TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Klaus Kaspar Kasak 134673IAPB

DEVELOPMENT OF POSITIONING SYSTEM ANALYSIS SOFTWARE ON THE EXAMPLE OF ULTRA-WIDEBAND TECHNOLOGY

Bachelor's thesis

Supervisor: Taavi Laadung
MSc

Gert Kanter
MSc

Tallinn 2020

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Klaus Kaspar Kasak

5.20.2020

Abstract

The aim of the thesis is to implement, validate and characterize the software developed by the author, which calculates and analyzes the statistical parameters of the output coordinates of real time positioning systems (RTLS). *Python* was chosen as the programming language to accomplish this task, along with several mathematics and graphics libraries that allowed for calculations and graph plotting respectively.

Two ultra-wideband (UWB) systems were chosen as the basis for which the software was developed and validated on, as they are currently one of the most popular technologies for indoor positioning. The validation consisted of a series of tests, during which the systems were utilized to collect positioning data.

The output of the software enabled the user to effectively evaluate and compare the performance of the systems, demonstrating and confirming its potential use for developers and end-users of real time positioning systems.

This thesis is written in English and is 50 pages long, including 8 chapters, 22 figures and 3 tables.

Annotatsioon

POSITSIONEERIMISSÜSTEEMIDE ANALÜÜSIMISE TARKVARA ARENDUS ÜLILAIRIBATEHNOLOOGIA NÄITEL

Käesoleva bakalaureusetöö eesmärk on luua, valideerida ja kirjeldada autori poolt arendatud tarkvara. Programm kogub reaalaja positsioneerimissüsteemide väljastatud koordinaate, töötleb andmeid ning väljastab graafikuid ja arvutatud parameetreid. Väljundi põhjal on võimalik analüüsida erinevate positsioneerimis-süsteemide täpsus- ja tabavusvõimekust.

Arenduseks valiti programmeerimiskeel *Python*, kasutades sisse-ehitatud matemaatika ja graafikute joonestamise teeke. Programmi sisendiks on positsioneerimis-süsteemi poolt arvutatud koordinaadid ning jälgitava seadme asukoha tõelised koordinaadid. Väljundiks on mitmesugused graafikud ning statistilised näitajad nagu koordinaadi keskmine- ning ruutkeskmine viga ja standardhälve, mis iseloomustavad testitava süsteemi positsioneerimise täpsust ja tabavust.

Töös on läbi viidud näidiskatsed, mille tulemusel kogutud andmeid kasutati programmi sisendina. Näidiskatsed ja analüüs viidi läbi kahemõõtmelise ja kolmemõõtmelise positsioneerimise režiimis, vajades sisendiks vastavalt (X; Y) ja (X; Y; Z) koordinaate.

Programmi arendusel ja katsete läbiviimisel kasutati ülilairibatehnoloogial põhinevaid ning kommertstootena saadaval olevaid siseruumide positsioneerimis-süsteeme KIO ja Sewio. Seadmete paigaldamisel lähtuti tootjapoolsetest soovitustest ning väljundandmete analüüsil arvestati tootjate tehnika konfiguratsiooni iseärasustega.

Bakalaureusetöö osana on ka toodud mõned näited, kuidas väljundgraafikuid ja -andmeid on võimalik tõlgendada. Tulemuste käsitlusel näidati, et antud tarkvara võimaldab positsioneerimissüsteeme põhjalikult hinnata ja võrrelda, lihtsustades protsessi ja vähendades aega tulemuseni jõudmiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 50 leheküljel, 8 peatükki, 22 joonist, 3 tabelit.

List of abbreviations and terms

API	Application programming interface
AVG	Average (arithmetic mean)
BLE	Bluetooth low energy
CDF	Cumulative distribution function
FCC	Federal Communications Commission
GPS	Global Positioning System
GUI	Graphical user interface
LoS	Line of sight
NFC	Near Field Communication
PoE	Power over Ethernet
RF	Radio frequency
RFID	Radio-frequency identification
RTLS	Real time location system
STDEV	Standard deviation
UWB	Ultra-wideband
VLC	Visible light communication

Table of contents

List of figures

List of tables

1 Introduction and motivation

As technology is steadily evolving all over the world, so is the need for increasingly smarter and automatic solutions. This necessity is felt in all fields, ranging from cattle breeding to industrial enterprises. New and innovative methods are constantly tested and implemented in these areas, one of which happens to be Real Time Positioning Systems (RTLS). By making use of an RTLS and acquiring information about the location of people or objects, it is possible to solve or reinforce solutions to complex problems such as safety, security, process optimization and digitalization, access control and presence detection.

There are numerous wireless technologies that can be utilized for real time positioning, such as Bluetooth Low Energy, Ultrasound, Wi-Fi, UWB and many more. Dependent on the various technologies, there are also different ranging methods i.e. ways the specific technology can be used to determine the location of the object being tracked. Furthermore, even the outputs of systems utilizing the same core technologies and ranging methods may differ due to unique position calculation algorithms.

Each technology holds potential for a certain level of positioning accuracy, however the factors stated above affect the actual outcome. In addition, there exist numerous other aspects that determine the performance of an RTLS besides accuracy, but are not talked about as often, such as precision, bias of the error and the ability of the technology to penetrate obstacles.

With such a wide array of options and factors, there exists a need to analyze and understand these aspects in detail when making RTLS-related commitments. Currently, there is no designated instrument for this and the analysis can only be done manually, which can be tedious and time-consuming. Hence, the author decided on developing a dedicated and automated tool for this purpose. The motive of the endeavor is to provide a means for the developers of RTLSs to validate performance and decide on courses of action, and provide end-users with information to fine-tune or choose between systems.

UWB-based RTLSs were chosen as the basis upon which the tool was initially built and validated, being one of the most widely offered and relevant positioning technologies in terms of reliability and accuracy. The fact that UWB, compared to most other technologies, is a relatively new presence in commercial positioning systems, adds further reason to study it [1].

The remainder of the thesis is organized as follows. Sections 2 and 3 give an overview of real time positioning systems and UWB, respectively. Section 4 explains the realization of the tool, section 5 describes the testing process, and section 6 gives the results of the analysis. Finally, the thesis closes with the conclusion.

2 Real time positioning systems

Real time positioning systems can be defined as networks of intercommunicating devices designed to provide the locations of objects or people with varying levels of accuracy, in real time. Although the term “RTLS” applies for systems that can be used both indoors and outdoors, the term is used for indoor positioning systems in the scope of this thesis.

The potential of location-based systems was first widely recognized after the successful deployment of the Global Positioning System (GPS). However, the solution cannot be used in indoor environments due to the roofs and walls of buildings interfering with the Line of Sight (LoS) between satellites and receivers, attenuating radio signals before they can reach their target. After a need to pinpoint the locations of indoor entities started to occur, the evolution of commercialized RTLSs began, which effectively solved the aforementioned problem. Since the purpose of an RTLS is roughly the same as GPS, a more easy-to-understand and informal way to define the former is to call it an “indoor GPS”.

2.1 RTLS technologies

Although RTLSs are comparable to GPS, the procedure of acquiring the position of an asset can be substantially different. GPS uses radio waves for communication, whereas RTLSs employ a diverse choice of methods besides the former, which can be divided into four major groups:

- Radio frequency (RF) based technologies
    - Wi-Fi
    - Bluetooth Low Energy (BLE)
    - Cellular networks
    - Radio-frequency identification (RFID)
    - Near Field Communication (NFC)
    - Zigbee
    - Ultra-wideband (UWB)

- Sound based technologies
    - Audible sound based
    - Ultrasound
- Optical technologies
    - Infrared
    - Visible Light Communication (VLC)
- Passive technologies
    - Magnetic field based
    - Ambient sound based
    - Ambient light based
    - Inertial navigation based (Dead Reckoning)

Out of these groups, RF-based systems are a popular choice for users and vendors alike. This has to do with the many perks and wide array of choices the category offers. For example, some RF technologies, like Wi-Fi, make use of already existing building infrastructure, enabling users to cut down on costs and setup time. Additionally, there are many options regarding location accuracy, location update frequency, scalability and complexity of integration. From the RTLS vendor perspective, most systems are cheap and easy to produce, as many RF technologies can be considered relatively mature and numerous component suppliers are available.

2.2 Workings of a UWB-based RTLS

The components of UWB-based RTLSs (Figure 1) can differ depending on the applied ranging method or the personal preference of a vendor, but certain elements are required in every variation. The cooperation of these elements makes it possible to provide the user with coordinates.

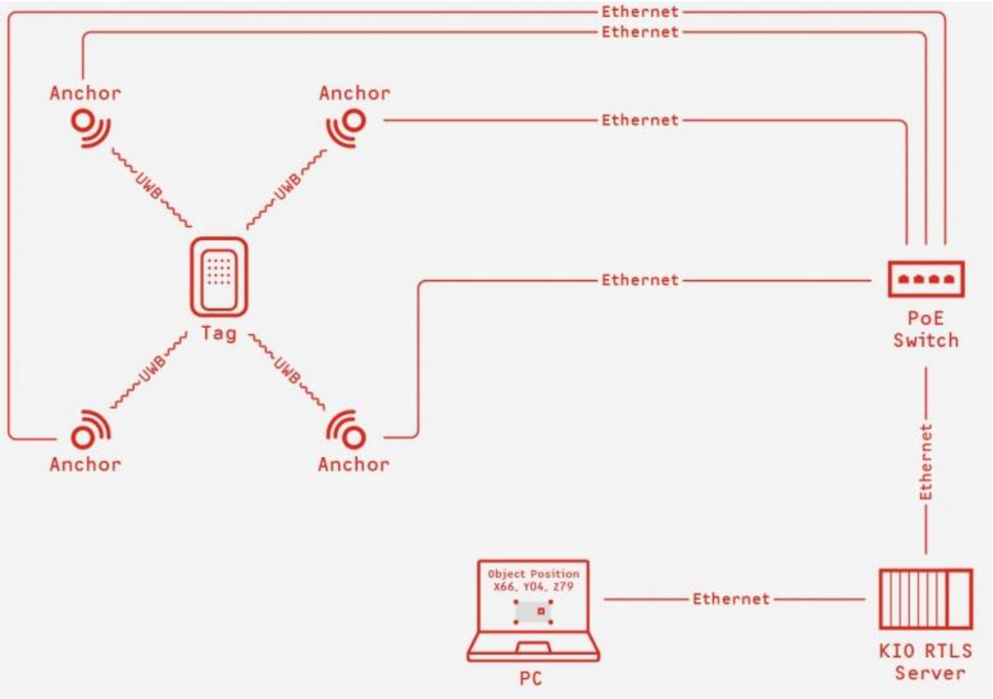


Figure 1. RTLS setup and elements [2].

2.2.1 Coordinate system

The user is required to define the area in which the tracking will take place, as a coordinate system. This can generally be accomplished by choosing a point of origin in the area, from which the axes of the three dimensions protrude. UWB-based RTLSs allow for virtually unlimited scaling of the coverage area as long as the user is able to provide power and data connection to the server and the anchors.

2.2.2 Tags

Tags are small, mobile devices that can be attached to objects of interest in order to track them in the coverage area of an RTLS. The location of the tag is what the system will eventually calculate. They can have different shapes, sizes, casings and Ingress Protection ratings depending on the nature of the tracked object and the deployment environment. Most tags have either a replaceable or rechargeable battery, but some may be required to run on external power.

Tags communicate with anchors, relaying pieces of information which can later be used for coordinate calculation. The communication can happen at various frequencies, customizable by the user, which essentially dictates how often the corresponding tags' coordinates are calculated.

2.2.3 Anchors

Anchors serve as reference points in the coordinate system defined by the user, as well as a means of communication with the server. Opposed to a tag, an anchor is a stationary device that ideally should never have to move after a position has been chosen for it. The user needs to place anchors over the area in which they wish to track something.

Anchors are usually powered through Power over Ethernet (PoE), which also provides a data connection to the server. Wi-Fi can be an alternative for data connection, but in this case, power needs to be provided separately via mains.

2.2.4 Anchor setup

To set up an anchor, after choosing a position for it, the user has to manually measure its location in the coordinate system, relative to the point of origin. The measured coordinates will be entered into the platform running on the server to be later used in tag coordinate calculation.

Some perquisites may need to be considered when choosing a location for an anchor, depending on whether the user wishes to acquire two-dimensional (X; Y) or three-dimensional (X; Y; Z) coordinates of the tag. Depending on the system and its configuration, typically the anchors need to be installed at different heights in order for the coordinate calculation algorithm to be able to calculate the tag height. For 2D mode, the height of anchors is not of critical importance.

Furthermore, the tag signal needs to be able to reach simultaneously a minimum of three anchors for the system to mathematically be able to calculate 2D coordinates of the tag and a minimum of four anchors for 3D positioning. Most UWB-based systems insist on the latter as default input for the coordinate calculation algorithm, as it is then possible to easily switch between 2D and 3D modes. Additionally, even when only using 2D mode, the fourth anchor can provide extra accuracy for the x and y coordinates of the tag due to the presence of an additional reference point in tag coordinate calculations. Since anchors have a defined working radius, it has to be taken into consideration when placing anchors – making sure the tag is able to connect to the prescribed amount of anchors in all locations where the user wants to know its coordinates.

Of course, environmental peculiarities also need to be taken into account when setting up anchors. In most cases, the rule of thumb is to try to guarantee LoS propagation conditions between anchors and tags. Often it translates into placing anchors higher above the ground.

A complete anchor setup can be called a grid. The grid is made up of cells – a cell being a fixed number of nearby anchors, covering a certain area. The number of anchors that make up a cell, i.e. cell size, is determined by the configuration of the RTLS – how many anchors the system expects the tag to communicate with simultaneously.

2.2.5 Server

The server is a hub, connecting all devices in the system, where the coordinates of tags are calculated.

Since the server is connected to all anchors, which in turn are wirelessly connected to any tags in their reach, it has access to virtually all the devices in the system. Hence, any necessary configurations of the RTLS can be made through the server, including assignment of anchor coordinates, setting tag coordinate refresh rates, choosing between 2D and 3D modes and many more.

In most cases, the server has a Graphical User Interface (GUI) for ease of use and real-time visualization of the system. The latter provides a means for the user to validate the coherence of the setup and inspect tag movements visually, without having to rely only on raw data. Depending on the vendor, there could be additional features and tools included, such as allowing the user to create event triggers based on tag movement or generating heat maps based on tag location history.

The server can be cloud-based or a local physical device, providing the user with flexible options for setup. Most commercial RTLSs also have a dedicated Application Programming Interface (API) which can be used to access the calculated tag coordinates and other data the server produces, and integrate the solution with users own existing platforms or information systems.

3 Characteristics of ultra-wideband

Despite only recently finding widespread usage and commercialization, UWB technology has existed already for over 100 years. The first known usage dates back to 1901, when Guglielmo Marconi, an engineer of Italian descent, used it to transmit Morse code sequences over the Atlantic Ocean. At that point in time, the potential for multiuser systems and the benefit of having an exceptionally large bandwidth were never considered.

After roughly 50 years, UWB was picked up by the US military, who first used it in the form of impulse radars to sense and measure distances to objects in proximity to each other. During the 1960s to 1990s, as the technology steadily advanced, usage of UWB was restricted only to the US military and the Department of Defense, who also coined the term “UWB”. It provided the military a covert method for communications and data links, as pulses being spread over a wide spectrum can be difficult to detect.

As the 2000s approached, other wireless technologies such as Bluetooth and Wi-Fi started becoming globally established. This showed the world the potential of wireless technologies and the wide array of applications they could have. In the following years, development breakthroughs in micro-processing and increasing pressure from developers of ultra-wideband systems to the Federal Communications Commission (FCC) finally caused UWB to be approved for commercial use in 2002. Since then, it slowly made its way into the field of real time positioning [3].

Nowadays, UWB is being widely used commercially and still by the military. Common practices include [4]:

Commercial:

- High speed LAN / WAN (>20 Mbps)
- Altimeter
- Presence detection
- Positioning systems

Military:

- Radar
- Covert communications
- Intrusion detection
- Precision positioning systems
- Data links

3.1 Advantageous properties for indoor positioning

Precise positioning can be considerably more difficult indoors than it is outdoors. This is because indoor locations usually have obstacles such as walls, people, goods and assets that interfere with RF waves, while outdoors there are hardly any obstructions of the aforementioned variety. These conditions have to be taken into account when deploying an RTLS to ensure a reliable solution. Thus, every company offering a positioning system has to consider the unique characteristics of the technology their system is based on. Like other technologies, UWB has distinct traits that have made it the preferred choice of many when it comes to high-accuracy positioning.

3.1.1 Multipath resistance

As there are numerous objects indoors and more often than not, LoS between anchors and tags can be hard to achieve, wireless signals can find divergent paths from one transceiver to another, reflecting off objects in the room. This effect is called “multipath propagation” and it can hinder precise positioning, as we are interested in the correct distance between transceivers. Thus, a signal path with an incorrect length measurement will translate into inaccurate coordinates.

Ultra-wideband transceivers are able to broadcast, as the name states, over a considerably wider bandwidth than most other radio wave-based telecommunication tools, specifically 500 MHz and higher [5]. In contrast, the general 2,4 GHz Wi-Fi channels offer a typical bandwidth of 22 MHz [6], along with 3G and 4G networks, which utilize only a maximum of 20 MHz spectrum [7]. The fact that UWB uses such a large bandwidth causes the temporal peaks of the signal to be much more sharp and distinct (Figure 2), opposed to narrowband signals (Figure 3) which are partially merged and hard to distinguish. Note: the timescale in the figures is identical. This enables us, by distinguishing separate signal peaks, to clearly define separate paths and choose the shortest one, which logically must have the most accurate length compared to the real world value [8].

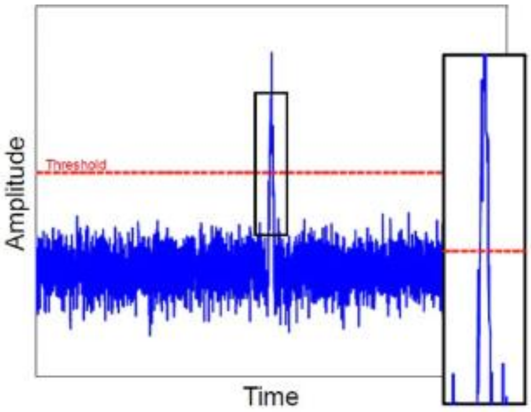


Figure 2. Wideband signal with noise [8].

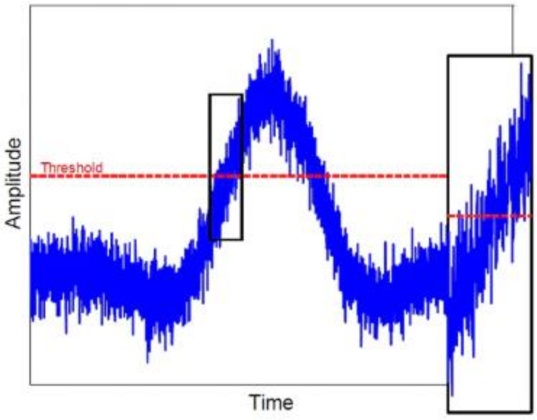


Figure 3. Narrowband signal with noise [8].

3.1.2 Accuracy

The level of coordinate accuracy UWB offers can be accounted to being resistant to the effects of multipath. Since a wide bandwidth enables us to choose the signal path with a distance closest to the real-world length between the transceivers, it allows to refine the distances fed into the coordinate calculation algorithm of an RTLS, therefore producing more accurate coordinates.

UWB-based real time positioning systems generally offer an accuracy from 2 to 30 cm, while other radio-based RTLSs offer significantly less [1].

3.1.3 Obstacle penetration

Although UWB is well-equipped against obstacles, some loss of LoS is usually inevitable. Since radio waves are slowed down or reflected entirely when trying to penetrate mass, it can cause the coordinates produced by a radio-based RTLS to become inaccurate.

This factor does not handicap UWB-based solutions as much as other radio systems. In the radio spectrum, lower frequency signals inherently possess longer wavelengths, which allows them to penetrate a large variety of materials. As previously mentioned, UWB makes use of an incredibly wide spectrum of frequencies. Therefore it is possible to reduce the effects of signal obstructions by using the lower end of the available spectrum, which most systems enforce by default [4].

3.1.4 Power efficiency

For the user, one of the most important aspects when deploying an RTLS is often the battery life tags are able to achieve. Many use cases require the tag to work sequentially for extended periods of time, therefore it is important to minimize the power consumption of the tag as much as possible.

Compared to most other positioning technologies, UWB transmitters consume very little power, since the complexity of communication is largely concentrated in the receiver. This is due to the transmitter having to operate only during pulse transmission, while the receiver needs to constantly listen for incoming signals. Certain ranging methods require the tag to act as a transceiver, somewhat crippling its battery life, but the ones that allow for the tag to require only transmitter capabilities provide a potential battery life up to several years [9].

4 Realization of the tool

Most RTLSs possess a GUI which is often used to visualize the movements of tags. Utilizing the GUI provides the user a vague idea of the accuracy of the system, as they are able to compare the location of the tracked object in the real world against the location on the screen. However, given that UWB RTLSs are predominantly meant to solve use cases that require centimeter-level accuracy, this sort of performance verification is often times not sufficient.

Validating the performance of an RTLS via coordinate analysis is a time-consuming process that involves collecting large quantities of data, running it through extensive calculations and drawing conclusions. This matter is reinforced by the fact that radio-based RLTS systems are prone to disturbances from seemingly insignificant physical changes in the setup location, meaning that several iterations of tests need to be conducted all over the tracking zone in order to validate performance in the whole area.

Taking the above into account, it was clear that a tool for analyzing the performance of an RTLS needed to be one that automates data collection and calculation processes, additionally providing the user with easy-to-read information in the form of graphs.

Hence, two programs were developed: *data_collector.py* for automating the process of data collecting and *analyzer.py* for data analysis. Accompanying *analyzer.py* are three subsidiary files, containing utility, calculation and graph plotting functions.

4.1 Python

Python is a high-level, general-purpose programming language, created by Guido van Rossum and released in 1991. It is one of the most popular programming languages in the world and is characterized by its versatile libraries and emphasis on syntax simplicity.

Python was chosen as the language for developing the programs for similar reasons. Among the many libraries, *matplotlib* stood out as an excellent means to plot various graphs, *scipy.stats* and *numpy* allowed for more complex calculations and creation of matrices, *os* enabled simple file handling and *socket* provided a means to receive RTLS data. Furthermore, since collecting and parsing data differs from system to system, requiring users to modify the data collection tool, providing the tool in an easy-to-understand programming language will prove helpful.

4.2 Data collection tool

The data collection tool (*data_collector.py*) is a simple program that provides an automated and methodological approach for collecting positioning data.

The user is expected to choose an area they wish to analyze the performance of the RTLS in, set up the system in accordance to the manufacturer’s recommendations and choose a number of test locations in the area where a tag will be placed to gather positioning data. The locations should be chosen in a manner dependent on the user’s objective: to evaluate the performance of the system in difficult locations, only these specific spots need to be tested, while acquiring information about performance in the whole area requires the user to densely cover the whole area with test locations. It is recommended to conduct several tests at each location to provide a more legitimate result when analyzing.

The program is designed to receive data from an RTLS server, parse out everything besides the raw coordinates and save them in text files in a format that allows them to be fed into the analyzer tool later on, simultaneously enabling the user to keep track of location indexes and test indexes.

4.3 Analyzer tool

The analyzer tool (*analyzer.py*) takes in the positioning data files generated by *data_collector.py*, compares them against their respective true positions, calculates various statistical measures describing the resulting errors and composes reports reflecting them, in addition to a variety of visual graphs.

Statistical measures are calculated for position errors and coordinate errors separately. Position errors are Euclidean distances between the measured position and the true position, while coordinate errors are the offsets of measured X, Y and Z coordinates against their true values. The calculations are done separately to provide easily understandable results in the form of position errors and information about each axis in the form of coordinate errors.

Analysis results are calculated and reported in three levels of scales: per the entire area (containing all locations and tests), per location and per test. This enables the user to single out problematic locations and tests, as well as get an idea of the overall performance.

4.4 Structure of the analyzer tool

The analyzer tool consists of one large function called *handle_data()* and several smaller functions, which exist in three separate files and are called by the former.

The main purpose of *handle_data()* is browsing through input files, reading in coordinates and fitting them into lists. The function is composed of separate levels of scale, containing operations executed per each system being tested, per area, per location, per test and per data row. While iterating through these levels, it calls the other functions to transform the coordinates, calculate statistical parameters and plot graphs when needed. Additionally, *analyzer.py* composes a report, into which all computed statistical parameters are printed per level of scale.

The three accompanying files are called *utility.py*, *statistical_measures.py* and *graphs.py*, named after the nature of the functions they contain. The first, *utility.py*, provides utility functions, such as converting number formats, finding unique values in a list and finding the greatest absolute value in a list. *Statistical_measures.py* contains functions for performing various calculative operations on data. The third, *graphs.py*, includes all functions that plot graphs.

The full structure of the analyzer tool can be seen in Appendix 1. The source code is public and can be accessed at https://bitbucket.org/KlausKaspar/rtls_analyzer/src.

4.5 Performance metrics

The correctness of calculated tag coordinates is arguably one the most fundamental aspects when evaluating a positioning system. Fortunately, coordinates possess properties such as accuracy and precision (Figure 4), which can be used to evaluate their compliance to the real-world situation.

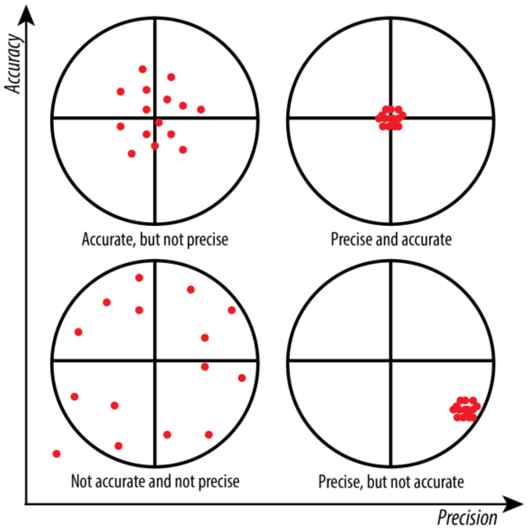


Figure 4. Illustration of accuracy and precision [10].

4.5.1 Accuracy

Accuracy has two accepted definitions. The conventional definition, which is used in mathematics, science and engineering states that accuracy only refers to a measurement's closeness to a true value, keeping the factor of precision separate. The International Organization for Standardization on the other hand, defines accuracy similarly to the aforementioned interpretation, but keeps precision as part of it, meaning that a group of measurements cannot be accurate unless it is also precise [11]. The thesis uses the conventional definition and analyses accuracy and precision separately.

4.5.2 Precision

The precision of a group of measurements indicates their closeness to each other, rather than to the true value. This means that the more similar the positions an RTLS calculates are, the more precise the system is [11]. Information about precision can be expressed via standard deviation of the measurements.

4.6 Statistical measures

Information about accuracy and precision is expressed through various statistical measures and functions, which are calculated in the tool to give the researcher a better idea of the performance of the system being analyzed.

4.6.1 Arithmetic mean

The arithmetic mean or average (AVG) is a value which represents the central tendencies of a group of numbers.

The equation of AVG:

$$A = \frac{1}{n}\sum_{i=1}^{n} a_i \tag{1}$$

A is the arithmetic mean, *n* is the number of entries in the set and a_i are the values in the set, is used in the tool to calculate average position- and coordinate errors, in order to evaluate the accuracy of results.

4.6.2 Standard deviation

Standard deviation (STDEV) describes the variability of a set of data around its average. In other words, it lets the researcher know how spread out measurements are. The more dispersed the measurements, the greater their variability. A low STDEV indicates a low dispersion of data, while a high STDEV signifies a large spread. Therefore, STDEV enables the researcher to gauge precision [12].

In the tool, STDEV is used to measure the variability of position- and coordinate errors. It is applied separately, as the STDEV of position errors gives information about the overall state of the results, while the STDEV of separate coordinates allows for in-depth analysis of each axis.

The tool applies the following STDEV equation:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{N}} \tag{3}$$

σ is the STDEV, x_i are the values in a data set, μ is the mean of the set, and N is the number of entries in the set.

4.6.3 Root-mean-square error

Root-mean-square deviation or error (RMSE) is an indicator used for detecting the presence and severity of outliers in a set of values. This makes RMSE a measure of precision. The equation for calculating RMSE is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{x} - x_i)^2}{N}}, \tag{4}$$

where $\hat{x}$ is the true value, x_i are measured values and N is the number of measured values.

The tool uses RMSE to provide the user information about the severity of outliers in coordinate errors.

4.7 Graphs

Once the aforementioned statistical measures have been obtained, they are utilized to plot several informative visual depictions of data, which characterize the results in detail and enable easier detection of anomalies.

4.7.1 Error distribution histogram

The error distribution histogram (Figure 5) is plotted for each test and location using their position errors, STDEVs and AVGs. It summarizes the results of each test by visualizing the distribution of errors. A researcher can use this graph to check whether the accuracy and precision of the system stay within satisfactory boundaries in specific tests or locations or if they cross the thresholds set in the system specifications.

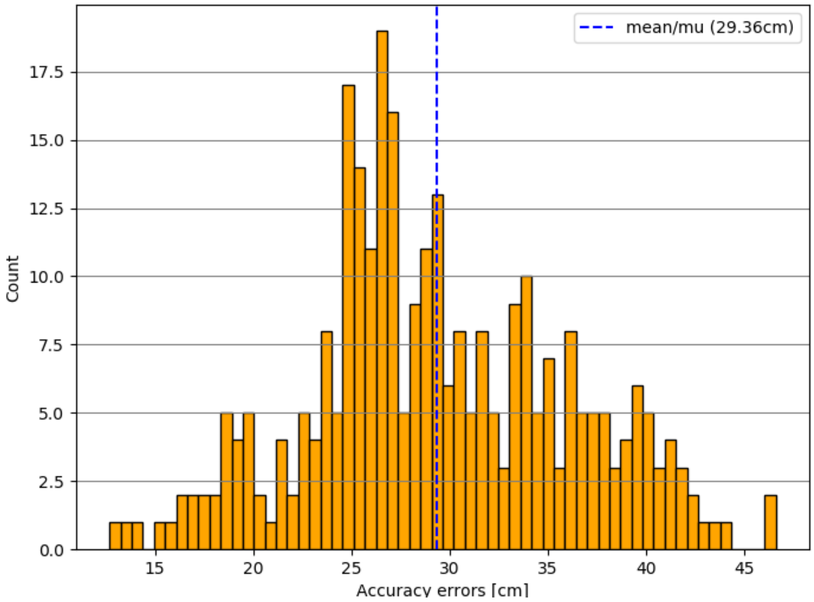


Figure 5. Error distribution histogram.

4.7.2 Scatter graph

Scatter graphs (Figure 6) are plotted for each test and location, similarly to the Gaussian distribution histogram. The inputs for this function are all the measured X and Y coordinates per test, which are used to plot points on a grid. The coordinates of the true location are also plotted. A color map is used to emphasize overlapping points. The graph effectively illustrates the distribution of measured points against the true location in an easily comprehensible way. Additionally, it can be used to deduce the bias of errors, which can in turn be used to make corrections in anchor placement. Currently, this graph is applicable only to 2D tests.

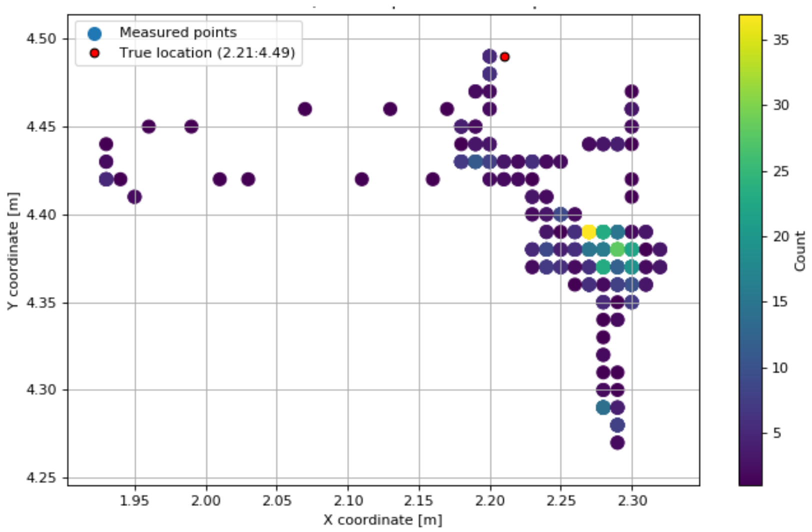


Figure 6. Scatter graph.

4.7.3 Cumulative Distribution Function

The Cumulative Distribution Function (CDF) (Figure 7) is a graph that is being used to characterize the entire area. It is plotted as a single graph comparing each RTLS's position error probabilities and as separate graphs per each system comparing the probabilities of X, Y and Z coordinate errors.

The CDF graph can be used to compare the overall accuracy of input systems, as well as see the range and likelihood of errors.

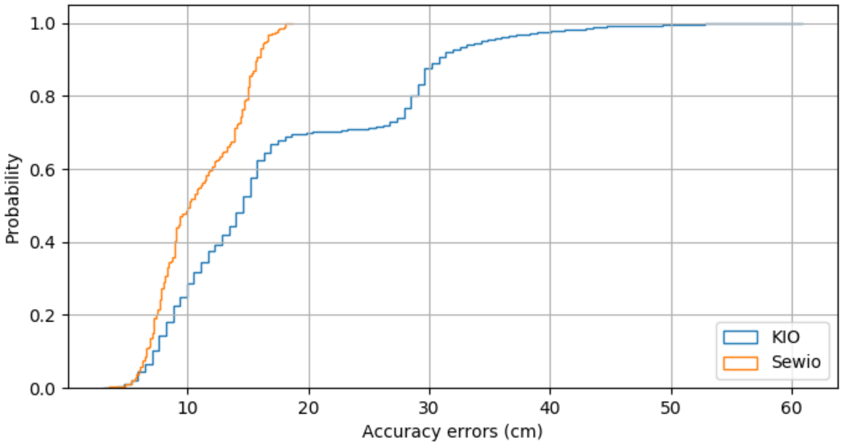


Figure 7. Cumulative distribution histogram.

4.7.4 3D histogram of X and Y errors

A 3D histogram of X and Y errors (Figure 8) is plotted for each RTLS that is being analyzed, for the entire area. Inputs are X and Y coordinate errors. Graph shows the distribution and quantities of these errors. By investigating which quarter most errors have concentrated into, the user can discover the general bias of the error and make corresponding corrections in the anchor setup to reduce the error. It provides information similar to scatter graphs, but instead of focusing on single tests or locations, it summarizes the errors across all tests.

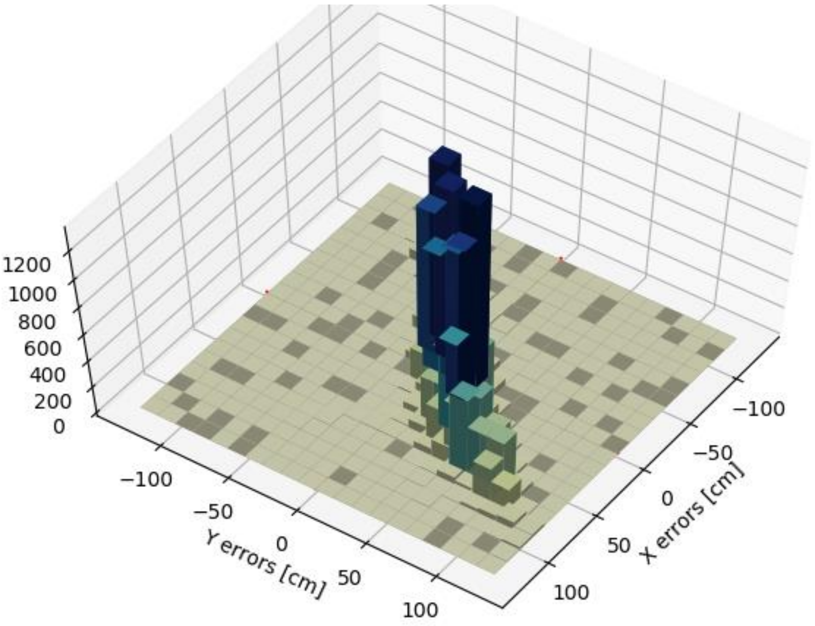


Figure 8. 3D histogram of X and Y errors.

4.7.5 Comparative histograms

The Comparative histogram graph (Figure 9) is useful for comparing the statistical parameters of RTLSs at each location. The tool plots two graphs: one for comparing AVGs and one for STDEVs.

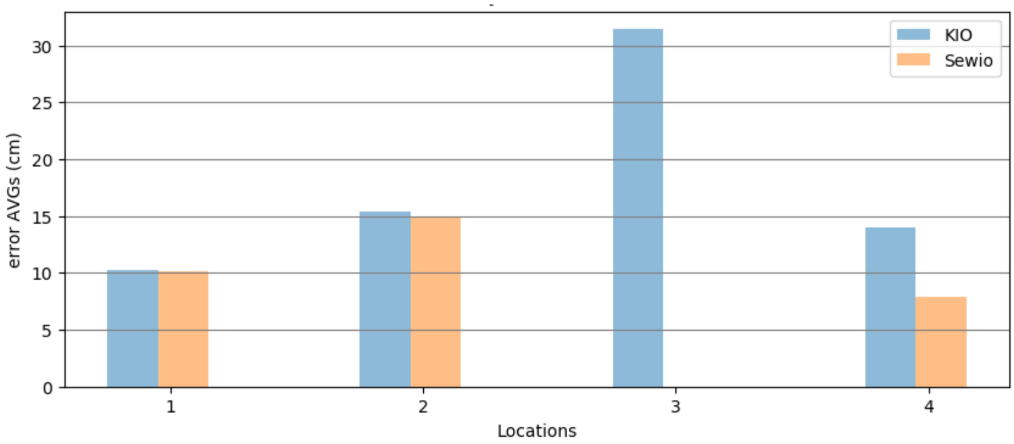


Figure 9. Comparative histogram of AVG errors per location.

5 Test setup

In order to evaluate the functionality of the software, several positioning experiments were conducted which allowed to apply the data collecting tool and simultaneously gather the necessary input data for *analyzer.py*. Two UWB-based RTLSs utilizing the same core technology but differing in some design choices were deployed and analyzed during the process.

5.1 Setting

The tests took place at the office of Eliko Tehnoloogia Arenduskeskus, in a 15 by 5 m meeting room with a table and chairs in the middle (Figure 10). Conditions for using UWB RTLSs were decent, as clear LOS between each anchor and tag at test locations was guaranteed, with a few exceptions (locations 6 and 7 on Figure 15).

Figure 10. Testing site at Eliko’s office.

5.1.1 KIO

KIO RTLS is a UWB-based location system developed in Estonia by Eliko Tehnoloogia Arenduskeskus OÜ.

KIO devices utilize UWB chips produced by Decawave to communicate and measure distances between each other using the time of flight ranging scheme. These distances are converted into coordinates for all axes.

According to the technical specifications sheet, the coordinates produced by KIO RTLS should stay within 30 cm from the tag's true position. Positioning precision is stated to be less than 7 cm in 70% of the cases, which roughly corresponds to a STDEV of 7 cm [2].

5.1.2 Sewio

Sewio is a UWB-based RTLS system which originates from the Czech Republic.

Sewio also uses the Decawave UWB chip. However, to calculate distances, they apply a different ranging scheme called time difference of arrival. Another noteworthy difference between the two systems is that Sewio uses built-in barometers in its devices to measure the z coordinate, while KIO estimates it solely by distance measurements.

Sewio claims up to 30 cm deviation from the tags true location. No information was found on precision [13].

5.2 Anchor layouts

As in many use cases the tag's height does not change, most RTLSs have separate modes for 2D (X; Y) and 3D (X; Y; Z) tracking, which also require a different anchor layout. To produce more accurate x and y coordinates in 2D mode, the z coordinate of the tag is fixed within the system to reduce its effect on the coordinate calculation algorithm. Both modes and layouts were tried and analyzed during the tests, meaning that a total of 4 separate tests were conducted.

Both the 2D and 3D setup utilized five anchors, covering the whole room. The anchor placement was identical for both systems regarding x and y coordinates. Also, both systems recommended anchors, relative to each other, to be at roughly the same height for 2D positioning. 3D positioning required KIO anchors to be placed at different heights, but Sewio anchors could be left at the same locations as during the 2D setup, with the condition of calibrating the barometer for each device.

Each setup was made following the vendor recommendations, to the extent that the shape of the room could allow.

5.3 Tag locations

A total of twelve locations in the room were chosen as tag test locations, at ten of which the tag was fixed to a tripod in an upright position. The remaining two spots were on the table, the tags lying flat on their backs (Figure 11).

Figure 11. Sewio tag lying on its back.

To reduce the impact of human error and test to repeatability, data needed to be gathered multiple times per location, changing the tags location after each collection event. A method was required to place the tag in the exact same position when returning to each location. For this purpose, a tripod with plastic vices screwed on top was used to keep the tag still (Figure 12).

Figure 12. Tripod with a tag attached on top.

To be able to place the tripod onto the chosen locations (Figure 13) in the exact same manner, the locations were marked on the ground with tape and the outlines of the tripod's bottom were drawn on it (Figure 14). The tripod was always rotated in a manner that the tag was facing the x axis wall. To acquire tag test location and the true coordinates of the anchor, a laser measuring tool, Leica Disto D110, was used.

Figure 13. Tripod standing on a chosen location.

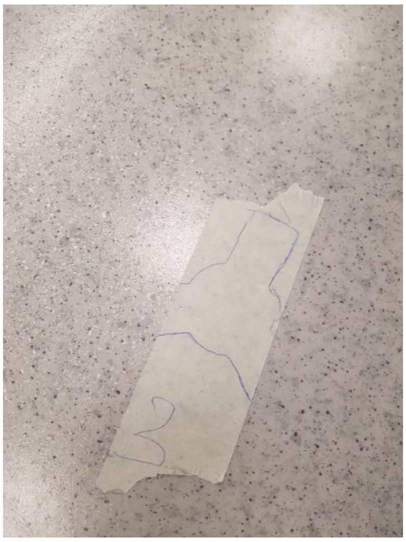


Figure 14. Marking on the ground.

5.4 Data acquisition

Prior to data gathering, tags of both systems were set to update their location at 100 ms intervals. As previously mentioned, a total of four distinct tests were conducted. During each of them, the respective anchor layout was made based on the system recommendations and the requirements of 2D/3D mode.

A tag of the system under test was carefully placed into the first chosen location. Then, the data gathering tool *data_collector.py* was used to connect to the server of the system and listen, record and transform the calculated coordinates into a format that the analyzer tool could accept, until a necessary amount of data had been collected. Then, the tag was moved to the second testing location, and the process was repeated again (Figure 15). After each location was visited in this manner, the process would start again from the first position, until the wanted number of iterations had been completed, which signified the end of the first test. Following this, the anchor positions would be updated to match the requirements of the next test. The previous operations were repeated, until all distinct tests were completed.

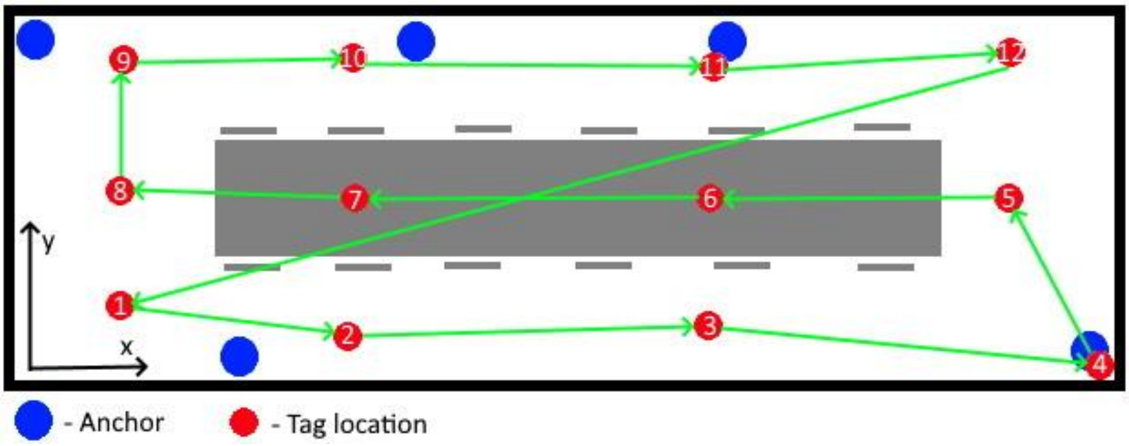


Figure 15. Data collecting sequence.

6 Analysis results

Once all necessary data had been collected, the *analyzer.py* tool could be used to analyze it. The following subsections discuss the nature of the findings that the tool generated.

The purpose of this section is to bring an example on how to interpret the results of the analysis and draw conclusions based on them. For the sake of brevity, only the analysis of the 2D tests is examined in-depth, while the most noteworthy discoveries of the 3D analysis are included in a small subsection. Additionally, only the graphs which convey the most information have been used in the example.

The results were examined through the perspective of a user possessing above-average knowledge of RTLSs and positioning, comparing two systems. This viewpoint was chosen, since it enables to demonstrate how to interpret the results analytically, as a researcher would to gauge the potential of their own system, as well as retain the element of comparison to keep in mind the interest of an end-user.

6.1 2D results

In the analysis and comparison of the 2D setups, the Z-coordinate was excluded from statistics calculations, since it was given a fixed value when testing, indicating that the user was only interested in X and Y-coordinates.

6.1.1 Accuracy

Looking at the comparative histogram of average position errors at all tag test locations (Figure 16), it is clear that the positions calculated by KIO are far closer to the actual tag positions at all test locations, with the exception of location 2.

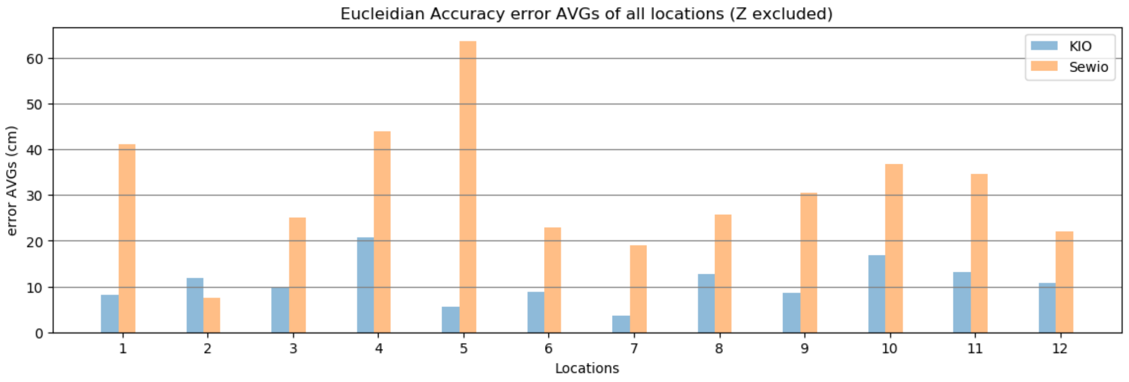


Figure 16. AVG accuracy errors at all test locations.

The average errors of KIO RTLS stay within boundaries promised in its technical specifications at all times, while Sewio, claiming to have the same accuracy, crosses the limit at half of the locations. Comparing the text-based report generated by the tool (Table 1) we can see that the overall positioning error average for the whole area was 10.81 cm for KIO and 31.06 cm for Sewio.

Table 1. Statistical parameters of the area.

	AVG error (cm)	**STDEV (cm)**
KIO Area Accuracy	10.89	4.83
Sewio Area Accuracy	31.06	17.54

It is evident that KIO RTLS has better accuracy in 2D mode but precision also needs to be analyzed before a proper assessment of the performances of the systems can be acquired.

6.1.2 2D Precision

The comparative histogram of test location STDEVs (Figure 17) indicates KIO's superiority again. While Sewio's STDEV fluctuates across all measured locations, KIO stays at a more stable level.

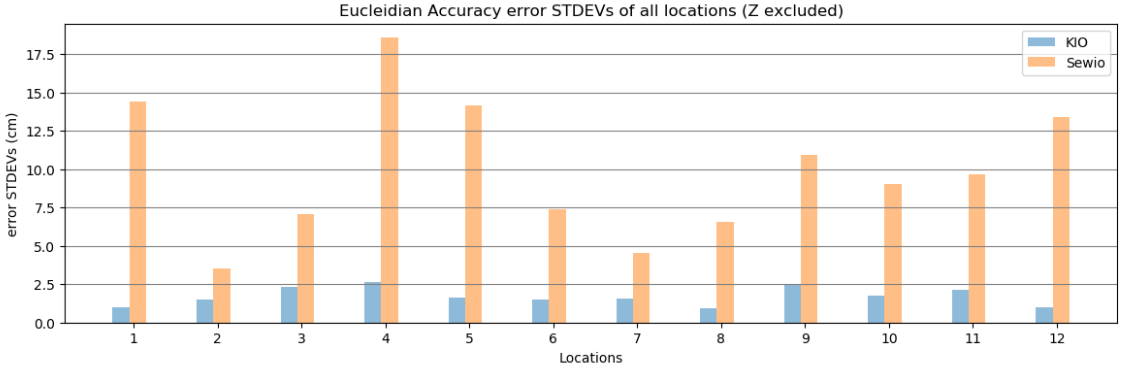


Figure 17. STDEVs at all test locations.

The text-based report informs us that the STDEV across all locations was 4.83 cm for KIO and 17.54 cm for Sewio. KIO stays within its stated precision limit of 7 cm.

Taking location 4 as an example, where KIO can be seen on Figure 18 and Sewio on Figure 19, comparing the scales of the respective scatter graphs confirms the difference in precision. We also get some insight on the reason for this difference, as there is a significant difference in the number of overlapping points between the two systems. In addition, the location where the most overlapping points are (yellow end of color map), is a single, well-defined area for KIO, but in Sewio's case, there are multiple far-apart high-concentration areas, decreasing precision even further.

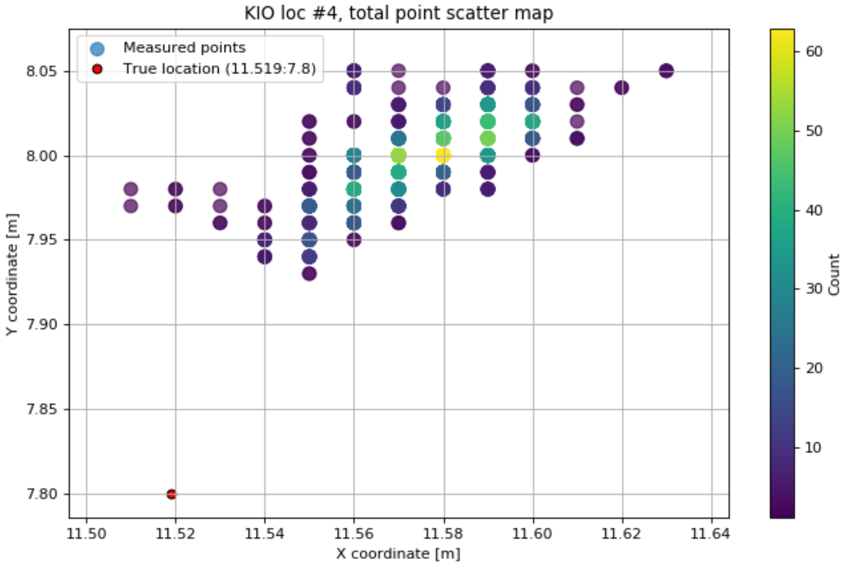


Figure 18. KIO scatter graph at location 4.

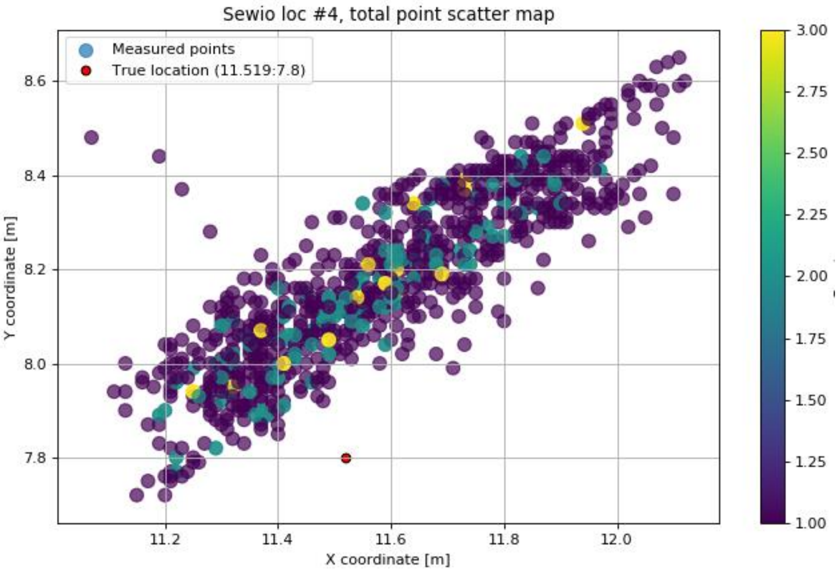


Figure 19. Sewio scatter graph at location 4.

6.1.3 Accuracy axis-wise

Delving into possible reasons what could have caused the fluctuations in accuracy and precision, the next thing to check would be the offsets of coordinates separately.

The differences in the statistical parameters of the coordinates (Table 2) confirm previous findings, the parameters of KIO's coordinates surpassing Sewio's.

Table 2. Statistical parameters of coordinates.

	AVG error (cm)	**STDEV (cm)**
KIO X	6.29	7.24
KIO Y	8.02	9.42
Sewio X	18.75	21.99
Sewio Y	22.74	27.76

The same notion is seen when looking at the coordinate error CDF graph (Figure 20). The lines representing x and y coordinates are plotted on a significantly smaller scale for KIO (roughly -17 to 24 cm) compared to Sewio (roughly -60 to 83 cm).

As the coordinate error CDFs are plotted without using the absolute values of coordinate errors, they also contain information about the bias of the errors. Although Sewio's coordinate errors are greater, the errors are divided more evenly compared to KIO, as both lines of the former cross the zero-tick of the errors axis almost exactly at 50%

probability, meaning that roughly the same amount of errors stay in both directions. In KIO's case, this division is marginally imperfect, possibly indicating a slightly poor calibration of the devices or imprecision in anchor placement or measured coordinates.

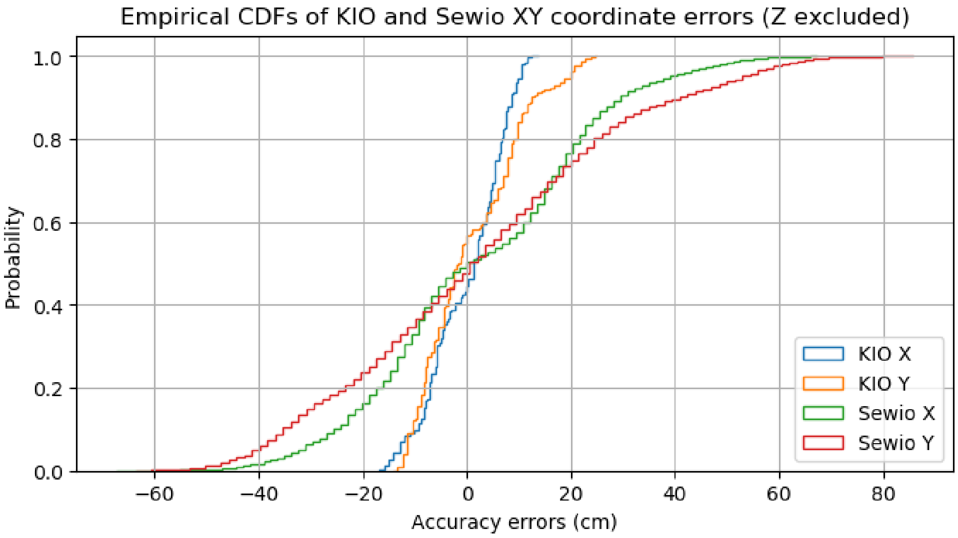


Figure 20. CDF of X and Y errors.

Knowing the parameters of each axis, the dominating direction of the errors and the ranging schemes of both systems enables the user to make educated decisions on how to change the anchor layout to hopefully improve the results and repeat the process until the desired level of accuracy has been met.

For example, in this analysis, an important fact to note is that for both systems, the x coordinates has superior parameters compared to y. Since anchors were placed at the exact same locations, it indicates that the current anchor layout may be unfavorable for the Y-coordinate.

6.1.4 2D Summary

Inspecting the comparative CDF of position errors (Figure 21) summarizes the results of the accuracy analysis by effectively illustrating the error ranges and probabilities of both systems on the same graph. During the given test, positioning errors stayed under roughly 27 cm in 100% of the cases for KIO. Compared to Sewio, the error happened to be under 27 cm in slightly under 50% and 100% of the cases for staying under 102 cm.

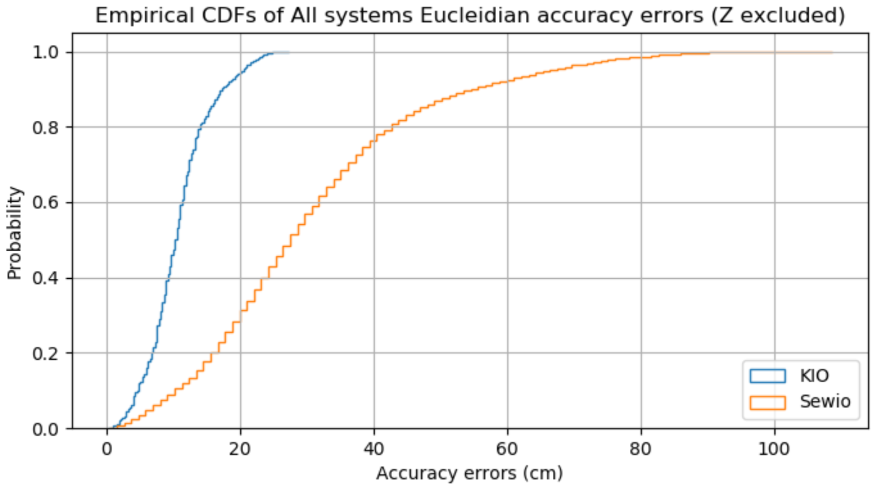


Figure 21. CDF of position errors.

Overall, the quality of KIO’s performance was considerably higher, judging by the accuracy and precision of both position and coordinate errors. The KIO system even operated, in both aspects, better than what was specified in its technical sheets, in all three levels of scale. In Sewio’s case, the average accuracy of the entire area was essentially in accordance with what was promised. However, the existence of severe outliers as seen in some test locations (1, 4, and 5) on Figure 16, where the average accuracy was 10 to 30 cm above the threshold, shows that the system performs inconsistently in some points of the room.

In conclusion, the results of the 2D data analysis stipulate that for the given setup, KIO would be the preferred system to use in terms of coordinate performance metrics.

6.2 Summary of 3D findings

The results of the 3D tests analysis were quite different from 2D, resulting mainly from the inclusion of the Z coordinate, but also from a slightly different setup for KIO and the usage of the barometer built into Sewio devices.

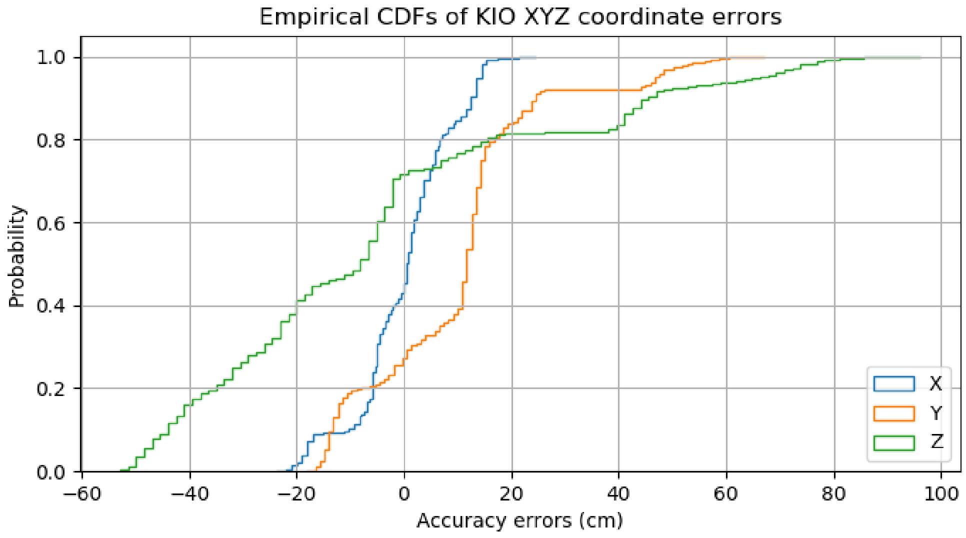
For both systems, major differences between the statistical parameters of X and Y-coordinates compared to Z can be seen in the report presented in Table 3. While the Z-coordinate appears to be the weak point for KIO, being the prime contributor to its inaccuracy and the X, Y-coordinates being more accurate, the opposite applies for Sewio.

The Z-coordinate of the latter is surprisingly accurate, more so than any other coordinate either system.

Table 3. Statistical parameters of the area and coordinates.

	AVG error (cm)	**STDEV (cm)**
KIO Area Accuracy	35.3	18.85
KIO X	6.94	8.86
KIO Y	15.49	16.58
KIO Z	27.13	33.51
Sewio Area Accuracy	45.66	32.38
Sewio X	30.01	36.16
Sewio Y	30.91	39.83
Sewio Z	5.56	5.34

The CDF of coordinate errors (Figure 22) also reflects this. Note the difference in the scales of the error axis. Sewio's Z-coordinate line is exceptionally steeper and covers a narrower area than X and Y lines, the situation being in reverse for KIO. This leads to a realization that using a barometer is extremely effective for measuring height between the anchors and tags.



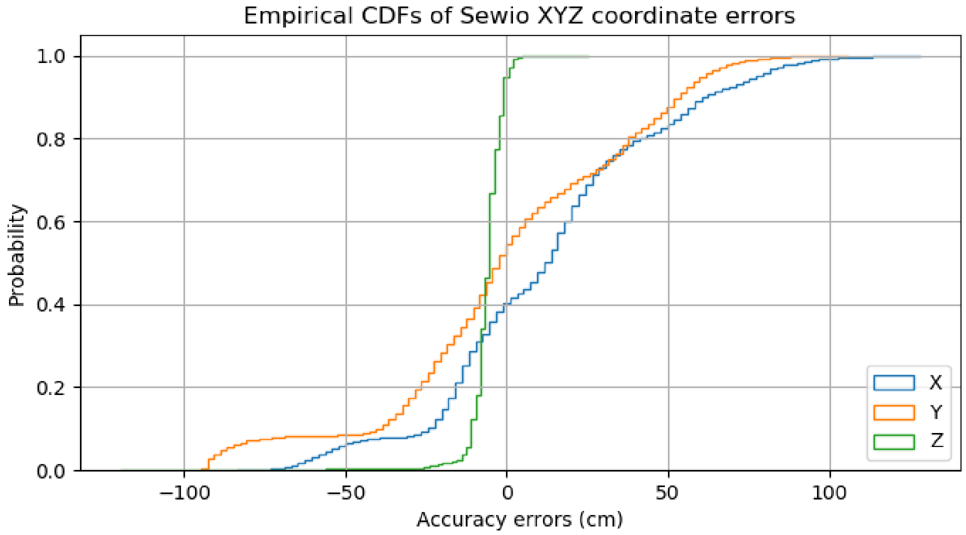


Figure 22. CDFs of KIO and Sewio coordinate errors

6.3 Benefits for potential users

From the perspective of an RTLS developer, the results provide a detailed overview of the current capabilities of the systems. The pros and cons of both systems are brought out, giving the researcher information on what needs to be improved and considered for future development of their devices. In addition to the statistical parameters in specific locations, the user also learned of the average performance of the systems in the whole given area and environment. Since testing is a daily occurrence in device and software development, automating the analysis saves up a significant amount of time, which can be used to focus on other tasks.

For an end-user of an RTLS, struggling to make a setup work while having little knowledge of the logic behind positioning algorithms and ranging schemes, the tool helps to gain an insight on the situation. The reports and graphs that the program outputs present information that a technician should be able to act on, until the best possible setup is achieved. If the user is trying to pick a system for their specific use case, the easily interpretable comparisons allow to make a justified choice.

7 Future work

Although the tool encompasses a wide variety of elements, there is still room for improvement and additional features that could be implemented in the future, to give the user more options and enable a more thorough analysis.

Firstly, in order to provide a better overview of the situation at each test and location, scatter graphs could be upgraded to also convey information about height. This would mean transforming the graphs into 3D models. They would also be needed to be made interactive, since a still image of a 3D model could turn out to be hard to interpret.

Secondly, the tool allows the user only to analyze positioning information that is collected from tags with fixed locations. However, the analysis of positioning data that has been collected from a mobile tag would also be beneficial, since most use cases involve tracking mobile objects/subjects and coordinates can behave differently compared to standing still. Hence, a possible idea for future development is an automated method of analyzing movement-based coordinates. Including the factor of mobility would allow to further assess the latency of the RTLS.

Finally, some improvements in overall ease of use would prove helpful. The present version of the tool requires a moderate to extensive level of understanding of the logic behind an RTLS to be able to make meaningful conclusions based on the results of the analysis. To make it more accessible for the end-users of RTLSs, the degree of this prerequisite could be lowered by presenting the results in ways that are more straightforward and easier to comprehend. The tool could have an option for, instead of showing numbers and graphs, directly instructing the user what he needs to do in order to improve the results. Additionally, developing a dedicated UI would make data collecting less tedious, as it is currently done via command line.

8 Conclusion

The aim of the thesis was to implement, validate and characterize a software tool which automates and speeds up the analysis of RTLS positioning data. The program was created by the author, using the programming language *Python* and utilizing several built-in mathematics and graphics libraries.

Out of the technologies utilized in real time positioning systems, UWB was chosen upon which the tool was built and validated through, due to being a popular choice in terms of positioning accuracy. Insight on the concepts of an RTLS and UWB are provided in the thesis.

To validate the tool, several tests were conducted, using two UWB-based positioning systems: KIO and Sewio. The author employed an iterative method for collecting positioning data from 12 separate locations in the testing area, which was then used as input for the software.

Reflecting on the analysis results, the tool was successfully able to grant the user insight on several impactful aspects of positioning via statistical parameters and illustrative graphs. Thus, considering the goals established in the introduction, it can be concluded that the developed tool fulfills its purpose of providing individuals working with RTLSs an automated method for obtaining detailed information on positioning performance and helping them make informed decisions regarding them.

Although the program was developed and demonstrated with UWB-based RTLSs, the usage of the tool also expands to systems utilizing other technology. The only prerequisite is the system having an output in the form of X, Y and Z coordinates and the existence of a measurable true location that the former can be compared against

Acknowledgements

I would like to express my sincere gratitude to the following:

- My supervisors, Taavi Laadung and Gert Kanter, for guiding me throughout the thesis. They were available each time I needed a helping hand and I could not have completed the thesis without them.
- My friends and colleagues, Alan Kalda and Enar Reilent, for boosting my motivation and putting up with my constant barrage of questions.
- Eliko Tehnoloogia Arenduskeskus, for providing me access to their rooms, positioning systems and coffee.
- My girlfriend and cat, for putting up with my long hours of working at night.
- My parents – for everything and for providing me a college education.

Thank You,

Klaus Kaspar Kasak

Bibliography

[1] B. Krulwich, "Sub-Meter Accuracy Indoor Location Positioning Technologies," Grizzly Analytics, LLC, 2018.

[2] E. T. A. OÜ, "KIO RTLS Regular Cell," 2019. [Online]. Available: https://www.eliko.ee/wp-content/uploads/2019/10/KIO-RTLS-Regular-Cell-Technical-Specifications-2019-10-24.pdf. [Accessed 2020].

[3] F. Nekoogar, Ultra-Wideband Communications: Fundamentals and Applications, Prentice Hall, 2005.

[4] S. Jogi and M. Choudhary, Ultra Wideband Demystified: Technologies, Applications, and System Design Considerations, River Publishers, 2009.

[5] I. Oppermann, M. Hämäläinen and J. Iinatti, UWB: Theory and Applications, Wiley, 2004.

[6] ElectronicsNotes, "Wi-Fi Channels, Frequencies, Bands & Bandwidths," ElectornicsNotes, [Online]. Available: https://www.electronics-notes.com/articles/connectivity/wifi-ieee-802-11/channels-frequencies-bands-bandwidth.php. [Accessed 2020].

[7] A. Elnashar, M. A. El-saidny and M. R. Sherif, Design, Deployment and Performance of 4G-LTE Networks: A Practical Approach, Wiley, 2014.

[8] C. Connell, "What’s The Difference Between Measuring Location By UWB, Wi-Fi, and Bluetooth?," ElectronicDesign, 2015. [Online]. Available: https://www.electronicdesign.com/technologies/communications/article/21800581/whats-the-difference-between-measuring-location-by-uwb-wifi-and-bluetooth. [Accessed 2020].

[9] B. Otis and J. Rabaey, Ultra-Low Power Wireless Technologies for Sensor Networks, Springer, 2007.

[10] "Precision Vs. Accuracy," St. Olaf College, 2020. [Online]. Available: https://wp.stolaf.edu/it/gis-precision-accuracy/. [Accessed 2020].

[11] International Bureau of Weights and Measures, "International vocabulary of metrology - Basic and general concepts and associated terms (VIM), 3rd edition," 2012.

[12] M. P. Barde and P. J. Barde, "What to use to express the variability of data: Standard deviation or standard error of mean?," NCBI, 2012.

[13] Sewio, "Indoor Location Tracking and Positioning," [Online]. Available: https://www.sewio.net/indoor-location-tracking-and-positioning/. [Accessed 2020].

Appendix 1 – Analyzer tool structure