access points, d_q is the weight for this access point and ϵ is a small number to avoid division by zero.

$$Y = \frac{\sum_{q=1}^{K} \frac{Y_q}{d_q+\epsilon}}{\sum_{q=1}^{K} \frac{1}{d_q+\epsilon}} \tag{28}$$

3.2.4 Dead Reckoning

Dead reckoning is a positioning method that can be used when the initial position, direction and distance moved is known. By moving this distance in the correct direction from the initial position, the new position can be found. This new position is then used as the initial position in the next step.

A common way to find the direction and distance moved is to gathered data from an accelerometer and a magnetometer. The data from the accelerometer can be used to detect steps to estimate the distance walked and the magnetometer can measure the direction in which the person is walking. Since both an accelerometer and a magnetometer are usually built into smartphones, this means that the users own device can be used for this positioning method without the need of additional hardware.

For the step detection it is good to filter the signal first to reduce noise and remove the constant acceleration from gravity[11]. First the signal is filtered with a high-pass filter to remove low frequency signals, such as gravity. Then a low-pass filter is used on the filtered signal to remove high frequency noise. The result is a gravity free signal with minimum noise.

The step detection algorithm is then used on this signal to count the user's steps. Two common step detection methods are peak detection and zero-crossing detection. Zero crossing counts signals crossing zero to determine if a step has occurred. Usually a timing threshold is used to not count false steps. This threshold varies depending on the person, which means that it's hard to use this method without calibrating it for the user. The peak detection method on the other hand, does not need to be calibrated. This method counts a step when a local maxima and a local minima is detected in sequence within a certain time interval. A maxima is detected when the signal exceeds a maxima threshold which is determined by adding $\Delta threshold$ to the last minima. In the same way a minima is detected when the signal drops below a minima threshold determined by subtracting $\Delta threshold$ from the last maxima.

The next step in the algorithm is to estimate the step length. This can be done in several different ways. Four of them are Weinberg's approach, Scarlet's approach, Kim's approach and a static approach where it's assumed that all steps are the same length. The equations 29, 30, 31 and 32 show the equations for these different methods. Equation 29 shows the static method where it is assumed that the step length is proportional to the height of the person. Equation 30 shows Weinberg's approach which uses the difference between the maximum and minimum acceleration to estimate the step length. Equation 31 shows Scarlet's approach which tries to eliminate the error caused by variation in pace by also taking the average acceleration into consideration. Equation 32 is Kim's approach which tries to estimate the step length based only on the average acceleration.

$$step\_length = height \cdot k \tag{29}$$

$$step\_length = k\sqrt[4]{a_{max} - a_{min}} \tag{30}$$

$$step\_length = k\frac{\frac{\sum_{k=1}^{N}|a_k|}{N} - a_{min}}{a_{max} - a_{min}} \tag{31}$$

$$step\_length = k\sqrt[3]{\frac{\sum_{k=1}^{N}|a_k|}{N}} \tag{32}$$

According to a study[11] the dynamic methods in equation 30, 31 and 32 produce a smaller error than the static method in equation 29 when tested over the distances 10, 20 and 30 meters. Furthermore Weinberg's and Scarlet's methods had better results than Kim's method for 20 and 30 meters.

3.3 Wireless Interface Technologies

There are several technologies available for collecting the data required for the positioning algorithms. In this section some common technologies for collecting data will be explained, and their strengths and weaknesses will be discussed.

3.3.1 Wi-Fi Positioning

Wi-Fi is a wireless local area network (WLAN) utilizing radio frequencies to transmit data. Today, most devices use a Wi-Fi connection which has a frequency of 2.4 GHz or 5.8 GHz. Using Wi-Fi for indoor positioning is quite common[15][16][17]. The most common way of using Wi-Fi for positioning is by collecting RSSI values from several access points together with fingerprinting[15][16].

Wi-Fi has the benefit of easy integration with smartphones and tablets since most of them already have the required hardware built in. Another benefit is that Wi-Fi is available in most larger buildings and office spaces. The main disadvantage with using Wi-Fi routers or access points for positioning is the precision. For reasonable precision more routers/access points are needed than is typically available in an office. To increase precision, the RSSI measurements are often combined with some other method such as dead reckoning[17]. The reason for having low precision when using Wi-Fi is due to the inconsistencies of RSSI, such as shadowing and multipaths.

3.3.2 Bluetooth Low Energy

Another interface for positioning algorithms is Bluetooth low energy (BLE) which is using 2.4 GHz, radio frequency, signals to communicate. The difference between standard Bluetooth and BLE is that BLE is designed for longer battery lifetime while maintaining the range. With the trade off being the reduced data rate. In positioning systems, BLE is used to collect RSSI values for range estimation which is independent on data-rate. Much like Wi-Fi, BLE positioning systems have reduced accuracy due to RSSI effects, such as shadowing and fading[18][19]. The main benefit of BLE is that it is compatible with most smartphones and tablets. Another benefit is that BLE tags/anchor-points are power and cost efficient. Because of this, more anchor-points can be integrated and used to improve the accuracy of the system.

3.3.3 Ultra Wideband Radio

Radio signals above 3.1 GHz and below 10 GHz are said to be within the ultra wideband(UWB) radio spectrum. The UWB spectrum are often used for positioning indoors[20]. It works by transmitting low spectral density signals in all frequencies of the band to create a pulse. The pulse is then received and the ToF or TDOA is computed. The width of the pulse is dependent on the width of the spectrum. A broader spectrum can yield a shorter pulse width. There exist indoor positioning systems that use UWB for distance measurements[21]. The benefit of using UWB pulses is the ability to separate the direct pulse from the multipath pulses. UWB systems are also able to distinguish between a pulse reflected by a distance close to the sensor and a distance far from it, eliminating the effects of shadowing. Since the effect of multipathing and shadowing can be removed, the precision of UWB systems are high. A problem with using UWB technology is that it is hard to implement. To use UWB efficiently all receivers and transmitters need to be able to determine time-differences in the picosecond range. A restriction with UWB is the regulated transmit power and restriction to only indoor use, which depends on country[21].

3.3.4 Radio Frequency Identification

Radio frequency identification tags can be used together with a technique, similar to fingerprinting, to position an object indoors[21][22][23][24]. Radio frequency identification (RFID) is often used to identify an object or person using either a passive or active tag[22]. An active tag is a powered device that transmits the identification code when triggered by a signal or with timed delays. The passive tag is not connected to a power source but uses another transmitted radio signal to power the chip. The chip in the tag reflects the identification code so that the tag can be identified. In positioning systems both active and passive tags can be used. The problem with using passive tags is that all tags in range will reflect the identification code. If active tags are used it could be programmed to only identify the closest tags. The major benefit with using passive tags is the cost. An active tag can cost as much as 50 times more than a passive tag[22].

In positioning systems the tags are distributed throughout the space as reference tags[22][23][24]. The object is equipped with another tag. To find the position of the object, the RSSI of the tag on the object is compared to the signals of all the other mapped tags. This results in a grid based solution where the position is found by knowing the surrounding reference tags. Compared to normal fingerprinting, the difference is that no calibration is needed and there is a live update to reduce the effect of slow fading. One major consideration of RFID positioning is that accuracy is increased proportional to the amount of reference tags and readers. Therefore, a precise solution can be very expensive.

3.3.5 Magnetic Field Positioning

There are two methods that can be used to utilize magnetic fields for indoor positioning[26]. Either an artificial field is generated using coils. If the position of these coils are known, the strength of the magnetic field could be converted to distance and the position estimated using trilateration. The other way is to use the earth's magnetic field to find the position. Indoors, the earth's magnetic field is altered by the steel structure of the building. These variations in the fields can be used as a fingerprinting map for indoor positioning. When using the earth's magnetic field, no hardware need to be placed. But at the same time the natural field might be disturbed by electronic equipment or moving ferromagnetic material, like elevators[27].

A disadvantage of using the magnetic field for fingerprinting is that there are only three components to map, the magnetic field strength in x-, y- and z-direction. To be able to use all three components for mapping, the magnetic north must be known. In many implementations this is not possible. For example, the way different users holds the device can differ and therefore the relation between the earth's coordinate system and the device's system is unknown. An accelerometer can be used to solve this problem, but even then it's only possible to use two of the components for mapping[26].

However, there are examples of positioning systems using three components for mapping. One of them is a system based on a particle filter using magnetic field fingerprinting together with an encoder system[27]. This system uses one map with the field strength in the horizontal plane, one with the field strength in z-direction and one with the direction of the field in the horizontal plane.

3.3.6 Optical Positioning

Using visible light for positioning is often done with a machine learning and big data approach. It can also be achieved with light beacons and smartphone devices[28]. One of the methods behind this is to find the AoA and use triangulation to determine the position. Camera properties can be used to find the angle of incidence on a captured image, see section 3.1.2. To be able to use this technology, identification of the anchor-points need to be identified. One way of doing this is by encoding data in the emitted light from the anchor-points[28][29]. This encoding is often done with simple On-Off keying. On-off keying works by symbolizing the off and on state by zeroes and ones respectively. A familiar use of on-off keying is the Morse code.

An advantage of using the visible spectra compared to radio waves is that it cannot be interfered by other common radio waves. Also, other light sources do not interfere with the positioning if the correct light source can be distinguished. Another advantage is the compatibility with smartphones which makes it easy to start using the technology. Although it is easy to get started for the users, the system has hard requirements on the buildings infrastructure. There need to exist light sources with LED keying to be able to use this technology.

4 Evaluation

In this section commercial systems as well as academically developed systems will be discussed and evaluated in terms of how good they are in different aspects of indoor positioning.

4.1 Metrics

This section will explain the three metrics that was investigated in the evaluation of the commercial and academic indoor positioning systems.

4.1.1 Accuracy

Accuracy is what decides how good the system is at estimating the position. The accuracy can be presented in multiple ways. The two most common is to either take the mean of the distance errors from the estimated position to the real position or to find the interval, starting from zero, where the confidence level is 90 %. How the accuracy is expressed for each of the systems are presented in table 1 together with the accuracy for each system.

4.1.2 Cost

The cost of the system is a metric that both depends on additional infrastructure needed for the system as well as the cost of the individual tags and the software. If the system can reuse any of the infrastructure already available in the building or doesn't need any additional, the expense of the system will decrease.

4.1.3 Scalability

Scalability is a metric that evaluates how easy or difficult it is to implement the system in a large scale. Some of the factors that affect the scalability are cost, work required to install new infrastructure as well as calibration and if it is possible to keep the same accuracy for a larger scale system.

4.2 Commercial Systems

Here, some available commercial indoor positioning systems will be evaluated. The accuracy, cost and scalability of the systems will be investigated and discussed.

4.2.1 Pozyx

Pozyx[30] provides an UWB based solution for indoor positioning and motion tracking. The simplest system consists of four anchors and one tag. The anchors are spread out in the room around the user. With accurate time measurements of the signal between the tag and the anchors the position of the tag is calculated with trilateration.

The Pozyx anchors can both be used by themselves or as an Arduino shield. Libraries with functions for the anchors and tags are provided by the company, both for Arduino and in python. All the tags consists of a microcontroller, an UWB chip, antenna and motion sensors such as gyroscope, magnetometer, accelerometer and altimeter. The microcontroller provides an interface to an external device through either I^{2}C or an interrupt line. The anchors have the same components as the tags except for the motion sensors. Multiple algorithms are included with the system, which has different accuracy and computation speed to fit different applications. The algorithms are running on a real time operating system to ensure stability and speed.

Studies have been made on the positioning accuracy of the Pozyx system[31]. It shows that it is possible to get below 0.5 m of accuracy with 90 % confidence. This study however, uses a single anchor and takes advantage of multipathing to get multiple range measurements. Pozyx claims that less than 0.1 m accuracy is possible, but no studies are found that support this except for Pozyx's own results on the website[32]. In their study they look at the mean error in positioning and the result is that 0.092 m accuracy is achieved for the best performing algorithm.

A starting package is available from their store for 600 euro. This includes one tag and four anchors. This system would be able to cover an area of about 21x21 m in a normal indoor space. From this it is possible to expand the area further by adding more anchors. Since there need to be one anchor in each corner of a 21x21 m square, only two more anchors are needed to increase the covered area by 21x21 m. Also, the anchors do not require any manual measuring to be calibrated since Pozyx features auto-calibration. This means that it is relatively easy to expand the covered area of the system if needed.

4.2.2 Infsoft

Infsoft[33] is a German company providing several solutions for indoor positioning and navigation. Their solutions are based on Wi-Fi, Bluetooth or RFID depending on the customer preferences. The solutions can be for navigation, tracking, analytics or positioning. The stated accuracies and ranges for the Infsoft systems are gathered from their website[34].

They have Wi-Fi based solutions that can communicate with any Wi-Fi connected device for client-based RSSI measurements. Using the measurements they can estimate the position based on fingerprinting technology. For server-based measurements the device is replaced with an Infsoft locator node. The Wi-Fi solutions they provide have an accuracy of 5-15 m and an approximate range of 150 m depending on the Wi-Fi access point. The accuracy is based on that Wi-Fi access points normally are placed for coverage of data transmission and not as access points. It is mentioned that increasing the number of access points will increase accuracy.

An additional solution is their BLE navigation system. The solution uses beacons for positioning and is advertised as a solution with higher accuracy than Wi-Fi with the ability to position smart devices. This solution is a client-based solution using RSSI distance measurements to locate the device. The accuracy of this solution is 1-3 m where the beacons have a transmission range of approximately 30 m.

For a more precise solution Infsoft offers a UWB based positioning system. The solution uses Infsoft locator nodes and tags with ToF measurements to find distances. Using at least three locator nodes, the tag can be positioned using trilateration. This system is advertised to have an accuracy of 10-30 cm. Each locator node is said to have a range of 10-150 m depending on the installation.

Which of the solutions is selected depend on what the customer desires and what type of application the system is intended for. An Infsoft white paper[35] summarizes the benefits and disadvantages with the different solution. The Wi-Fi solution has the benefits of using existing infrastructure with good coverage, but lacks in precision. BLE has better precision than Wi-Fi but needs to have integrated hardware which have a low cost but with a quite low range. The benefits of the UWB system is better precision but has disadvantages of high cost and energy consumption.

4.2.3 IndoorAtlas

IndoorAtlas[36] is a positioning system founded by Janne Haverinen to commercialize the use of magnetic positioning. IndoorAtlas has a system that uses several sensors to achieve better accuracy. They have positioning algorithms that use the earths magnetic effect, BLE, and Wi-Fi. They also use an accelerometer for dead reckoning positioning. The difference to many other platforms is that IndoorAtlas combines a mixture of several positioning technologies into one solution. The combination make use of benefits from all technologies. The main sensors to find the positions is the magnetic field fingerprinting technique which has the best accuracy of the three. The accuracy of the whole system is stated to be between 1-3 m[37]. The magnetic sensors is backed up with BLE sensors and Wi-Fi for positioning using RSSI. Using extra sensors give redundancy and robustness to the system. Magnetic positioning can be quite slow to achieve good accuracy. This is complemented by using Wi-Fi to have some fast but inaccurate positioning while the other positioning algorithm is converging to an accurate result.

One customer of IndoorAtlas is Mumbai Airport that uses their system to help travellers navigate through one of their terminals. Another customer is Yahoo! Japan who is using the system to help commuters in large metro stations. These examples show that it is possible to use IndoorAtlas in large indoor spaces. The IndoorAtlas positioning application is completely free and a map of a building can be created. To use the system, fingerprinting of the area needs to be done which only some smartphones have the possibility and precision to do well enough. While it is possible to use IndoorAtlas completely free it is difficult to obtain fast positioning and good accuracy. For better accuracy BLE beacons and Wi-Fi access points need to be integrated to the infrastructure.

4.2.4 Eliko

Eliko[38] has developed hardware and software for indoor positioning systems. Their solution is based on UWB technology using time of arrival to measure the distances and trilateration to estimate the position of a tag. The solution consists of three main devices. The first type of device is called KIO anchor tags which has a range of 50 m. Several anchors can be added to improve the tracking area. The tracking area when using four, correctly positioned, anchors is around 1000 m^2. The second device is a battery powered KIO tag which should be attached to the object being tracked. They also provide a server that handles the anchors and tags for real time tracking. Eliko provides an API for integration of the tags into the clients normal working space. The system is commercialized to be used as a real time tracking system for objects such as tools.

The accuracy, according to their website[38], of their system is between 5-30 cm and can be used in non-line-of-sight environments. No studies have been found to support this claim but when looking at studies of systems with the same technology this seems reasonable[31][32]. The cost of a system with four anchor tags, one tracking tag and a server with API is 2100 euro. The system can be expanded with extra target tags and anchors tags. Target tags cost 250 euro each and anchor tags cost 320 euro each.

4.2.5 OpenRTLS

Another positioning system using UWB positioning is OpenRTLS[40] (Open Real Time Locating System). Apart from UWB, OpenRTLS uses microelectromechanical systems to provide acceleration, rotation, magnetic field and pressure measurements. The positioning is done via ToF measurements for indoor use and TDOA for outdoor use. Just as other UWB systems OpenRTLS uses tags that are spread out around the object to be tracked. The accuracy of OpenRTLS, according to OpenRTLS's own studies[41], is between 1-30 cm. Fine accuracy of 1 cm can only be achieved with precise calibration and line-of-sight measurements when the object is close to an anchor. For normal use with line-of-sight measurements the accuracy is within 10 cm. The range of an anchor point is approximately 30 m for indoor use. Each tag can transmit one hundred signals per second but this setting can be reduced to only transmit once every day for energy efficiency. Using the lowest power consumption settings for the network, the UWB tags have a battery life of around 9 years. A package containing eight anchors and four tags with power supplies costs 7000 euro which makes a full installation of OpenRTLS cost 1-5 euro per square meter.

4.3 Academic Positioning Systems

In this section academically developed systems will be evaluated. The evaluation will investigate which technologies are used and then look at some tested values for the accuracy of the systems. The system will also be evaluated for cost and scalability if possible.

4.3.1 BlueSentinel

BlueSentinel[42] is an indoor positioning system with the aim to keep track of occupants in each room of a building. The system makes use of BLE signal strength measurements from devices called iBeacons to decide the room in which the user is located. The system has two apps, one that is used in the offline phase to collect data to set up the system and one that is used by the occupants during the online phase. In the offline phase the first app is used to measure the RSSI of all iBeacons in range of each room and connect them to a room label. It is enough to measure the data for a few seconds. The second app measures the RSSI values of each iBeacon and compares this data to the data gathered by the first app to detect in which room the user is located.

Two algorithms were tested to classify in which room the user was located, K-nearest neighbors and a tree-based algorithm. Both of the algorithms performed equally well in terms of accuracy with 83 % and 84 % correct classifications respectively. However, the k-nearest neighbors algorithm was considered better since it had less false negatives than the tree based algorithm. Less false negatives are favourable since the system was meant to be used in a smart home and it was assumed that the user would have a worse experience from false negatives than false positives.

The system is relatively easy to expand to include more rooms since only one tag is required for each room and the cost of each tag is low. Also the calibration process is fast because only a few seconds of measurements are required for each room. Compared to regular fingerprinting where the whole room needs to be mapped out, this makes the calibration of this system a lot simpler for large buildings.

4.3.2 Luxapose

Luxapose[28] is an indoor positioning system that uses the camera of a smartphone and visible light to find the position. It uses AoA measurements collected through the photos of a Nokia Lumia 1020 smartphone to find anchor-points. The anchor-points are LED-light sources that are scattered throughout the room. The LED-lights use on-off keying to transmit their location to the smartphone. From one picture an on-off sequence can be read due to the rolling shutter effect. Luxapose can triangulate the position from the location of the anchors, the AoA and the camera properties. This is done by comparing the distances and angles on the projected image to the distances between the anchors in the real coordinate system.

A study[28] of the Luxapose system shows that the accuracy is 10-50 cm. The accuracy is varies depending on how many anchor-points are in sight. If five well placed anchors are used the mean error will be below 10 cm. When three badly placed anchors are used, the mean error will be around 50 cm. However, the system can reliably position the smartphone only if the anchors can create a triangle. Orientation accuracy is tested to be around 3° for several positions with five anchors.

Some disadvantages with the Luxapose system are that infrastructure is needed, at least three line-of-sight anchors are needed for positioning, and the user need an active camera facing the ceiling. However, they are looking at a solution using a larger light fixture working as four anchors that would make it easier to integrate into the infrastructure. The benefit of the system is that it has good accuracy while still being a low-cost system. A LED anchor-point can cost as little as 3 euro for larger quantities. But the price for each user will be larger as they need a smartphone with a high resolution camera.

4.3.3 Cricket

Cricket[44] is an indoor positioning system developed at Massachusetts Institute of Technology. It consists of Cricket nodes, which is a microcontroller with hardware to transmit and receive RF and ultrasonic signals. These nodes can either be configured as beacons or listeners. The beacons are fixed in space with known coordinates while the listeners are attached to the objects that are being tracked. RF and ultrasonic signals are used to find the distance from the listener to the beacons. The beacons periodically transmits an RF message with information about the node, for example an identification number and coordinates. At the beginning of this RF message it also transmits a short ultrasonic pulse. Since the ultrasonic signal will propagate slower than the RF signal there will be a time difference when they arrive at the listener. This time difference is measured and the distance to the beacon can be calculated. Cricket also features orientation estimation by using multiple receivers on the same node. The orientation is estimated by calculating the difference in distance between the receivers.

The system includes algorithms that uses beacon measurements to automatically calibrate the beacon positions. There are also algorithms to detect and avoid interference between beacon transmissions, as well as outlier rejection algorithms. These algorithms improves the accuracy of the system, which would otherwise be subject to large positioning errors due to interference between beacons. The positioning accuracy of the system is 10-12 cm and the orientation accuracy is 3°.

The performance of the Cricket system does not decrease much with an increased covered area as long as the beacon density is kept low. This is because the ultrasonic range is quite low and won't overlap much unless there are a lot of beacons in the same area. Since the range of RF signals is longer than the ultrasonic signals they will cause interactions that slow down larger systems, but not much can be done about this.

4.3.4 LearnLoc

LearnLoc[45] is a mobile positioning application. The android application uses RSSI Wi-Fi measurements from the smartphone to position the device. This is done with fingerprinting. However, the radio map is created with a machine learning algorithm instead of calibrating it manually. The smartphone collects some fingerprints and these are sent to a server. The server trains the neural network with a supervised training algorithm. Supervised training means that the algorithm trains from collected sets of inputs and outputs that correlate with each other. When the training phase is done the testing phase start, which is when the algorithm is used to predict the position. During this phase the computation is transferred to the smartphone instead. The LearnLoc system uses K-nearest neighbours or regression models to estimate the position of the smartphone.

A study[45] of the system's performance is made by the creators of the system. The accuracy is compared to the frequency of Wi-Fi scans. Best accuracy was achieved using K-nearest neighbours with a scan frequency of one scan per second. The average error for this test was 1.138 m. The disadvantage with using frequent Wi-Fi scans and many calculations is the energy efficiency of the system. When decreasing the scan frequency to once every eighth second, the power consumption was reduced to half of the previous test. The cost in accuracy when using less frequent scans are significant. With a scan frequency of once every eighth second, the average error is around 4 m. The testing of the system was done on the campus of Colorado State University with the available Wi-Fi access points.

4.3.5 LANDMARC

LANDMARC[24] is an indoor positioning system developed to investigate RFID positioning in terms of accuracy and cost. Active RFID reference tags are spread out in the the desired detectable area with one tag covering approximately one square meter. The reference tag positions are calibrated with the system.

Some RF readers are needed to communicate with the network of active tags. Another active tag is used as the tracking tag to be positioned. The position is found by distinguishing the four closest reference tags to the tracking tag. Then the RSS is measured to find the position within the area using trilateration.

The accuracy of the LANDMARC system was tested by the founders in different test cases. Tests of accuracy for different tag locations were done. Some of which were close to physical obstructions to test shadowing of the reference tags. Also, the accuracy depending on reference tag density was tested. A high density test was done with one meter in between reference tags and a low density test with two meters between reference tags. The result from this study gave a 90th percentile accuracy error of 1.5 - 3.0 m for different target positions with the lower reference tag density. With the higher density the 90th percentile was approximately 1.5 m for all target positions. The conclusion is that accuracy is increased for non-line-of-sight with higher density of reference tags. However, increasing the number of active reference tags will substantially increase cost.

A newer study[46] has improved the LANDMARC positioning system with an improved neighbouring algorithm and newer hardware. They conduct testing with 50 cm spacing between the active reference tags. With the new conditions the accuracy was improved to an average error of 15 cm. The testing area that was used for the study was approximately 17 m^2 and used 117 active tags. The newer system has increased the accuracy of the LANDMARC system but the cost still remain high.

4.4 Summary

The commercial and academic indoor positioning systems have been evaluated using three comparison metrics. These metrics are explained in this section and the results are compiled in table 1.

Table 1: This table summarizes the results from the evaluation of the commercial and academic indoor positioning systems.

System	Accuracy	Technology	Positioning Algorithm	Cost	Scalability
Pozyx	10 - 20 cm 90 % Confidence	UWB ToF	Trilateration	High	Easy but High Cost
IndoorAtlas	1 - 3 m	Magnetic Field	Fingerprinting	Low	Substantial Calibration needed
Eliko	5 - 30 cm	UWB ToF	Trilateration	High	Easy but High Cost
OpenRTLS	1 - 30 cm	UWB ToF	Trilateration	High	Easy but High Cost
BlueSentinel	Room 83 % Confidence	BLE RSSI	Fingerprinting	Low	Easy and Low Cost
Luxapose	10 - 50 cm	Optical Light AoA	Triangulation	Medium	High Infrastructure Requirements
Cricket	10 - 12 cm Mean Error	RF + Ultrasound TDoA	Trilateration	Low	Low cost but slower
LearnLoc	1 - 3 m	Wi-Fi RSSI	Fingerprinting	Low	Substantial Calibration needed
LANDMARC	1.5-3 m 90 % Confidence	RFID RSSI	Fingerprinting + Trilateration	High	High Infrastructure Requirements and High Cost

4.5 Conclusion

In table 1 there are a relation between the technologies used and the characteristics of the system. Generally all commercial and academic systems that use the same technology achieves approximately the same results. This means that the traits of different technologies can be extracted from the study. These traits are presented in table 2.

Table 2: This table concludes the traits of different technologies used in indoor positioning systems.

Technology	**Error**	**Cost**	**Scalability**
UWB	<30cm	High	Easy but High Cost
WiFi	<3m	Low	Easy, might need calibration
BLE	<3m	Low	Easy, might need calibration
Magnetic Field	<3m	Low	Calibration Needed

From table 2 it is possible to conclude that UWB-technology gives the best accuracy of the most common indoor positioning systems but it is also the most costly. The WiFi, Bluetooth and magnetic field alternatives have a lower cost but are also much less accurate. The scalability of BLE and WiFi depends on what positioning algorithm is used. If a trilateration algorithm is used, scaling is as easy as adding more anchors and save their positions in a database. A fingerprinting based system it is harder to scale since calibration is needed for all of the area that is added which will be a lot of work.

5 Method

In this section the method of the project will be explained. This includes the design choices made for the system as well as the reasoning behind them. There are also descriptions of all the different parts of the indoor positioning system. Lastly the descriptions of the tests to evaluate the system are also included in this section.

5.1 Choice of Measurement Technology, Hardware and Software

5.1.1 Measurement Technology

The selection of measurement technology was done based on a study of previous work and commercial systems done in section 3. From the study it was concluded that UWB based positioning using ToF to collect the distance measurements was the most fitting method. The reasoning behind this was that the accuracy of UWB positioning is substantially better than other technologies such as RSS measurements of Wi-Fi or BLE. Since the positioning system are supposed to be used with an autonomous robot it is important to have a high accuracy. Otherwise the robot won't know it's position well enough to be able to avoid obstacles. The main disadvantage of this system compared to the others is the cost of the UWB-modules. The cost of one module is approximately 35 euro and need a processor to handle the communication. With this cost the technology might be too expensive to implement in large buildings such as shopping malls or storage buildings. But for a smaller office the cost is reasonable. With an accuracy below 30 cm the systems possibilities are widened. Meaning that the system can not only be used for human navigation, but also for autonomous navigation.

5.1.2 Hardware

The system needs hardware to send short radio pulses and store the timestamps in both the tag and the anchor node. The DW1000 module from Decawave[47] is designed for this application area. Transmission and reception timestamps are stored in the modules registers with a resolution of 15.65 ps, which corresponds to a distance of approximately 5 mm. To get this low resolution a very high clock frequency is required to update the value of the timestamp. This value needs to be updated 64 billion times per second to achieve a resolution of 15.65 ps. This means that a 64 GHz clock is required. This can be compared to a high-end computer processor which clocks at about 4 GHz. Settings for the DW1000 module can be chosen to achieve the required trade-off between accuracy, range, power consumption and transmission speed.

In this project the pulse repetition frequency was set to 64 MHz for maximum resolution of the timestamp. Higher pulse repetition frequency result in less time between pulses which in turn results in higher resolution. The preamble length of the module was set to 128 symbols for low overhead during transmission, resulting in high transmission rates. For high speed data transmissions the data rate was set to 6.8 Mbps. The module also has channel settings that decide the operational frequency of the device. Channel 5 was chosen which operates with a bandwidth of 499.2 MHz and a center frequency of 6489.6 MHz. Note that these settings need to consider the radio regulations of the country of operation. Further information about the operational settings of the DW1000 module can be found in the DW1000 user manual[48].

Microcontrollers were used to control the DW1000-modules. The microcontrollers selected for the project was Raspberry Pi 3 B+[52] for the tags and Arduino Pro Mini[51] for the anchor nodes. A Raspberry Pi is used for the two tags to achieve faster processing. This choice was made since the controller for the tags will handle the positioning algorithm and the filtering. These extra computation together with the fact that this processor will handle two DW1000-modules makes processor speed important. The processor of the Raspberry Pi clocks with a frequency of 1.4 GHz, while the Arduino clocks at 8 MHz.

5.1.3 Software

An open source library for the DW1000-module was used to setup and communicate between the processors and UWB-modules. The library is available in both Python code and Arduino(C++) code and can be found at Github[49][50].

5.2 System Design

5.2.1 System Overview

The chosen system design includes a minimum of three anchor nodes and one tag with two UWB-modules. The anchors were spread out in the room and could be at any position as long as they are in the same horizontal plane and not in a line. The anchor nodes handled the computation of the time of flight and sent the result back to the tag. The tag converted the ToF to distance and computed the position with trilateration while filtering the result. These filtered positions were then used to find the orientation of the object. This overview of the system is illustrated in figure 7.

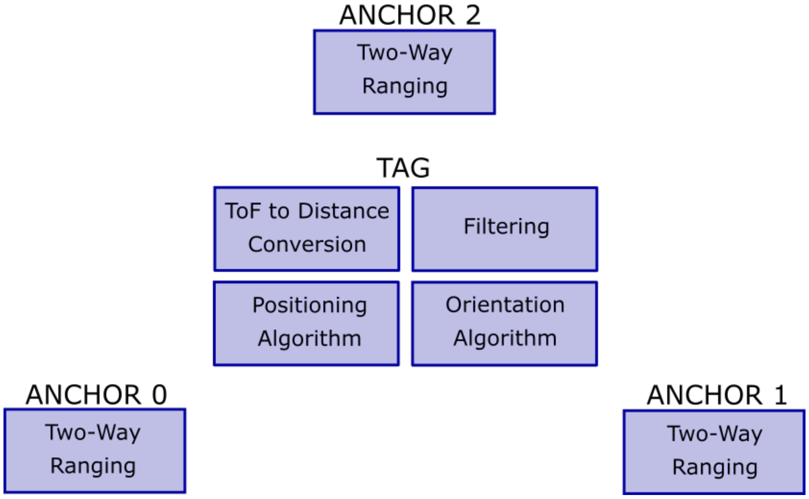


Figure 7: The figure shows the system overview with the functions of each node.

5.2.2 Time of Flight Measurements

It was decided that an asymmetric double-sided two-way ranging scheme was going to be used to calculate the ToF between an anchor and a tag. It has the advantage of minimizing the error in ToF while not having the requirement of constant reply times. To use this scheme, three messages need to be sent between the devices. In figure 8, these messages are POLL, POLL_ACK and FINAL. The tag stores the timestamps TS_1, TS_4 and TS_5 in the payload of the message sent between the devices while the anchor stores the timestamps TS_2, TS_3 and TS_6 locally. When the FINAL message is received, the time of flight can be calculated by the anchor using the timestamps from both devices. This is done with the expressions in equation 33. Finally, the ToF is sent to the tag in the message called TOF_REPORT.

$$
\begin{aligned}
T_{round1} &= TS_4 - TS_1 \\
T_{reply1} &= TS_3 - TS_2 \\
T_{round2} &= TS_6 - TS_3 \\
T_{reply2} &= TS_5 - TS_4 \\
T_{tof} &= \frac{T_{round1}T_{round2} - T_{reply1}T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}}
\end{aligned}
\tag{33}
$$

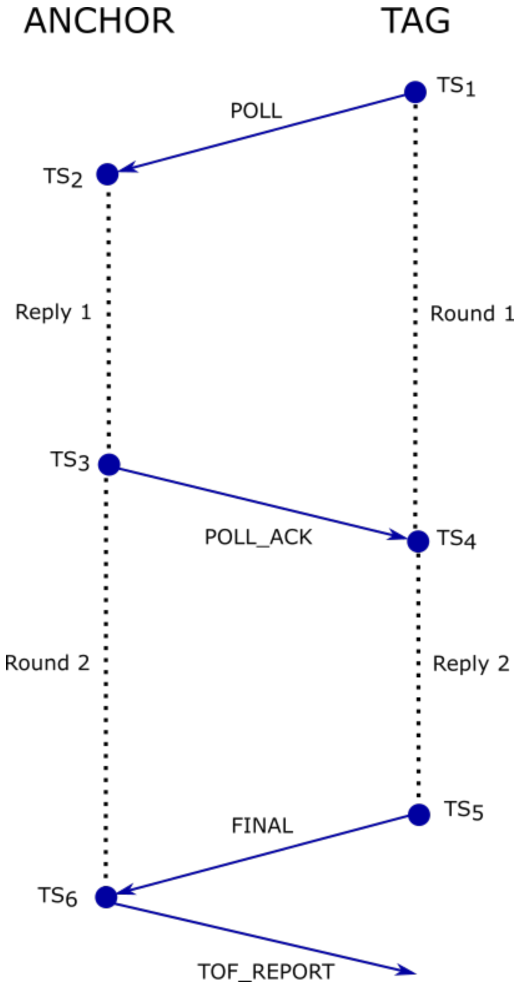


Figure 8: The figure illustrates the communication flow between an anchor and a tag when calculating the time of flight.

5.2.3 Calibration

The UWB-modules record the timestamp when the signal is transmitted and received in the module. The problem with this is that the timestamps can't accurately measure the time of flight. This is because when a signal is transmitted a timestamp is registered, then the signal spend some time in the antenna before it enters the air and actually starts travelling towards the receiver. The same is true when a signal is received. Then the signal instead spends time in the antenna before the timestamp can be recorded. This leads to transmit timestamps that are too early and receive timestamps that are too late. This antenna delay depends on the individual hardware. Hence a calibration constant needs to be introduced that is subtracted or added to the timestamps to get an accurate result. This means that each module used in the system has to be calibrated individually. Since the modules in this project are either a tag or an anchor they always send the same sequence of messages, which has the same amount of received and transmitted packets. This is good since there is no need to look at the antenna delay for a transmitted message and a received message separately. Instead it is possible to consider a general antenna delay for the whole sequence which makes the calibration process less elaborate.

The process to calibrate the UWB-modules began with choosing one of the tags and give it an arbitrary antenna delay. The three anchors were then calibrated using this tag. This was done by placing an anchor 5,01 m away from the tag. This is the distance recommended by Decawave in the user manual for the DW1000 module [48]. 500 measurements of the range were performed and the mean were noted. The difference between this mean and the true distance was used to calculate a reasonable adjustment of the antenna delay for the anchor. This was repeated until the mean range was the same as the true range. After all of the anchors had been calibrated using this method, the second tag was calibrated the same way using one of the anchors as the other device. The results from the calibration is presented in appendix C. These results include the final antenna delays for each module and the standard deviation and error of the range measurements between each tag and the anchors with the final delays.

5.2.4 Position and Orientation Computations

The position of each tag was calculated from the measured distances using trilateration as explained in section 3.2.1. These calculations were done individually for each UWB-module while simultaneously being filtered by a particle filter explained in section 5.2.5. The filtered position values were then used to calculate the orientation of the object using equation 34, where (x_n, y_n) are the coordinates of the UWB-modules. Index 0 denotes the UWB-module placed in the front of the object and index 1 denotes the module placed in the back.

$$arctan2(y_0 - y_1, x_0 - x_1) = \begin{cases} arctan(\frac{y_0 - y_1}{x_0 - x_1}), & \text{if } x > 0 \\ arctan(\frac{y_0 - y_1}{x_0 - x_1}) + \pi, & \text{if } x < 0 \; and \; y \geq 0 \\ arctan(\frac{y_0 - y_1}{x_0 - x_1}) - \pi, & \text{if } x < 0 \; and \; y < 0 \\ \frac{\pi}{2}, & \text{if } x = 0 \; and \; y > 0 \\ -\frac{\pi}{2}, & \text{if } x = 0 \; and \; y < 0 \\ undefined, & \text{if } x = 0 \; and \; y = 0 \end{cases} \tag{34}$$

5.2.5 Particle Filter

A particle filter was implemented to filter the range measurements and give a smoother change in position with less error. The particle filter was chosen mainly because of its capability to capture non-linearities in a system. The exact particle filter that was chosen was the sequential importance resampling version. This particle filter's algorithm consists of five parts. These are initialization, prediction, update weights, estimation and resampling. All of these steps will be explained in this section. Each of the UWB-module has its own particle filter since they have different positions and is therefore filtered separately. The algorithm of the particle filter is illustrated graphically in figure 9.

Initialization
In the initialization step all of the particles are initialized. Each particle consists of a state vector which contain x- and y-coordinates, speed and heading. Since the initial position and velocity of each module is unknown, these state variables are initiated uniformly distributed on the expected area the module is located.

Each particle is also assigned a weight which corresponds to how likely that particle is to represent the true position. These weights are initialized to $\frac{1}{N}$, where N is the number of particles in the filter, since all of the randomly distributed particles are equally likely to be the true position when the filter is initialized. In the filter used for the system in this project, N was chosen to be 500. This number of particles gave the filter a good balance between computation time and estimation accuracy.

Prediction
The next step is to make a prediction of the next position of the module. This is done by giving each particle a small normally distributed perturbation in heading and speed and then use Newton's laws of motion together with the time since the last prediction to calculate the new position. The particle filter then ends up with N guesses of where the true position of the module might be. Each of these guesses with its own weight.

Update weights
In this step a measurement is acquired and the weight of each particle is updated depending on how far away from the measurement they are located. This is done by defining a Gaussian curve with zero mean and standard deviation σ, $f(x) = N(0, \sigma)$. Then the weights of each particle is adjusted according to equation 35, where $\widetilde{w}_n$ is the new weight for particle n, w_n is the old weight for particle n and d_n is the Euclidean distance from the measurement to particle n. This increases the relative weight of particles close to the measurement and decreases the relative weights of particles further away. The last step is to normalize the weights to get a total weight of 1.

$$\widetilde{w}_n = w_n f(d_n) \tag{35}$$

Estimation
Here the filtered position is calculated from the particles and their weights. This is done by taking the weighted mean of the particles' positions as in equation 36, where μ is the estimated position average, w_n is the weight of the n:th particle, x_n is the position of the n:th particle and N is the total number of particles.

$$\mu = \sum_{n=0}^{N} w_n x_n \tag{36}$$

Resampling
To prevent particle degeneracy a resampling step is performed. Particle degeneracy happens because the process of updating the weights makes a few particles acquire a very high weight compared to most other particles which will get a weight close to zero. This is a problem because if the filter would continue with these weights it would essentially be like using a particle filter with only a few particles. This problem is solved with an algorithm called resampling. Resampling can be done in multiple ways. In this project it's done by multinomial resampling. This algorithm samples the particles with probabilities corresponding to their weights. When a particle is sampled a copy of that particle is created. This process is repeated until N particles have been sampled. This way the number of particles are kept constant. When N particles have been sampled the weights of the particles are normalized again before the process restarts from the prediction step.

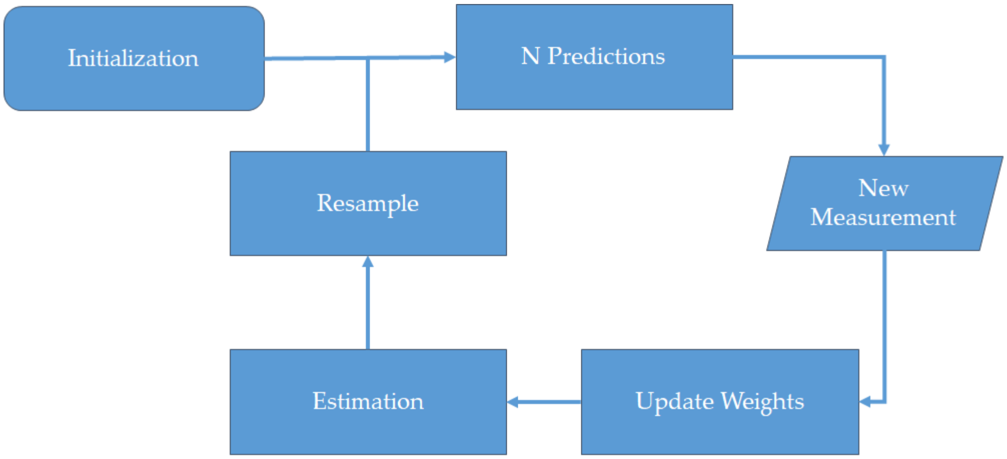


Figure 9: Graphical representation of the particle filter algorithm

5.3 Tests

In this section the test setups and their purpose will be described. This includes descriptions of the test procedures, tag and anchor movements placement as well as movement patterns of the tags. All of the tests used the same version of the software. Meaning that neither the filter nor the software for the anchors and tag were optimized for a specific test.

All of the tests were conducted in an line-of-sight indoor environment. A Cartesian coordinate system with the origin in one of the anchors was defined for the tests. The orientation in the coordinate system was defined as 0 degrees along the positive x-axis. The tag was mounted on a robot while the tests were conducted. The robot's position in the coordinate system was defined as the position in the center point between the two UWB-modules. These were placed at the back and front of the robot with equal distance to the center of rotation. Also the orientation of the robot was defined as the orientation of the vector from the back module to the front module. The placement of the anchors, in Cartesian coordinates, are described by:

$$
\begin{aligned}
(X_0, Y_0) &:= (0, 0) \\
(X_1, Y_1) &:= (4, 0) \\
(X_2, Y_2) &:= (2, 10)
\end{aligned}
$$

Each test was repeated for three distances between the UWB-modules. This was done so that the dependency between UWB-module separation and accuracy in orientation could be investigated. These distances were 20 cm, 30 cm and 40 cm.

5.3.1 Stationary Tag

In the first test the system was tested with the robot placed stationary at (2, 6) and the orientation 0 degrees. 400 measurements of each module's position was made with approximately 250 ms in between each measurement. These measurements were used to calculate the orientation and position of the robot in real time. While all of the statistical analysis of the data was performed after the actual test was done.

5.4 Moving Tag

Apart from the test with a stationary tag, tests were also conducted where the tag was moving along a predetermined path. To move the tag along the path a robot was used. The path was preprogrammed into an Arduino Mega 2560 which controlled the robot. The robot was equipped with treads instead of wheels to be able to rotate on the spot. This made it possible to perform 90° turns. Two different paths were investigated which are meant to represent common movement patterns of an autonomous robot. These paths are described in further detail below. In all of the tests with a moving tag the orientation and position was sampled with a frequency of about 4 Hz. Also, all of the calculations and filtering required were done in real time.

The first path consisted of the robot starting at (0.5, 6.5) and then going in a straight line for 3 m along the positive x-axis. In figure 10 an illustration of this path is presented.

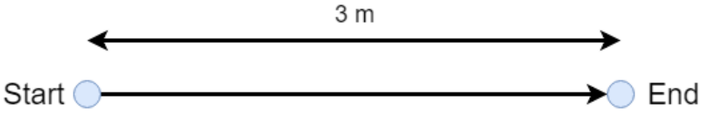


Figure 10: This figure illustrates the second path used in the tests.

The second path consisted of the robot starting at (0.5, 3.5) and going straight for 3 meters along the positive y-axis, then doing a clockwise turn of 90°. It then continued straight for 3 meters before doing another clockwise turn of 90° and then finish by going straight for another 3 meters. In figure 11 an illustration of this path is presented.

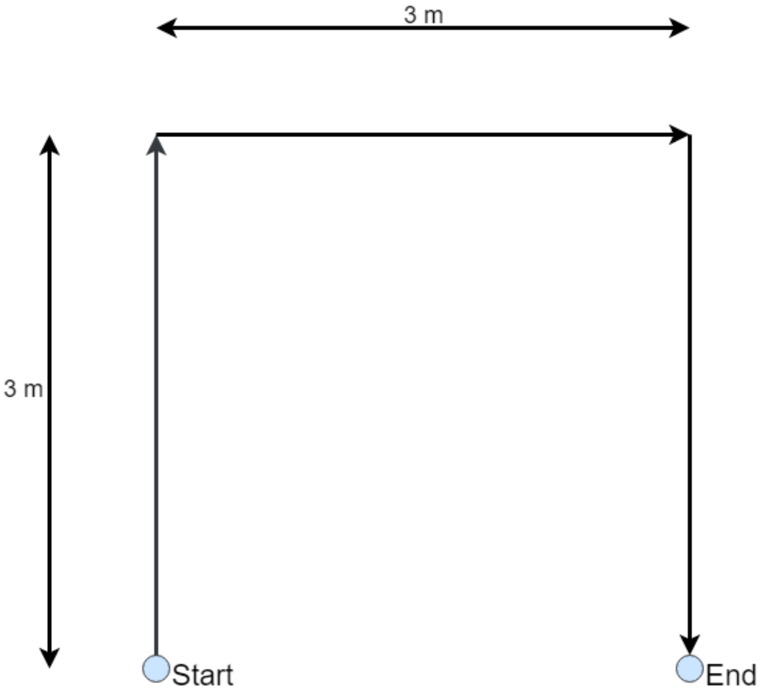


Figure 11: This figure illustrates the third path used in the tests.

5.5 Data Analysis

When the tests were completed the data gathered from the different tests was imported into Matlab to be analyzed. A few different metrics were calculated to be able to analyze the performance of the system. These calculations were not done in real time since they are not necessary in a positioning system for an autonomous robot. The calculations to obtain these are explained below.

When writing the equations for the metrics, two functions are used to save space. These are Euclidean distance in two dimensions, $d(\cdot)$, and expectation value, $E(\cdot)$. The Euclidean distance is the L^2 norm which takes two coordinates as input and returns the distance from the origin to this point. It is defined as in equation 37. The expectation value is the mean position of a vector of coordinates passed into it. This is defined as in equation 38.

$$d(x, y) = ||(x, y)|| = \sqrt{x^2 + y^2} \tag{37}$$

$$E(\overrightarrow{x}, \overrightarrow{y}) = \left(\frac{1}{N} \sum_{i=1}^{N} x_i, \frac{1}{N} \sum_{j=1}^{N} y_j \right) \tag{38}$$

For the stationary tag test, four metrics were calculated. These were the mean position error, standard deviation of the position error, mean orientation error and the standard deviation of the orientation error.

The mean position error was calculated by taking the mean of the measured positions and then find the Euclidean distance from this point to the true position. This is described in equation 39 where $d(\cdot)$ is the function for Euclidean distance and $E(\cdot)$ is the expectation value. $(x_{Meas,i}, y_{Meas,i})$ is the measurement corresponding to the true position $(x_{True,i}, y_{True,i})$.

$$e_{pos} = d(E(x_{Meas}, y_{Meas}) - (x_{True}, y_{True})) \tag{39}$$

The standard deviation of the position was calculated by taking the standard deviation for the distances from each measurement to the mean of the measurements. This is described in equation 40 where N is the number of measurements, $d(\cdot)$ is the function for Euclidean distance and $E(\cdot)$ is the expectation value. The i:th measurement of the position is denoted by $(x_{Meas,i}, y_{Meas,i})$.

$$\sigma_{pos} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (d(E(x_{Meas}, y_{Meas}) - (x_{Meas,i}, y_{Meas,i})))^2} \tag{40}$$

The mean orientation error is the mean value of the angle minus the true orientation as in equation 41, and the calculation of the standard deviation of this angle is shown in equation 42. Here N is the total number of measurements and $\theta_{Meas,i}$ is the measurement corresponding to the true angle $\theta_{True,i}$.

$$e_\theta = \mu_\theta - \theta_{True} = \frac{1}{N} \sum_{i=1}^{N} \theta_{Meas,i} - \theta_{True} \tag{41}$$

$$\sigma_\theta = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (\mu_\theta - \theta_{Meas,i})^2} \tag{42}$$

For the tests with moving tags two metrics were calculated. These were the mean distance error and the mean absolute orientation error.

The mean distance error was calculated by summing the distance from each measurement to the true position for each measurement and dividing by the number of measurements. This is shown in equation 43 where N is the total number of measurements and $d(\cdot)$ is the Euclidean distance. $(x_{Meas,i}, y_{Meas,i})$ is the measurement corresponding to the true position $(x_{True,i}, y_{True,i})$.

$$\frac{1}{N}\sum_{i=1}^{N} d((x_{Meas,i}, y_{Meas,i}) - (x_{True,i}, y_{True,i})) \tag{43}$$

The mean absolute orientation error was calculated by taking the absolute value of the difference between each true orientation and the measured orientation and then calculate the mean of them. This is shown in equation 44 where N is the total number of measurements and $\theta_{Meas,i}$ is the measurement corresponding to the true angle $\theta_{True,i}$.

$$\frac{1}{N}\sum_{i=1}^{N} |\theta_{Meas,i} - \theta_{True,i}| \tag{44}$$

5.6 Simulations

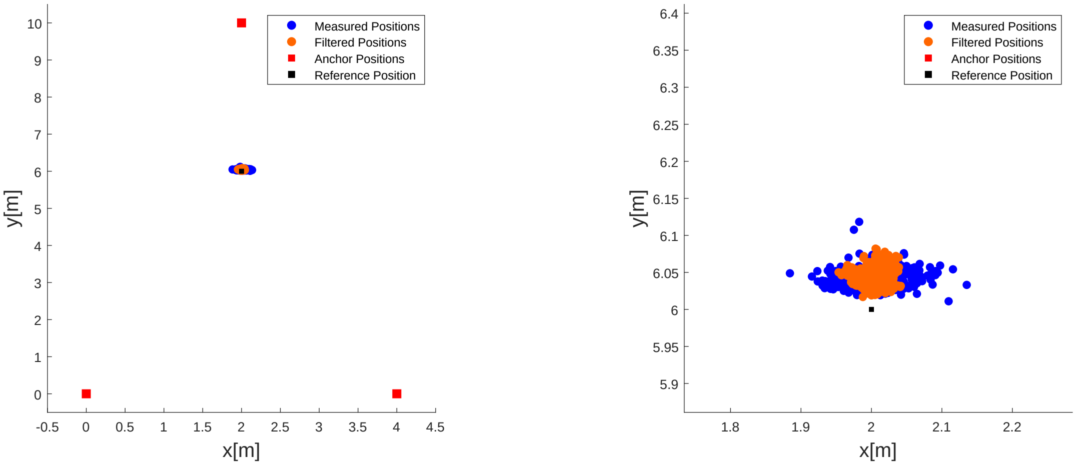
To complement the test runs, simulations of the robot going along the test paths were performed. The measurements were simulated by adding a random normally distributed perturbation to the range from each of the anchors to the true position of the modules. The standard deviation of this perturbation was chosen as 0.05 m. This was determined from the results of the calibration, which can be seen in appendix C, where 0.05 m was the highest standard deviation acquired for any of the anchor-tag combinations. These simulated measurements were then analyzed in the same way that the real measurements were.

6 Results

In this section the results from the data analysis are presented as well as the plots for some of the test cases. The plots that are not presented here can be found in appendix D.

6.1 Stationary Test

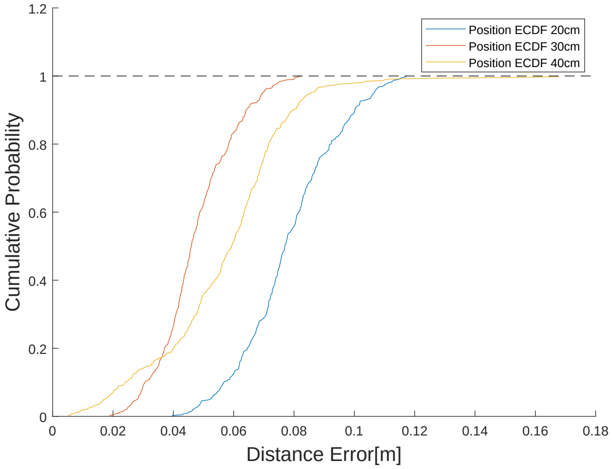
In figure 12 the position measurements and the filtered positions are plotted with a tag separation of 30 cm. The red squares are the anchor positions, the black square is the true position, the blue dots are the measured values and the orange dots are the filtered values. The graphs for 20 cm and 40 cm tag separation has been moved to appendix D since they all look similar. In figure 13 the empirical cumulative distribution functions of the orientation and position error are plotted for each of the tag separation distances. Finally the values acquired from the data analysis are presented in table 3.



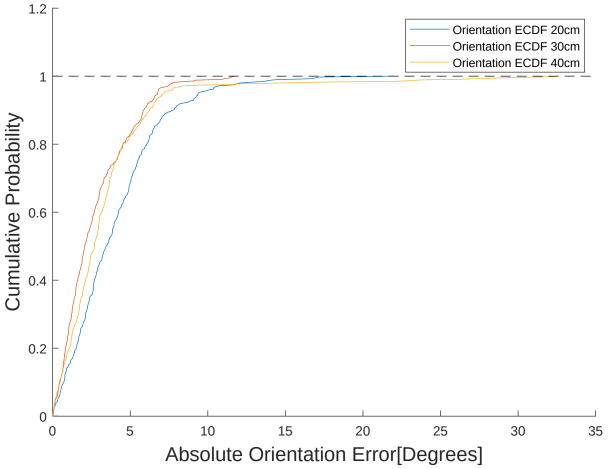
(a) Position results for the stationary test with a tag separation of 30 cm.

(b) A close-up of the stationary measurements.

Figure 12: Unfiltered and filtered measurements for the stationary test with a tag separation of 30 cm.



(a) Empirical CDF for the error in distance.



(b) Empirical CDF for the absolute error in orientation.

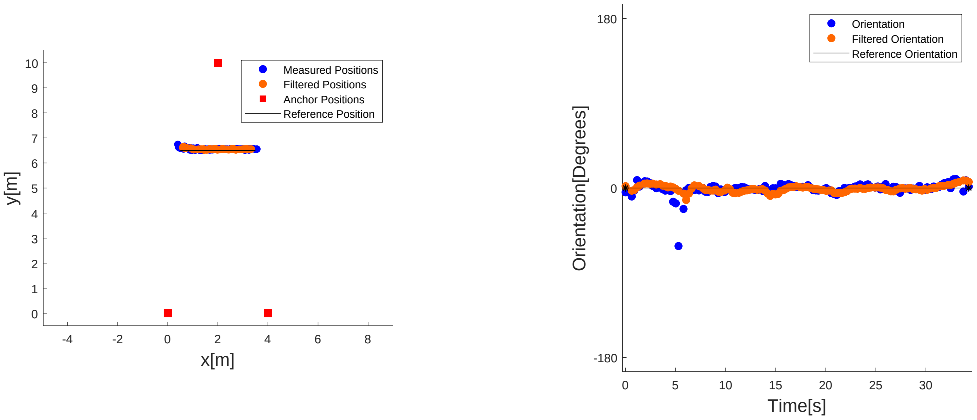
Figure 13: The figure shows the empirical cumulative distribution functions with regard to the error in position and orientation for the stationary test. The blue line represents 20 cm separation between the tags, the red line represents 30 cm separation between the tags and the yellow line represents 40 cm separation between the tags.

Table 3: In this table the results from the tests with a stationary tag are presented. Each column contains the result from one of the three separation distances of the tags.

Tag Separation	20 cm	30 cm	40 cm
Mean Position Error[m]	0.077	0.045	0.053
Position Standard Deviation[m]	0.024	0.021	0.031
Mean Orientation Error[deg]	2.41	1.49	2.46
Orientation Standard Deviation[deg]	4.47	3.21	4.53

6.2 Line Path

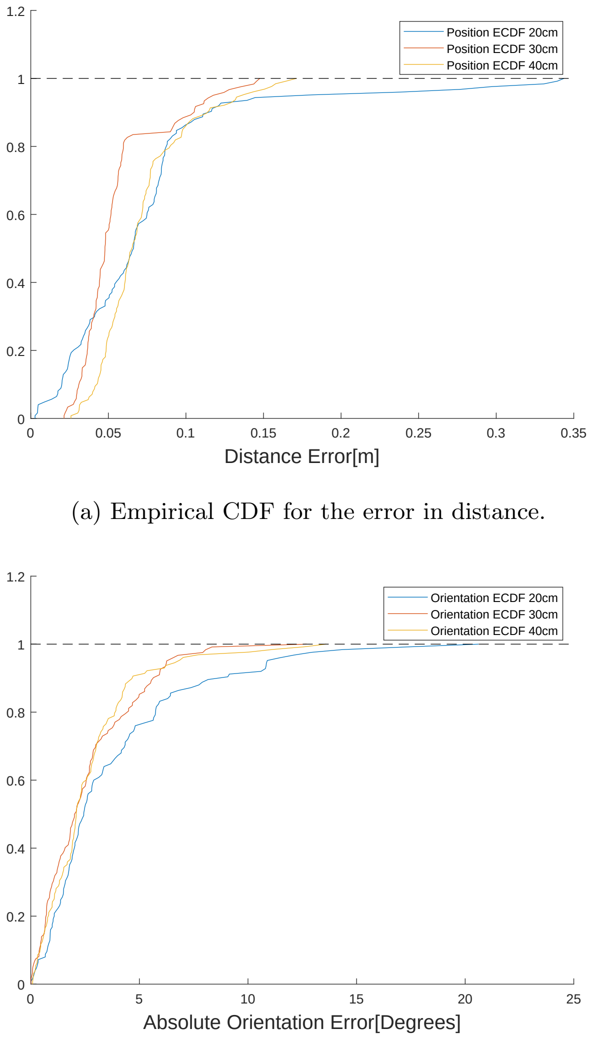
In figure 14 the position and orientation measurements as well as the filtered positions and orientation are plotted with a tag separation of 30 cm. The red squares are the anchor positions, the black line is the true position, the blue dots are the measured values and the orange dots are the filtered values. The graphs for 20 cm and 40 cm tag separation has been moved to appendix D since they all look similar. In figure 15 the empirical cumulative distribution functions of the orientation and position error are plotted for each of the tag separation distances. Finally the values acquired from the data analysis are presented in table 4.



(a) Position results for the line path with a tag separation of 30 cm.

(b) Orientation results for the line path with a tag separation of 30 cm.

Figure 14: Unfiltered and filtered measurements for the line path.



(a) Empirical CDF for the error in distance.

(b) Empirical CDF for the absolute error in orientation.

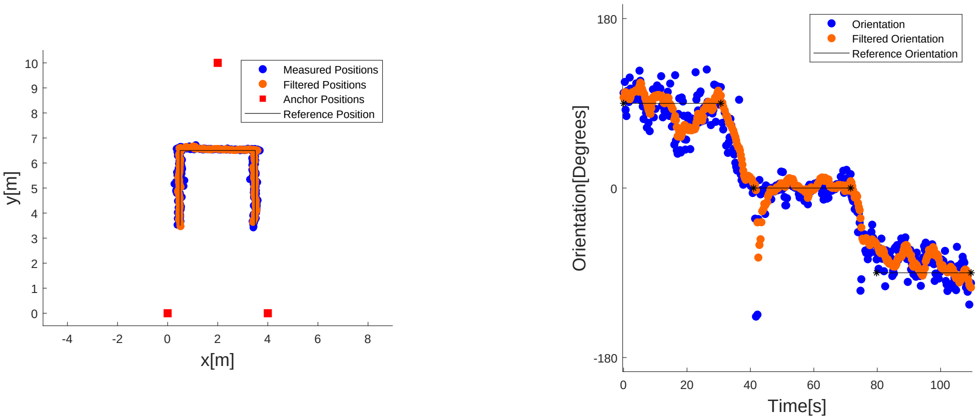
Figure 15: The figure show the empirical cumulative distribution functions with regard to the error in position and orientation for the line path. The blue line represents 20 cm separation between the tags, the red line represents 30 cm separation between the tags and the yellow line represents 40 cm separation between the tags.

Table 4: In this table the results from the line path test are presented. Each column contains the result from one of the three separation distances of the tags.

Tag Separation	20 cm	30 cm	40 cm
Mean Distance Error[m]	0.073	0.056	0.072
Mean Absolute Orientation Error[deg]	3.72	2.58	2.64

6.3 Square Path

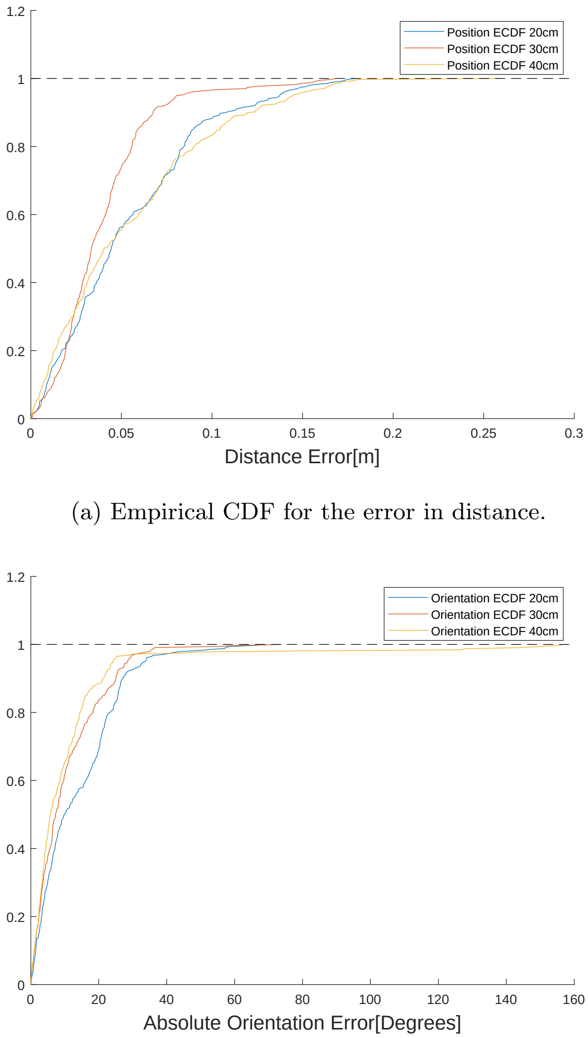
In figure 16 the position and orientation measurements as well as the filtered positions and orientation are plotted with a tag separation of 30 cm. The red squares are the anchor positions, the black line is the true position, the blue dots are the measured values and the orange dots are the filtered values. The graphs for 20 cm and 40 cm tag separation has been moved to appendix D since they all look similar. In figure 17 the empirical cumulative distribution functions of the orientation and position error are plotted for each of the tag separation distances. Finally the values acquired from the data analysis are presented in table 5.



(a) Position results for the square path with a tag separation of 30 cm.

(b) Orientation results for the square path with a tag separation of 30 cm.

Figure 16: Unfiltered and filtered measurements for the square path.



(a) Empirical CDF for the error in distance.

(b) Empirical CDF for the absolute error in orientation.

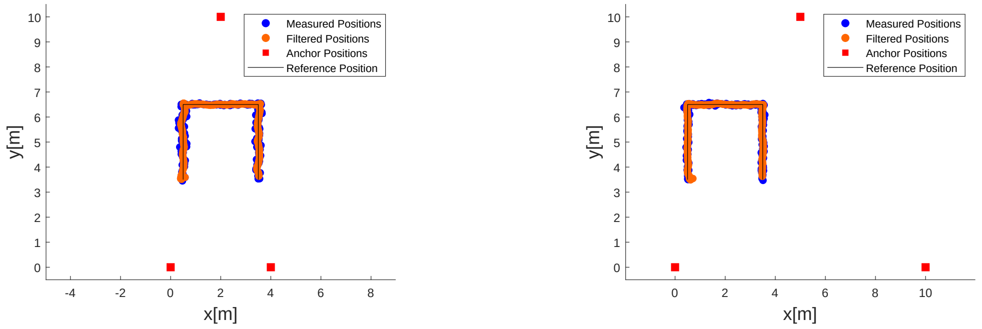
Figure 17: The figure show the empirical cumulative distribution functions with regard to the error in position and orientation for the square path. The blue line represents 20 cm separation between the tags, the red line represents 30 cm separation between the tags and the yellow line represents 40 cm separation between the tags.

Table 5: In this table the results from the square path test are presented. Each column contains the result from one of the three separation distances of the tags.

Tag Separation	20 cm	30 cm	40 cm
Mean Distance Error[m]	0.054	0.040	0.055
Mean Absolute Orientation Error[deg]	13.8	10.5	11.1

6.4 Simulations

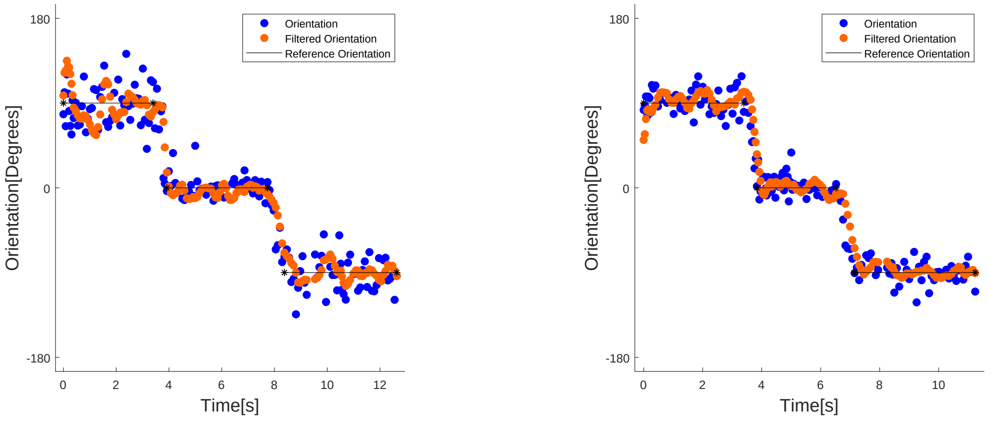
Two simulations of the square path were done with different anchor placements. Both simulations had a tag separation of 40 cm. The simulated positions and orientations from both simulations, with the anchors placed similarly to the test runs, can be seen in figure 18 and 19 respectively. Where in sub-figures (a) show results with narrow anchor placements at the coordinates (0, 0), (4, 0) and (2, 10). Sub-figures (b) show results with wide anchor placements at the coordinates (0, 0), (10, 0) and (5, 10). Data analysis for both anchor placements was done and the results are shown in table 6.



(a) Position results for the simulated square path with narrow anchor placements.

(b) Position results for the simulated square path with wide anchors placements.

Figure 18: Unfiltered and filtered positions, simulated for the square path with different anchor setups.



(a) Orientation results for the simulated square path with narrow anchor placements.

(b) Orientation results for the simulated square path with wide anchor placements.

Figure 19: Unfiltered and filtered orientations, simulated for the square path with different anchor setups.

Table 6: In this table the data analysis results from the simulations of the square path tests are presented. The two columns contains the result for the two mentioned anchor placements.

Anchor Placement	Narrow	Wide
Mean Distance Error[m]	0.028	0.019
Mean Absolute Orientation Error[rad]	9.17	6.07

7 Discussion

7.1 System Accuracy

To understand the accuracy of the system the metrics used in the tests need to be motivated, see section 5.5. In the stationary tests the position and orientation mean was computed. These show the biased error in position and orientation. Meaning, it will not take into account the spread of the measurements. Instead the spread is captured by the standard deviation in position and orientation. In the tests with moving tags these are combined into one metric for position and another for orientation. These are the mean distance error and mean absolute orientation error. They will include both the spread of the measurement and the bias offset of each measurement. Giving an understanding of the general performance of the system.

Tables 3, 4 and 5 displays the accuracy of the system for the three tests. Generally the system has high accuracy in position and distance which is below 8 cm in mean error for all test setups. The standard deviation in position can be seen in table 3 where it is around 3 cm for all separation distances. This standard deviation shows that the test are precise and that reliability of a single measurement is quite high. Compared to the mean position error, the mean distance errors seen in the tables for the path tests, 4 and 5, are large. The reason for this is the earlier mentioned choice of metrics. In order to understand the standard deviation of the moving test, the incline of the ECDF graphs, figures 15 and 17, can be studied. These graphs show that the standard deviation remained in the same magnitude for the path tests as the stationary tests except for some sporadic values. Generally the system performs beyond expected in position. With an accuracy better than similar UWB systems discussed in section 4, where the systems had an accuracy between 10 cm and 30 cm. The best performing tag separation for positioning seemed to be 30 cm. Figures 13a, 15a, and 17a compare the distance errors between the tag separations. During the stationary tests, a tag separation of 30 cm increased the accuracy by approximately 10 % and 30 % compared to the other tag separations. However, this increase in accuracy corresponds to a few centimeters. Even without the increase the position results compare well to other indoor positioning systems.

Improvements due to tag spacing for orientation can be seen in figures 13b, 15b, and 17b. In the moving tests the optimal tag separation seem to be 30 cm with an increase of orientation accuracy of approximately 30 % compared to 20 cm tag spacing. A tag separation of 40 cm seemed to perform similar to a tag separation of 30 cm. This hints that the performance increase might be saturated at 30 cm separation. The separation of 30 cm is said to be better than 40 cm as lower dimension of the system is preferred. The improvement in orientation may be a result of few realizations of the measurements. To make a statement on the true optimal tag spacing, further study is needed.

Orientation results for the system were different for the stationary and line path tests compared to the square test. The results from the former tests were below 4 degrees in mean orientation error and mean absolute orientation error respectively. A mean error of this magnitude is comparable to other positioning systems that compute orientation, such as Cricket[44] and Luxapose[28]. However, the standard deviation for the stationary test were determined to be approximately 3-5 degrees. This is large in comparison to the difference in mean error between the separation distances. This means that the comparison between different separation distances might be uncertain. Although, when looking at the results for the square path, in section 6.3, the orientation errors are in the magnitude of 10-15 degrees which is significantly larger than in the other tests. In figure 16b it can be seen that the error is mostly present during the beginning and end of the path.

7.2 System Errors

Overall, the performance of the system is good except for some errors and problems. The first noticeable error is the mean position error in the stationary test, seen in table 3. Measuring a stationary position with 400 measurements should result in an accurate mean position. Though, in figure 12b we see clearly that the measurements are shifted from the reference position which results in the mean position error of 4-8 cm. An explanation to this error could be that the background disturbances were different in the testing area compared to where the calibration was done. Therefore an offset would be induced in the ranging process and result in a trilateration bias. Further, the calibration does have errors in some connections that can also be an explanation for the error. Another factor that affect the stationary offset is the accuracy of the reference point. During the measurements the robot was placed with the center of the robot on the reference point. The error induced in this method could be in the magnitude of centimeters. This would result in a biased error in position. Another factor that could cause a bias in the stationary test is the change of signal strength. Receiving a pulse with more energy will result in a shorter ToF and less energy will result in a longer ToF. This phenomenon is caused by a shifting threshold power level depending on the amplitude of the pulse. Meaning that the calibration is not fully correct when the signal strength is changed.

Another significant error was the orientation error during the square test, see table 5. Seen in figure 16b, the orientation error of the system is largest in the beginning and the end of the test. These times correspond to the paths along the defined y-axis, see section 5.4. From figure 17b it seems that the high absolute orientation error is because of large standard deviation. The source of these errors could be the positions of the tags in comparison to the placement of the anchors. Placement of the anchors was done in regard to the testing area to achieve line-of sight. Therefore, the anchors placed on the defined x-axis were only separated by 4 m while the third anchor was separated from the other anchors by more than 10 m. The asymmetry of the anchors' placements can be the reason for large error in orientation for the square path. This leads to an uneven standard deviation that can be seen in the stationary test, see figure 12b, where the measured positions of the robot has a higher spread along the x-axis than the y-axis. High standard deviation for the square path can be an explanation for the oscillations in the orientation graph, figure 16b. Also the filter was not optimized for measurements with this standard deviation. Given a higher spread, the filter tries to follow the measurements and starts oscillating around the reference orientation. The problems that occurred during the square path can also be a result of change in signal strength levels as mentioned earlier. The change of signal strength is large when the reference path is close to the anchors, affecting the square path measurements the most.

A last minor problem is a delay in the measurements caused by dual tag communication. There will be a minor offset in the position measurements during the path test due to the fact that measurements for both tags can not be done simultaneously. When the object is moving, the true position will change between the two measurements. The error caused by this will depend on the velocity of the object. With the velocity, used in the tests, of 10-15 cm/s, this effect would only cause a minor error in position.

7.3 Improvements

Some of the above mentioned errors can be solved or avoided for further testing of the system. To reduce the issue of a bias in the stationary position, the calibration could be done in the same location as the measurements were done. This might improve the accuracy of the position, hence improving the stationary results as well as the results for the path measurements.

One major improvement of the system could be the anchor placement. Placing the anchors to form a equilateral triangle would presumably make the measurements more symmetrically distributed with improved accuracy. The reasoning behind this change is that with a wider angle between the anchors would make the overlapping areas of error smaller. A smaller and symmetrical error area would make the orientation measurements of the system more stable and accurate. This change can be seen in the simulation results in section 6.4 where two anchor placements were tested. In figure 19 the orientation of a simulation with the same anchor placements as in the tests were used. The results from this simulation shows that the large standard deviation during the beginning and end of the path is still apparent in the simulations. The other simulation has a wider placements of the anchors and give the orientation result shown in figure 19b. In this result the orientation is more stable than the previous simulation. This supports our hypothesis that the orientation performance during the square path measurements would improve with a wider anchor placement. Comparing the data analysis of the simulation, in table 6, show that the accuracy of orientation is increased by more than 30 % with a wider anchor placement. It also shows that the position performance is increased with a wider anchor placement. In summary, it seems that the performance could be increased substantially by changing the anchor placement. Though further study is needed to find the optimal placement. Another approach to change the overlapping area of errors would be to position the object as close to the center of the the anchors, as possible. Conceivably, when the tags are located in the middle of the anchors the symmetry of the error area should be optimal. This is achieved by choosing the locations of the anchors to have an overlapping area as close to the middle as possible. These presumptions can be tested by stationary measurements in different locations.

Having a larger separation between the anchors might also result in less change of signal strength level as the positions will be further away from the anchors. Resulting in less errors from the change of signal strength. Another way to reduce these errors is to construct an algorithm that compensates for this affect. The DW1000 modules have registers that store the signal strength of the received message, in which can be used for the compensation. But a study need to be conducted to know how the compensation function should be modeled.

7.4 System Performance Summary

The general idea behind the scope is that the system should be able to navigate a robot in an indoor environment. Meaning that the system need to have high accuracy while still processing in real-time. The sampling frequency of the system was 4 Hz. Sampling frequency is determined by the frequency at which the center position and orientation is calculated. A sampling frequency of 4 Hz combined with mean error of below 10 cm and 6 degrees are of reasonable magnitude to handle an robot traveling in an indoor environment. Though, it will depend on the application and required speed of the robot. The statement assume that the mentioned improvement has been done and the precision of orientation is improved in the square path.

In comparison to other indoor positioning systems mentioned in section 4 the system has reasonable cost and scalability for indoor environments. The cost of the system is comparable with similar UWB positioning systems, with the added cost of one tag. One additional tag will increase the cost per tracked object. However, the processing for both tags is currently done in one single microprocessor, so the added cost is only of one DW1000 transceiver. Further, increasing the number of tags will affect the scalability of the system. On the other hand, in the application of an indoor robots, the number of robots to be tracked is likely few compared to the area of effect. To fully evaluate the scalability of the implemented system, an algorithm for added tags and anchors need to be studied and implemented.

8 Conclusion

During the project an indoor positioning system with a dual tag setup has been developed. This system had some flaws that lead to an uneven standard deviation of the measurements. If these problems were to be solved with the methods suggested in the report, the system would likely perform in parity with existing systems. Which means that it would be feasible to use a dual-tag system to achieve a similar accuracy and precision as existing indoor positioning systems in terms of orientation and position.

9 Future Work

Here, some ideas for further expanding upon the project will be brought up and discussed.

9.1 Anchor Placement

As mentioned in the discussion, section 7, the system was more accurate when driving in x-direction than in y-direction. This is believed to be because of the anchor placement. The distance between the anchors in x-direction is only 4 m compared to y-direction where the distance is 10 m. This could lead to a higher standard deviation in x-direction.

It would be interesting to investigate if the problem is solved by instead placing the anchors at (0, 0), (10, 0) and (5, 10). It would also be interesting to increase the amount of anchors to see if a better result could be achieved this way. However, to make this possible an algorithm to take more than three anchors into account when calculating the position has to be developed. In addition to the possibility of better measurements the scalability of the system also increases if more anchors can be added easily. If it's possible to use an arbitrary number of anchors, it's possible to easily scale the area where the system can operate by just adding more anchors.

9.2 Movement Patterns

It is also relevant to investigate how the system performs in a wider variety of movement patterns. Only two different paths were investigated in this thesis. Both of which were based on movement in straight lines parallel to the x- and y-axis. To expand upon these, more paths could be added. For example with diagonal or circular elements to see how the system perform under those conditions. Since the system performs differently when going in x direction versus y-direction it's reasonable to think that it would perform differently in other directions as well.

If it's possible to find a correlation between the error and the direction of the tag, this could be implemented into the filter to make an attempt at correcting the error.

9.3 Sampling Rate

One of the drawbacks of the system was that the sampling rate of the position and orientation was very low, only 4 Hz. With an increased sampling rate there would be more measurements available for the filter. Which would mean that the filter can adjust to change better and also have an increased accuracy overall. One of the reasons for the low sampling rate was that the two UWB-modules could not be run simultaneously. This was because the same microcontroller, the Raspberry Pi, was used to control both of them. One way to go about solving this would be to use a separate microcontroller to control each of the modules. Then have these communicate the measured distances to a main microcontroller which takes care of processing the measurements. Using this method, it should be possible to double the sampling rate since both of the modules can be run simultaneously.

This also has the benefit of more current measurements. With our method the measurements from the different modules are never measured at the same time, but about half a period apart. This was not really a problem with our test setup since the tag moved so slowly that this delay was not more than a few centimeters. But if this problem is solved the tag can move at a faster speed without the system having problems because of this.

9.4 Multiple Tags

To further expand the system support for multiple tags could be implemented. As of now the system doesn't support this. However, it's still possible to use a few more tags without any notable increase in signal collisions. But at some point a coordinator tag will have to be implemented to coordinate the air time of each tag. Otherwise the collisions of signals would result in such a low successful sampling rate that the system would be rendered slow to be used.

The support for multiple tags is not required to test the performance of the system. But for most real world applications more than one tag is desirable. If the system is going to be used in such an application, this feature would be useful.

10 References

[1] Santiago Mazuelas, Alfonso Bahillo, Ruben M. Lorenzo, Patricia Fernandez, Francisco A. Lago, Eduardo Garcia, Juan Blas, and Evaristo J. Abril, "Robust Indoor Positioning Provided by Real-Time RSSI Values in Unmodified WLAN Networks", October 2009

[2] Yihong Qi, Princeton University, "Wireless Geolocation In a Non-Line-Of-Site Environment", November 2003

[3] Eiman Elnahraway, Xiaoyan Li, Richard P. Martin, "Poster Abstract: The Limits of Localization Using RSSI", November 2004

[4] Oguejiofor O.S, Aniedu A.N, Ejiofor H.C, Okolibe A.U, "Trilateration Based localization Algorithm for Wireless Sensor Network", International Journal of Science and Modern Engineering, September 2013 ISSN: 2319-6386, Volume-1, Issue-10

[5] Richard B. Langley, University of Brunswick, "Dilution of Precision", GPS WORLD, May 1999

[6] Charles Cohen and Frank V. Koss, University of Michigan, "A Comprehensive Study of Three Object Triangulation", May 1993

[7] João Sena Esteves, Adriano Carvalho, Carlos Couto, "Generalized Geometric Triangulation Algorithm for Mobile Robot - Absolute Self-Localization"

[8] Vincent Pierlot, Marc Van Droogenbroeck, University of Liège, "A New Three Object Triangulation Algorithm for Mobile Robot Positioning", June 2014

[9] Josep Maria Font, Joaquim A. Batlle, Technical University of Catalonia, "Mobile Robot Localization. Revisiting the Triangulation Methods", 2006

[10] Bluetooth positioning based on weighted K-nearest neighbors and adaptive bandwidth mean shift [Online], Available: https://journals.sagepub.com/doi/full/10.1177/1550147717706681, [Accessed: Jan. 18, 2019]

[11] Azkario Rizky Pratama, Widyawan, Risanuri Hidayat, "Smartphone-based Pedestrian Dead Reckoning as an Indoor Positioning System", International Conference on System Engineering and Technology, 2012

[12] Rong Peng, Mihail L. Sichitiu, "Angle of Arrival Localization for Wireless Sensor Networks", 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, 2006

[13] Approaches for Angle of Arrival Estimation [Online], Available: https://www.cs.utexas.edu/~swadhin/reading_group/slides/AoA.pdf, [Accessed: Jan. 22, 2019]

[14] Fredrik Gustafsson, Fredrik Gunnarsson, Positioning Using Time-Difference Of Arrival Measurements, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003

[15] Nicolas Le Dortz, Florian Gain, Per Zetterberg, "WIFI Fingerprint Indoor Positioning System Using Probability Distribution Comparison", 2012

[16] Beom-Ju Shin, Kwang-Won Lee, Sun-Ho Choi, Joo-Yeon Kim, Woo Jin Lee, and Hyung Seok Kim, Sejong University, "Indoor WiFi Positioning System for Android-based Smartphone", 2010

[17] Sebastian Hilsenbeck, Dmytro Bobkov, Georg Schroth, Robert Huitl, and Eckehard Steinbach, Technische Universitat München, "Graph-based Data Fusion of Pedometer and WiFi Measurements for Mobile Indoor Positioning", September 2014

[18] Elena Simona Lohan, Jukka Talvitie, Pedro Figueiredo e Silva, Henri Nurminen, Simo Ali-Löytty, Robert Piché, Tampere University of Technology, "Received Signal Strength models for WLAN and BLE-based indoor positioning in multi-floor buildings", 2015

[19] Myungin Ji, Jooyoung Kim, Juil Jeon, Youngsu Cho, "Analysis of Positioning Accuracy corresponding to the number of BLE beacons in Indoor Positioning System", July 2015

[20] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A. Al-Ammar, Hend S. Al-Khalifa, "Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances", 2016

[21] Mari Saua Svalastog, University of Oslo, Indoor Positioning - Tehnologies, Servies and Arhitetures, May 2007

[22] S.L. Ting, S.K. Kwok, Albert H.C. Tsang, George T.S. Ho, The Hong Kong Polytechnic University, "The Study on Using Passive RFID Tags for Indoor Positioning", International Journal of Engineering Business Management, Vol. 3, No. 1, 2011

[23] Abdelmoula Bekkali, Horacio Sanson and Mitsuji Matsumoto, "RFID Indoor Positioning based on Probabilistic RFID Map and Kalman Filtering", Wireless and Mobile Computing, Networking and Communications (WiMob 2007)

[24] LIONEL M. NI, YUNHAO LIU, YIU CHO LAU, ABHISHEK P. PATIL, "LANDMARC: Indoor Location Sensing Using Active RFID", Wireless Networks 10, 701–710, 2004

[25] S Ravindra, S N Jagadeesha, "Time Of Arrival Based Localization in Wireless Sensor Networks: A Linear Approach", Signal and Image Processing: An International Journal (SIPIJ) Vol.4, No.4, August 2013

[26] Binghao Li, Thomas Gallagher, Andrew G Dempster, Chris Rizos, "How feasible is the use of magnetic field alone for indoor positioning?", International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012

[27] Han-Sol Kim, Woojin Seo, and Kwang-Ryul Baek, Indoor Positioning System Using Magnetic Field Map Navigation and an Encoder System, 2017

[28] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, Prabal Dutta, University of Michigan, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light", September 2014

[29] Weizhi Zhang, M. I. Sakib Chowdhury, Mohsen Kavehrad"Asynchronous indoor positioning system based on visible light communications", Optical Engineering 53, April 2014

[30] Pozyx website [online], Available: https://www.pozyx.io/, [Accessed: Jan. 24, 2019]

[31] Josef Kulmer, Stefan Hinteregger, Bernhard Großwindhager, Michael Rath, Mustafa S. Bakr, Erik Leitinger, Klaus Witrisal, "Using DecaWave UWB Transceivers for High-accuracy Multipath-assisted Indoor Positioning", IEEE International Conference on Communications Workshops, 2017

[32] Pozyx Website, System Performance [Online], Available: https://www.pozyx.io/Documentation/Datasheet/SystemPerformance, [Accessed: Jan. 25, 2019]

[33] Infsoft website [online], Available: https://www.infsoft.com/, [Accessed: Jan. 24, 2019]

[34] Infsoft Sensor Technology [online], Available: https://www.infsoft.com/technology/sensors, [Accessed: Jan. 25, 2019] https://www.infsoft.com/technology/sensors

[35] Infsoft White Paper [online], Available: https://cdn.infsoft.com/www/images/solutions/basics/whitepaper/infsoft-Whitepaper-EN-Indoor-Positioning_download.pdf, [Accessed: Jan. 25, 2019]

[36] IndoorAtlas website [online], Available: https://www.indooratlas.com/, [Accessed: Jan. 25, 2019]

[37] IndoorAtlas Technologies Website [online],
Available: https://www.indooratlas.com/positioning-technology/, [Accessed: Jan. 25, 2019]

[38] Eliko website [online], Available: https://www.eliko.ee/, [Accessed: Jan. 25, 2019]

[39] Eliko KIO RTLS Specification [online], Available: https://www.eliko.ee/products/kio-rtls/, [Accessed: Jan. 25, 2019]

[40] OpenRTLS Website [online], Available: https://openrtls.com/, [Accessed: Jan. 25, 2019]

[41] OpenRTLS Website [online], Available: https://openrtls.com/page/rtls, [Accessed: Jan. 25, 2019]

[42] Giorgio Conte, Alessandro Antonio Nacci, Massimo De Marchi, Vincenzo Rana and Donatella Sciuto, "BlueSentinel: A first approach using iBeacon for an energy efficient occupancy detection system", 1st ACM International Conference on Embedded Systems For Energy-Efficient Buildings (BuildSys), 2014

[43] Apple Developer Website on iBeacons [Online], Available: https://developer.apple.com/ibeacon/, [Accessed: Jan. 28, 2019]

[44] Nissanka Bodhi Priyantha, "The Cricket Indoor Location System", PhD Dissertation, 2005

[45] Sudeep Pasricha, Viney Ugave, Charles W. Andersson, Qi Han, "LearnLoc: A Framework for Smart Indoor Localization with Embedded Mobile Devices", ISBN 978-1-4673-8321-9/15, 2015

[46] He Xu, Ye Ding, Peng Li, Ruchuan Wang, Yizhu Li, "An RFID Indoor Positioning Algorithm Based on Bayesian Probability and K-Nearest Neighbor", Sensors 17.8, August 2017)

[47] Decawave Website DW1000 module [Online],
Available: https://www.decawave.com/product/dwm1000-module/, [Accessed: Feb. 18, 2019]

[48] DW1000 User Manual [Online],
Available: https://thetoolchain.com/mirror/dw1000/dw1000_user_manual_v2.05.pdf, [Accessed: Feb. 26, 2019]

[49] DW1000 Python Library [Online],
Available: https://github.com/ThingType/DW1000_Python_library, [Accessed: Mar. 21, 2019]

[50] DW1000 Arduino Library [Online], Available: https://github.com/ThingType/arduino-dw1000, [Accessed: Mar. 21, 2019]

[51] Arduino Pro Mini Datasheet [Online],
Available: https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/ProMini8MHzv1.pdf, [Accessed: Mar. 22, 2019]

[52] Raspberry 3B+ Specifications [Online],
Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/, [Accessed: Mar. 22, 2019]

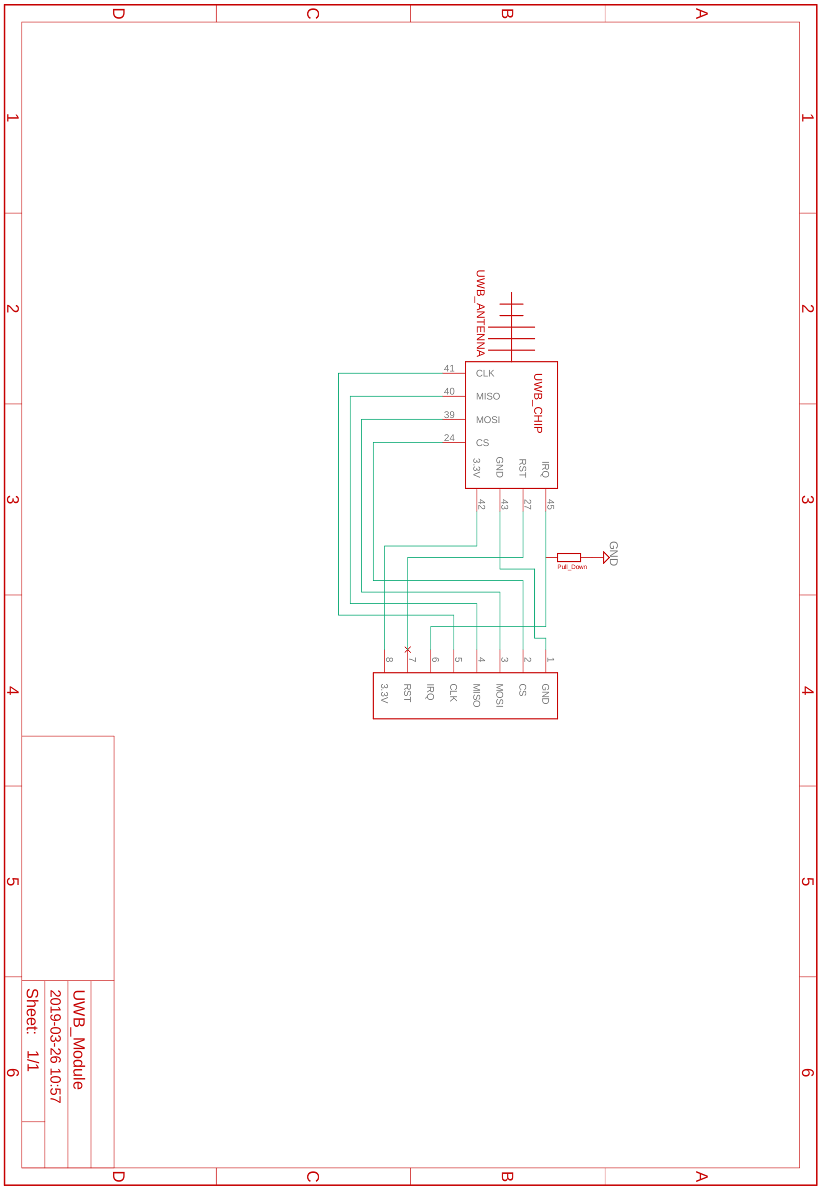
A Code

All of the code for this project can be found on Github: https://github.com/oscjoh95/Indoor-Positioning-System-DW1000-with-Orientation

B Circuit diagrams

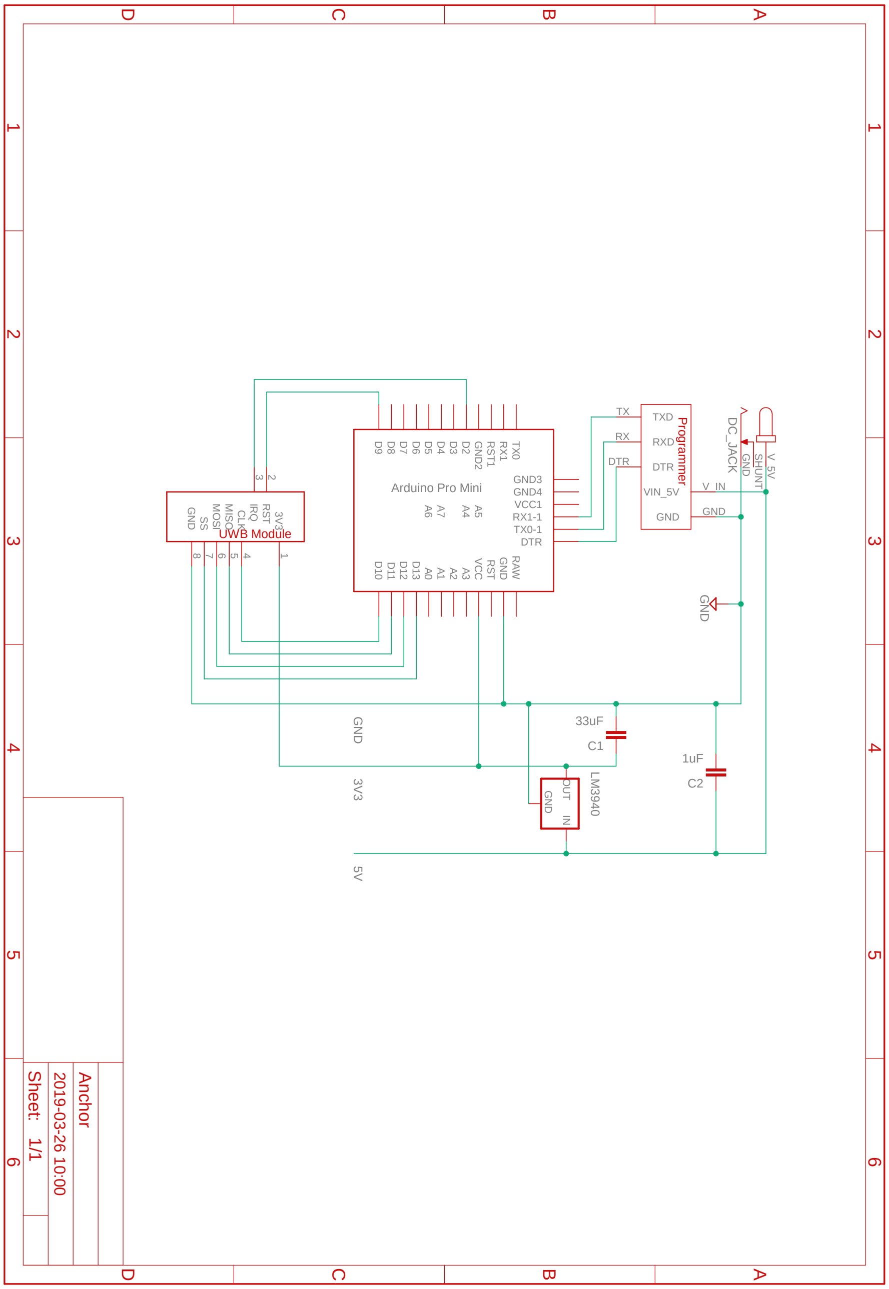
These are the circuit diagrams for the project.

B.1 UWB-module



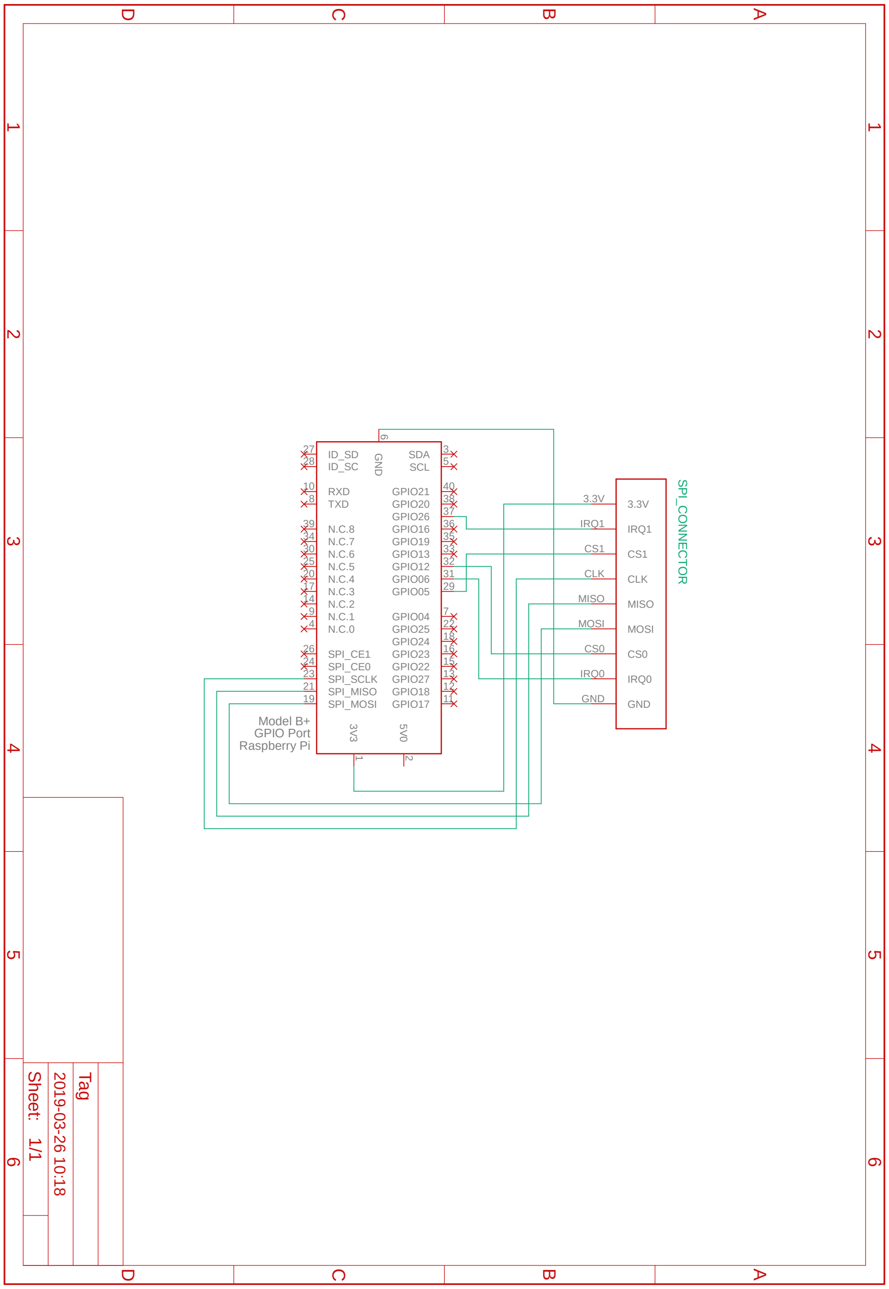
Circuit diagram of the UWB-module used in the project.

B.2 Anchors



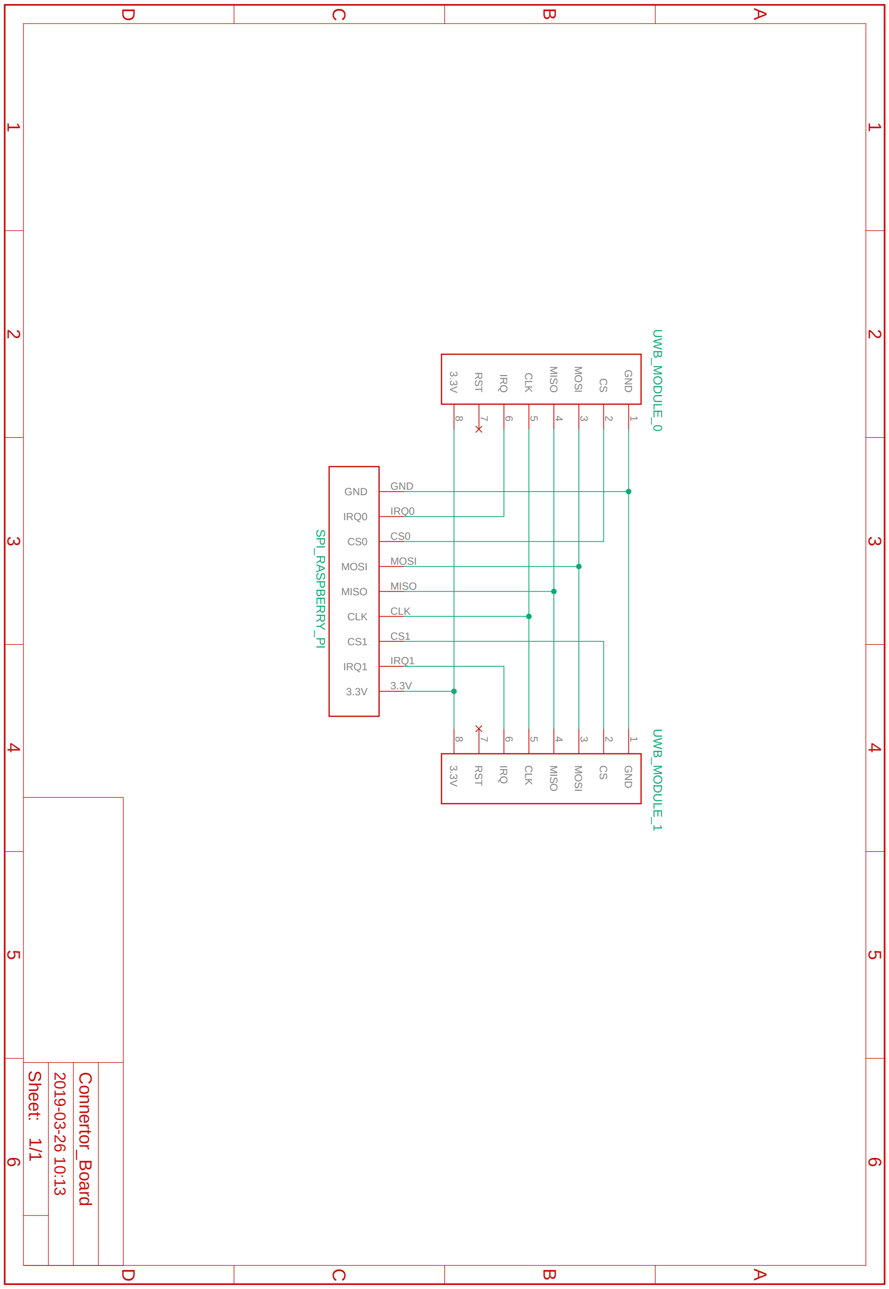
Circuit diagram of the anchor used in the project.

B.3 Tag



Circuit diagram of the tag used in the project.

B.4 Connector Board



Circuit diagram of the connector board used in the project.

C Calibration results

This table show the calibrated values for the five modules, anchors $A0$ to $A2$ and tags $T3$ and $T4$. The antenna was calibrated with $T3$ as reference, with antenna delay at 250400.00 ps.

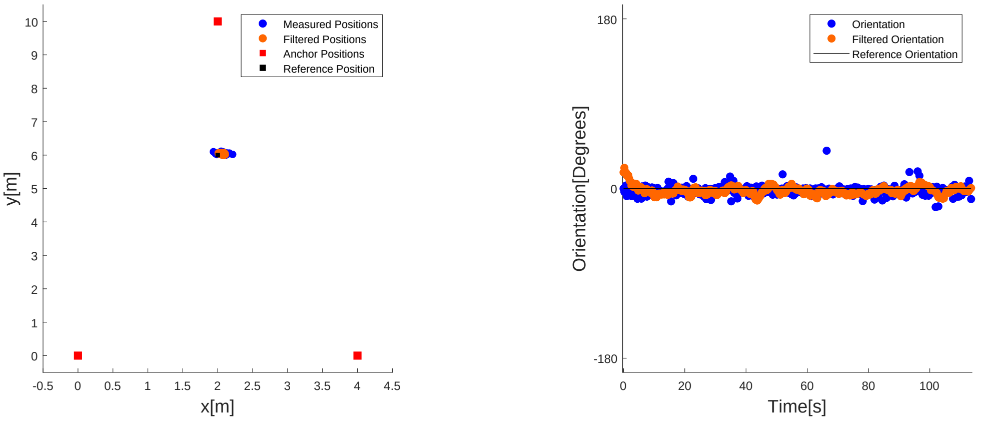
Module	**Antenna Delay Constant[ps]**
A0	259883.90
A1	258851.00
A2	259085.75
T3	250400.00
T4	250869.50

This table notes the standard deviation and mean error during the calibration. Where $\Delta_{n,p}$ represents the communication between module n and module p. The mean error is the mean of the difference between measured distance and its true distance with the calibrated antenna delay using 500 measurements.

Communication	**Mean Error**	**STD[m]**
$\Delta_{0,3}$	0.00	0.03
$\Delta_{1,3}$	0.00	0.02
$\Delta_{2,3}$	0.00	0.03
$\Delta_{0,4}$	0.01	0.03
$\Delta_{1,4}$	0.05	0.04
$\Delta_{2,4}$	0.00	0.05

D Plots

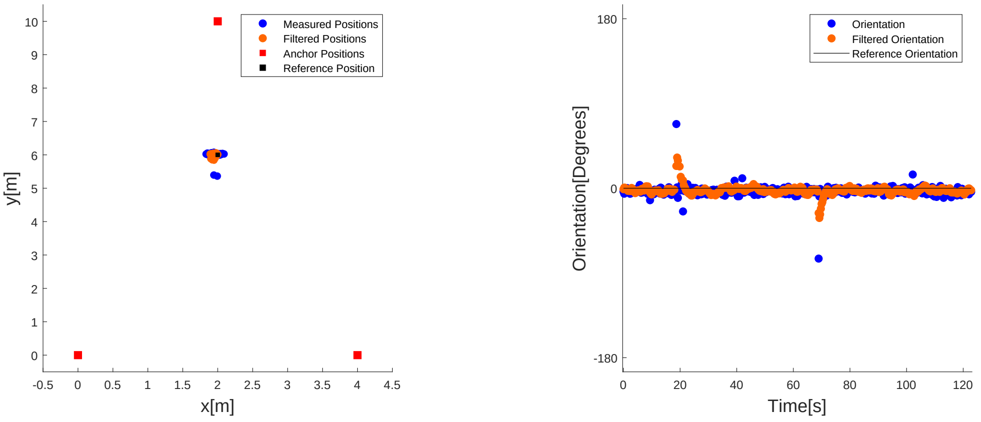
D.1 Stationary Test 20 cm



(a) Position results for the stationary test with a tag separation of 20 cm.

(b) A close-up of the stationary measurements with a tag separation of 20 cm.

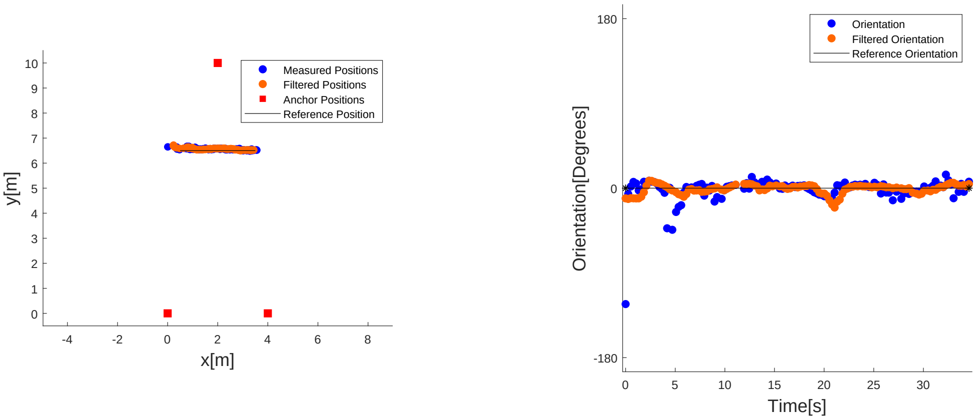
D.2 Stationary Test 40 cm



(a) Position results for the stationary test with a tag separation of 40 cm.

(b) A close-up of the stationary measurements with a tag separation of 40 cm.

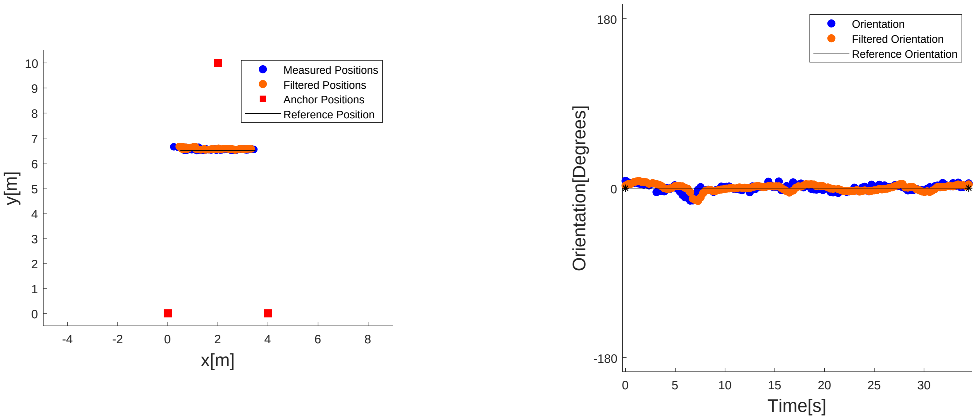
D.3 Line Path 20 cm



(a) Position results for the line path with a tag separation of 20 cm.

(b) Orientation results for the line path with a tag separation of 20 cm.

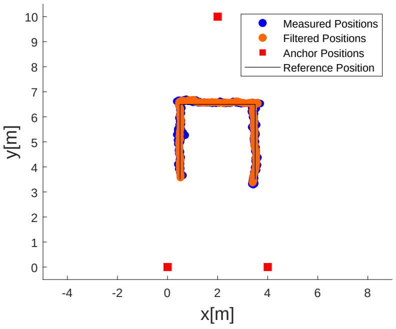
D.4 Line Path 40 cm



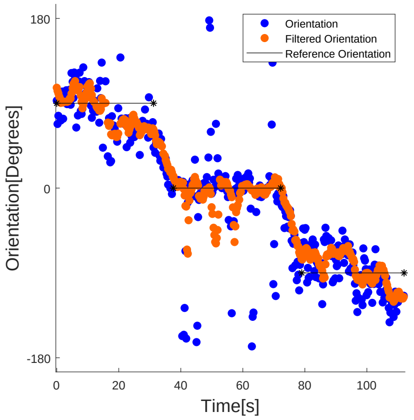
(a) Position results for the line path with a tag separation of 40 cm.

(b) Orientation results for the line path with a tag separation of 40 cm.

D.5 Square Path 20 cm

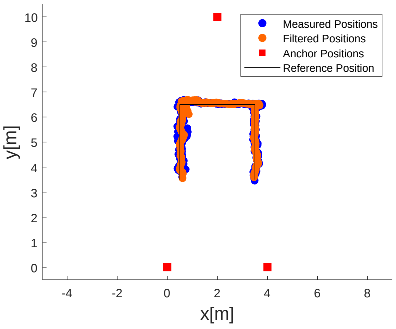


(a) Position results for the square path with a tag separation of 20 cm.

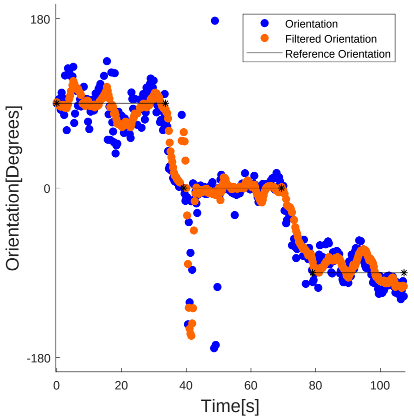


(b) Orientation results for the square path with a tag separation of 20 cm.

D.6 Square Path 40 cm



(a) Position results for the square path with a tag separation of 40 cm.



(b) Orientation results for the square path with a tag separation of 40 cm.