POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Elettronica

Tesi di Laurea Magistrale

UWB Tracking System for Patient Monitoring in Home Environment

Relatori:
Prof. Danilo Demarchi
Dr. Paolo Motto Ros
Dr. Stefano Sapienza

Candidato:
Maurizio Capra

Aprile 2018

Acknowledgments

Table of contents

List of figures

Summary

The purpose of this thesis work is the development and testing of a position algorithm, based on Time of Flight (TOF) of UWB (Ultra WideBand) radio signals, for autonomous indoor navigation. The system is composed by two types of transceivers: anchors and tags. The former are fixed in known positions and work as a reference while the latter are the units to track . Analysing the TOF of the UWB pulses among transceivers, the algorithm is able to derive the distances (called ranges) of the tags respect to all the anchors and consequently, applying a trilateration technique, the position of the tags in 2D. The tracking code has been translated and embedded in a java library and implemented on a Home Robot monitoring system. The final purpose of the platform is providing a rapid intervention and assistance to elder people in case of health issues. The system will work in synergy with wearable sensors which continuously record personalized health parameters depending on the subjects pathology.
Since in this context false positives (i.e. health status reported as good when it is not) are not acceptable, the systems sensitivity is extremely high. However this inevitably leads to lot of false negative(health status reported as critic when it is not) that translates in unnecessary stress for the subject and the clinicians. The robot works as a filter for the alarms. The basic idea is to have a tag placed on the patient and another on the robot, with several anchors spread around the flat (one per room).
In presence of potentially harmful symptoms , the robot activates and, guided by the UWB tracking algorithm, travels to the patient establishing in the meanwhile a web call with an operator that, thanks to a camera and a monitor, will verify the subjects conditions. The content of this thesis describes step by step all the design process of the UWB tracking platform together with the selection of the hardware.

The first chapter introduces some basics concept about TOF and the trilateration process. Furthermore UWB technology and the reason why it is suitable for indoor application are explained in detail.

In the second chapter the hardware of the system and the robot are introduced in detail. The UWB tracking platform is the Decawave Real Time Localization System (RTLS), TREK 1000 based on the DW1000 module that creates the radio pulses. This kit comes with a pre embedded tracking algorithm configured for 3 anchors and 1 tag(since each kit contains 4 devices). In order to improve the performance exploiting redundant information others two TREK 1000 has been used, having a total of 8 anchors and 2 tags. The main hardware component of the Home Robot is represented by the iRobot Create Development kit, this robotic platform is based on robotic vacuum cleaners produced by iRobot. This robot has two wheels driven by motors and a third one for support with encoders that measure the distance travelled and the orientation. The control software is in a computer that communicates with the iRobot through a 7 pin MiniDin connector. The platform is also equipped with a webcam and a monitor that allow it to establish a web connection with an operator. Furthermore the operator can manually move the iRobot thanks to a remote control.

The study of the TREK 1000 system and of the errors that affect it are presented in the third chapter. Here the tracking speed and the sample frequency are investigated with the antenna performance.

The fourth chapter includes a detailed description of the position algorithm and all the steps performed during its development. First of all is analysed the new trilateration algorithm implemented instead of the Decawaves one that allows to rely on the ranges of only three anchors. The new algorithm considers n anchors(with n equal or greater than 3), exploiting the redundant information to increase the resolution of the tracking. In the second part of the fourth chapter are explained all the corrective functions and the filters applied. The corrective functions allow to mitigate the error due to non linear effects that could affect the UWB system, while the filters are exploited to increase the robustness discarding range measurement that are not considered consistent with respect to the current position of the Robot.

The fifth chapter is dedicated to the static and dynamic measurements and test performed to verify the performance of the algorithm. Here measurements obtained

placing the tag on test points forming a grid in the 2D plan are presented in graphic and numeric way. It is shown how the error changes with different boundary conditions and according to the corrective functions and filters applied.

The sixth chapter describes how the algorithm is embedded in the Robot application, and it introduces the calibration process and the role played by the Simultaneous Localization And Mapping (SLAM) procedure to obtain the floor plan of the house. The calibration process is fundamental in case the anchors position is unknow, so prior to perform the trilateration, they must be found. This operation performed with the help of the SLAM can lead to obtain anchor positions with high accuracy respect to the flat dimensions.

Finally the last chapter presents the conclusions, with the discussion of the results obtained, and possible future improvements for the Home Robot application.

Chapter 1

Introduction

World's population is aging and the number of people over the age of 65 is increasing. It's important for elder people to feel free and independent in their home environment. On the other hand, with aging several health issues and deseases can occur, then is also importat to provide them with a reliable mean to support and prevent tragic events during daily activities.

The idea of the Home Robot monitoring system is to track the health status of the elder people that want to live in an independent way in their habitations. The Home Robot is a tool used by the caregivers to track the patient's status thanks to a sensor and a webcam placed on the robot. The sensor is placed on the patient in order to supervise his status according to his patology. The robot remains in a stand-by condition connected to its charge base, and in case of helth issues communicated by the sensor, it navigates in autonomous way throught the home environment establishing a connection with a caregiver using the webcam.
The caregiver has then the duty to decide if it is the case to call the 911, in fact since the system must be very sensitive, some false negative alarms (health status reported as critic when it is not) could occur.

The aim of the thesis is to support the Home Robot with a reliable indoor navigation system and a position application able to locate the bot and the patient in any home environment.
The system is composed by two kind of device: anchor and tag. The anchors are devices positioned in fixed points while tags are devices free to move in the area delimited by the anchors. Positioning one tag on the bot and one on the patient is possible to track their positions and to direct the former toward the latter.

The localization problem is composed by two phases: ranging and localization. In the next paragraph those two steps will be introduced.

1.1 Technology

There is increasing demand of accurate localization system for many applications such as robot control, goods and people tracking, indoor navigation and support during critic events like earthquake or avalanche for people rescue. Nowadays current technology offers a lot of possible implementation for positioning system: GPS, infrared, ultrasound and RF (radiofrequency) techniques.

Global Positioning System (GPS) is a satellites based technology that provides position and time information. This kind of systems are very accurate and are used in a lot of devices such as mobile phone, tablet and on-board car console. However GPS cannot provide position with high resolution in dense urban and indoor environment casue signal coming from satelites is reflected by the buildings and walls and in many cases is too weak to penetrate them.

Infrared and ultrasound techniques are among the most used in indoor environment thanks to their semplicity and low cost. However in order to work properly they require the Lign Of Sight(LOS) condition between anchor nodes and tags, so this means that in scenario populated by many obstacles or walls these technologies have a limited usage. This is due to the fact that light and sound are not able to penatrate objects and walls, on the countrary RF signals are able to easily propagate in home environment.

The four most used ranging techniques based on RF are: Angle of Arrival (AoA), Received SIgnal Strength (RSS), Time of Arrival (ToA) also called Time Of Flight (TOF) and Time Difference Of Arrival (TDOA).
In AoA method, anchors exploit the angle of arrival of the RF signal coming from the tag to determine the position of the latter. In general this approach requires directional antennas or antenna arrays, and many times it is used together with

ToA in order to obtain a better accuracy. RSS methods are based on the fact that an electromagnetic signal decays with the distance between the transmitter and the receiver, but they are very susceptible to multipath and No Lign Of Sight (NLOS), this means that they are not very reliable.
Time based technique (ToA) requires clock synchronization among nodes, generally achieved by means of wired connections. When synchronization is not possible it is called asynchronous ToA, also known as two way ranging.
When synchronization is achieved only between anchors and no information are available about tag's clock is possible to use TDOA.

1.2 ToF Ranging

In Time of Flight ranging the distance between two devices (anchor and tag) is obtained measuring the propagation time of a signal that is travelling at a known speed. The principle is very simple, knowing the travelled time τ and the wave speed c, the distance d between to object is:

$$d = \tau * c \tag{1.1}$$

One of the simplest way to exploit this principle is using a source of light, tipically infrared, and a sensor capable to identify the presence of this kind of light. Emitting a IR ray in a certain direction and recording the time of emission *t1*, this will travel till the object covered with reflectors and, once reflected, backward to the sensor that can assert the time of reception *t2*. Tof will be calculated as:

$$\tau = \frac{t2 - t1}{2} \tag{1.2}$$

In fact the difference between *t2* and *t1* represent the round trip time, i.e. the time spent by the signal travelling from the emitter to the reflector and from the object to the light sensor (that is tipically placed near the emitter). This is twice the ToF.

Since IR light cannot travel through walls and objects (NLOS), RF signals are preferred in harsh indoor environment full of possible obstacles such as can be an home environment. The main difference using RF is that the tag is not represented

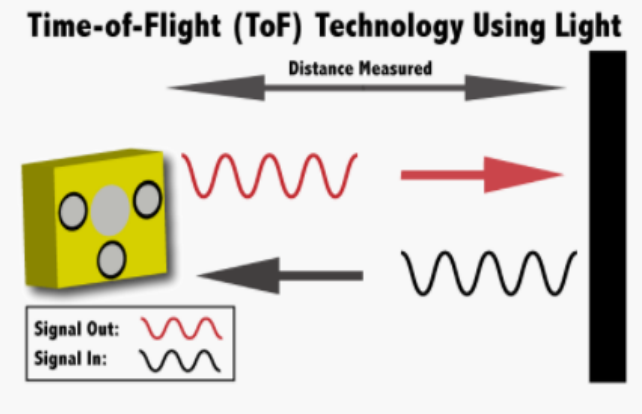


Figure 1.1: Time of Flight using IR light [20].

by a reflector, but is another device able to receive and send a message from/to the anchor.
The speed of an EM signals is comparable to the speed of light (300000 km/s), this means that the measure of τ must be very accurate, in fact even a small uncertainty multiplied by the speed of light could lead to big error in the range estimation.
In order to reach the maximum possible accuracy, ToA requires to have all the anchors synchronized among them. In this case, a single signal(or better a message) is sufficient for ranging [2].
The receiver, once it gets the message, is able to estimate the range by subtracting the timestamp of the transmission, recorded by the transmitter, from its own receive timestamp. This approach is very simple, but since the anchors must be synchronized, and in general this process is done by wiring all the anchors together, is very expensive.

In the absence of global synchronization, the ranging process is called *two way ranging*. Though it doesn't require the synchronization, at least two messages are required and it's very sensitive to imperfections of the reference crystal of the nodes.

Setting t_{replyB} and t_{replyA} to constant values, when node A receives the response message from B, it can calculate the ToF by subtracting the transmitted timestamp from the reived one, as well as t_{replyB}:

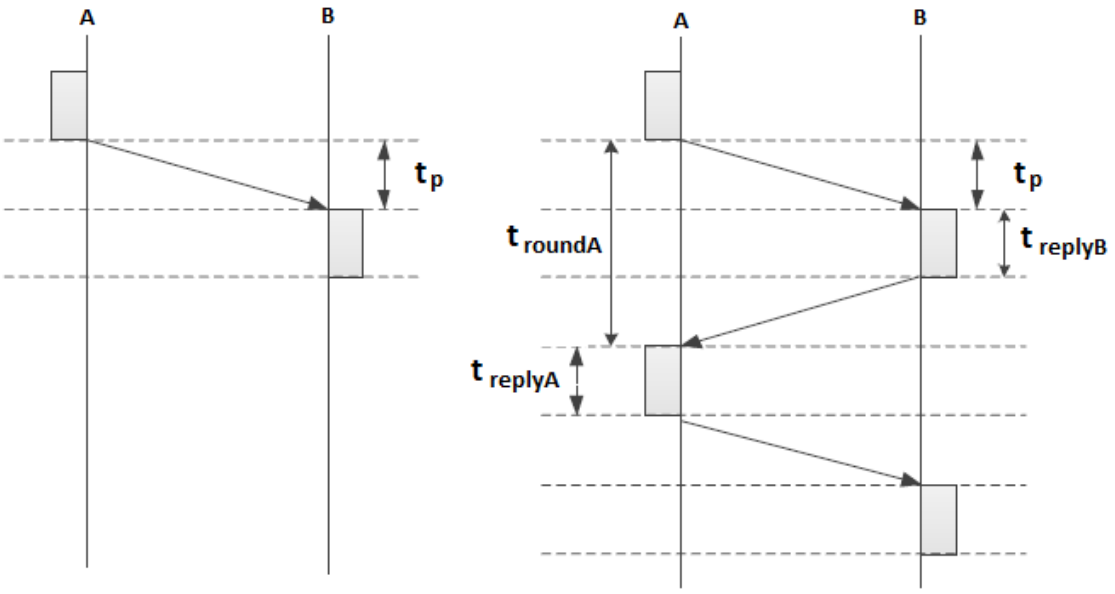


Figure 1.2: ToA and TWR approaches [2].

$$\tau = t_p = \frac{t_{roundA} - t_{replyB}}{2} \tag{1.3}$$

Where t_{roundA} is:

$$t_{roundA} = 2t_p + t_{replyB} \tag{1.4}$$

the roundtrip time respect to node A.
A more robust approach consists in including one more message as depicted in figure 1.2, in this way the propagation time (ToF) is less affected by the errors introduced by the crystals. This approach is called *symmetrical double sided two way ranging* (SDS-TWR), and the ToA is:

$$\tau = t_p = \frac{t_{roundA} - t_{replyA} + t_{roundB} - t_{replyB}}{4} \tag{1.5}$$

The system used in this thesis exploits the *symmetrical double sided two way ranging* approach since its anchors are not synchronized. As will be explained later, the fact that the anchors are not wired is a point of strength for this application,

since it make it more portable and easier to use in an home environment.

1.3 Positioning process

Position, generally speaking, is calculated using measurements of distances or angles referred to fixed points (anchors) whose positions are known. A certain number of fixed nodes are necessary to find a position of a tag that is free to move in the area delimited by the formers. In order to estimate the position in 3D, at least four anchors are needed, while in 2D just three.
In fact the problem in 2D consists of finding the intersection between three circumferences, while in 3D consists of finding the interception of four spheres.

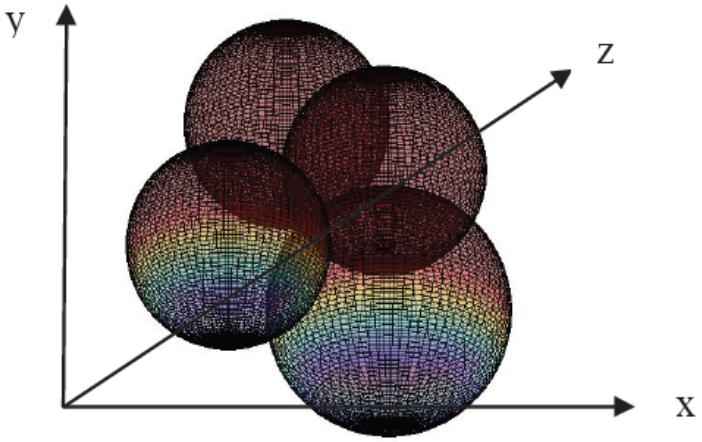


Figure 1.3: Spheres intersection [2].

The ranges obtained during the ranging process are always affected by errors, like any other measurements. This means that during the positioning process the intersection will not be perfect, or better the intersection will not result in a specific point, but rather in a delimited area. In fact considering the ranges as measured distances with random errors, their intersection represents an area in which the tag is situated. The errors that affect the range measurements result in an uncertainty in the tag's position. In RSS and ToA these errors results in an uncertainty in the circle shapes, while in AoA the uncertainty affects the shape of a line [1].

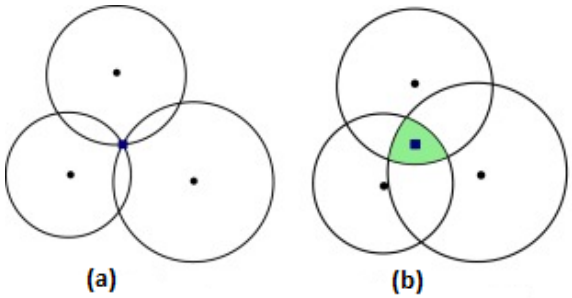


Figure 1.4: a) ideal interception, b)interception with noisy ranges. [17]

There are two main positioning method: geometric approach and statistical approach.

1.3.1 Geometric approach

In the geometric approach the tag's position is determined by the intersection of uncertainty regions. In case of RSS or ToA, as described above, the distances between the tag and at least three anchors are necessary to compute the position. In AoA just two anchors are needed.

Positioning using ToF Time based positiong methods rely on the range measurements obtained using the ToA. Once obtained the distances between the tag and the anchors, by using a geometric approach, is possible to find the unknown position. By collecting at least three range measurements from three fixed nodes in known positions, is possible to locate the tag by interception.

As depicted in figure 1.5 the 2D location is obtained using three ranges d_1,d_2,d_3 referred to three different source nodes. By solving these equations jointly is possible to derive the position of the tag:

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad i = 1,2,3 \tag{1.6}$$

where x and y are the tag's coordinates in 2D, $[x_i,\ y_i]$ is the position of the *i-th* anchor and d_i is the *i-th* range.

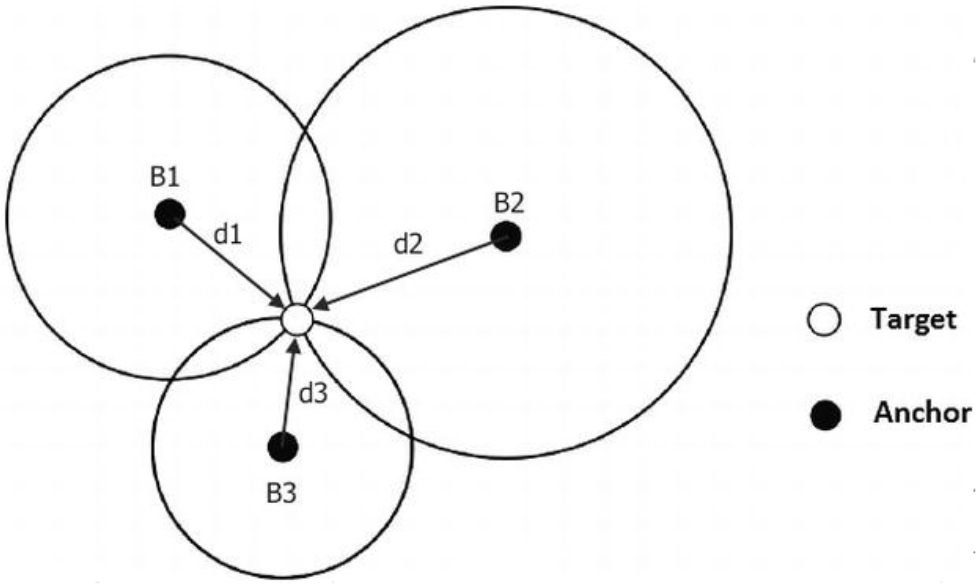


Figure 1.5: RSS and ToA used for the range estimation. Target node's position obtained by intersection [3].

When synchronization is achieved only between anchors and no information are available about tag's clock, is possible to use TDOA. In this approach, the parameter is the difference between two time of arrival. Multiplying this factor by the speed of light, the position of the target is obtained with uncertainty on the shape of a hyperbola as shown in figure 1.6 [3].

ToA is measured at two anchor nodes, τ_1 and τ_2. Since anchors and tag are not synchronized, an offset on τ_1 and τ_2 exist, and since the anchors are synchronized, this offset is equal on both the nodes.

$$\tau_{TDOA} = \tau_1 - \tau_2 \tag{1.7}$$

τ_{TDOA}, that is the estimation of TDOA, is offset free thanks to the difference operation [9], [10], [3].

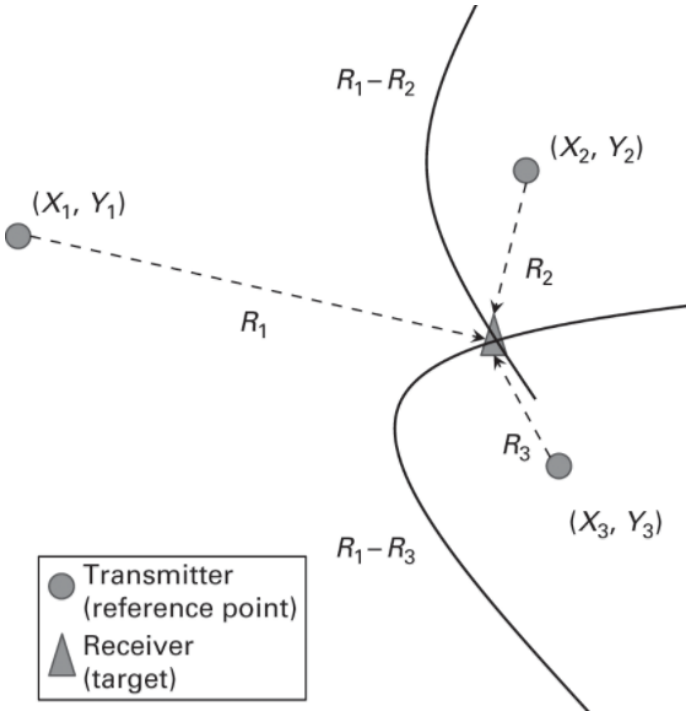


Figure 1.6: TDOA used for the range estimation. Target node's position obtained by intersection of hyperbola [3].

Positioning using RSS In RSS ranges are estimated measuring the signal strength (the power). Knowing that the signal power decreases with the distance (path loss), is possible to understand the distance from a source that transmits with a known power.

$$P(d) = P_0 - 10 \cdot n \cdot log_{10}(\frac{d}{d_0}) \tag{1.8}$$

where *P(d)* is the received power at distance d, n is the path loss exponent, d_0 is the reference distance. Once obtained the range measurements, the positioning process is the same as seen for the ToA.

Positioning using AoA Using the AoA approach only two source nodes are required to estimate the position of the tag [1].

As depicted in figure 1.7 the tag's position is found by intersection of these two

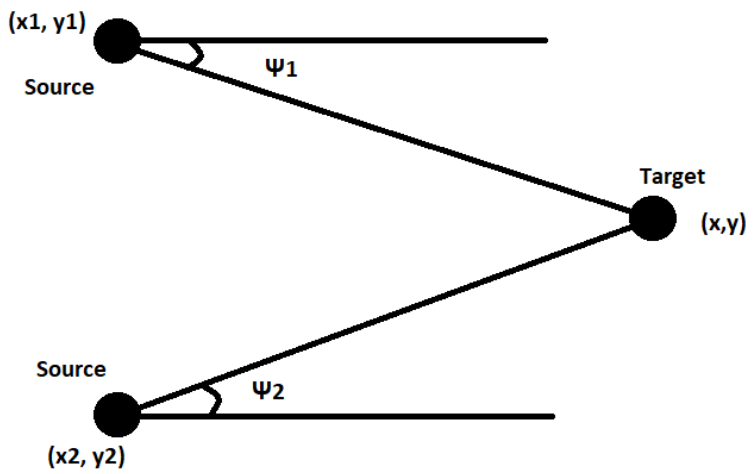


Figure 1.7: AoA used for the range estimation. Target node's position obtained by intersection [3].

lines. Each angle of arrival gives an equation like:

$$tan\psi = \frac{y - y_i}{x - x_i} \quad i = 1{,}2 \tag{1.9}$$

1.3.2 Statistical approach

Since geometric approaches are not able to solve the problem in presence of noisy ranges, a statistical approach is preferred. Literature is plenty of source localization techniques based on additive measurement error model.
A primary distinctions can be between iterative and closed-form algorithms. By definition an iterative algorithm is an algorithm that needs to be performed several times in order to reach a result that satisfies certain parameters, while the closed-form one can compute the result istantly. In general for real time applications, closed-form approach is preferred since is faster.
Other important distinctions are likelihood-based versus least-squared. All these approaches consist of formulating an error or cost function and trying to minimize it.

1.4 UWB technology

Ultra Wide-Band is a type of RF technology characterized by a large bandwidth and low energy level transmission protocol.
By definition stated by the Federal Communication Commission (FCC) an Ultra Wide-Band system has a bandwidth larger than 500 MHz and the frequency in which the system has the maximum power density f_c must be greater than 2.5 GHz. In case f_c is lower than 2.5 GHz, B_{frac} must be larger that 0.2, where:

$$B_{frac} = \frac{B}{f_c} \tag{1.10}$$

where B is the bandwidth of the system [1].
Being f_H and f_L the frequencies at which the power spectral density is 10 db below the one in f_c:

$$f_c = \frac{f_H + f_L}{2} \tag{1.11}$$

$$B_{frac} = \frac{2(f_H - f_L)}{f_H + f_L} \tag{1.12}$$

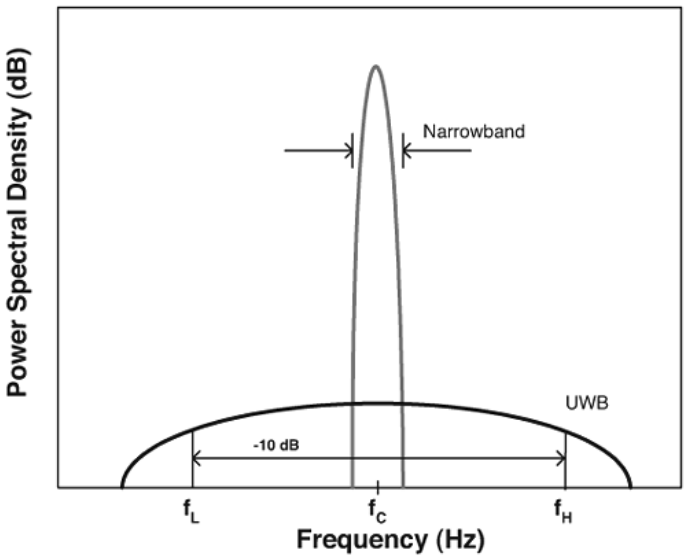


Figure 1.8: Typical bandwidth of a UWB signal [1].

For any RF signal, Shannon's theory characterizes the capacity of the signal channel. It states that a direct relationship between capaciry and bandwidth exists as well as an inverse relationship between bandwidth and power consumption:

$$C = B \cdot log_2(1 + SNR) \tag{1.13}$$

where C is the chanel capacity, B the bandwidth and SNR the signal to noise ratio. For a specific capacity is possible to consume less power using a larger bandwidth, this is one of the most important feature of the UWB: low power.

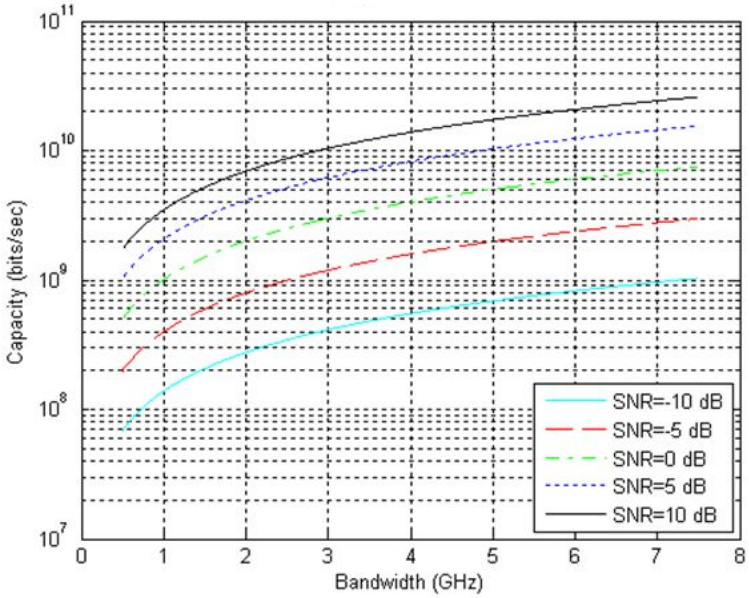


Figure 1.9: Channel-Bandwidth relationship [1].

1.4.1 Features and characteristics

UWB has many important characteristic that makes this technology suitable for ranging and consequently for positioning systems. The most important feature is the large bandwidth in comparison with the other RF systems that are based on narrow-band.
Thanks to the reverse relationship existing between frequency and time domain, the

direct result of the large bandwidth is the short life time of a UWB signal. This means high time resolution that is an important characteristic for positioning since makes the signal robust against multipath.
High bandwidth means also high speed comunication. Furthermore, UWB allows to use low carrier frequencies, where signals can more easily travel through objects.
High time resolution and short wavelength result in robust technology against multipath and fading.
Last but not least, since UWB can be transmitted in base band allows the hardware to be simpler and cheaper.
Conventional RF systems transmit information by modulating frequency, power or phase tipically of sinusoidal wave, while UWB transmits data by generating radio energy at specific time intervals, this meakes it suitable for time position or time modulation.

1.4.2 Impulse Radio (IR)

One of the most used method for UWB transmission is the Impulse Ratio (IR). In this method information of the simbol is uttered by position and/or polarity of the signal. The UWB pulse can be generated from a Gaussian pulse:

$$pulse(t) = \pm\frac{\sqrt{2}}{\alpha}e^{-\frac{2\pi t^2}{\alpha^2}} \tag{1.14}$$

where

$$\alpha^2 = 4\pi\theta^2 \tag{1.15}$$

where α is the pulse factor, while θ is the variance.
As stated in equation 1.14 α is responsible of the pulse width, greater its value, the narrower the width. Pulses of the duration of the order of fractions of nanoseconds are UWB signals.
UWB signal can be modulated using Time-Hopping(TH) and Direct Sequence(DS) modulation. Here an example of TH transmission:

Two consecutive signals form a symbol. IR signal occupies one of the chip-intervals(T_c) inside a frame (T_f). In the figure depicted in 1.10 the stream represents

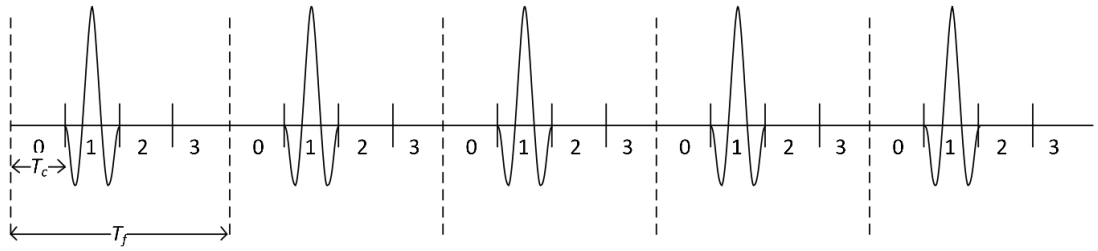


Figure 1.10: IR UWB signal [1]

a binary data.
Impulse UWB is a good technology for location sensor networks, thanks to its high quality communication. The fine time resolution of some sub-nanosecond pulses allows an accuracy of a few centimeters in distance measurements.

1.4.3 Multipath

By definition multipath is the propagation phenomenon that results from a radio signal reaching the receiving antennna through two or more paths. Typical causes of multipath are reflection and refraction of the ionosphere, or more in general in an indoor environment, walls and objects. Signals affected by multipath result in constructive or destructive interference. The latter causes fading, that is the variation or attenuation of the signal.
Signal reflected by multiple objects results in several signals that reach the receiver in different moments causing interference that makes the receiver unable to distinguish the original signal that travels in LOS.

Thanks to the large bandwidth that results in high time resolution(short pulses), UWB is a robust technology respect to this problem. This enables the receiver to identify multipath reflections from the original signal. This property makes the UWB suitable for indoor environments full of obstacles and objects.

1.4.4 UWB regultion

UWB technology is license free, so anyone can implement an UWB system without the necessity of a patent. However since UWB cover a wide band of frequency a

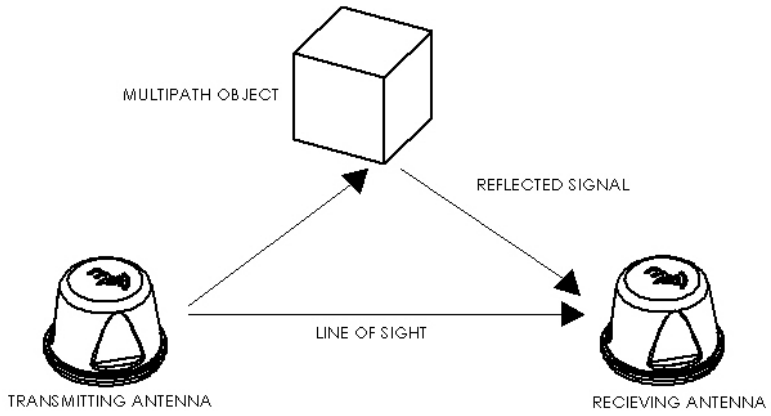


Figure 1.11: Mutipath example [18].

regulation must exist to avoid collision and interference among different communication methods.
Federal Communication Commission (FCC) is one of the most important organizations that provides regulations. This was the first organization to design rules about UWB in 2002.

The IEEE 802.15.4-2011 states that the standard output level for UWB transmission(Equivalent Isotropically Radiated Power EIRP) is -41$\frac{dBm}{MHz}$, and the spectrum from 3.1 to 10.6 GHz is divided in 14 channels of 500 MHz each.

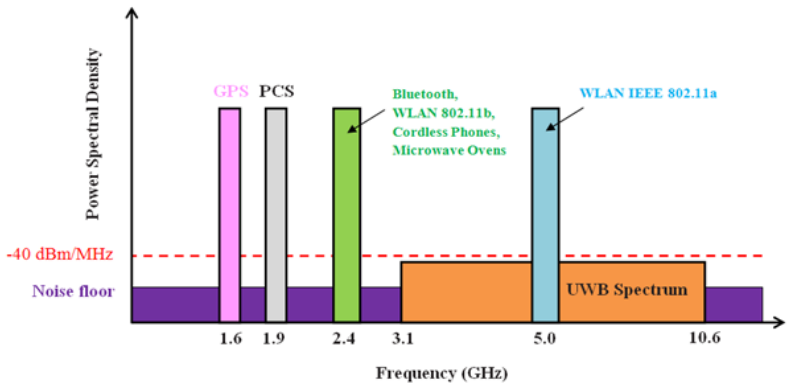


Figure 1.12: FCC regulation for emitted signal power [3].

Figure 1.12 demonstrates the FCC's regulation about the transmitted UWB power.

Many countries, as result of the regulations, have allocated frequency spectrum for UWB use. As stated above, the UWB spectrum is divided in 14 channels, as it is possible to see from figure 1.13 many countries don't allow the use of several channels since they are already used by other applications such as could be satellites communication.

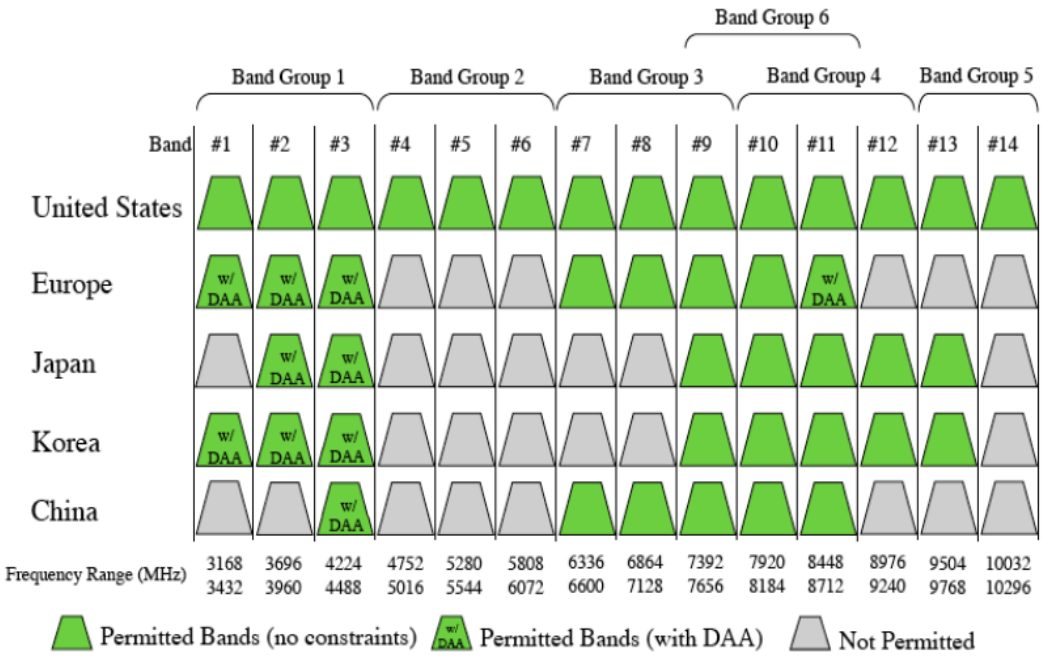


Figure 1.13: Allocated channels in different countries [5].

DAA stands for *Detect and avoid*, and it is a technique to avoid interference between transmitter and wireless environment. Thanks to this UWB can use the designed channel without interfeering with other technologies that work on the same frequencies.

1.4.5 Commercial UWB systems

Nowadays many commercial UWB systems are present on the market. They differ in accuracy, cost, dimension and maximum covered range. Some examples are presented in the next paragraphs.

Zebra Zebra Dart UWB [13] (figure 1.14) is designed for real time position in 2D and 3D applications . This system is compliant with the international standard IEEE 802.15.4.f.
Its features include high perormance in environments affected by strong multipath, battery life up to 7 years at 1Hz blink rate, range capability up to 200m, programmable frequency sample of the tag up to 200Hz, waterproof antenna and an accuracy around 30cm.
Zebra tags have a diameter of anly 4 cm and thanks to their long battery life they can last for 7 years. The major limitation is the cost that is 12,000 USD.

Figure 1.14: Zebra Dart UWB kit [13].

Pozyx Pozyx system, figure 1.15, provides positioning and motion information. The system is composed by 5 devices, 4 anchors and 1 tag. 4 anchors are required for 3D positioning, 3 for 2D. This board is Arduino compliant and it's equipped with accelerometer, gyroscope and magnetometer that are used to obtain the orientation of the device. All these sensors are affected by bias and error, however together these kind of errors can be mitigated.
The frequency sample of the system is up to 140Hz with an accuracy of 10cm. The major limitation is the maximum range that is typically around 30m. The cost is aound 740 USD.

Decawave Decawave is an Irish company that designed the *DW1000 IC*(1.16) [7], an integrated circuit designed for indoor real time location. It is IEEE 802.15.4-2011 compliant. It has an accuracy of 20cm even with moving tag up to 5m/s.

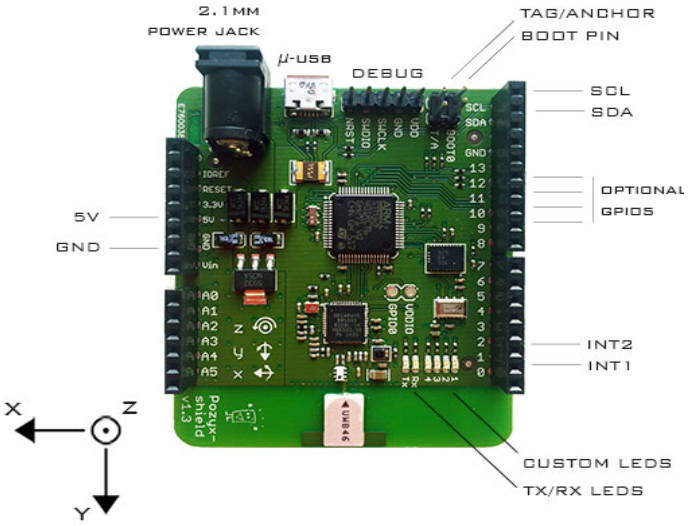


Figure 1.15: Pozyx board [19].

Communication range up to 290m and high fading environment robustness.
Its low power consumption makes it suitable for monitoring aplication. TREK1000 is Decawave kit composed by 4 configurable EVB1000 that mount the DW1000. The cost for the entire kit is 900 USD.

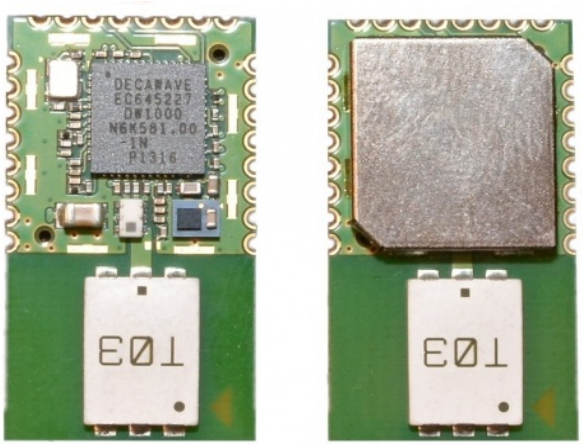


Figure 1.16: DW1000 with antenna [7].

According to the low cost and good performance such as high accuracy and the programmable kit, Decawave system was the designated one for this thesis work.

1.5 SLAM

Mobile robots must be able to position theirself inside working environment. In ideal case GPS can be used to obtain absolute position of the bot, but in many cases and application, the GPS is not suitable such as an indoor environment. Sometimes also a problem related with cost, size and weight could limit the use of that technology. In some cases, knowing details about the working area, a robot can locate itself using exteroceptive sensors like laser scanner or camera for relative position measurements to known landmarks.

The problem of Simultaneous Localization And Mapping (SLAM) [4] is of high interest in many applications, ranging from planetary exploration to transportation and environmental monitoring. SLAM problem consists of constructing and updating a map of a certain environment and at the same time tracking the position of an object within it.
SLAM uses several types of sensors such as: laser rangefinders, LiDAR, sonar sensors, cameras. According to sensors many different algorithms have been exploited based on Kalman filters or particle filters. They elaborate a probability function for the position of the robot and the shape of the map.

In 2D cases the kinematics of the robot is given by a mixture of rotation and move forward commands. All this command are implemented with some additional motor noise. An alternative approach consists of reading odometry data from the robot's encoders of the wheels after each command.

So thanks to SLAM a robot can navigate in an indoor enviroment, constructing a map of this, and track its position. Giving this a bot is able to navigate in an unknown surroundings.

Chapter 2

Hardware description

This section describes the hardware used starting from the robot and its feature finishing to the UWB Real Time Location System (RTLS).

2.1 Home Robot

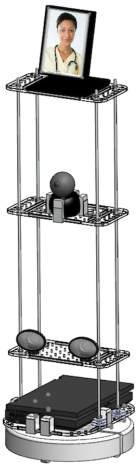


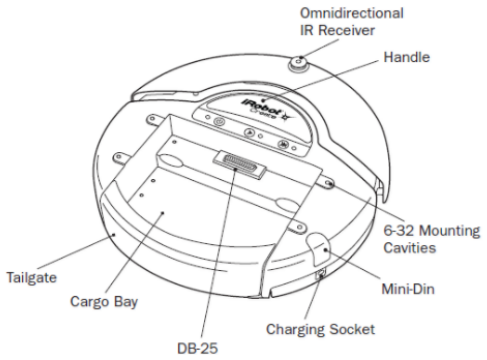
Figure 2.1: Home Robot Hardware [5] .

The Home Robot is composed by three main parts: iRobot, monitor and webcam. The iRobot [14] represents the most important element, the one that is in charge of the movement and it is also the physical base that support all the rest. The monitor allows the patient to interact with the axternal oparator connected by videoconference. The webcam reprents the *"eyes"* of the caregiver.
All these elements require an external processing unit such as a computer in order to interact with each other. The PC has the duty of coordinate all the processes by means of a dedicated java software.

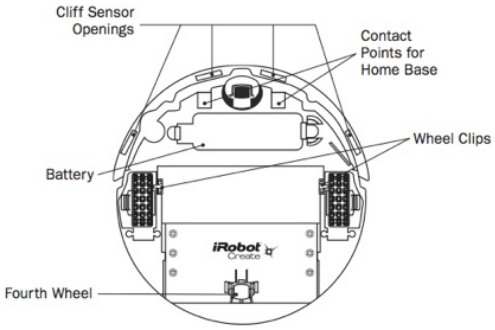
2.1.1 Hardware

iRobot Create

iRobot Create is a development kit produced by Roomba that allows the user to program it without knowing the low level code. iRobot Create's Open Interface (OI) provides to the user with a set of drive commands. It is possible to attach external electronics parts to it like light display or ranging sensors.



(a) iRobot Create top view [14].



(b) iRobot Create bottom view [14].

This platform is based on typical vacuum cleaners. It is equipped with two differential driven wheels. Thanks to optical encoders placed on the wheels is possible to track the motion and perform accurate movements. The OI provides the travelled distance within 1 mm of resolution, as well as the turned angle within 1 degree of accuracy.
Each wheel can rotate independently up to 500 $\frac{mm}{s}$. Thanks to simple formulas is possible to obtain the geometric center speed and the angular velocity:

$$v = \frac{\omega_l + \omega_r}{2} r \quad [\frac{m}{s}] \tag{2.1}$$

$$\omega = \frac{\omega_l - \omega_r}{2d} r \quad [\frac{rad}{s}] \tag{2.2}$$

where v is the geometric speed, ω_l and ω_r are the angular speed of the left and right wheel respectively, r the wheel radius and d the distance between one wheel and the geometric center.

iRobot Create is equipped with two communication port: cargo bay connector and a serial connector. Connecting the last one with the serial port of a PC is possible to start a serial terminal program capable of sending commands. This communication is set to work at 57600 baud, 8 data bits and 1 stop bit.

MIMO monitor um 720s

Figure 2.3: Mimo monitor.

Mimo Um-720s is a compact foldable 800x480 monitor that can be tilted till 90 degrees. Video, touch and power are provided via USB connection. This 7-inch resitive LCD display is compatible with Windows, Linux and Mac OS.

Webcam Logitec Orbit

Quickcam Orbit is a webcam by Logitec mounted on the bot in order to provide the user with sight of the current environment around it.
It's a 2 MegaPixel camera equipped with autofocus and USB 2.0 connection. The latter allow the remote control thanks to which is possible to rotate the camera of 102 degrees.

Figure 2.4: Logitec webcam

2.1.2 Software

The PC used for this application is equipped with Linux distribution, Ubuntu 17.10. This provides the serial communication with the robot, and it has also all the libraries needed to control the webcam and the external monitor.

The software is in charge of managing the movements of the robot transmitting commands through the serial port and receiving the feedback form the sensors. Another important task is the management of the videoconference and the remote control. In fact the caregiver has the ability of controlling the robot, in particular iRobot movements and webcam orientation, thanks to an easy interface.

The framework used for the software interface is *Node.js*. This is based on *JavaScript language* that is one of the most used programming language nowadays. Node.js is a JavaScript interpreter with low level APIs that allows the user to access processes and files. Thanks to its event-driven approach, its asynchronous architecture never blocks the execution.

Unlike the traditional programming where the execution is sequential and in case of blocking function the entire execution is stopped, Node uses callback functions that are invoked when the result from a blocking function is available. Thanks to this the execution is always in a running mode. In the Home Robot application this particular approach is preferred since allows the interface to monitor multiple features at the same time, avoiding the interruption of the execution that will cause loss of information coming from onboard sensors.

Software architecture

In order to exploit the event-driven asynchronous approach the software is divided in two parts: the *Controller* that is in charge of control the robot hardware and the *Player* that manages the web application used by the caregiver to establish a communication channel with the patient and the romote control [5].

The Player is designed to run on an external server, in this way it could work as a server-client framework allowing multiple robots to connect with externl operators.

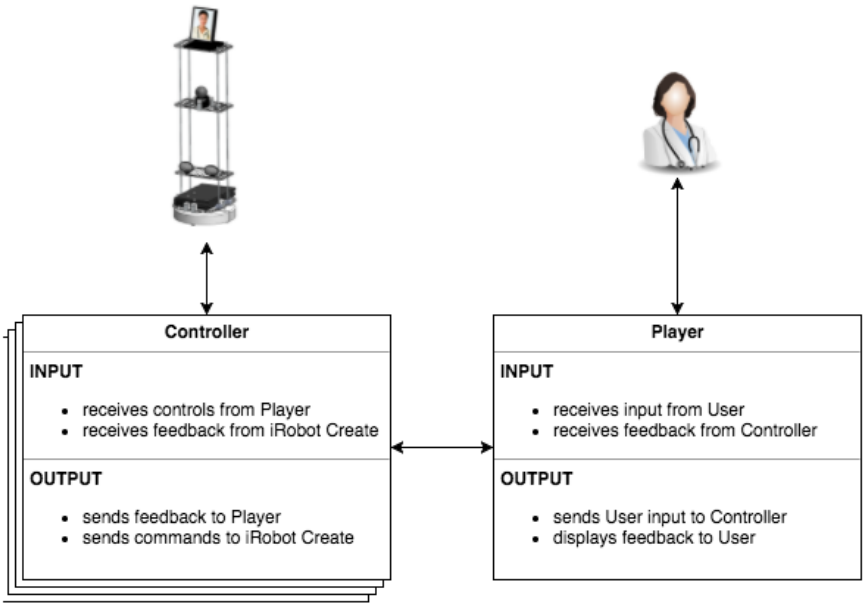


Figure 2.5: Software architecture of the home robot [5].

In this specific application both Player and Controller run on the PC.
The communication from Player to Controller is called *publish-subscibe* and allows the former to communicate with one robot at time, group of them or all of them.
The technique used to communicate from the Controller to the player is instead a producer-consumer link. The producer is the robot that starts sending messages on the communication channel, while the Player is the consumer that reads information present on the channel that acts like a FIFO buffer memory.

2.2 Decawave RTLS

The Wireless Sensors Network (WSN) represented by the Decawave RTLS is used for two main reasons in this application: localize the robot and the patient in the indoor environment and transmit the alarm messages from the wearable sensor placed on the patient to the caregiver by means of the software present on the PC connected to the robot.

Decawave RTLS is composed by anchors and tags that are represented by configurable devices. For this application TREK1000 development kit has been chosen. Its features are presented in the following sections.

2.2.1 Hardware

TREK1000

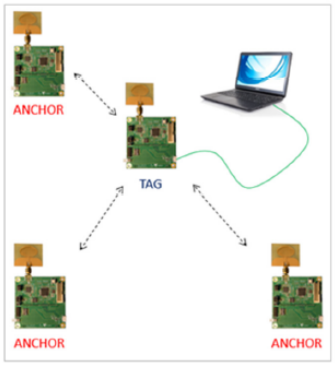


Figure 2.6: Navigation use case [8].

TREK1000 is composed by 4 EVB1000, evaluation boards for the DW1000 IC, stands and USB cables. TREK stands for Two Way Ranging(TWR) RTLS IC Evaluation Kit[7]. This kit provides a solution to the RTLS problem using UWB technology. UWB based RTLS use time based measurements like ToA to reach high accuracy position. TREK1000 exploits its software application and UWB ToA measurements for 2D and 3D location. Since for 3D positioning 4 anchors are needed and only 4 devices are provide (3 anchors and 1 tag), Decawave software takes as solution of the sphere intersection the one below the anchor's plan. Furthermore exists the possibility to expand the kit using more EVB for a maximum of 4 anchors and 8 tags. The fourth anchor doesn't increase the accuracy, but according to Decawave algorithm, is used to identify if the tag is above or belove the anchor's plan.

It works in three different use cases: tracking use case, geo-fancing use case and navigation use case. In the first one the tag is located respect to fixed anchors, the second one determines when tags leave or enter specific areas near an anchor positioned in the center (suitable for child monitoring, security bubble and personal safety). Navigation use case is the same of the tracking one, but the processing unit (PC) is connected to the tag instead of the anchor, and is the configuration used for the final application. The robot is equipped with one tag connected to the PC, another tag is placed on the patient and the anchors are placed around the home environment.

Anchors placement Decawave gives some hints to the user about the position of the fixed anchors using the TREK1000 in order to obtain the maximum possible accuracy[6]. The three anchors must be mounted at the same high, at height around 2-3 meters, above people's head to avoid interference. Anchors mounted to form a triangle, in fact if placed on a straight line the uncertainty intersection area of the three circle would be too wide resulting in poor accuracy. Antennas more than 15 cm far from the nearest wall to avoid interference.

EVB1000

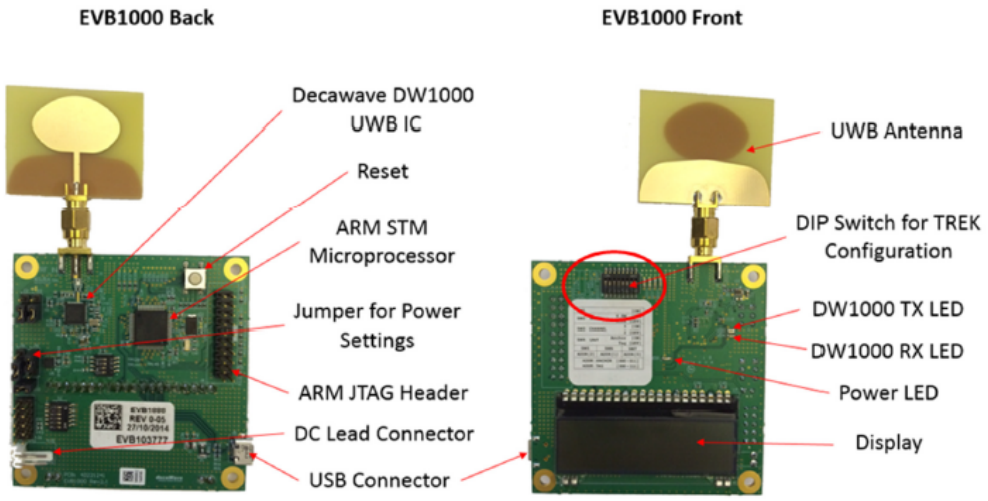


Figure 2.7: EVB1000 front and riar view [7].

EVB1000 is an 70mmx70mm evaluation board equipped with DW1000 IC. Everything is coordinated thanks to a STM32F105 ARM Cortex M3 processor that communicates with DW1000 IC by means of a SPI interface. In order to be programmed, the processor is connected to a twenty pin JTAG header. Two power connectrors can supply the system, a micro USB and a DC 3.3V power input.
This board allows the development of RTLS applications and WSN useful in many different fields like healthcare, warehousing, logistic and automation.

As depicted in figure 2.8 on the display side the Printed Circuit Board (PCB) is provided with an LCD display and an eight(8) pin DIP switch. Thanks tho the latter

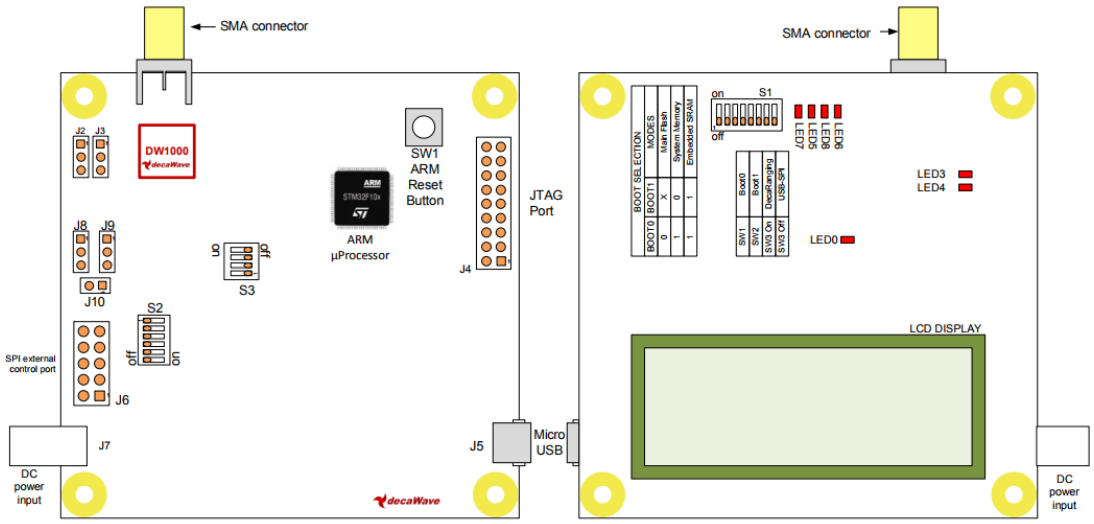


Figure 2.8: Component side (left) and display side (right) of the EVB1000 [6].

is possible to configure the device. Each DIP switch selects a certain configuration and in the up position is considered ON, while in the down position is OFF:

- The first one must always be on;
- the second allows the selection of the data rate, 6.8Mbps [ON] or 110Kbps [OFF];
- the third selects the UWB channel, channel 5 [ON] or chanel 2 [OFF];
- The fourth establish if the device is an anchor [ON] or a tag [OFF];
- DIP from 5 to 7 represent the number of anchor[0-2] or the number of tag[0-7] in binary code;
- The last one is reserved.

As explained above the board is capable to work on two different channels: channel 2 at 3.6GHz and channel 5 at 6.5GHz. Each country as its own regulation about the allocated spectrum. In USA both channels 2 and 5 are allowed, while in Europe only channel 5 is permitted.
On the top of the device there is a SMA connector for the antenna. The antenna

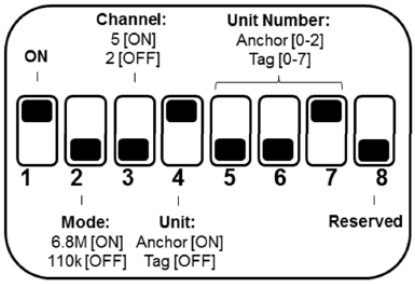


Figure 2.9: DIP switch for device configuration [7].

used is omni-directional with 1 dBi of gain for channel 2 and 3 dBi gain for channel 5.

DW1000

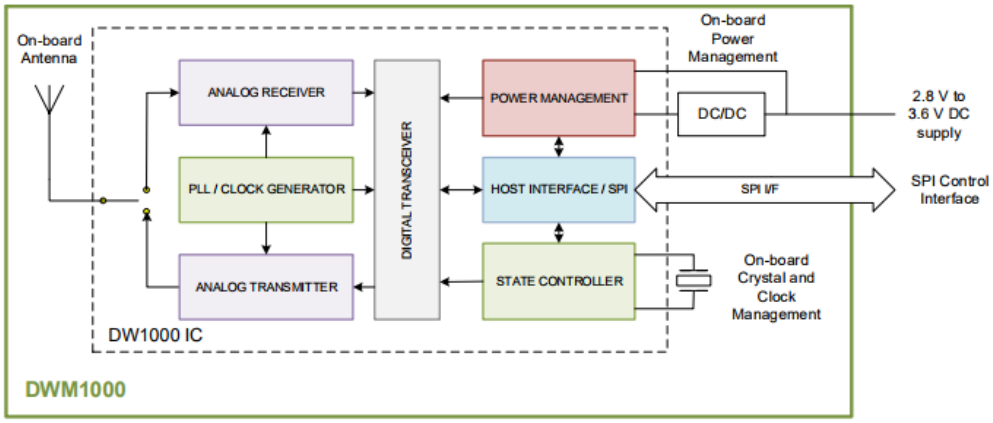


Figure 2.10: High level scheme of the DW1000 IC [6].

DW1000 is Decawave UWB transceiver IC. It integrates RF circuitry, power management, antenna and clock temporization ciruitry all on board. The one used on EVB1000 uses an external antenna in order to exploit the benefits of a bigger one. Since it is IEEE 802.15.4-2011 compliant can be used in many countries exploiting its 4 RF bands from 3.5 GHz to 6.5 GHz. The 23 mm x 13 mm x 2.9 mm 24 pin package host also a SPI interface to allow the communication with an external processor.
Thanks to its integration, no external RF circuit are required, so this makes its

usage simpler. Furthermore its low power consumption with its low cost make this module one of the best choise for the design of a RTLS system.
The on board 38.4 MHz crystal has been trimmed during the production process to reduce the initial error to 2 ppm. This is important for the ranging accuracy where anchors and tags are not synchronized among them. The board hosts an OTP(One Time Programmable) memory that the user can exploit to save the antenna calibration information.

2.2.2 Software

In this section the software relative to TREK1000 system will be analyzed. This comprends the WSN messages exchange, Two Way Ranging process and the GUI interface for PC designed by Decawave.

WSN

The Wireless Sensors Network provides mean through which anchors and tags exchange messages that are foundamental for the localization process. In this way Tag0 connected to the robot can receive the position of Tag1 placed on the patient. The communication protocol is organized in a Time-Division Multiplexing (TDM) approach. The superframe is represented by a TDM cycle, which is composed by a fixed number of time-slots of equal duration. For each time slot only one tag is allowed to transmit using UWB channel, while the other is sleeping. The Anchor0 is the TDM master, and it is on charge of maintaining tags into their own time-slots.

The message format is the IEEE 802.15.4 standard figure 2.11[6]. Since TREK application always uses data frames with 2 octet for source and destination address, the two frame control slots are constant.

The sequence number of the octet is incremented modulo 256 every time a frame is sent. The source and destination values depend on EVB1000 board's configuration of the DIP switches, anchor or tag mode and number. The 2 byte FCS are a CRC frame check sequence that is generate by the DW1000 IC and attached to the transmitted message.
The content of the ranging message depends on the type of the message sent. Three

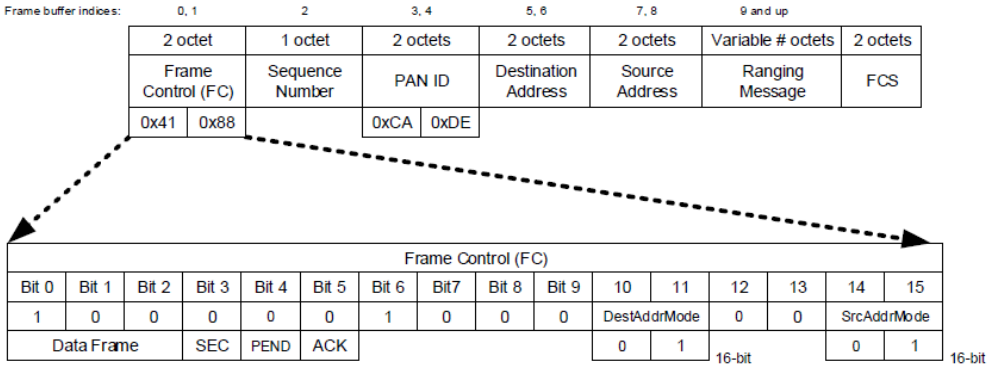


Figure 2.11: Message standard format [6].

types exist according to the sequence exchange for the TWR and are presented in the next paragraphs.

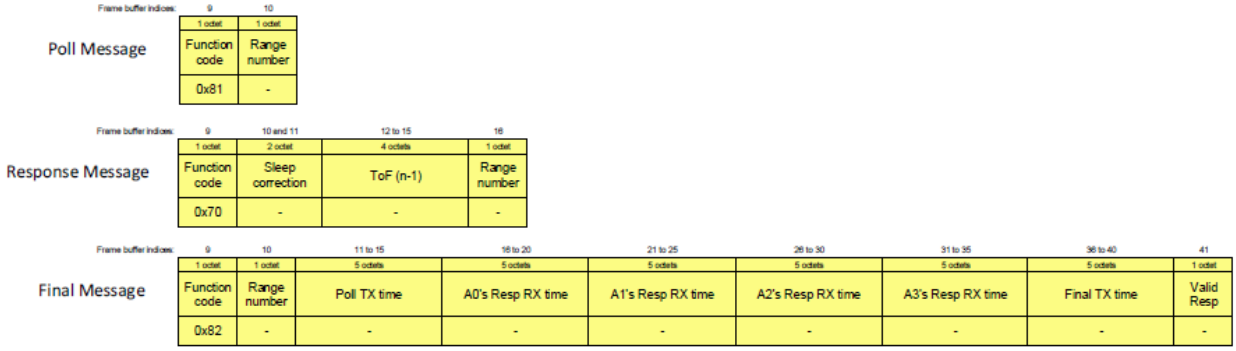


Figure 2.12: TWR message types [6].

Poll message Poll message is sent by the tag and has the purpose to initiate range measurement. The function code is 0x81, this octet identifies the Poll message. The range number depends on the sequence and each time is incremented.

Response Message Anchor is responsible of sending the Response meassage after the Poll message. It is composed by 8 octets. The first is the function code 0x70, the second and the third together form the sleeping correction, a parameter that adjusts the tag's sleep duration so the tag's activity can be synchronized with the time slots and avoid, in this way, to interfere with other tags. Then 4 octets, 32

bits, that represents the ToF value measured during the previous message exchange. The last octet is the range number.

Final message Final message is sent by the tag as conclusion of the message exchange. It is 44 octets long. The first is the function code 0x82, the second the range number, then 5 octets for the Poll TX time that is the timestamp, so the precise time the Poll frame was trasmitted by the tag. Then 4 equal fields composed by 5 octets, each one representing the Resp RX time, the timestamp for the response time from anchor 0, 1, 2, 3. Final TX time is a 5 octets field that contains the timestamp of the final message. The last frame is 8 bit and specifies which response time is valid.

The final application is based on two foundamental steps: the measure of the distances between anchors and tags, and the positioning algorithm that determines the 2D location of the Home Robot and the patient in the environment. The first step is performed thanks to the TWR process analyzed in the next section.

TWR

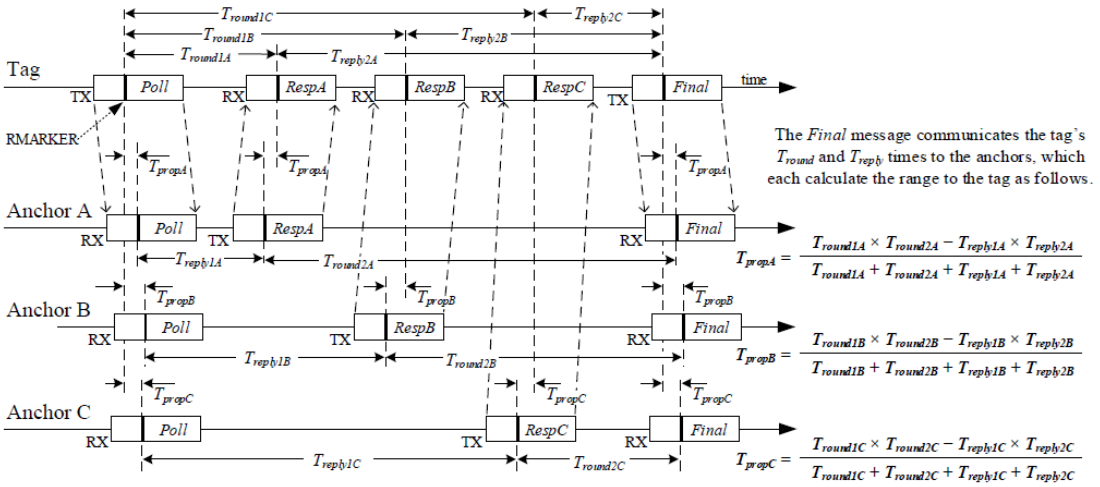


Figure 2.13: TWR algorithm.

The Tag, that is in charge of starting the TWR process, attemps to range to four anchors and then, once sent the final message, go to deep sleep mode to save energy. After a superframe time, it will attemp again.
In figure 2.13 is depicted the TWR algorithm scheme, the tag sends a Poll message which is received by three (or four) anchors. Then anchors reply sequentially with packets RespA, RespB and RespC (RespD, not present in the picture). At the end the tag sends the Final message received by all the anchors. This allows to locate the tag after sending only 2 messages and receiving 3. The low number of exchanged messages represents a saving in terms of battery power and air-time.

Recording the timestamps of the messages is possible to calculate the ToF between the tag and the anchors as shown in picture. Times called T_{reply} are known since are constant written in the code used to synchronize the response process. $T_{reply1A} < T_{reply1B} < T_{reply1C}$ in this way the tag knows the exact sequence of the response messages. These time periods consider also an important issue that is represented by the time during which the message in not in air but is handled by the device. In fact there is a small time that elapses from when the processor sends the message to when the message is in air. The same happens when a message is received, there is a delay from when the message reaches the antenna to when it is processed by the processor. So these time periods are tuned in order to cosider this delay.

Positiong algorithm

Decawave's positioning algorithm is a *trilateration* algorithm. it exploits 3 anchors to determine the 2D position and the fourth (if present) to choose the right solution on the z axis, otherwise the solution is always the one below the anchor's plan. The input of the algorithm are the 3D position of the anchors and the measured ranges obtained by means of the TWR process.
The trilateration process represents the steps necessary to find the solution of the intersection of three spheres. These spheres are located on the plane z = 0 and their radius are represented by the ranges. Trilateration will lead to two different solution with the same x and y value, but one with z value below the anchor's plane and the

other above. The range coming from the fourth anchor could be used to chose the right solution.

The tag position is found by formulating this geometric problem as intersection of spheres and then solving for the unknown x, y and z.

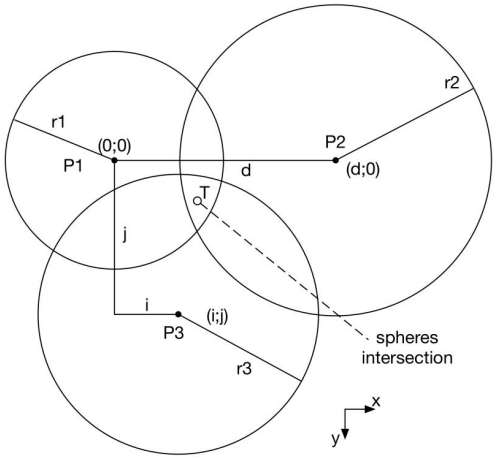


Figure 2.14: Three spheres intersection.

According to figure 2.14 the equation of the three spheres are:

$$r_1^2 = x^2 + y^2 + z^2 \tag{2.3}$$

$$r_2^2 = (x - d)^2 + y^2 + z^2 \tag{2.4}$$

$$r_3^2 = (x - i)^2 + (y - j)^2 + z^2 \tag{2.5}$$

Where d, i and j are the distances from the center of the first sphere located in (0;0) as depicted in figure 2.14. Spheres intersection is located in (x, y, z), obtained solving the three equations simultaneously. First of all equation 2.3 and 2.4 to find

x:

$$\begin{aligned} r_1^2 - r_2^2 &= x^2 - (x-d)^2 \\ r_1^2 - r_2^2 &= 2dx - d^2 \\ x &= \frac{r_1^2 - r_2^2 + d^2}{2d} \end{aligned} \tag{2.6}$$

Now x is known since the formula is composed of known value. Next step is to find y using equations 2.3 and 2.5, writing the first as $z^2 = r_1^2$ - x^2 - y^2:

$$\begin{aligned} r_3^2 &= (x-i)^2 + (y-j)^2 + r_1^2 - x^2 - y^2 \\ y &= \frac{r_1^2 - r_3^2 - x^2 + (x-i)^2 + j^2}{2j} \\ y &= \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x \end{aligned} \tag{2.7}$$

Knowing x also y is known. Last step consists in finding z value. Once again from equation 2.3:

$$z = \pm\sqrt{r_1^2 - x^2 - y^2} \tag{2.8}$$

As explained before this equation admits two possible solutions, one above the anchor's plane and one below.

The values of i, j and d are found considering the centers of the spheres as vectors from the origin and building a cartesian plane consistent with their directions. The x axis is in the direction from P1 to P2:

$$\hat{e}_x = \frac{P2 - P1}{||P2 - P1||} \qquad d = ||P2 - P1|| \tag{2.9}$$

i is the x component of the distance between P1 and P3, so geometrically speaking is the scalar product between the distance vector and the x unit directional vector:

$$i = \hat{e}_x \cdot (P3 - P1) \tag{2.10}$$

The unit directional vector for the y axis is:

$$\hat{e}_y = \frac{P - P1 - i\hat{e}_x}{||P - P1 - i\hat{e}_x||} \tag{2.11}$$

j represents the value of the distance between P3 and P1 along the y direction, so the scalar product between the distance vector and the y unit directional vector:

$$j = \hat{e}_y \cdot (P3 - P1) \tag{2.12}$$

Once obtained the unit directional vectors of x and y, the z one is simply the cross product between these two.

$$\hat{e}_z = \hat{e}_x \times \hat{e}_y \tag{2.13}$$

Finally the intersection point is calculated as:

$$\vec{P_T} = P1 + x\hat{e}_x + y\hat{e}_y \pm z\hat{e}_z \tag{2.14}$$

This result is expressed respect to the origin and using coordinate obtained respect to the anchors position.

Log File

Using the Pc application provided by Decawave is possible to save Log file produced by this application. This file is a text one in which all the information about the anchors position, range measurements and timestamps are written.

Figure 2.15 shows how a log file is made. In the first raw several information about the system configuratior are written. Then raw by raw range values appear with range sequence number and the anchor and tag ID representing to which devices the range belongs to.

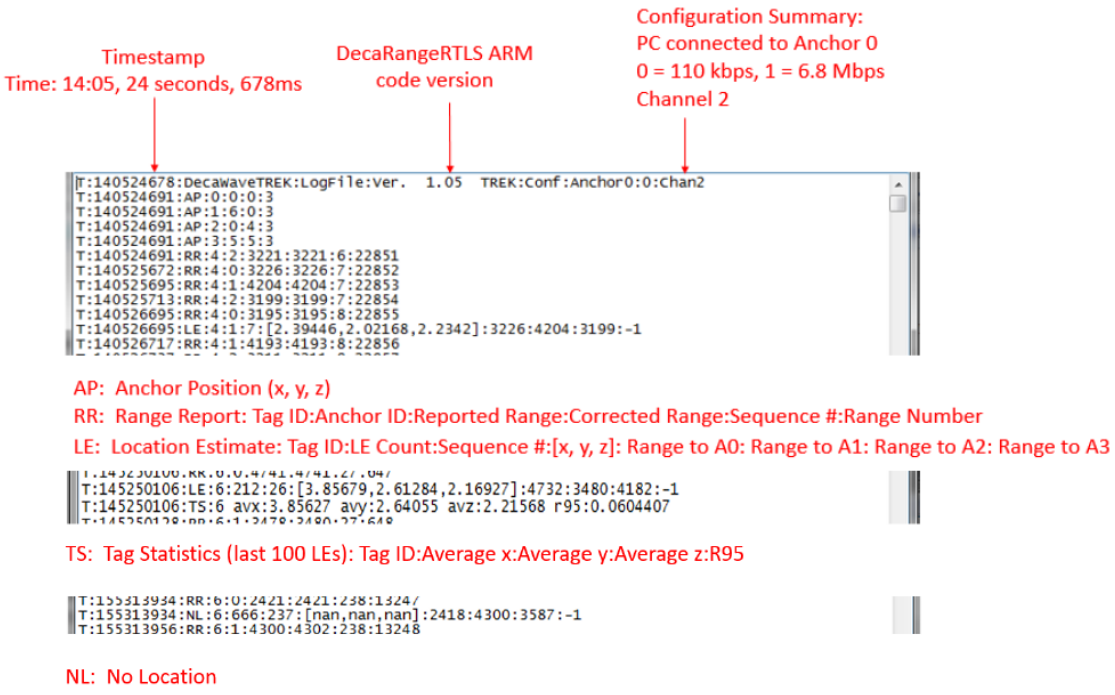


Figure 2.15: Log file [7].

Chapter 3

Study of the TREK1000 system and errors

In order to verify the performance of the system and to be sure of the feasibility of the final application using Decawave TREK1000 development kit several tests have been made.
The first test is a static one in order to understand which anchors configuration would lead to the best accuracy. This means placing the anchors in different fixed constellations and compare measurements to figure out when the smallest error is reached.
Then since the range measurements can be seen as the real value plus a random one that represent the error, from measure to measure the position of the tag varies randomly of a certain value. This variation lead to an oscillation of the tag position around the meadian value. This oscillation is shown in the next sections.
The last test is a dynamic one and allows to verify the ability of the system to track a moving object.

Note: all measurements have been obtained by means of the log file from decawave and then sucbsequently eleborated using designed ad hoc MatLab scripts.

3.1 Static error

3.1.1 Static error respect to anchor's position in the space

In this section static measurement errors will be analyzed comparing different anchor constellations. In order to do so, a grid of 6 meters x 4 meters is built on the floor. The anchors are positioned at the vertices of this rectangular grid using 1,17

m tall tripods, while the tag is moved(on the floor) from measure to measure inside and on the side of this rectangle and on the vertices of the small square of 1m x 1m size forming the rectangle.

Four different anchor configurations have been investigated in order to figure out which is the best solution, analyzing measurements dispersion and error distribution.

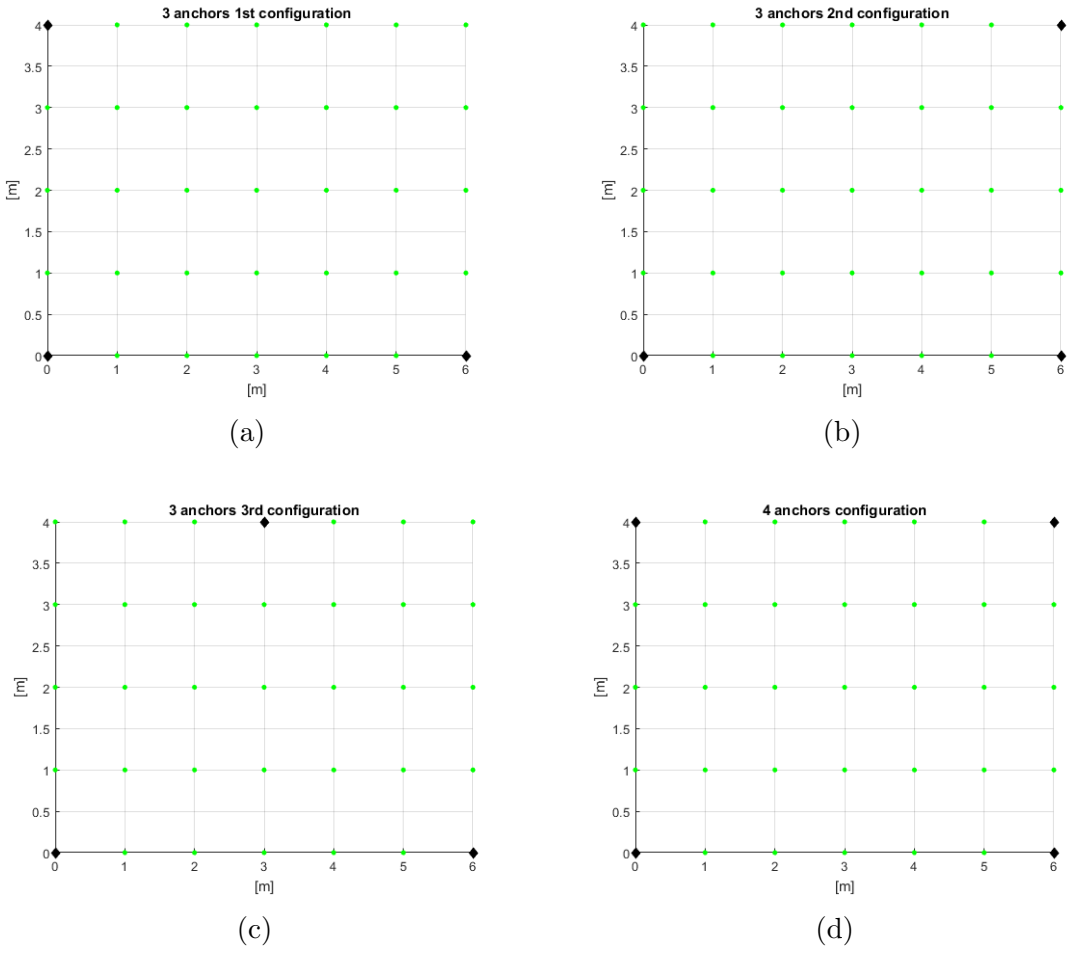


Figure 3.1: Anchor configurations and grid.

In figure 3.1 the four configurtion are showed, where the black diamond represent the anchors, while the green dots are the test points where the tag is placed. First 3 configurations exploit only 3 anchors positioned forming a triangle, while the last one exploits 4 anchors. Anchor 0 always in position (0,0), anchor 1 always in

position (6,0), anchor 2 changes from (0,4) to (3,4) or (6,4), anchor 3 only in the last configuration in position (6,4) on a tripod 1,64 m tall. In fact, as explained by Decawave, the fourth anchor must be on a different plane respect to the other anchors to be able to choose the right solution among the two obtained thanks to the spheres intersection

Note: in the following pictures anchors are represented by black diamonds, test points are green dots, each blue dot is a measurement. Each cluster of measurements belonging to the same test point has a red line that represents the linear interpolation of this cluster and helps the observer to understand in which direction it extends. Furthermore the area among the anchors is highlighted.

3 anchors, first configuration results

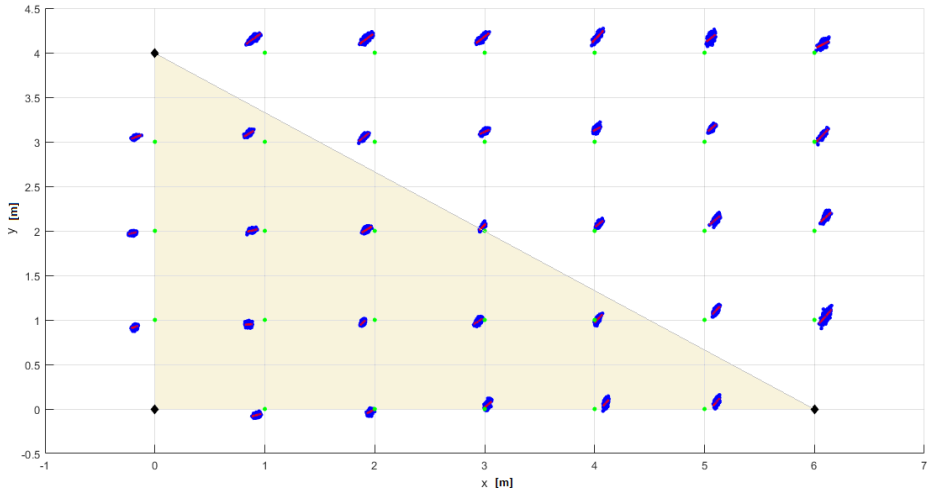


Figure 3.2: 2D plane and 3 anchors (first) configuration measurements.

In figure 3.2 is shown the first anchor constellation, where anchors form a rectangle triangle. Clusters of points are more accurate in the core of the highlighted area, so their average point is closer to the ideal point(green dots). This has a geometric explanation, in the middle of the area covered by the anchors the intersection area

is tight, while on the border the uncertainty area obtained by intersection is spread. This lead to an higher uncertainty on the border and in the outside region.

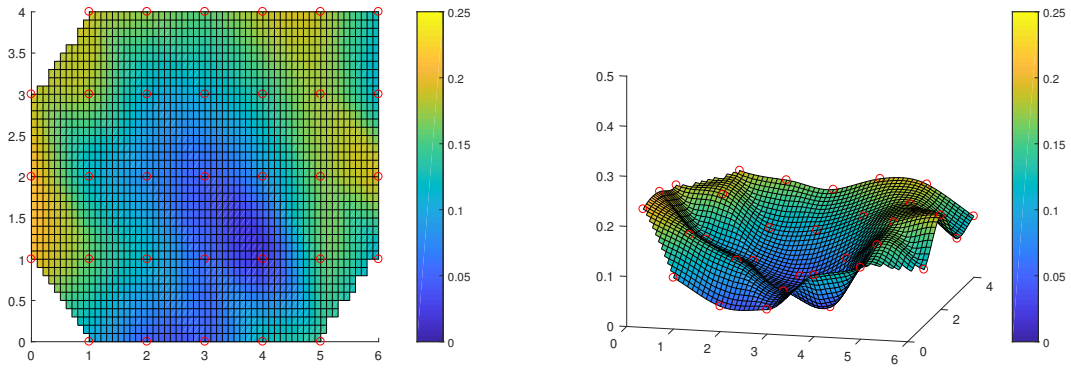
The following table shows the error on the x and y axis and the absolute error calculate as the euclidean distance between the average point of the cluster and the ideal point. At the end maximum absolute error, Root Mean Square Error (RMSE) and standard variation (σ) of the absolute error are presented.

x[m]	y[m]	X err [m]	Y err [m]	Abs err [m]
0	1	-0,1854	-0,0806	0,2022
0	2	-0,2034	-0,0258	0,2050
0	3	-0,1788	0,0503	0,1857
1	0	-0,0757	-0,0690	0,1024
1	1	-0,1473	-0,0511	0,1559
1	2	-0,1146	0,0037	0,1146
1	3	-0,1478	0,0887	0,1724
1	4	-0,1026	0,1574	0,1879
2	0	-0,0404	-0,0299	0,0503
2	1	-0,1060	-0,0288	0,1098
2	2	-0,0762	0,0113	0,0771
2	3	-0,0951	0,0557	0,1102
2	4	-0,0723	0,1583	0,1740
3	0	0,0239	0,0425	0,0488
3	1	-0,0517	-0,0116	0,0530
3	2	-0,0145	0,0491	0,0512
3	3	-0,0005	0,1111	0,1111
3	4	-0,0242	0,1583	0,1601
4	0	0,1023	0,0673	0,1224
4	1	0,0260	0,0047	0,0264
4	2	0,0429	0,0811	0,0918
4	3	0,0207	0,1424	0,1439
4	4	0,0309	0,1842	0,1868
5	0	0,1140	0,0860	0,1428

5	1	0,1087	0,1115	0,1557
5	2	0,1064	0,1314	0,1691
5	3	0,0697	0,1586	0,1732
5	4	0,0635	0,1692	0,1808
6	1	0,1034	0,0425	0,1118
6	2	0,1050	0,1560	0,1881
6	3	0,0774	0,0772	0,1093
6	4	0,0775	0,0942	0,1220
max_err[m]	0,2050		σ[m]	0,0506
RMSE [m]	0,0844			

Table 3.1: Static errors of the anchor constellation (a) figure 3.1.

The sign of the errors derives from the the fact that is calculated as *measured position - test point position*.
Following pictures are 3D representation of absolute error as difference between test point and average cluster position. The surface is an interpolation of test points with absolute error as height in the z axis direction.



(a) (b)

Figure 3.3: 3D surface error for constellation (a) figure 3.1. (a) top view, (b) side view.

Figure 3.3 shows the surface error, where x-y plane is the area covered by the

experiment, while in z direction the absolue value of the error is represented. The legend goes from 0 to 0.25 meters and colors change according to the gravity of the error.
Zone with lowest error, highlighted in blue, is the zone among the anchors. Error is greater outside this area.

3 anchors, second configuration results

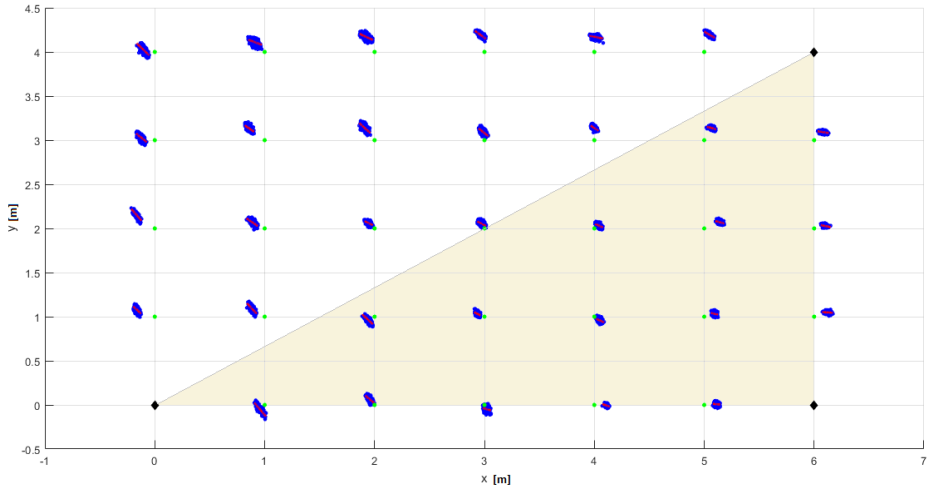


Figure 3.4: 2D plane and 3 anchors (second) configuration measurements.

In figure 3.4 is shown the second anchor constellation, where anchors form a rectangle triangle. Even in this case clusters of points are more accurate in the core of the highlighted area, so their average point is closer to the ideal point(green dots). Respect to the first configuration the clusters and error distributions are similar, the only difference is the orientation of the clusters. While in the first case they are oblique with a positive slope, in this case is the opposite.

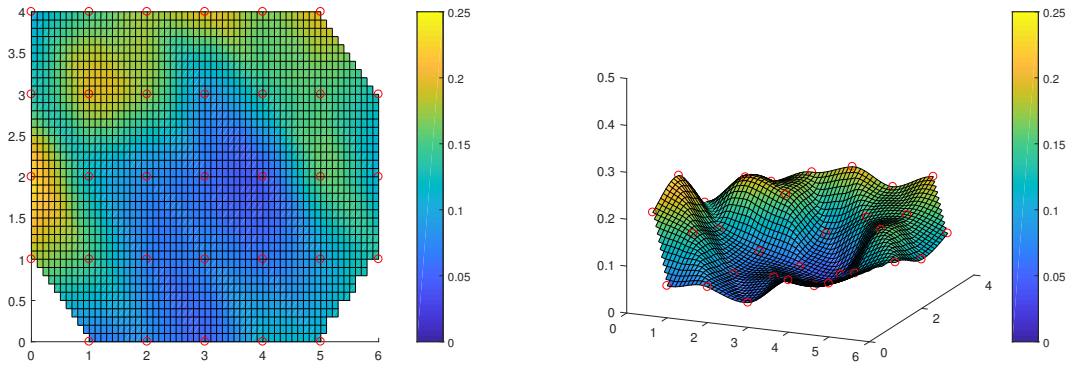
The following table shows the error on the x and y axis and the absolute error calculate as the euclidean distance between the average point of the cluster and the ideal point.

x[m]	y[m]	X err [m]	Y err [m]	Abs err [m]
0	1	-0,1618	0,0764	0,1789
0	2	-0,1637	0,1501	0,2221
0	3	-0,1253	0,0309	0,1291
0	4	-0,0992	0,0196	0,1011
1	0	-0,0391	-0,0556	0,0680
1	1	-0,1138	0,0881	0,1439
1	2	-0,1004	0,0587	0,1163
1	3	-0,1402	0,1330	0,1932
1	4	-0,0961	0,1129	0,1483
2	0	-0,0425	0,0627	0,0757
2	1	-0,0543	-0,0404	0,0677
2	2	-0,0532	0,0599	0,0801
2	3	-0,0905	0,1392	0,1660
2	4	-0,0671	0,1656	0,1786
3	0	0,0198	-0,0488	0,0526
3	1	-0,0635	0,0326	0,0714
3	2	-0,0249	0,0530	0,0585
3	3	-0,0116	0,0920	0,0927
3	4	-0,0394	0,1964	0,2003
4	0	0,1105	-0,0071	0,1107
4	1	0,0467	-0,0413	0,0623
4	2	0,0412	0,0315	0,0518
4	3	-0,0011	0,1382	0,1382
4	4	0,0230	0,1662	0,1678
5	0	0,1135	0,0080	0,1138
5	1	0,0944	0,0325	0,0998
5	2	0,1413	0,0699	0,1576
5	3	0,0712	0,1360	0,1535

5	4	0,0498	0,1933	0,1996
6	1	0,1246	0,0481	0,1335
6	2	0,1005	0,0272	0,1041
6	3	0,0852	0,0907	0,1244
max_err[m]	0,2221		σ[m]	0,0487
RMSE [m]	0,0893			

Table 3.2: Static errors of the anchor constellation (b) figure 3.1.

Error values are similar to the previos case. Both maximum error an RMSE have the same order of magnitude of the first arrangement.



(a) (b)

Figure 3.5: 3D surface error for constellation (b) figure 3.1.(a) top view, (b) side view.

Again the blue zone is the one among the anchors. The behavior of this constellation is similar to the first one considering the error distribution.

3 anchors, third configuration results

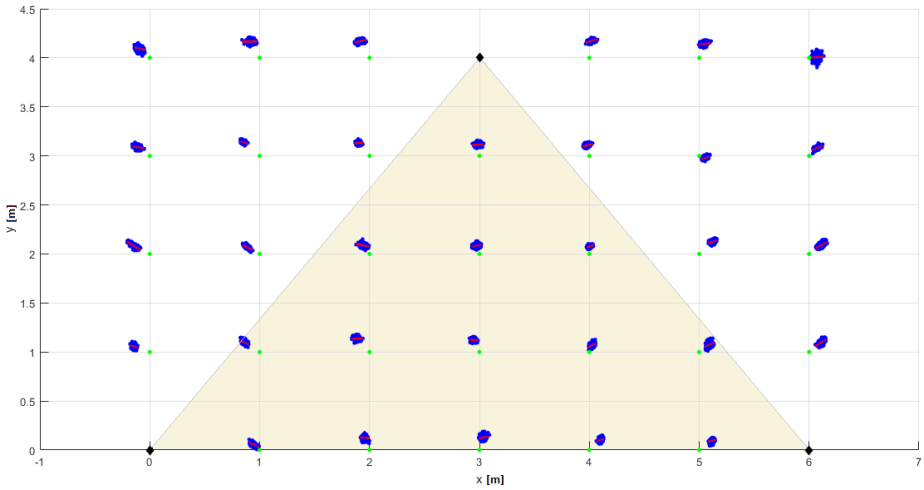


Figure 3.6: 2D plane and 3 anchors (third) configuration measurements.

In figure 3.6 is shown the third anchor constellation, where anchors form a isosceles triangle. Respect to the first two cases this time the configuration is very symmetrical and the cluster, especially in the center zone, are tight and round.

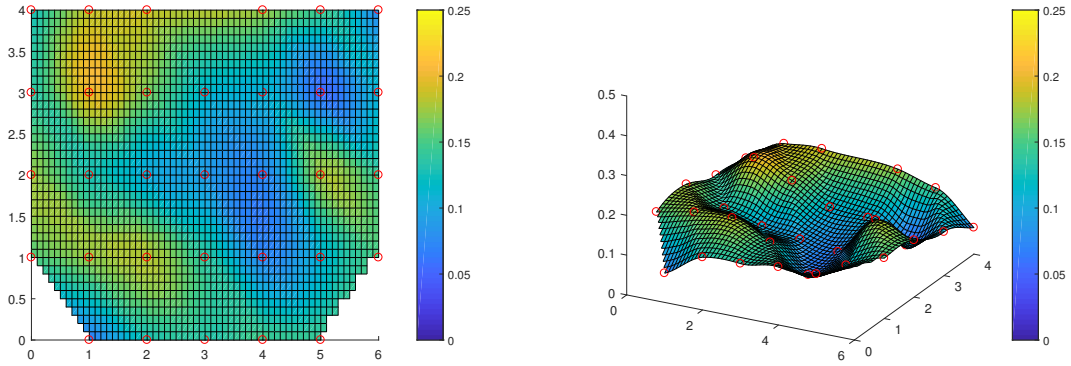
The following table shows the error on the x and y axis and the absolute error.

x[m]	y[m]	X err [m]	Y err [m]	Abs err [m]
0	1	-0,1441	0,0558	0,1545
0	2	-0,1483	0,0837	0,1703
0	3	-0,1084	0,0877	0,1394
0	4	-0,0887	0,0920	0,1278
1	0	-0,0509	0,0529	0,0734
1	1	-0,1409	0,0998	0,1726
1	2	-0,1105	0,0669	0,1292
1	3	-0,1467	0,1387	0,2019
1	4	-0,0756	0,1669	0,1832
2	0	-0,0441	0,1248	0,1323

2	1	-0,1129	0,1367	0,1773
2	2	-0,0539	0,0873	0,1026
2	3	-0,0968	0,1312	0,1631
2	4	-0,0835	0,1699	0,1893
3	0	0,0387	0,1296	0,1353
3	1	-0,0543	0,1234	0,1349
3	2	-0,0196	0,0865	0,0887
3	3	-0,0118	0,1145	0,1151
4	0	0,1022	0,1037	0,1456
4	1	0,0231	0,0678	0,0716
4	2	0,0045	0,0758	0,0759
4	3	-0,0244	0,1051	0,1079
4	4	0,0237	0,1740	0,1756
5	0	0,1124	0,0945	0,1468
5	1	0,0894	0,0722	0,1149
5	2	0,1229	0,1242	0,1747
5	3	0,0521	-0,0224	0,0567
5	4	0,0413	0,1416	0,1475
6	1	0,1150	0,0991	0,1518
6	2	0,1110	0,0902	0,1430
6	3	0,0732	0,0816	0,1097
6	4	0,0664	0,0054	0,0667
max_err[m]	0,2019		σ[m]	0,0389
RMSE [m]	0,0884			

Table 3.3: Static errors of the anchor constellation (c) figure 3.1.

The shape of the blue zone is quite triangular. Looking at the errors is possible to notice a certain symmetrical behavior respect to the 2D plane. this means that with a symmetric constellation the behavior of the error in the space is predictable.



(a) (b)

Figure 3.7: 3D surface error for constellation (c) figure 3.1. (a) top view, (b) side view.

4 anchors configuration results

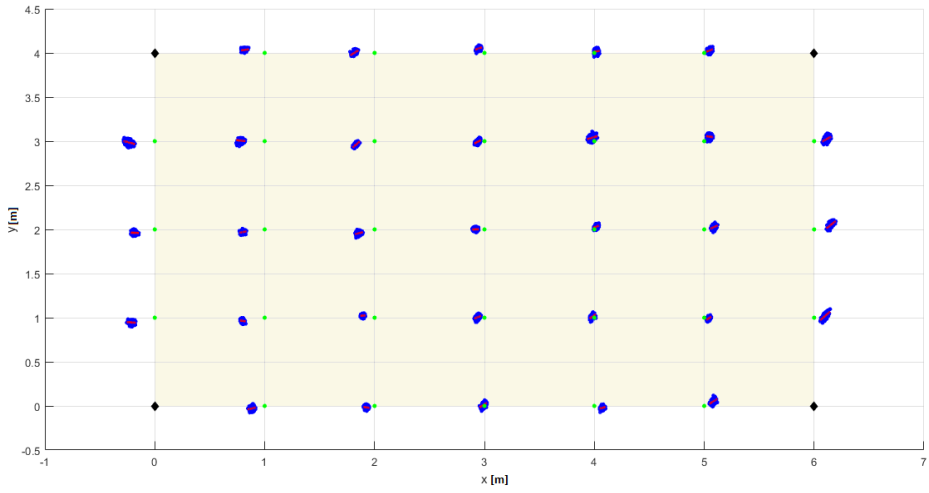


Figure 3.8: 2D plane and 4 anchors configuration measurements.

In figure 3.8 is shown the fourth anchor constellation, where anchors form a rectangle. Respect to the first three cases this time the configuration exploits 4 anchors in a symmetrical position. The fourth anchor placed in the top right corner is 1.64m tall, on a different plane rispect to the other anchors. This time clusters of points are

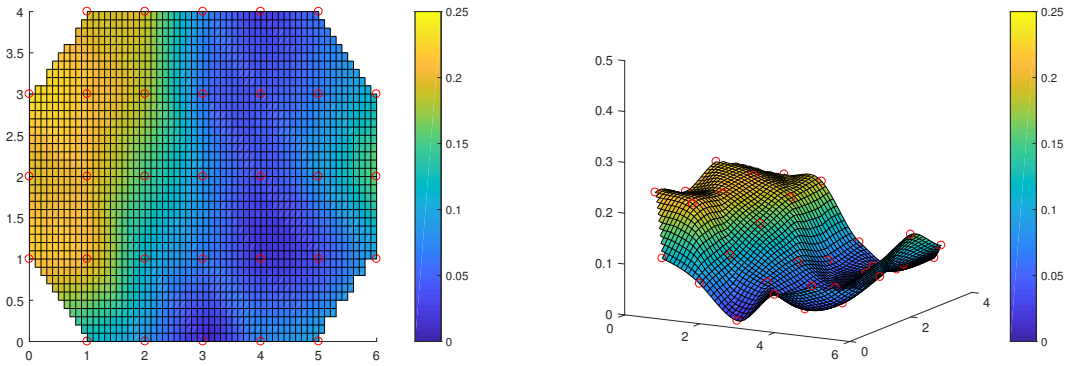
round and compact and the error reaches its minimum in the middle of the 2D plane

The following table shows the error on the x and y axis and the absolute error.

x[m]	y[m]	X err [m]	Y err [m]	Abs err [m]
0	1	-0,2097	-0,0544	0,2167
0	2	-0,1906	-0,0381	0,1943
0	3	-0,2282	-0,0173	0,2289
1	0	-0,1163	-0,0285	0,1198
1	1	-0,1996	-0,0369	0,2030
1	2	-0,1969	-0,0297	0,1991
1	3	-0,2100	0,0053	0,2101
1	4	-0,1856	0,0335	0,1887
2	0	-0,0769	-0,0160	0,0785
2	1	-0,1079	0,0226	0,1103
2	2	-0,1422	-0,0422	0,1484
2	3	-0,1706	-0,0413	0,1755
2	4	-0,1832	0,0016	0,1832
3	0	-0,0066	0,0128	0,0144
3	1	-0,0659	0,0030	0,0659
3	2	-0,0809	0,0007	0,0809
3	3	-0,0602	-0,0063	0,0606
3	4	-0,0512	0,0515	0,0727
4	0	0,0709	-0,0222	0,0743
4	1	-0,0176	0,0125	0,0216
4	2	0,0211	0,0334	0,0395
4	3	-0,0176	0,0400	0,0437
4	4	0,0204	0,0181	0,0273
5	0	0,0847	0,0521	0,0994
5	1	0,0411	-0,0074	0,0418
5	2	0,0873	0,0251	0,0909
5	3	0,0466	0,0518	0,0696

5	4	0,0517	0,0279	0,0588
6	1	0,1009	0,0194	0,1028
6	2	0,1538	0,0523	0,1625
6	3	0,1141	0,0236	0,1165
max_err[m]	0,2289		σ[m]	0,0660
RMSE [m]	0,0894			

Table 3.4: Static errors of the anchor constellation d figure 3.1.



(a) (b)

Figure 3.9: 3D surface error for constellation (d) figure 3.1. (a) top view, (b) side view.

With this configuration almost all the area among the anchors is blue, this means low error. This is probably the best configuration considering the compact clusters and the low error zone size.

Changing configuration doesn't change the maximum or the RMSE, but using symmetrical constellation clusters are more compact and the error has a symmetric and predictable behavior on the 2D plane. This important fact will be exploited in the fourth chapter.
In the following section the oscillatory behavior of clusters is analyzed.

3.1.2 Oscillation distribution respect to anchor's position in the space

The aim of this experiment is to analyse the oscillation of the tag position during static measures. In order to do so, the set up is the same of the previous experiment, a grid of 6 meters x 4 meters is built on the floor. The anchors are positioned at the vertices of this rectangular grid using 1,17 m tall tripods, while the tag is moved(on the floor) from measure to measure inside and on the side of this rectangle and on the vertices of the small square of 1m x 1m size forming the rectangle.

Four different anchor configurations have been investigated in order to figure out which is the best solution, analyzing oscillation distributions.

In order to figure out which kind of oscillation affects the tag, for each point of the grid, the 2D plane is divided in 9 zones. For each cluster of points obtained during a measurement in a test point, the average value is calculated, and placed in the center of the plane division. Then each measure of the cluster is compared respect to the average to determine if it is moved and in which direction.

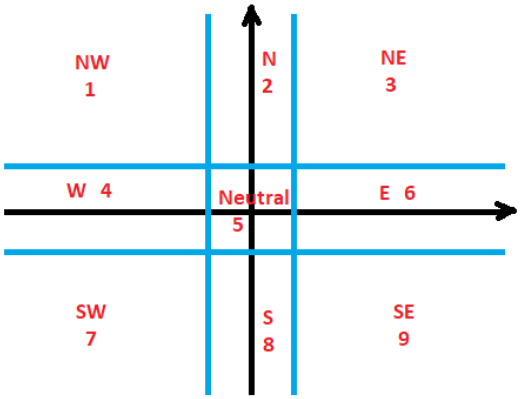


Figure 3.10: 2D plane division. The centre is represented by the average position of the measurement cluster. Blue lines represent limit boundaries to discriminate between different zones.

As it is possible to see from the picture above, some zones are greater than others, because the aim of this test is to investigate significant oscillations, so the small ones around the average point are not considered. Blue lines delimit zone boundaries, in

this case these lines are 1 cm far from the axis.
There are 5 kinds of oscillation:

- vertical;
- horizontal;
- positive slope;
- negative slope;
- neutral.

Vertical Vertical oscillation is when there is a vertical movement among subsequent measurements. According to the plane division a vertical oscillation occurs when the tag's position moves :

- from 2 to 8 or from 8 to 2;
- from 2 to 5 or from 5 to 2;
- from 8 to 5 or from 5 to 8;
- from 1 to 7 or from 7 to 1;
- from 3 to 9 or from 9 to 3.

Horizontal Horizontal oscillation is when there is an horizontal movement among subsequent measurements. According to the plane division a vertical oscillation occurs when the tag's position moves :

- from 4 to 6 or from 6 to 4;
- from 6 to 5 or from 5 to 6;
- from 4 to 5 or from 5 to 4;
- from 1 to 3 or from 3 to 1;
- from 7 to 9 or from 9 to 7.

Positive slope Positive slope oscillation is when there is an oblique movement among subsequent measurements with a positive *cosine* value of the angle obtained respect to the horizontal axis. According to the plane division a vertical oscillation occurs when the tag's position moves :

- from 3 to 7 or from 7 to 3;
- from 3 to 5 or from 5 to 3;
- from 7 to 5 or from 5 to 7.

Negative slope Negative slope oscillation is when there is an oblique movement among subsequent measurements with a negative *cosine* value of the angle obtained respect to the horizontal axis. According to the plane division a vertical oscillation occurs when the tag's position moves :

- from 1 to 9 or from 9 to 1;
- from 1 to 5 or from 5 to 1;
- from 9 to 5 or from 5 to 9.

Neutral When there are no movement of the tag's position, it is considered a neutral movement, so is not counted as an oscillation.

In order to evaluate the oscillations, each measurement has been scheduled in tables. Each row represent a point of the grid, and each type of oscillation is expressed in percentage.

3 anchors, first configuration results

From picture 3.2 is possible to see how clusters tend to be dense and compact in the highlighted zone, while outside (outside the zone covered by anchors) cluster

tend to elongate. The other key point is that cluster are closer to test points in the orange area. It means that the system accuracy is better in this zone. Cluster interpolations have a positive slope and tend to flat in the selected zone.

Oscillations are analysed in the following table:

Position	Vertical [%]	Horizontal [%]	Positive slope [%]	Negative slope [%]	Neutral [%]
(0,0)	anchor 0	anchor 0	anchor 0	anchor 0	anchor 0
(0,1)	1,72	3,45	1,03	0,00	93,80
(0,2)	1,35	7,07	1,68	0,34	89,56
(0,3)	1,63	4,58	2,61	0,00	91,18
(0,4)	anchor 1	anchor 1	anchor 1	anchor 1	anchor1
(1,0)	4,78	3,75	1,02	0,34	90,11
(1,1)	9,21	2,96	0,98	0,66	86,19
(1,2)	3,70	3,03	1,01	0,00	92,26
(1,3)	3,40	2,72	1,02	0,00	92,86
(1,4)	2,69	2,02	8,75	0,00	86,54
(2,0)	8,85	1,97	1,96	0,00	87,22
(2,1)	5,21	1,63	6,19	0,00	86,97
(2,2)	5,30	2,65	12,58	0,66	78,81
(2,3)	4,12	1,37	7,56	0,00	86,95
(2,4)	4,62	3,30	8,58	0,00	83,50
(3,0)	8,45	1,35	3,04	0,00	87,16
(3,1)	4,15	4,79	7,98	0,00	83,08
(3,2)	2,04	0,34	3,74	0,00	93,88
(3,3)	4,30	6,62	3,31	0,33	85,44
(3,4)	3,09	3,09	9,96	0,00	83,86
(4,0)	16,15	2,17	1,55	0,00	80,13
(4,1)	7,86	1,89	3,14	0,00	87,11
(4,2)	4,45	1,71	5,48	0,00	88,36
(4,3)	6,71	4,03	1,34	0,00	87,92
(4,4)	4,27	1,97	14,75	0,00	79,01
(5,0)	13,07	0,98	1,63	0,00	84,32

(5,1)	7,00	0,96	6,05	0,00	85,99
(5,2)	9,32	1,61	3,21	0,00	85,86
(5,3)	5,02	0,67	3,34	0,00	90,97
(5,4)	12,86	2,89	3,54	0,00	80,71
(6,0)	anchor 2	anchor 2	anchor 2	anchor 2	anchor 2
(6,1)	4,39	2,02	9,80	0,34	83,45
(6,2)	4,39	2,03	9,80	0,33	83,45
(6,3)	7,24	0,99	8,88	0,00	82,89
(6,4)	8,61	5,63	5,30	0,66	79,80

Table 3.5: Oscillation table referred to constellation (a) in figure 3.1.

In the table are reported the percentage of oscillations, for each row the highest percentage (excluded the neutral) is highlighted in red. Results show that oscillations in this configuration are most vertical or in the positive slope direction.

3 anchors, second configuration results

Once again, from figure 3.4, clusters are more focused in the highlighted zone than respect to the rest. In this case cluster interpolations have a negative slope, opposite behaviour respect to the first configuration, and again they tend to flat among the anchors.

Position	Vertical [%]	Horizontal [%]	Positive slope [%]	Negative slope [%]	Neutral [%]
(0,0)	anchor 0	anchor 0	anchor 0	anchor 0	anchor 0
(0,1)	11,92	0,66	0,00	2,32	85,10
(0,2)	6,98	0,33	0,00	11,30	81,39
(0,3)	5,24	2,10	0,00	7,69	84,97
(0,4)	6,80	2,59	0,00	13,92	76,69
(1,0)	0,92	0,31	0,00	16,00	82,77
(1,1)	9,06	2,35	0,00	7,05	81,54
(1,2)	5,61	0,99	0,33	9,24	83,83
(1,3)	7,57	0,99	0,00	4,28	87,16
(1,4)	8,53	8,53	0,00	4,78	78,16

(2,0)	6,43	0,96	0,00	5,15	87,46
(2,1)	7,28	1,98	0,33	11,26	79,15
(2,2)	3,69	4,03	0,34	2,35	89,59
(2,3)	7,90	3,78	0,34	8,59	79,39
(2,4)	7,67	5,75	1,28	6,39	78,91
(3,0)	8,33	3,67	1,00	1,00	86,00
(3,1)	8,55	0,00	0,00	0,33	91,12
(3,2)	5,28	5,61	0,66	7,26	81,19
(3,3)	3,95	2,96	0,00	11,51	81,58
(3,4)	2,88	0,64	0,00	11,22	85,26
(4,0)	5,79	0,32	0,00	0,96	92,93
(4,1)	5,48	1,94	0,00	3,23	89,35
(4,2)	5,56	2,94	0,00	1,96	89,54
(4,3)	3,99	0,66	0,00	4,32	91,03
(4,4)	5,03	5,03	1,34	2,68	85,92
(5,0)	5,23	5,88	0,98	1,96	85,95
(5,1)	14,77	1,68	1,34	0,34	81,87
(5,2)	6,00	2,67	0,33	1,00	90,00
(5,3)	2,94	2,29	0,65	1,31	92,81
(5,4)	3,98	0,92	0,00	5,50	89,60
(6,0)	anchor 1	anchor 1	anchor 1	anchor 1	anchor 1
(6,1)	6,05	5,10	0,64	0,96	87,25
(6,2)	4,22	7,47	0,32	0,65	87,34
(6,3)	2,96	4,28	0,33	0,33	92,10
(6,4)	anchor 2	anchor 2	anchor 2	anchor 2	anchor 2

Table 3.6: Oscillation table referred to constellation (b) in figure 3.1.

Oscillations in this configuration are directed in the vertical and negative slope direction.

3 anchors, third configuration results

From figure 3.6 is possible to see that clusters are more focused in the highlighted zone than respect to the rest. In this case cluster interpolations have an horizontal behaviour in the orange zone, while on the left they have a negative slope, the opposite of the right side. In respect to the first two configurations cluster are more focused and compact.

Position	Vertical [%]	Horizontal [%]	Positive slope [%]	Negative slope [%]	Neutral [%]
(0,0)	anchor 0	anchor 0	anchor 0	anchor 0	anchor 0
(0,1)	5,76	3,05	0,34	2,71	88,14
(0,2)	2,21	0,63	0,00	6,94	90,22
(0,3)	4,43	6,65	0,00	0,95	87,97
(0,4)	5,76	7,80	1,36	2,71	82,37
(1,0)	1,64	3,28	0,00	8,20	86,88
(1,1)	4,01	2,68	0,33	2,01	90,97
(1,2)	1,36	3,06	0,00	7,14	88,44
(1,3)	1,72	2,06	0,00	2,06	94,16
(1,4)	4,89	7,49	0,98	1,30	85,34
(2,0)	10,56	1,98	1,32	1,32	84,82
(2,1)	6,33	5,70	0,95	1,58	85,44
(2,2)	4,29	7,30	0,61	2,45	85,35
(2,3)	3,49	4,76	1,27	0,00	90,48
(2,4)	1,92	7,69	2,24	0,64	87,51
(3,0)	11,22	4,76	1,02	1,02	81,98
(3,1)	4,97	6,95	0,00	0,99	87,09
(3,2)	6,60	4,40	2,20	0,94	85,86
(3,3)	5,59	5,26	0,66	0,99	87,50
(3,4)	anchor 2	anchor 2	anchor 2	anchor 2	anchor 2
(4,0)	7,12	0,34	0,68	1,02	90,84
(4,1)	8,25	1,65	2,64	0,00	87,46
(4,2)	4,00	3,00	1,33	0,00	91,67
(4,3)	2,96	4,61	2,30	0,33	89,80

(4,4)	3,50	5,73	2,54	0,00	88,23
(5,0)	3,32	2,66	1,33	0,00	92,69
(5,1)	12,75	3,59	1,63	0,33	81,70
(5,2)	3,10	3,73	1,24	0,00	91,93
(5,3)	4,00	4,33	1,67	0,00	90,00
(5,4)	5,36	6,62	1,26	0,63	86,13
(6,0)	anchor 1	anchor 1	anchor 1	anchor 1	anchor 1
(6,1)	7,24	1,97	2,96	0,66	87,17
(6,2)	6,78	3,73	3,39	0,34	85,76
(6,3)	3,63	3,30	2,31	0,33	90,43
(6,4)	14,18	5,32	3,55	2,48	74,47

Table 3.7: Oscillation table referred to constellation (c) in figure 3.1.

Oscillations have an horizontal and vertical behaviour, this explain why cluster are so focused and they didn't extend in a oblique way.

4 anchors configuration results

In this configuration, as in the third, clusters are focused and close to test points. Cluster interpolation are most in the horizontal direction.

Position	Vertical [%]	Horizontal [%]	Positive slope [%]	Negative slope [%]	Neutral [%]
(0,0)	anchor 0	anchor 0	anchor 0	anchor 0	anchor 0
(0,1)	7,00	6,67	0,00	1,00	85,33
(0,2)	7,02	4,68	0,67	0,00	87,63
(0,3)	6,57	5,54	0,00	3,81	84,08
(0,4)	anchor 2	anchor 2	anchor 2	anchor 2	anchor 2
(1,0)	3,91	3,26	1,95	0,00	90,88
(1,1)	4,89	1,95	0,00	0,00	93,16
(1,2)	9,76	2,69	0,34	0,00	87,21
(1,3)	3,17	3,17	0,00	2,46	91,20
(1,4)	4,32	3,32	0,33	0,00	92,03

(2,0)	9,84	1,59	0,00	0,32	88,25
(2,1)	4,71	1,01	0,00	0,00	94,28
(2,2)	7,64	2,66	1,33	0,33	88,04
(2,3)	6,50	0,68	9,25	0,00	83,57
(2,4)	5,14	2,05	5,14	0,00	87,67
(3,0)	8,67	2,48	2,78	0,00	86,07
(3,1)	6,02	3,01	0,67	0,00	90,30
(3,2)	6,67	2,00	0,00	0,00	91,33
(3,3)	6,25	2,78	1,39	0,00	89,58
(3,4)	5,41	1,35	1,35	0,34	91,55
(4,0)	5,08	2,03	0,00	0,34	92,55
(4,1)	8,79	0,65	0,32	0,00	90,24
(4,2)	5,25	1,64	0,00	0,00	93,11
(4,3)	5,61	4,62	4,29	1,32	84,16
(4,4)	4,97	1,98	0,33	0,99	91,73
(5,0)	8,11	1,48	1,10	0,00	89,31
(5,1)	5,48	0,00	0,97	0,00	93,55
(5,2)	8,31	1,99	2,32	0,00	87,38
(5,3)	8,97	1,66	0,00	2,33	87,04
(5,4)	6,91	0,66	1,97	0,33	90,13
(6,0)	anchor 1	anchor 1	anchor 1	anchor 1	anchor 1
(6,1)	5,90	1,96	3,61	0,33	88,20
(6,2)	5,30	1,99	2,98	0,00	89,73
(6,3)	9,33	4,33	1,33	0,00	85,01
(6,4)	anchor 3	anchor 3	anchor 3	anchor 3	anchor 3

Table 3.8: Oscillation table referred to constellation (d) in figure 3.1.

In this configuration oscillation are almost completely in the vertical direction. This allows cluster to be compact.

All the configurations are characterized by a blue low error area located in the

middle of the grid. This means that for the first two configurations this area is asymmetric respect to the anchors distribution.
Among the 4 configurations tested in this experiment, the last two are better respect to the rest from the cluster concentration point of view. 3 anchors, third configuration has focused cluster, and his low error zone is placed at the centre of the area covered by anchors, while in the first two configurations this area is asymmetric. Furthermore oblique oscillations are drastically reduced.
4 anchors configuration is most likely the most precise according to its low error zone size and to the fact that the oscillation are only in the vertical direction allowing clusters to be more focused.

3.2 Dynamic error

The study of the dynamic error is related to the ability of the TREK1000 develpment kit to track a moving tag. Since the final application has to track a robot and a patient is foundamental to prove that this system is able to track till a certain speed. The test concern the use of a pendulum with variable length. The tag attached to the end of the pendulum is left free to oscillate and the speed of the movement is varied changing the angle of oscillation.
Since the Home Robot can travel at maximum speed of 0.5 $\frac{m}{s}$, a very low speed, the limit is represented by the patien's speed. A persor walks with an average speed that varies from 1 $\frac{m}{s}$ to 1.5 $\frac{m}{s}$, so this is the upper limit that the system must reach.

3.2.1 Pendulum

The movement of the pendulum is periodic and deterministic. The choice to use a pendulum is explained in the fact that this kind of movement emulates the one done by a person that is walking.
The set up for this experiment is composed by the 4 EVB1000 of the development kit and the pendulum represent by a tiny rope 2.24 m long attached to the roof. The anchors are positioned in:

- anchor 0 (0.0m; 0.0m, 0.0m);

- anchor 1 (1.80m; 0.0m, 0.0m);
- anchor 2 (0.0m; 3.0m, 0.0m);
- tag in resting position (0.9m; 1.50m, 0.76m)

The following chapter presents some measurements done, pendulum oscillation in time and the Root Mean Square Error (RMSE) of the oscillation peaks. Four measurements are performed keeping constant the oscillation period and varying the amplitude of the oscillation, in this way the average speed changes.

The damping factor is evaluated using an exponential interpolation of the oscillation peaks as time constant of the function:

$$d \cdot exp^{(\frac{-t}{\tau})} \tag{3.1}$$

where τ is the damping factor.
The table is composed of:

- L represents the lenght of the pendulum;
- T is the period;
- d is the oscillation amplitude;
- v is the average speed estimated like $T \cdot 1/2 \cdot 1/d$;
- θ the angle between the pendulum and the vertical axis at the moment of the release of the pendulum;
- $\frac{sin\theta}{\theta}$ index of the validity of the pendulum model;
- τ damping time constant.

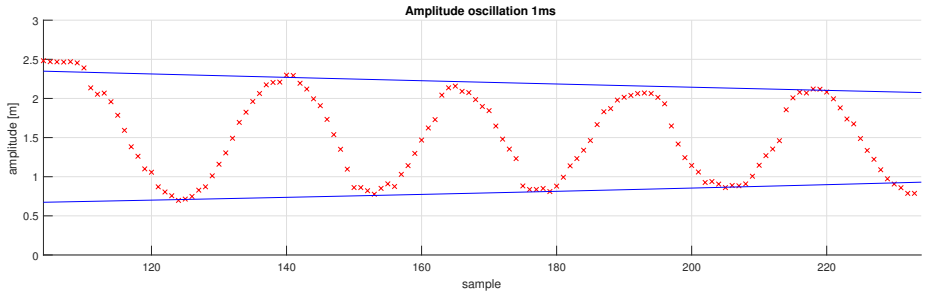
L [m]	T [s]	d [m]	v [$\frac{m}{s}$]	θ [°]	$\frac{sin\theta}{\theta}$	τ [s]
2.24	3	0.75	1	18.51	0.983	54.1
2.24	3	1	1.333	24.06	0.971	43.8
2.24	3	1.10	1.467	26.15	0.966	51.2
2.24	3	1.20	1.60	28.18	0.960	41.7

Model limits The limits of this measurements are due to the model and the pendulum movement.

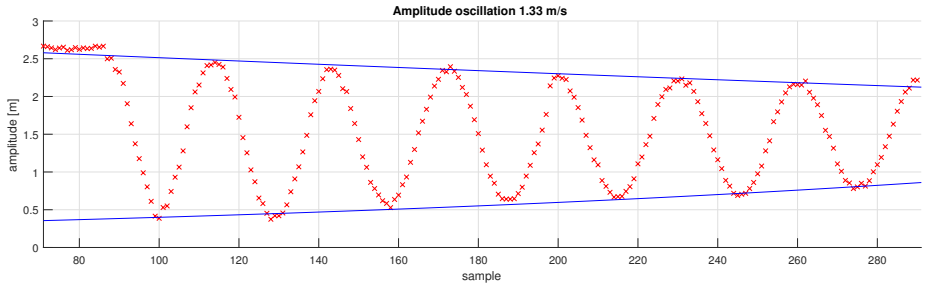
In order to the model to be valid, the ratio $\frac{sin\theta}{\theta}$ must be as much as possible equal to one, this limits the ascillation angles and so the amplitude that means low speed. The other limit is represented by the movement done by the pendulum and on the fact that the speed of it depends on the angle. The speed is maximum in the middle of the oscillation, when the pendulum converts all the potential energy in kinetic energy, while it is null at peaks. This is an advantage for the TREK1000 since it can better sample the peaks where the pendulum first decelerate, then stops and then accelerate again, while between one peak and the other the number of sample per unit length is lower.
Last limit is represented by the damping factor that reduces the amplitude of the oscillations in time.

In figure 3.11 are depicted different amplitude oscillation. Number of samples on the x axis, while amplitude in meters is placed on y axis. Each red cross reprsents one sample, while the blue lines are the exponential interpolations of the peak values from which the damping factor is obtained. The amplitude in time gives that sinusoidal shape represented in the picture. No filters or any kind of data eleboration is applied to the amplitude values, they are exactly as the system sampled them.

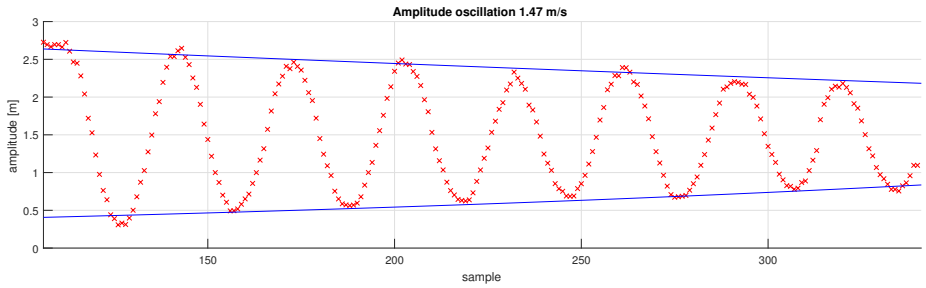
This experiment proves that the TREK1000 is able to track a moving tag from 1 $\frac{m}{s}$ to 1.6 $\frac{m}{s}$ respect to all limit presented by the model. Considering that iRobot has a maximum velocity of 0.5 $\frac{m}{s}$ and the patient in the apartment cannot move around at a speed greater than 1.5 $\frac{m}{s}$, this system is suitable for the final application.



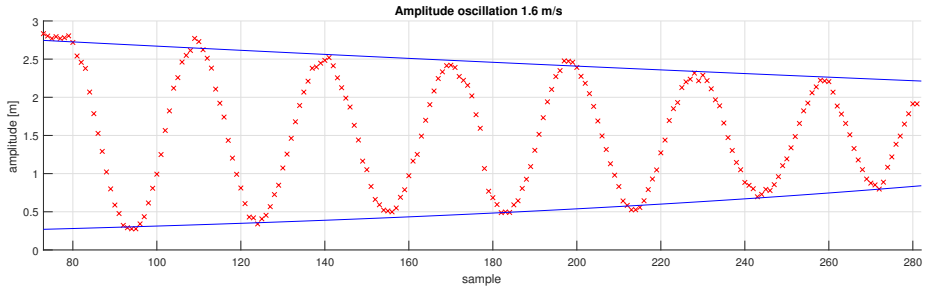
(a)



(b)



(c)



(d)

Figure 3.11: Amplitude oscillation for different release angles.

Chapter 4

Positioning Algorithm

This section is dedicated to the description of the positioning algorithm developed using MatLab environment. All the parts that compose the final software are explained step by step presenting the problem and the relative solution.

4.1 Trilateration algorithm

Analyzing the trilateration algorithm implemented by Decawave and explained in chapter II, section "Positioning algorithm", is possible to identify the limits that affect its performance. In fact this code exploits only three anchors for the trilateration process, while the fourth, when used, selects the right solution among the two obtained by the spheres intersection. Furthermore the first three anchors must be positioned at the same height, on the same z plane.

These constrains are limits since don't allow the code to use more than 3 anchors for the trilateration, so no redundant information, and intodruce a physical constrain on the anchor's arrangement. Since the final application must be used in apartment, it would be better to have a certain freedom degree on the anchor's placement. This means having a system that the customer can place in his home environment without the help of technicians. This include also a calibration process that will be discussed in chapter VI.
In order to solve this problem a new positioning algorithm is used. Many different types of algorithm are used for positioning problem [9], [10], [3]. After consulting many papers and different types of algorithm a closed form one has been chosen.
Since it's important to mantain the same performance in terms of sampling frequency, the use of a closed form solution is preferred respect to a iterative one as

proposed in [17]. In this way the position can be identified in just one code cycle. This positioning algorithm exploits a statistic approach by formulating an error function and finding the minimum of this one.

4.1.1 Closed-form Localization from Range-difference measurements

N denotes the number of anchors, while d_{ij} denote the range difference (RD) between anchors i and j with i and j varying from 1 to N. The positioning vector(x; y; z) of the i-th anchor is denoted as x_i and the unknown position of the tag is denotes as x_s.
The distance between the i-th anchor and the tag is $D_i = ||x_i - x_s||$ and the distance from the origin of the cartesian axis to the position of the i-th anchor is denoted as R_i.
The range differences are denoted as:

$$d_{ij} = D_i - D_j \tag{4.1}$$

where i = 1, . . . , N and j = 1, . . . , N.
The localization problem is to find the tag position x_s given d_{ij}. In total there are :

$$\frac{N(N-1)}{2} \tag{4.2}$$

distinc range differences. The formula above calculate all the possible combination of rage difference combining anchors excluding i=j and each equal pair d_{ij} = -d_{ij}.
The position of the tag is estimated as the one that best fits the RD measurements. It is found in closed form considering the tag's position as the minimizer of the error equation, so the minimizer of the difference between functions of RDs and function of the hypothesized source location.

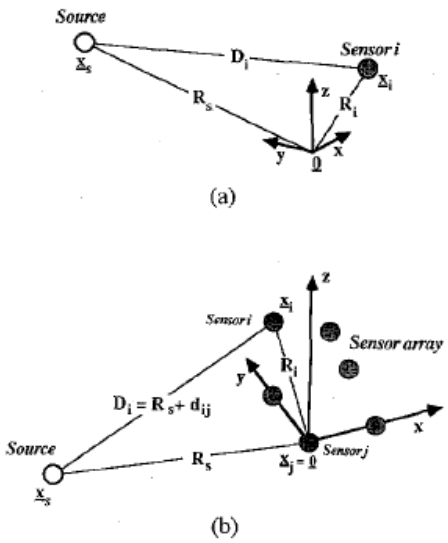


Figure 4.1: 3D scheme of the geometric distances between anchors (sensors) and tag (source).

First step is to map the origin of the cartesian space as the position of one anchor.

$$x_j = 0 \rightarrow \begin{array}{c} R_j = 0 \\ D_j = R_s \end{array} \tag{4.3}$$

From Pythagorean theorem and using 4.3 in 4.1

$$(R_s + d_{ij})^2 = R_i^2 - 2x_i^T x_s + R_s^2 \tag{4.4}$$

and

$$0 = R_i^2 - d_{ij}^2 - 2R_s d_{ij} - 2x_i^T x_s \tag{4.5}$$

Considering equation 4.5 involving the N - 1 range differences relative to anchor j, N - 1 equations in the three unknowns are obtained for the position x_s.
The delays affecting the range measurements is not null and is not negligible, so the so called error equation is introduced on the lefthand of eq. 4.5. Then is possible to obtain a position estimation of x_s minimizing the equation using a least square approach.

Let be j = 1. Then 4.5 is:

$$\epsilon_i = R_i^2 - d_{ij}^2 - 2R_s d_{ij} - 2x_i^T x_s \quad i = 2,3,\ldots,N \tag{4.6}$$

where ϵ_i is the error equation to be minimized. Using the matrix notation:

$$\epsilon = \delta - 2R_s d - 2Sx_s \tag{4.7}$$

where

$$\delta = \begin{bmatrix} R_2^2 - d_{21}^2 \\ R_3^2 - d_{31}^2 \\ \vdots \\ R_N^2 - d_{N1}^2 \end{bmatrix} \quad d = \begin{bmatrix} d_{21} \\ d_{31} \\ \vdots \\ d_{N1} \end{bmatrix} \quad S = \begin{bmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{bmatrix} \tag{4.8}$$

Error equation 4.10 is linear respect to x_s and to R_s. The closed form solution is:

$$x_s = \frac{1}{2} S_w^* (\delta - 2R_s d) \tag{4.9}$$

where

$$S_w^* = (S^T S)^{-1} S^T \tag{4.10}$$

So this algorithm use one anchor as reference point, placing the origin of the axis in its position. Every range measurement is transformed in a range difference respect to the reference anchor, then an error equation is formulated considering the fact that measurements are affected by error. Minimizing this equation an estimation of the tag's location is obtained. The validity of this algorithm is proved in [12].
The advantages of using this approach is that now the localization process can exploit multiple anchors in a redundant way and the anchors can be positioned even at different height.

4.2 Corrective function

Looking at the static measurements presented in chapter III, is possible to notice that the behavior of the anchors is affected by errors. Figure shows the static measurements obtained on a surface of 6m x 4m using 4 anchors and the positioning algorithm presented in the previous section. Anchors positioned in:

- anchor 0 (0.0m; 0.0m, 0.09m);
- anchor 1 (6.0m; 0.0m, 0.09m);
- anchor 2 (0.0m; 4.0m, 0.09m);
- anchor 3 (6.0m; 4.0m, 0.09m);
- tag placed on a tripod 1.64m tall.

The tag is moved on each green dot that represents a test point, while the blue point are clusters of measurements obtained for each test point.

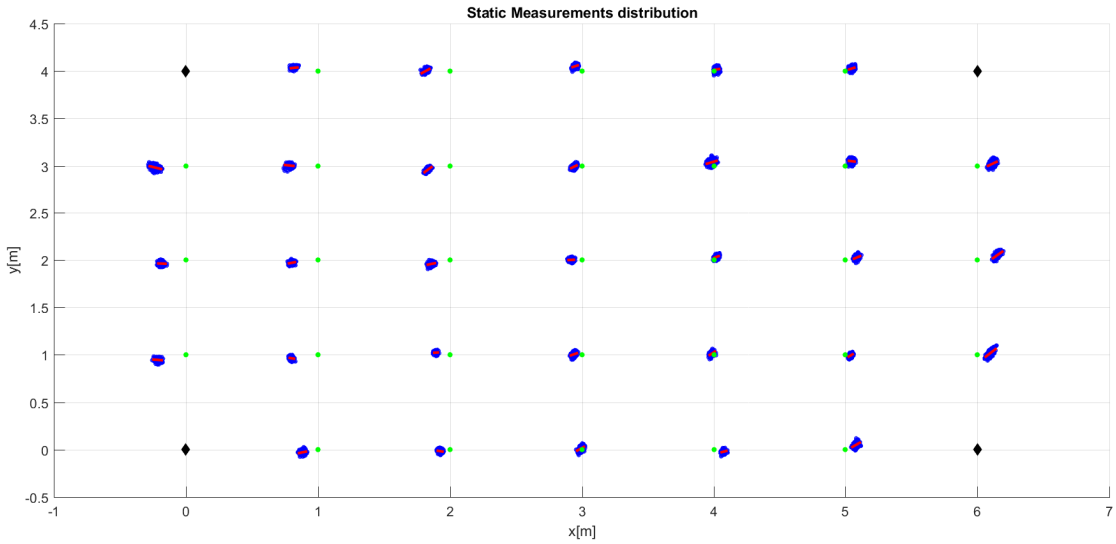


Figure 4.2: Static measurements obtained using 4 anchors and the closed-form algorithm

Taking in consideration one random anchors is possible to notice a cerrtain behavior of the measurement clusters. For example let's consider anchor 0, positioned

in (0.0, 0.0) in the figure, as reference node, how is possible to see the clusters near this node are underestimated respect to the relative test points, while distant clusters are overestimated. This problem is related to the fact that ranges coming from anchors tend to be underestimated for values smaller than 2m, while are overestimted for values greater than 2m. Ranges around 2 m are accurate respect to the real value.
In order to compensate this error a corrective function is built.

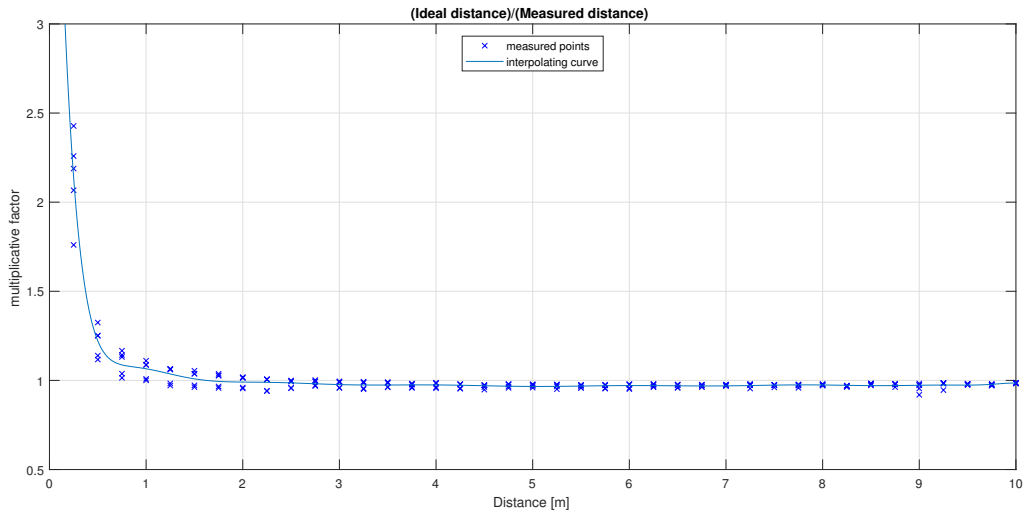


Figure 4.3: Corrective function.

Figure 4.3 shows the corrective function. This function corrects the range value obtained as a measure from the Decawave TWR process. It is basically a multiplicative factor that multiplies the range correcting its value. Since the range is underestimated for values smaller than 2m and viceversa for values greater than 2m, the multiplicative factor is greater than 1 before 2m and smaller than 1 over 2m. This means that the corrective function is function of the only range value, so it is a one dimensional function.

$$range' = range \cdot f(range) \tag{4.11}$$

In order to build this function an anchor and a tag is used. Placing the anchor in a fixed point and moving the tag by steps of 25cm far from the anchor on a straight

line, range values have been recorded using Log File. This process is done starting 25cm far from the anchor till 10m far, for different orientation of the antennas. Four different antenna configuration have been used. First the antennas in a parallel configuration, each antenna directed toward the other. Then the antenna of the tag is rotated of 90 degrees in order to be perpendicular to the other. Third configuration with both the antenna rotated of 90 degrees, and finally the tag's antenna directed toward the anchor and the anchor's one rotated of 90 degree.
For each configuration a several measurements have been recorded. In figure 4.3 for each step, 4 blue crosses are reported, one for each antenna configuration. Each blue cross represents the average value of a cluster of points obtained for that step.

The function is obtained dividing the ideal (real) distance values by the measured ones, and then interpolating those points. It is one dimensional since it only needs range value as input and gives the multiplicative factor as output. Multiplying the latter for the initial rage value, the corrected one is obtained.

Results and performance of this operation are presented in the next chapter.

4.3 Double system: 8 anchors

In order to increase the redundant information and try to increase the accuracy of the positioning process another TREK1000 development kit is used for a total of 8 anchors. The use of more than 4 anchors allows to go deeper into the study of the localization problem and permit to develop new techniques able to mitigate errors due to problems that will be explained in the following sections such as the No Line Of Sight (NLOS) one.

Since the system is designed to allocate a maximum of 4 anchors and also the TWR algorithm allows only this number of nodes a trick has been used. The system has two working channels, channel 2 around the frequency of 3.6 GHz and channel 5 of frequency 6.5 GHz. The trick consists in using 4 anchors working on channel 2 and the other 4 on channel 5, while the tag switches between these two in order to collect measurements and save them in Log File format. Then using a MatLab script the

measurements are analyzed and using the closed-form localization seen above, the position of the tag is estimated.

For the following experiments the anchors are distributed in a symmetric way, forming an octagon (considering the 2D plane). From now on all the techniques and methods developed are applied to a system with redundant information, in paticular with 8 anchors.

4.4 Best anchors filter

As explained in chapter I, "positioning process" section, the spheres intersection produces an uncertainty area in which the tag is situated. Using 8 anchors and so 8 range values for the localization problem could lead to a better accuracy, but in some cases a range value could expand the uncertainty area. This could be caused by the non ideal position of the tag respect to the tag position, or a peak of noise component during the measure. In order to tight this uncertainty area a simple algorithm has been developed.

The approach used is iterative. It consists in non considering one of the range value and use the others to obtain the tag position. Once estimated the latter, errors are calculated as difference between the euclidean distance between the tag's position and the anchors' one and the range value. This error is calculated for each node and the process is repeated neglecting one anchor, and one range value, at turn. Once identified the range measurement responsible of the bigger error, this one is eliminated from the positioning algorithm.
This approach is repeated until the maximum error is below a certain static threshold, or the number of remaining range values is 4 (the minimum numbe of anchors for 2D positioning plus one redundant anchor).

Even if the positioning algorithm is statistical one beased on error equation, so not based on geometrical and spheres intersection, this approach to select the

best range values is still valid. In fact instead of tightening the uncertainty area it minimize the error equation neglecting terms that are affected by a greater error respect to the others.

4.5 Offset correction

Like every positioning algorithm, even the "Closed-form Localization from Range-difference measurements" one introduces some bias and variance in the tag position. This error derives from linearization and approximation operation that simplifies the mathematical model, but are an approximated representation of the reality.

The variance is mitigated thanks to a median operation performed on the tag position results. The bias instead, is related to which anchor is used as the refence node in the algorithm. So in order to mitigate this error every anchor is used as reference one and then the tag position is obtained by means of an average operation. In this way the bias factor is compensated and redistributed in the 2D plane.
Results and performance are shown in the next chapter.

4.6 NLOS correction

No Line Of Sight (NLOS) is one of the major problem for localization system. If the direct path between an anchor and the moving tag is obstructed from an object, obstacle or even a person, the ToA of the signal is delayed, which means that a positive bias is introduced. The first step in NLOS mitigation is identify the presence of this problem in the measurements obtained, then try to mitigate it.
In general NLOS identification is done by means of probabilistic techniques [18], working with probability density function, noise variance and static thresholds. One of the aim of this thesis is to keep a simple approach to the positioning problem, so the probabilistic way is discarded and a more simpler method is preferred.

As discussed above, NLOS introduces a bias factor in the range value, so this means that there is a variation in the range measure when an obstacle moves between the

achor and the tag. The first operation done by the algorithm is a derivative operation on the range value. In this way when the derivative value overcome a static threshold the algorithm understands that there is presence of NLOS.
The static threshold must hold in consideration that both the robot and the patient can move inside the home environment, so must be able to discriminate between the variation due to a NLOS and the variation due to the movement of the tag.
Since the robot moves at a maximum speed of 0.5 $\frac{m}{s}$, and the system sampling frequency is 10 Hz, the static threshold is :

$$max - robot - variation = 0.5\frac{m}{s} \cdot \frac{1}{10}s = 0.05m \tag{4.12}$$

The same for the patient:

$$max - patient - variation = 1.5\frac{m}{s} \cdot \frac{1}{10}s = 0.15m \tag{4.13}$$

However considering the variance of the tag position and the error due to the TREK1000 non synchronization, a factor of 10cm must be added to these threshold. In this way the algorithm doesn't recognize the error introduced by the system as a NLOS one. So the thresold for the robot is 15cm, while the one for the patient is 25cm. Adding 10 cm on the thresholds is not a problem since the variation due to metallic object or walls is in the order of meters.

The derivative method applied above allows the algorithm to identify variation in the range measure that are not due to the tag movement, but to some object that is situated between the tag and the anchor LOS. Thanks to the redundant information coming from the eight anchors, the number of ranges for the trilateration is higher than the minimum required, so ranges affected by NLOS are discarded from the positioning computation.

If we consider the case in which an obstacle moves between the tag and an anchor and then stay still, the derivative is not able to identify it anymore. In fact the derivative approach is valid when the range suffers a change. Let's consider a wall, the robot moving in the apartment can change room and an anchor that before was in LOS, now is subjected to NLOS. The derivative notices this only in the moment

of the change of room, but then, since the wall is still, no variations occur.

After the derivative processs the NLOS mitigation algorithm performs a check on the consistency of the range measurements. Using the position obtained during the previous measurement session, the code produces an estimation of the future range values. Once obtained the new measurements, these are compared with the estimated one, and the range values that are not consistent are filtered from the positioning process.
The estimation is obtained using the previous position and the known location of the anchors. The difference of these two gives a range value that then is compared to the new one, and if their difference is over a certain threshold, like in the derivative case, they are discarded. Even in this case the threshold hold in consideration the possibility that the tag is moving.

4.7 Algorithm overall

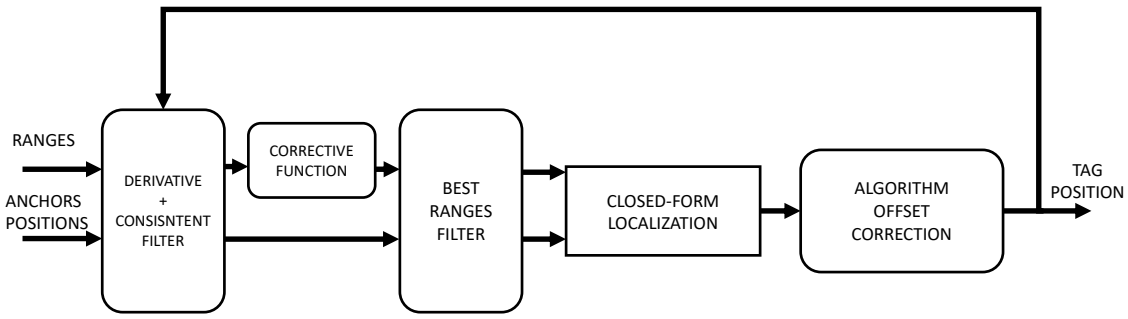


Figure 4.4: Algorithm overall.

In figure 4.4 is depicted a scheme relative to the developed algorithm. On the left are present the inputs of the algorithm that are the measured ranges, and the position of the anchors. The first stage is represented by NLOS filter that performes the above explained derivative and consistent filter. This stage receives as input also the previous tag position that is represented by the arrow that goes from the output to this stage.
The second stage is the corrective function that operates only on the ranges since depends only on them. The third stage is a best ranges filter, that discards range values affected by errors considered too big respect to a treshold.
The central stage is represented by the closed-form positioning process that identifies the tag positiong taking as input the value of non discarded ranges and the relative anchor positions. The last step consists in correcting the offset introduced by the localization algorithm.
The output is the tag position.

The algorithm has been traslated from Matlab to a java library and then implemented on the existent robot software described in chapter II.

Chapter 5

Test and Performance

In this chapter tests performed on the algorithm exposed previously are presented. All the tests were run in the main lab of the Motion Analysis Lab (MAL) at the Spaulding Rehabilitation Hospistal in Boston, Massachussets.
The log Files recorded by means of the Decawave system are analyzed using MatLab scripts.

Let's start from a system composed by 4 anchors and one tag. Anchors positioned in:

- anchor 0 (0.0m; 0.0m, 0.09m);
- anchor 1 (6.0m; 0.0m, 0.09m);
- anchor 2 (0.0m; 4.0m, 0.09m);
- anchor 3 (6.0m; 4.0m, 0.09m);
- tag placed on a tripod 1.64m tall.

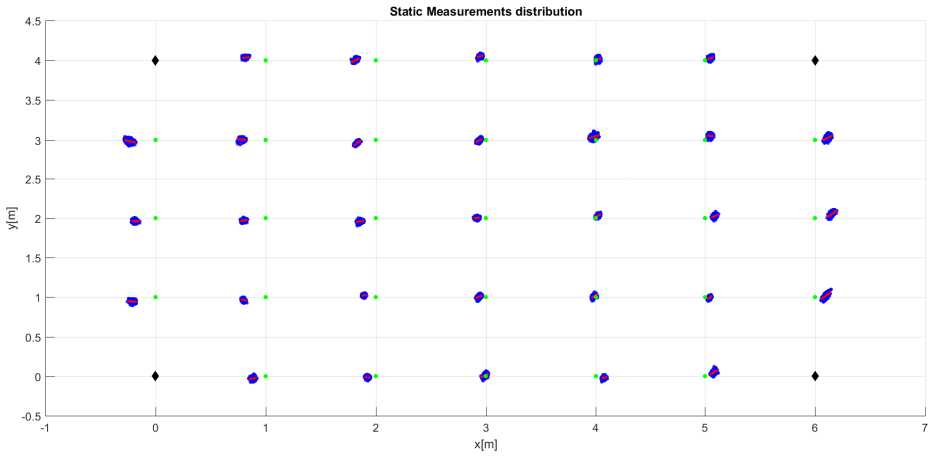


Figure 5.1: Static measurements obtained using 4 anchors and the closed-form algorithm

Again the black diamond represent the anchors, the green dots the test points and the blue cluster are the measured point. Figure 5.1 shows the starting point.

No correction				
x[m]	y[m]	X err [m]	Y err [m]	Abs err [m]
0	1	-0,1271	-0,1899	0,2285
0	2	-0,1835	-0,0153	0,1841
0	3	-0,1836	0,0388	0,1876
1	0	-0,1200	-0,2121	0,2437
1	1	-0,1437	-0,1514	0,2087
1	2	-0,1734	-0,0569	0,1825
1	3	-0,1567	0,0437	0,1627
1	4	-0,1985	-0,0280	0,2005
2	0	-0,0626	-0,1488	0,1615
2	1	-0,0526	-0,1069	0,1191
2	2	-0,1083	-0,0420	0,1161
2	3	-0,1134	-0,0036	0,1135
2	4	-0,0688	0,0794	0,1051
3	0	0,0038	-0,1322	0,1323
3	1	0,0275	-0,1071	0,1106

3	2	0,0022	-0,0272	0,0273
3	3	0,0014	0,0138	0,0139
3	4	-0,0054	0,0568	0,0570
4	0	0,1274	-0,1059	0,1657
4	1	0,0971	-0,0795	0,1255
4	2	0,1002	-0,0036	0,1003
4	3	0,0843	0,0187	0,0864
4	4	0,0934	0,0317	0,0987
5	0	0,1598	-0,1508	0,2198
5	1	0,1901	-0,0772	0,2052
5	2	0,2177	-0,0025	0,2177
5	3	0,1645	0,0369	0,1686
5	4	0,2086	-0,1187	0,2400
6	1	0,2222	-0,0345	0,2249
6	2	0,2291	-0,0201	0,2300
6	3	0,3514	-0,2880	0,4544
max_err [m]	0,4544		σ[m]	0,0824
RMSE [m]	0,1832			

Table 5.1: Static measurement error table.

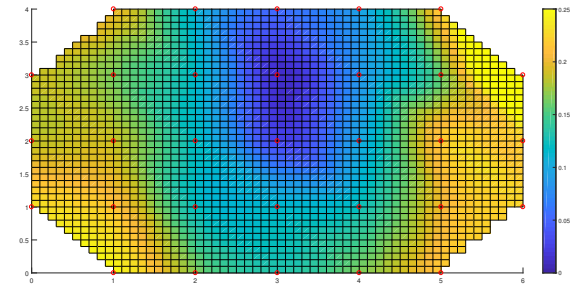
Table 5.1 shows the static error obtained without any correction function or filtering. This represents the starting point of the positioning system. At the end of the table maximum error, Roor Mean Squared Error and variance are reported. Colors change according to the magnitude and sign of the error as depicted in the following table.

Legend	
color	limit [m]
cyan	-0.10≤x<-0.05
blue	X<-0.10
yellow	0.05<x≤0.10
red	x>0.10

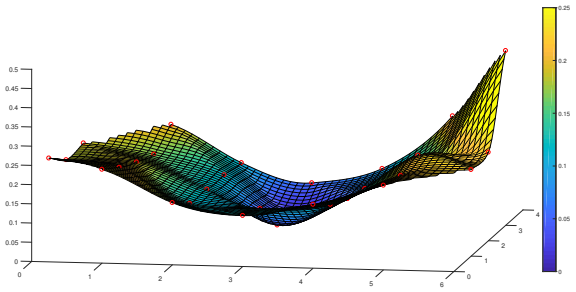
green	-0.05≤x≤0.05

Table 5.2: Color legend relative to static error.

Errors from -0.05 to 0.05 meters are green cause considering the final application, this magnitude is acceptable. In fact since the scope is to track a person and a robot in a home environment, high accuracy under 5 cm is not required. Errors greater than 5 cm but smaller than 10 cm in magnitude are still accepted, but they are borderline, while is better to avoid errors greater than 10 cm. The reason is that knowing the position of the center of the robot with an uncertainty greater than 10 cm in magnitude could be a problem for the algorithm present on the bot which duty is to draw the path from the actual position to the patient. Considering that in a home environment many are the possible obstacles, an uncertainty of the order of 15-20cm could drive the robot to collide with them.



(a)



(b)

Figure 5.2: 3D error surface for no error correction. (a) top view, (b) side view

Figure 5.2 is a 3D representation of the absolute error. x-y plane shows the position of the test points, while the absolute error is in the z direction. Each error related to the relative test point is then interpolated forming a surface.
As it is possible to see the error is higher near the border of the area covered by the anchors. The explanation is that the ranges are affected by underestimanting/overestimating error, and in the center area, due to the simmetry of the anchors position, these errors compensate. Near the border istead these error tend to add each other.

5.1 Corrective function

This section shows the results obtained after the use of the corrective funtion. The latter is one dimensional function that depends only on the range value. It operates as a multiplicative factor that modifies the range value, correcting the underestimanting/overestimating effect presented in the previous chapter and noticeable in figure 4.2

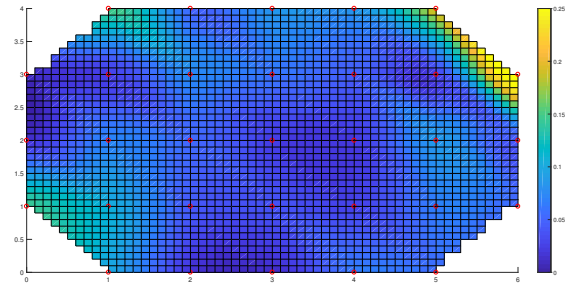
Correction				
0	1	0,0442	-0,1340	0,1412
0	2	-0,0056	-0,0141	0,0152
0	3	-0,0049	-0,0098	0,0110
1	0	0,0011	-0,0944	0,0944
1	1	-0,0141	-0,0990	0,1000
1	2	-0,0415	-0,0539	0,0681
1	3	-0,0281	-0,0025	0,0283
1	4	-0,0787	-0,1324	0,1540
2	0	-0,0001	-0,0173	0,0173
2	1	0,0150	-0,0398	0,0426
2	2	-0,0328	-0,0400	0,0517
2	3	-0,0393	-0,0629	0,0741
2	4	-0,0096	-0,0475	0,0484
3	0	0,0037	0,0150	0,0154
3	1	0,0257	-0,0264	0,0369
3	2	0,0020	-0,0250	0,0251

3	3	0,0013	-0,0605	0,0605
3	4	-0,0051	-0,0863	0,0865
4	0	0,0605	0,0154	0,0625
4	1	0,0264	-0,0113	0,0287
4	2	0,0249	-0,0037	0,0252
4	3	0,0122	-0,0454	0,0470
4	4	0,0331	-0,0852	0,0914
5	0	0,0364	-0,0456	0,0583
5	1	0,0581	-0,0334	0,0670
5	2	0,0843	-0,0023	0,0843
5	3	0,0355	-0,0054	0,0359
5	4	0,0883	-0,2105	0,2283
6	1	0,0460	0,0205	0,0504
6	2	0,0484	-0,0189	0,0520
6	3	0,1623	-0,3240	0,3624
max_err [m]	0,3624		σ[m]	0,0706
RMSE [m]	0,1008			

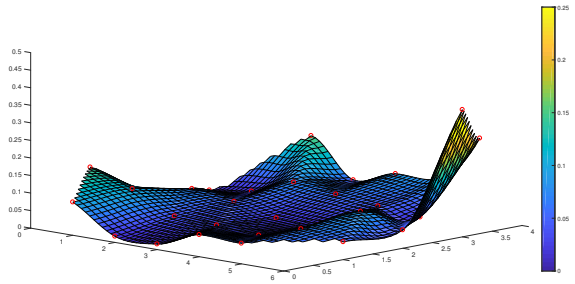
Table 5.3

Table 5.3 if compared to the previous one 5.1, shows an increased accuracy. The maximum error is decreased of almost 10 cm, while the RMSE is 8 cm smaller, almost halved. The RMSE, that now is 10 cm, is compliant with the final application.

Comparing figure 5.3 with figure 5.2 is possible to notice an important improvement of the absolute error. The top view shows how the error is drastically reduced. The side view instead is flattered and more uniform respect to the previous one.



(a)



(b)

Figure 5.3: 3D error surface for error correction. (a) top view, (b) side view

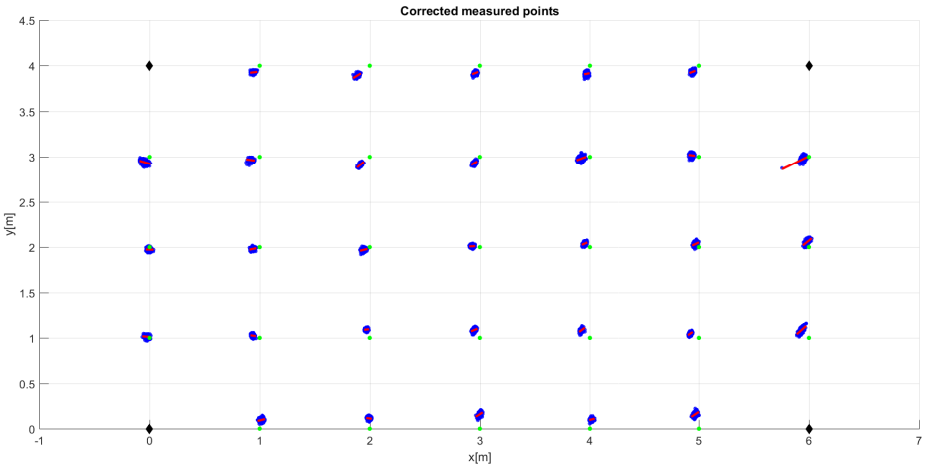


Figure 5.4: Static measurements corrected using the corrective function.

Figure 5.4 shows the clusters of point distribution once applied the corrective

function. If compared with figure 4.2 is evident how the clusters are moved toward the test point, decreasing the euclidean distance.

5.2 Double system: 8 anchors

In order to have redundant information, another TREK1000 development kit is added to the previous one. A total of eight anchors and one tag are used in the following tests. Anchors are positioned forming an octagon in order to exploit the symmetry of the constellation.
Anchors positioned in:

first system

- anchor 0 (0.0m; 0.0m, 0.09m);
- anchor 1 (2.82m; 2.82m, 0.09m);
- anchor 2 (-2.82m; 2.82m, 0.09m);
- anchor 3 (0.0m; 5.64m, 0.09m);

second system

- anchor 0 (-2.0m; 4.82m, 0.09m);
- anchor 1 (-2.0m; 0.82m, 0.09m);
- anchor 2 (2.0m; 4.82m, 0.09m);
- anchor 3 (2.0m; 0.82m, 0.09m);
- tag placed on a tripod 1.64m tall.

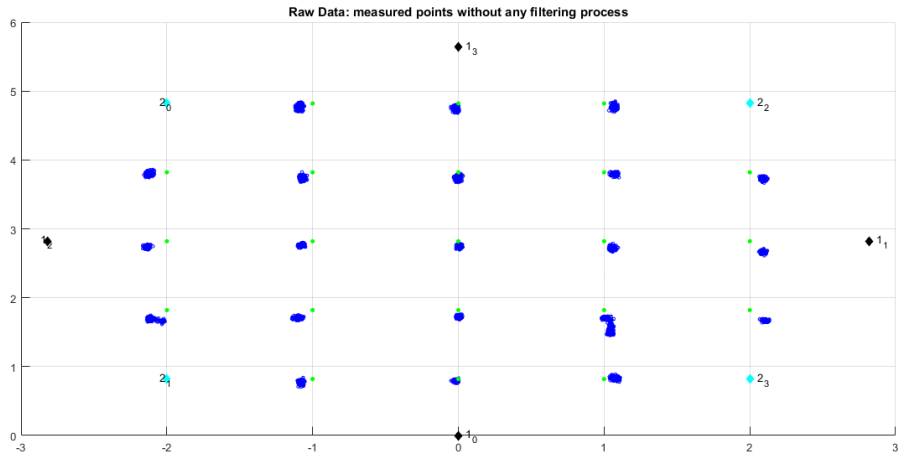
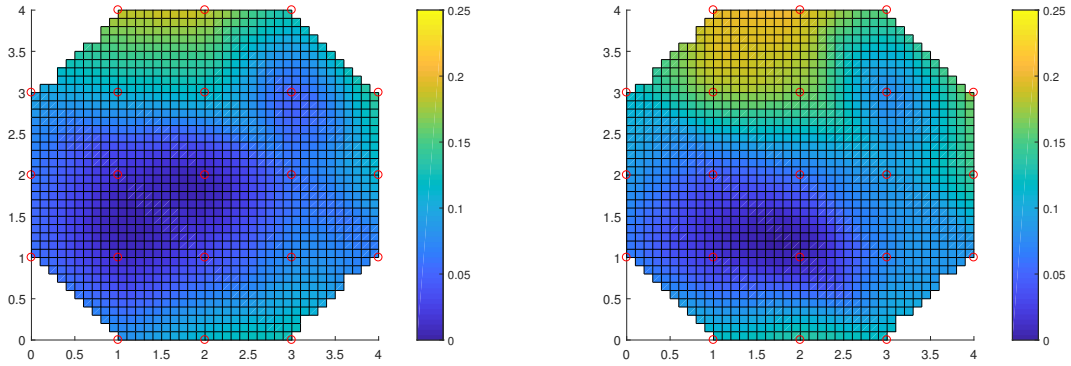


Figure 5.5: Static measurements with 8 anchors witout any correction.

Figure 5.5 shows the measurement distribution with 8 anchors. One system is represented by black diamond, while the second one by cyan diamonds.

5.3 Offset correction

The offset introduced by the closed-form localization algorithm is a bias errors that pulls the measurement toward the reference anchor. In order to mitigate this error all the anchors are used as reference one, repeating the positioning algorithm for each of them. Then the final position is obtained averaging all the results.



(a) (b)

Figure 5.6: Top view of the error surface. (a) shows the error after the correction, while (b) is the error before.

In figure 5.6 is reported the top view of the surface error before and after applying the offset correction. In the first case the error is ditributed along the border far from the anchor 0 located in (0; 0) that is the reference node. In fact since all the points are pulled toward this node, the error is present along the border in the opposite direction. After averaging the results get from all the anchors used as reference, the error is spread around the 2D area.

No correction				
X_err[m]	Y_pos[m]	x_err[m]2	y_err[m]	abs_err[m]
0	1	0,0570	-0,0278	0,0634
0	2	0,0426	-0,0754	0,0866
0	3	0,0517	-0,1100	0,1216
1	0	0,0095	0,1178	0,1182
1	1	-0,0048	-0,0167	0,0174
1	2	0,0184	-0,0463	0,0499
1	3	0,0210	-0,1720	0,1732
1	4	-0,0335	-0,1956	0,1985
2	0	-0,0149	0,1332	0,1341
2	1	0,0034	-0,0052	0,0062

2	2	0,0057	-0,0664	0,0666
2	3	-0,0012	-0,1652	0,1652
2	4	-0,0282	-0,1970	0,1991
3	0	-0,0066	0,1215	0,1217
3	1	-0,0744	-0,0213	0,0774
3	2	-0,0354	-0,0844	0,0915
3	3	-0,0774	-0,0216	0,0804
3	4	-0,0168	-0,1619	0,1628
4	1	-0,0668	-0,0543	0,0861
4	2	-0,0655	-0,1357	0,1507
4	3	-0,0680	-0,1664	0,1798
max_err [m]	0,1991		[m]	0,0562
RMSE [m]	0,1247			

Table 5.4

Correction				
X_pos[m]	Y_pos[m]	x_err[m]	y_err[m]	abs_err[m]
0	1	0,0585	0,0022	0,0585
0	2	0,0431	-0,0434	0,0612
0	3	0,0529	-0,0852	0,1003
1	0	0,0114	0,0807	0,0815
1	1	-0,0049	0,0100	0,0112
1	2	0,0184	0,0027	0,0186
1	3	0,0210	-0,1110	0,1129
1	4	-0,0110	-0,1881	0,1884
2	0	-0,0160	0,1135	0,1147
2	1	0,0034	0,0330	0,0332
2	2	0,0057	-0,0115	0,0129
2	3	-0,0012	-0,1179	0,1179
2	4	-0,0177	-0,1857	0,1865
3	0	-0,0142	0,1276	0,1284

3	1	-0,0744	0,0146	0,0758
3	2	-0,0354	-0,0487	0,0602
3	3	-0,0460	-0,0275	0,0537
3	4	-0,0187	-0,1401	0,1414
4	1	-0,0669	-0,0039	0,0670
4	2	-0,0698	-0,0953	0,1181
4	3	-0,0776	-0,1376	0,1580
max_err [m]	0,1884		[m]	0,0528
RMSE [m]	0,1042			

Table 5.5

Analyzing tables 5.4 and 5.5 is possible to notice how the error is redistributed in the area covered by the anchors. Peaks of error are reduced, the maximum error is smaller after the correction as well as the RMSE.

5.4 NLOS correction

The NLOS is a common problem in environments full of objects and obstacles. In the following picture is represented a static measure, where the anchors are forming an octagon and the tag is placed in the center of the constellation in (0; 2.82). During the measuments a person is walking around the tag moving an obstacle represented by a metal cylinder. Since they both represent obstacles, some ranges are biased as effect of the NLOS. The cluster of points is very spread all around the 2D plane and the measurements don't convey toward a unique solution.

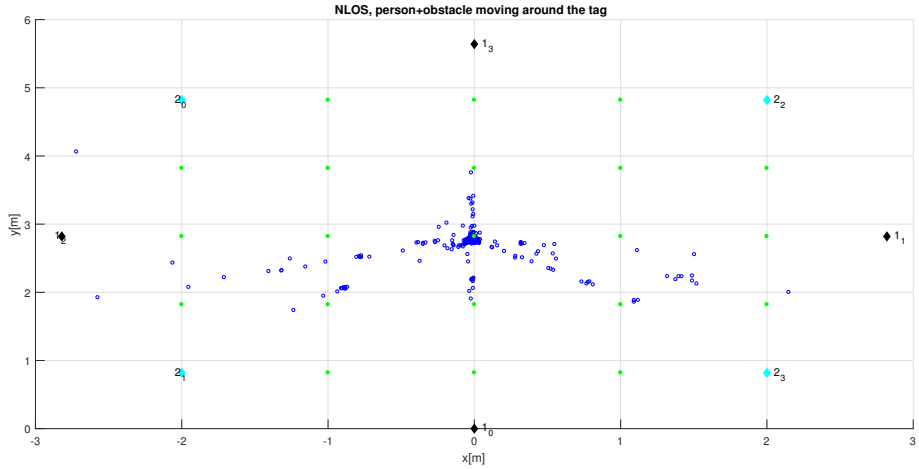


Figure 5.7: Moving obstacle, no NLOS correction applied.

After applying the NLOS filter a big improvement is noticeable. Now the cluster is compact and the points convey toward the center. All the point are included in a radius smaller than 20cm from the real position (test point).

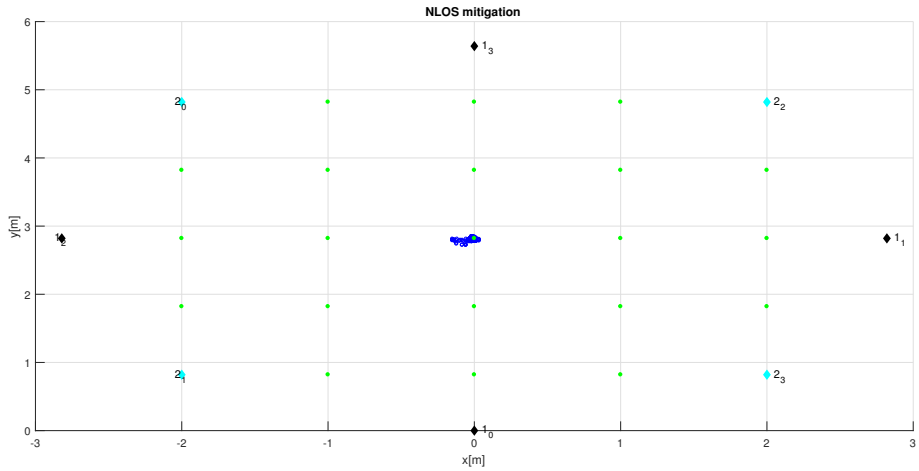


Figure 5.8: Moving obstacle after applying the NLOS filter.

Chapter 6

Anchors Calibration

The final application is designed to work in home environment. One aim is to make this application ready to go as soon as the patient get the system, without any external interventon by technicians. Once the patient places the anchors inside the apartment, the position of the nodes are not known to the system, and it would be unpleasant if the patient has to measure the exact position of them, or ask for the help of some technicians.
In order to avoid all this a calibration phase has been developed. During this phase the robot is able to construct a map of the environment and locate the anchors in it. Thanks to this the final user only needs to know how to place the nodes all around and start the robot.
The calibration process requires three steps to be performed in sequence in order to be effective:

- step 1: use on board SLAM process to construct the environment map;
- step 2: during step 1, record where the minimum range values are measured;
- step 3: move in positions recorder during step 2 and perform a reverse trilateration.

6.1 Step 1: mapping

Many vacuum cleaner home robots have the SLAM feature. Exploiting their on board sensors they are able to draw a map of the environment in which they operate and also to autolocate theirselves.
The idea is to use this feature to obtain a detailed map of the house, knowing in this way where obstacles are located and the shape of the apartment. Then thanks

to the map, the robot is able to trace the shorter path between it and the patient and navigates to him/her.
When the robot has to reach the patient, first of all it establishes a path to follow in order to travel till destination. Then this path is decomposed in a list of movement commands and sent to the drive interface that controls the wheels. As the robot proceeds, it monitors its position and retraces the map, and if it is the case it modifies the list of commands.

During the acquisition of the shape of the environment the robot is able to locate also some obstacles such as a couch or a table, and then avoid it during the path determination process. Furthermore in the case the patient decides to change the enterior design of the habitation, he/she just needs to lunch the calibration process again and the system will be upgraded with the new map. The same if he/she decides to move the anchors.

6.2 Step 2: locate minimum ranges

As explained above, during step 1, the robot must record its position on the map when the minimum distance between its tag and an anchor is measured. In this way is possible to obtain an approximation of the anchor distribution in the house. The latter is foundamental for the last step that is in charge of locating the anchors with high accuracy.
Step 2 is a simple operation that can be executed using a simple MatLab script that analyzes the range values, and every time the minimum distance between the tag and one anchor si found out, the position obtained from the robot must be saved.

Decawave system has the problem that when the tag and the anchor are closer than 20 cm, the range value is not consistent anymore, in the sense that the ToA value is affected by an error that is greater that the measure itself. When two devices are too close, the ToA measure requires a very high resolution, but since TREK1000 doesn't have it, the measure is not valid.
This problem can be simply avoided by placing the anchors near the ceiling, above people head. In this case the signal avoid the interference due to people, and since

the tag is positioned on the robot, that travels on the floor, the range will be always greater than 20cm.

6.3 Step 3: reverse trilateration

Once obtained an approximation of the position of the anchors, the robot can perform the third and last step to locate with an error smaller than 10 cm the anchors in the home environment.
First it has to move in the same room where the minimum distance with respect to the anchor has been recorded and then perform the reverse trilateration.

The trilateration process consists in locate a certain tag knowing its distance from three fixed points positioned in known coordinates. In this case the robot can only measure the distance from the tag to the anchor, but it has no fixed points. To overcome this problem, the reference nodes are obtained by moving the robot in three different points during the calibration process.

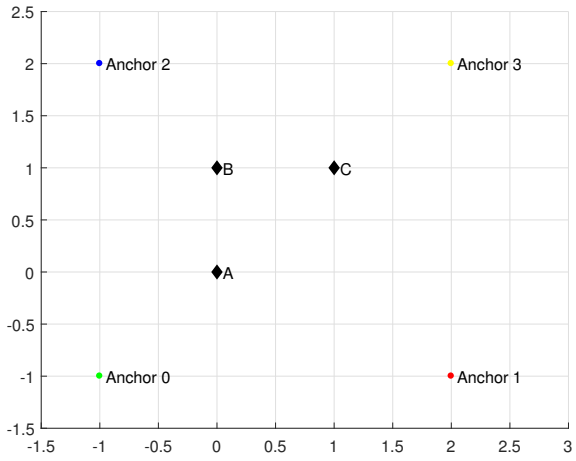


Figure 6.1: Calibration procedure, L movement.

As shown in figure 6.1 the robot starts from point A, travels straight till point B, then roates of 90 degrees and travels till point C. In each point A, B and C the robot performs some range measurements. At this point the system knows the distances

from the tag and the anchor in three different points at a known distance. In fact the location of the point is not known, but the relative distances are known.
So the reverse trilateration is subdivided in several different phases:

- record range distances in point A;
- travel straight till point B,
- record range distances in point B;
- turn 90 degrees clockwise;
- travel staright till point C;
- record range distances in point C;
- trilateration.

The range distances are recorded by the computer connected to the robot and to the tag. The L movement described by the robot is accurate, with uncertainty of millimiters thanks to the high resolution of the wheel encoders. So the uncertainty due to the robot movement is negligible respect to the uncertainty obtained from the UWB system.

Note: in the following example is reported the case in which the reverse trilateration is performed on 4 anchors. In the final application the reverse trilateration is applied on one anchor at time, in order to avoid the NLOS problem introduced by walls.

Point A is considered the origin of the system ($x_A = 0$; $y_A = 0$; $z_A = r_h$), where r_h is the height of the tag positioned on the robot. Thi value is measured once the structure of the robot is designed and recorded as constant in the software. Range measurements in pont A are indicated as [$d0_A$, $d1_A$, $d2_A$, $d3_A$]. Then the robot travels till point B of a distance equal to m_{AB} provided by the optical encoders. The new point B is located in ($x_B = 0$; $y_B = m_{AB}$; $z_B = r_h$) and the ranges are

$[d0_B, d1_B, d2_B, d3_B]$.
After a rotation of 90 degrees, the robot travels a distance of m_{BC} measured again by the encoders. So point C is ($x_C = m_{BC}$; $y_C = m_{AB}$; $z_C = r_h$). Once last measurements $[d0_C, d1_C, d2_C, d3_c]$ are obtained, reverse trilateration can be performed.

$$D = \begin{bmatrix} d0_A & d1_A & d2_A & d3_A \\ d0_B & d1_B & d2_B & d3_B \\ d0_C & d1_C & d2_C & d3_C \end{bmatrix} \tag{6.1}$$

$$P = \begin{bmatrix} 0 & 0 & r_h \\ 0 & m_{AB} & r_h \\ m_{BC} & m_{AB} & r_h \end{bmatrix} \tag{6.2}$$

Where D is the matrix of the range values and P is the matrix of the relative positions of the anchors. Matrix D is divided as follow:

$$D_0 = \begin{bmatrix} d0_A \\ d0_B \\ d0_C \end{bmatrix} \quad D_1 = \begin{bmatrix} d1_A \\ d1_B \\ d1_C \end{bmatrix} \quad D_2 = \begin{bmatrix} d2_A \\ d2_B \\ d2_C \end{bmatrix} \quad D_3 = \begin{bmatrix} d3_A \\ d3_B \\ d3_C \end{bmatrix} \tag{6.3}$$

Exploiting the Matrix P and vectors D_{i-th} is possible to perform the reverse trilateration and figure aout the position of the anchors:

- Anchor 0 = trilateration (P, D_0);
- Anchor 1 = trilateration (P, D_1);
- Anchor 2 = trilateration (P, D_2);
- Anchor 3 = trilateration (P, D_3);

Using three fixed points to trilaterate the position of the anchors gives two solutions, one above and one below the height of the tag positioned on the bot. Since this height is not modifiable, is not possible to obtain a measure on a different plane, so the solution to be chosen is the one above the tag. In fact placing the anchors just under the ceiling allows to know a priori that the anchors are higher than the tag.

Table 6.1: Calibration error vs relative distance between reference points. Errors obtained as euclidean distance between real point and measured one.

	anchor 0			anchor 1			anchor 2			anchor 3		
distance	x_err0	y_err0	abs_err0	x_err1	y_err1	abs_err1	x_err2	y_err2	abs_err2	x_err3	y_err3	abs_err3
0,10	0,2205	-1,4320	1,4489	-0,2541	1,1725	1,1997	-0,4379	-0,1890	0,4769	0,8130	-0,4375	0,9232
0,25	-0,5416	0,0797	0,5475	-0,1471	0,2259	0,2695	-0,1798	0,0538	0,1877	-0,0481	0,3795	0,3825
0,50	-0,4564	0,0883	0,4649	-0,0399	0,2473	0,2504	-0,1381	-0,0054	0,1382	0,1022	0,0996	0,1427
0,75	-0,3754	0,1048	0,3898	0,0486	0,1095	0,1198	-0,0830	-0,0120	0,0839	0,1367	0,2199	0,2589
1,00	-0,4354	0,0346	0,4368	0,0779	0,0761	0,1089	-0,0726	-0,0241	0,0765	0,1610	0,0987	0,1888
1,25	-0,2589	0,0260	0,2601	0,0960	0,0239	0,0989	-0,0988	-0,0427	0,1077	0,1610	0,0211	0,1624
1,50	-0,1740	0,0180	0,1749	0,1258	0,0170	0,1270	-0,0593	-0,0461	0,0751	0,1595	0,0319	0,1627

Table 6.1 shows calibration errors for different relative distance between reference points. Errors decrease as the relative distance increase, this is due to the fact that since the reference points are more distant, the intersection of the spheres gives a tight area.
For anchors 0 and 3 the errors are greater. This is due to the fact that for these nodes the reference points A and C are in line, resulting in a bigger uncertainty area given by the intersection. Instead for anchors 1 and 2 the reference points are perfectly distribuited, in fact the error is in the order of 10 cm or less starting from a relative distance of 50 cm. from this is possible to notice of the reference points distribution is foundamental to reach an high accuracy.

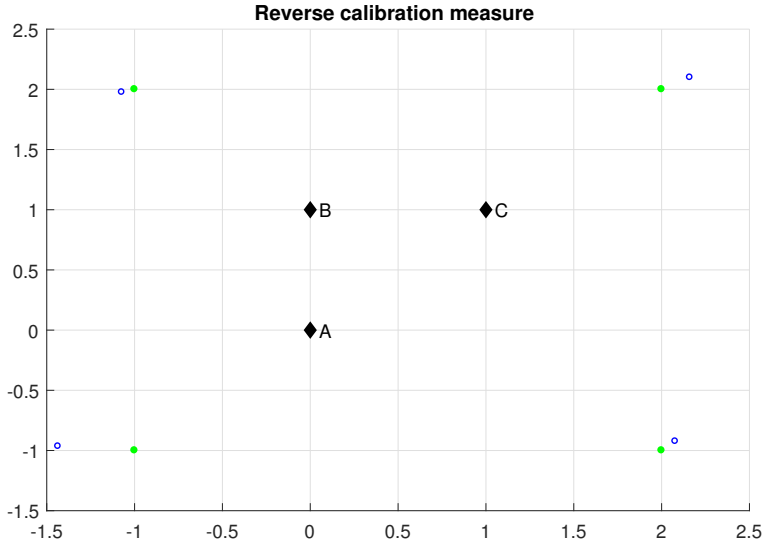


Figure 6.2: Calibration measuremet, L movement.

Figure 6.2 shows the anchor calibration for a relative distance of 1 m. Black diamonds are the reference points while the green dots represent the ideal position of the anchors, while the blue dots are the one obtained by means of the reverse calibration.
The reverse calibration introduces an error on the position of the anchors of the order of 10 cm that must be add to the error of the algorithm presented in chapter IV.

Chapter 7

Conclusion and future work

The final application is an home robot monitoring system able to assist a patient in his/her home environment in case of critic health status. The aim of this thesis work is to provide the robot with an indoor navigation system based on Ultra Wide-Band. In chapter IV is shown how the algorithm works and in the following one are reported test and results.
Starting from a declared error of 20 cm in positioning, due to the UWB Decawave system, the algorithm is able to reduce this to 10 cm, so basically is able to halve the initial error. Furthermore it introduces a mitigation of the NLOS error thanks to redundant information coming from a second system.
Since the final application is designed to become a commercial device one day, it requires a calibration process able to locate anchors devices inside the house without the help of external technicians. This calibration phase is presented in chapter VI. Thanks to its three steps process is able to locate anchors with an accuracy of 10 cm, that is very high considering the dimension of an apartment.

Many improvements can be done to this application in order to transform it in a commercial device.
A possible future work consists in substituting the PC with Raspberry board that can reduce the power consumpion, the occupied volume and the total weight.
The current robot prototype is not equipped with a functioning SLAM, so a possible improvement is to design a SLAM procedure providing the necessary sensors to the bot. Furthermore SLAM could be usefull to help the autolocalization of the bot, so could be integrated with the actual localization algorithm in order to make the software, and so the total system, more robust.
Since the Decawave UWB system is a development kit, it doesn't allow to connect

an external sensor to its Wireless Sensor Network (WSN). So the next step would be to design a dedicated hardware PCB that allocates the DW1000 from Decawave and alows also to communicate with external devices through a certain protocol. This is foundamental, especially for the tag to be placed on the patient that has the duty to transmit the health status to the central working unit represented in this case by the PC or by the raspberry board.
Furthermore even the TWR algorithm could be improved, so instead allowing the communication only between 4 anchors, it could be extended to an indefinite number, allowing to use redundand information on the same channel and with more than one system.

Bibliography

[1] Mohammadreza Yavari and Bradford G. Nickerson Ultra Wideband Wireless Positioning Systems Technical Report TR14-230 March 27, 2014

[2] Bruno Silva 1, Zhibo Pang 2, Johan kerberg 2, Jonas Neander 2 and Gerhard Hancke 1 Experimental Study of UWB-based High Precision Localization for Industrial Applications 1 Dept. of EECE, University of Pretoria, South Africa 2ABB AB, Corporate Research, Sweden 2014

[3] Francesco Bandiera,1 Angelo Coluccia,1 Giuseppe Ricci,1 Fabio Ricciato,2 and Danilo Spano1 TDOA localization in asynchronous WSNs 1. DII, University of Salento Via Monteroni, 73100 Lecce, Italy name.surname@unisalento.it 2. Austrian Institute of Technology Vienna, Austria fabio.ricciato@ait.ac.at

[4] Anastasios I. Mourikis and Stergios I. Roumeliotis Analysis of Positioning Uncertainty in Simultaneous Localization and Mapping (SLAM) 2004 IEEElRSJ International Conference on Intelligent Robots and Systems

[5] Edoardo Bonizzoni A remote assistive technology for elderly safety based on indoor localization Master of Science in Engineering Master Thesis

[6] Decawave SOURCE CODE GUIDE DECARANGE RTLS ARM SOURCE CODE Understanding and using the DecaRangeRTLS ARM source code Version 2.1 2015

[7] Decawave TREK1000 User Manual HOW TO INSTALL, CONFIGURE AND EVALUATE THE DECAWAVE TREK1000 TWO-WAY RANGING (TWR) RTLS IC EVALUATION KIT Version 1.06

[8] Decawave TREK1000 Quick Start Guide Two Way Ranging(TWR) Evaluation Kit Version 1.3

[9] Ankush Vashistha, Rejina Ling Wei Choi, Choi Look Law A TDOA Measurement Technique for Asynchronous Indoor Localization System using UWB-IR The Center for Infocomm Technology (INFINITUS) Nanyang Technological University (NTU, Singapore) vash0001@e.ntu.edu.sg, rejinachoi, ecllaw@ntu.edu.sg

[10] Jacek Stefanski synchronous time difference of arrival (ATDOA) method Gdansk University of Technology, Faculty of Electronics, Telecommunications

and Informatics, Department of Radio Communication Systems and Networks, Narutowicza 11/12, 80-233 Gdansk, Poland

[11] Jia Wang, Asad Khalid Raja, Zhibo Pang Prototyping and Experimental Comparison of IR-UWB based High Precision Localization Technologies ABB AB, Corporate Research, 72178, Vsters, Sweden jiaw, asadraja@kth.se, pang.zhibo@se.abb.com

[12] Julius O. Smith, Jonathan S. Abel IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-35, NO. 12, DECEMBER 1987 1661 Closed-Form Least-Squares Source Location Estimation from Range-Difference Measurements

[13] Zebra technologies dartuwb tech datasheet

[14] iRobot Create Owners guide www.irobot.com

[15] R. Xu, W. Chen, Y. Xu, and S. Ji, A New Indoor Positioning System Architecture Using GPS Signals, Sensors, vol. 15, no. 5, pp. 1007410087, 2015.

[16] Tingcong Ye, Michael Walsh, Peter Haigh, John Barton, Alan Mathewson, Brendan OFlynn Experimental Impulse Radio IEEE 802.15.4a UWB Based Wireless Sensor Localization Technology: Characterization, Reliability and Ranging Tyndall National Institute University College Ireland, UCC, Cork

[17] Shuai He, Member, IEEE, Xiaodai Dong, Senior Member, IEEE, and Wu-Sheng Lu, Fellow, IEEE Asynchronous Time Difference of Arrival Positioning System

[18] Ismail Guvenc, Chia-Chin Chong, and Fujio Watanabe NLOS Identification and Mitigation for UWB Localization Systems DoCoMo Communications Laboratories USA, Inc. 3240 Hillview Ave, Palo Alto, CA, 94304 iguvenc, cchong, watanabe@docomolabs-usa.com

[19] https://www.pozyx.io/Documentation

[20] https://www.terabee.com/time-of-flight-principle/