

POLITECNICO DI TORINO

---

DIPARTIMENTO DI INGEGNERIA MECCANICA  
E AEROSPAZIALE (DIMEAS)

Master's Degree in Biomedical Engineering

No-Line-Of-Sight Identification and  
Mitigation in a Real Time Indoor  
Ultra-Wide Band  
Localization System



**Relatore**

Prof. Danilo DEMARCHI

**Correlatore:**

Dr. Paolo MOTTO ROS

Alessio SERRANI

matricola: 240529

---

March 2019

*Alla mia famiglia.*

# Acknowledgements

This thesis has been completed thanks to the help, the support, and the inspiration of several people.

First I would like to express my sincere gratitude to my thesis supervisors Prof. Danilo Demarchi (Politecnico di Torino) and Prof. Paolo Bonato (Harvard Medical School). The first in Turin and the second in Boston, the door of their offices were always open whenever I ran into problems or had a question about my work. They allowed this research to be my own work, but steered me in the right direction.

I would also like to thank Dr. Stefano Sapienza for his support and inspiring suggestions that have been fundamental for the development of this thesis. Without his passionate participation and advices, the entire work could not have been successfully completed.

I would also like to acknowledge Dr. Federico Parisi and Prof. Paolo Motto Ros as second readers of this thesis and I am gratefully to them for their very valuable suggestions.

In addition, all the Motion Analysis Lab members in Boston and Paolo Civera lab members in Turin deserve special thanks for their support throughout all my thesis period.

Finally, I must express my very profound gratitude to my family and friends for providing me with unconditional support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This achievement would not have been possible without them. Thank you.

Author  
Alessio Serrani

# Abstract

Nowadays accurate indoor localization is becoming a widespread necessity to develop services and products for commercial, military and health care applications. For example, from a commercial point of view it is crucial to keep track of goods, from a military point of view it is useful to track logistics as well as tracking potential dangers and from a health care perspective it could be fundamental to track and monitor people with special needs via mobile platforms. This thesis will deal with problems associated to indoor localization; it is part of a larger project, the Home Robot project, focused on reducing common issue of false alarms coming from fall sensors. The purpose of the project is to provide a reliable tool that can check the status of patients using a mobile robot when a fall sensor detects an event, navigating through the home environment to the last known patient position and beginning a video conference between caregiver or physician and the potentially injured patient. The aim of the thesis is to support the Home Robot project with a reliable navigation system and a positioning algorithm able to locate both the robot and the patient in any home environment.

To implement the localization application, a commercial Ultra-Wide Band (UWB) system will be used, composed by one tag (the entity that needs to be localized) and several anchors (entities that are in a fixed position to exploit the trilateration process). The main problem faced in this work is the No-Line-Of-Sight (NLOS). NLOS situations occur when there is an obstacle between the tag to be localized and one or more fixed anchors. In those cases, the final localization of the tag will be wrong thanks to the interaction between the obstruction and radio signals.

The overall proposed algorithm is composed by a classification algorithm to identify NLOS affected measurements and a subsequent mitigation algorithm to correct the final position estimations that present an error due to the presence of the NLOS effect. The identification algorithm consists of a Support Vector Machine (SVM) classifier based on three informative features extracted from the Channel Impulse Response (CIR), the diagnostic tool made available from the used commercial UWB kit. These three features describe the situation of the analyzed UWB channel (wireless connection among transmitter and receiver), exploiting differences and similarities between the first received radio signal and the one with the biggest amplitude during the same message exchange session. The algorithm has been tested among different datasets, showing an overall discrimination capability between NLOS and LOS condition greater than 99%.

The second part of the thesis concerns the mitigation problems: once NLOS affected measurements are discovered, it is needed to correct related trilateration error. The proposed method is a Taylor Series (TS) based Least-Square (LS) correction algorithm that is a good trade-off between accuracy and low-complexity, crucial feature since the final application is a real-time one. The performance of the proposed algorithm is evaluated experimentally through both static and dynamic simulations and it shows a significant improvement in respect to the original NLOS affected position estimations. In all the studied cases, the final average positioning error will be reduced under the threshold of 20cm, taken as a benchmark since it is the declared accuracy under LOS condition of the used UWB kit, with a further important decrease in the variability (in terms of standard deviation) inside the measurements.

The overall result is an algorithm able to provide a reliable and robust position estimation method for the Home Robot application, that can achieve high NLOS identification and mitigation performance in different environments and inside different working platforms.

# Contents

<b>Acknowledgements</b>	I
<b>Abstract</b>	II
<b>List of Figures</b>	VI
<b>List of Tables</b>	IX
<b>1 Introduction</b>	1
1.1 Summary	1
1.2 Falls detection problem	3
1.3 Home robot project	4
1.3.1 Hardware	5
1.3.2 Software	8
<b>2 State of the art</b>	10
2.1 Overview of different localization technologies	10
2.1.1 Indoor localization	11
2.2 Definition of the NLOS problem	12
2.3 Related studies about NLOS identification and mitigation in UWB systems	13
2.3.1 NLOS identification	13
2.3.2 NLOS mitigation	14
<b>3 Materials and methods</b>	16
3.1 UWB Technology	16
3.1.1 UWB features and characteristics	18
3.1.2 Impulse Radio Ultra-Wide Band (IR UWB)	18
3.1.3 UWB ranging techniques	20
3.1.4 UWB trilateration process	23
3.1.5 UWB regulations	26
3.1.6 Commercial Ultra-Wide Band infrastructures	27
3.2 Decawave TREK1000 Evaluation Kit	30

3.2.1	Hardware features . . . . .	30
3.2.2	Software features . . . . .	34
3.3	Channel Impulse Response . . . . .	39
<b>4</b>	<b>NLOS identification and mitigation algorithm</b>	<b>43</b>
4.1	Overall algorithm working principle . . . . .	43
4.2	Features extraction method . . . . .	46
4.3	Identification algorithm . . . . .	52
4.3.1	Features robustness evaluation . . . . .	52
4.3.2	Data collection for the training set . . . . .	54
4.3.3	Classifier construction . . . . .	56
4.3.4	Classifier validation phase . . . . .	57
4.4	Mitigation algorithm . . . . .	59
4.4.1	Taylor Series-based Least-Square algorithm . . . . .	60
4.4.2	Static simulations . . . . .	64
4.4.3	Dynamic simulations . . . . .	66
<b>5</b>	<b>Measurements and results</b>	<b>75</b>
5.1	Features robustness evaluation . . . . .	75
5.2	Identification algorithm performance . . . . .	81
5.3	Mitigation algorithm performance . . . . .	83
<b>6</b>	<b>Conclusions</b>	<b>90</b>
6.1	Future improvements . . . . .	92
	<b>Bibliography</b>	<b>93</b>

# List of Figures

1.1	Different commercial fall detection systems . . . . .	4
1.2	Example of a Home Robot monitoring system. . . . .	4
1.3	Rendering of the Home Robot used in the thesis . . . . .	5
1.4	iRobot Create2 programmable robot . . . . .	6
1.5	iRobot Create2 schematic components view . . . . .	7
1.6	Logitech webcam . . . . .	7
1.7	MIMO monitor . . . . .	8
1.8	Home Robot software architecture . . . . .	9
2.1	Multipath concept explanation. . . . .	12
3.1	Typical bandwidth of an UWB signal. . . . .	17
3.2	Capacity-Bandwidth relationship given a fixed SNR. . . . .	17
3.3	IR UWB signal with TH modulation. . . . .	19
3.4	SDS-TWR working principle [15]. . . . .	22
3.5	Position estimation in 2-D and 3-D spaces. . . . .	23
3.6	Intersection of three spheres in a 2-D space. . . . .	24
3.7	Trilateration with ideal measurements (a) and with noisy measurements (b). . . . .	25
3.8	FCC regulation for emitted signal power in UWB systems. . . . .	26
3.9	Allowed UWB channels in different countries. . . . .	27
3.10	Zebra Ultra-Wide Band Technology. . . . .	28
3.11	Pozyx UWB board. . . . .	29
3.12	Decawave TREK1000 Evaluation Kit. . . . .	29
3.13	Decawave TREK1000 Evaluation Kit . . . . .	31
3.14	TREK1000 Tracking Use Case . . . . .	31
3.15	TREK1000 Geo-Fencing Use Case . . . . .	32
3.16	TREK1000 Navigation Use Case . . . . .	32
3.17	EVB1000 front and back view . . . . .	33
3.18	DW1000 UWB transceiver schematic view . . . . .	34
3.19	Superframe organization in the Decawave RTLS . . . . .	35
3.20	Decawave message format . . . . .	36

3.21	Different type of messages exploited during TWR application . . . . .	36
3.22	Decawave TWR algorithm overview . . . . .	38
3.23	Decawave Channel Impulse Response . . . . .	39
3.24	Comparison between CIR content in LOS and NLOS condition . . . . .	40
4.1	Overall algorithm block diagram . . . . .	44
4.2	CIR_PWR variable in the RX Frame Quality Information register . . . . .	46
4.3	RXPACC variable in the RX Frame Information register . . . . .	46
4.4	FP_INDEX variable in the Receive Time Stamp register . . . . .	47
4.5	LDE_PPINDX variable in the Leading Edge Detection Interface register . . . . .	47
4.6	FP_AMPL1 variable in the Receive Time Stamp register . . . . .	48
4.7	FP_AMPL2 and FP_AMPL3 variables in the RX Frame Quality Information register . . . . .	48
4.8	LDE_PPAMPL variable in the Leading Edge Detection Interface register . . . . .	48
4.9	Diagnostic data are inserted at the end of the ranging part of the Final Message . . . . .	49
4.10	Disposition of diagnostic data inside Final Message . . . . .	50
4.11	Introduced editing in the original TREK1000 firmware . . . . .	50
4.12	Teraterm view with ranging information ( <i>ma</i> , <i>mc</i> and <i>mr</i> ) and diagnostic information (all the variables described above). . . . .	51
4.13	Evaluation grid for the training set construction . . . . .	55
4.14	Comparison between a correct trilateration and a wrong trilateration . . . . .	62
4.15	Example of TS-LS mitigation effect . . . . .	62
4.16	Example of TS-LS correction with Anchor2 in NLOS . . . . .	65
4.17	Vicon output for Trajectory 1 . . . . .	68
4.18	Identification algorithm performance for Trajectory 1 . . . . .	69
4.19	Mitigation algorithm performance for Trajectory 1 before and after transition phase correction . . . . .	71
4.20	Trilateration absolute error for Trajectory 1 before and after transition phase correction . . . . .	72
4.21	Final estimation for Trajectory 1 . . . . .	74
5.1	Comparison between features behaviour in LOS and NLOS condition . . . . .	76
5.2	Features behaviour in a dynamic NLOS condition . . . . .	76
5.3	Boxplots representing features behaviour varying acquisition length . . . . .	78
5.4	Boxplots representing features behaviour varying obstacle position . . . . .	79
5.5	Boxplots representing features behaviour varying antenna orientation . . . . .	80
5.6	Scatterplot graph represents clearly the differentiation between the two situations. . . . .	81
5.7	Identification algorithm performance on training set . . . . .	82

5.8	Comparison between original error, error after discarding mitigation algorithm application and LOS error . . . . .	84
5.9	Graphic representation of mitigation results in static conditions over different tested datasets . . . . .	85
5.10	Barplot representing achieved mitigation results for analyzed trajectories in terms of absolute trilateration error for both LOS and NLOS parts of each trajectory. . . . .	87
5.11	Trajectories reconstruction starting from original estimations, with TS-LS algorithm and smoothing algorithm application (Trajectory2, Trajectory3, Trajectory4) . . . . .	88
5.12	Trajectories reconstruction starting from original estimations, with TS-LS algorithm and smoothing algorithm application (Trajectory5, Trajectory6 and Trajectory7) . . . . .	89

# List of Tables

4.1	Anchor-Tag-Obstacle coordinates in the static and dynamic evaluation	52
4.2	Anchor-Tag-Obstacle coordinates in the acquisition length influence evaluation, with an indication of tested recording times . . . . .	53
4.3	Anchor-Tag-Obstacle coordinates in the obstacle position influence evaluation, with an indication of tested obstacle positions . . . . .	53
4.4	Anchor-Tag-Obstacle coordinates in the antenna orientation influence evaluation, with an indication of tested antenna orientations . . . . .	54
4.5	Tested combinations of kernel functions and C values . . . . .	57
4.6	Anchors-Tag coordinates in the data collection with multiple NLOS anchors . . . . .	58
4.7	Anchors-Tag coordinates in the data collection made in a different recording room . . . . .	58
4.8	Anchors-Tag coordinates in the data collection with a platform with a different number of anchors . . . . .	59
4.9	Anchors disposition for the tested trajectories . . . . .	67
5.1	Classification results on validation datasets . . . . .	82
5.2	Overall NLOS identification performance over training and validation sets . . . . .	83
5.3	Mitigation results on tested datasets in terms of improvement and average trilateration error . . . . .	84

# Chapter 1

## Introduction

### 1.1 Summary

During last decades, world's population is getting old more and more, with an increasing number of over-65-years old people. This phenomenon of aging is leading to the emergence of new kind of situations and needs: if, on one side, it is undeniable that senior people feel more comfortable living in their own home without too many intrusions in their own life, on the other hand, it is still necessary to provide them a safe ensemble of tools that can ensure a continuous health monitoring.

The continuous health monitoring can cover all kinds of diseases. Chronic diseases (such as cardiovascular diseases, epilepsy, schizophrenia, chronic respiratory diseases) usually benefit a lot from a continuous monitoring, but also injuries that are not predictable could be mitigated and serious consequences could be avoided. In this second category, for example, falls can be included. Especially for elder people and especially in a home environment, falls are a common problem: they could cause serious damage to the elderly when they are not detected quickly, thus avoiding having to wait hours before someone becomes aware of the problem and therefore being the trigger of further health issues.

To provide a quick and reliable detection of sudden falls, a lot of sensors and commercial alert systems are available. As it is easy to understand, these systems are sensible: from the point of view of the patient, not to detect a fall is much more serious than detecting one that does not exist. This is why often the problem of false positives has to be faced: in this context a false positive means that sensors report a tragic event that has never happened. To mitigate this kind of unwanted effect, the Home Robot project has been developed: it consists in an autonomous mobile robot that is triggered by the detection of a fall by the sensor and it has the task to reach the patient to let his caregiver/physician check his health status through a video call, before calling for help.

To allow the robot to do its job, the first thing is to provide it with a reliable indoor navigation tool. The aim of the thesis is to support the Home Robot with

a trustworthy navigation system and a positioning algorithm able to locate the robot and the patient in any home environment. In this thesis a commercial UWB localization kit by Decawave (<https://www.decawave.com>) has been used. The kit is composed by four electronic boards: three of them are fixed in the room (they will be called "anchors" from now on) and exchange Ultra-Wide Band (UWB) signals with a fourth that is the mobile one (that can be placed both on the robot and on the patient) and that has to be located in the space (it will be called "tag" from now on). The main problem faced in this work is the No-Line-Of-Sight (NLOS) problem: NLOS situations occur when there is an obstacle between the tag to be localized and one or more anchors. In those cases, the final localization of the tag will be wrong thanks to the interaction between the obstruction and radio signals.

The thesis will be organized as follows. In the continuum of this chapter the fall detection process will be introduced, with a quick overview about used technologies and few words about false positives issue. Then, the Home Robot project will be explained in detail: both hardware components and software infrastructure of the robot will be described to better contextualize the introduction of the localization system and the core of the thesis that is the NLOS identification and mitigation. The second chapter is a description of the state of the art. In this part an overview about localization technologies is given. Then an explanation of NLOS effect and related studies for its identification and mitigation are provided.

The third chapter deals with the materials and methods used in this thesis. A detailed description of UWB technology and Decawave kit (both from the hardware and software points of view) is given. A big part of this section is about kit working principle to better understand the changes in the firmware that will need to be done later. At the end Channel Impulse Response (CIR, the diagnostic tool used to face NLOS identification) is introduced.

The fourth chapter is the one with the description of the overall proposed algorithm. A block diagram with all the steps is given. Then the three main parts are described in detail: features extraction from Decawave firmware, classifier construction for NLOS identification and mitigation algorithm to reduce NLOS error if it were present, according with previous classifier output.

The fifth chapter is about the presentation of collected measurements and results. Features robustness evaluation under different conditions is provided. Then identification and mitigation performance both in static and dynamic situations are described in detail.

The sixth chapter deals with some final considerations. First overall performance are discussed in depth, then, to summarize, personal conclusions of the author and some ideas about future improvements and/or additions for the proposed infrastructure are provided.

## 1.2 Falls detection problem

Falls are one of the biggest problems for elder people during their daily activities in the home environment. In the most of cases, falls cause injuries and they can also lead to death. Statistically about 30% of people that are over 65-years-old fall every year: among these reported falls, more than one out of ten lead to serious injuries such as fractures, hematomas, brain trauma. There are a lot of risk factors that can be associated with falls (i.e. vision deficits, gait problems, chronic health issues) and all of these often worsen with increasing age [1]. Another related problem is the time the elder has to wait before someone becomes aware of his situations. The more time he will pass on the ground the more serious the consequences will be. Some of the effects of lying on the ground for more than one hour are, for instance, the appearance of pressure ulcers, incontinence, dehydration [2].

In this context, automatic falls detection is becoming increasingly important. Nowadays there is a wide variety of sensors and alarm systems that can provide this feature. Most of them rely on the usage of wearable sensors (especially inertial sensors), but there are also some others that use cameras, microphones, pressure sensors on the floor (Figure 1.1) [3].

Usually falls detection alarm systems include an entire platform that is based on the sensor measurement. Sensor output is processed continuously by the platform and if a fall is reported a call center is informed and the call at 911 is automatic. Regardless of the technology used in the sensors, it is natural to understand that all of them are designed to have a high sensibility: not even reporting an event could lead to tragic consequences for the patient. Having high sensibility often means also having a problem of false positives. A false positive means that the sensor reports a detected fall when indeed the event has never happened. These errors are not so infrequent: they can happen, for example, in case of an improper use of the device or during an action similar to a fall, like falling on the sofa. If false positives number is not negligible, several problems have to be faced. On one side, they lead to stressful situations both for familiars and caregivers and for patients, on another side, from a more practical point of view, the overall cost of the service is going to increase: there will be more useless calls to the call center and therefore more unnecessary interventions of the first aid. This translates into a waste of money for the health system.

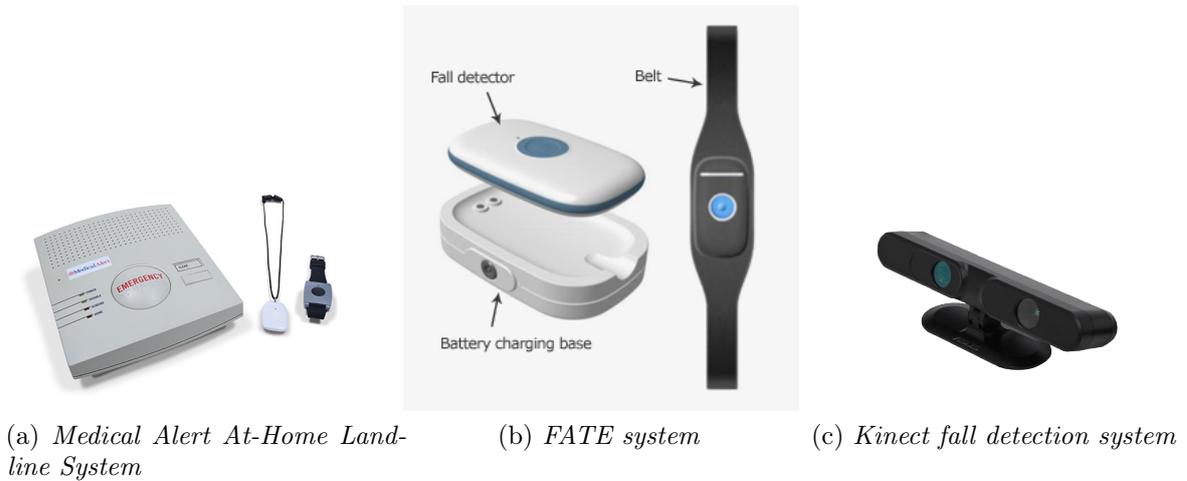


Figure 1.1: Different commercial fall detection systems

### 1.3 Home robot project

To mitigate the effect of false positives presence in falls detection systems, the idea is to insert inside the platform a filtering system. This "filter" has the task to identify when sensor reports an event that has never happened and, in that case, it will not trigger the rescue route, from the call to the call center to the ambulance sending. This filter can be implemented using a mobile robot. The concept of a Home Robot monitoring system is becoming more widespread, considering that it could be an excellent non-invasive way to track the health status of elder people that want to live in an independent way in their habitations. The Home Robot is a tool used by caregivers to check the patient's health status when a fall is detected by sensors (Figure 1.2).



Figure 1.2: Example of a Home Robot monitoring system.

The working principle of a Home Robot is intuitive: when sensors report the fall event presence, instead of starting standard rescue activities, the robot is in charge to reach the patient wherever he is in the house in that moment and, in the meanwhile, it must start opening a video chat with the related caregiver or physician. When it arrives close to the patient, the caregiver/physician has the possibility to check health status of the patient and, if necessary, to start rescue operations. All this kind of actions can be possible only if the robot is able to reach the patient location in a very precise way around the home environment. This means that, when sensor detects the fall, the position of both the patient and the robot must be known by the robot itself to have the possibility to navigate autonomously through the house and reach the patient.

This section describes both hardware and software parts of the robot used in this thesis to implement the Home Robot monitoring system application.

### 1.3.1 Hardware

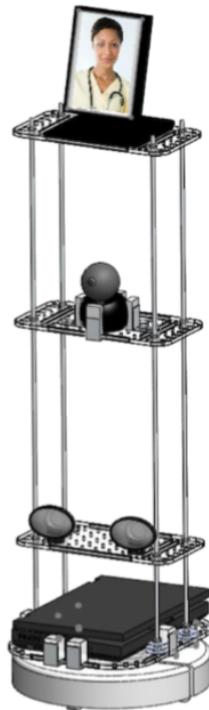


Figure 1.3: Rendering of the Home Robot used in the thesis

The Home Robot monitoring system is implemented putting together different components to give life to the desired structure (Figure 1.3). The prototype of the

robot present in the Motion Analysis Lab is composed of four main parts: iRobot Create2 Programmable Robot, that is the mobile part of the structure, a notebook, that manages all the features of the software and coordinates all the other parts through a Java interface, a monitor and a webcam, that are the parts to allow the videoconference between patient and caregiver/physician.

## iRobot Create2



Figure 1.4: iRobot Create2 programmable robot

iRobot Create2 is a development kit that allows user to program movements and sounds or to add other external electronic parts. It is produced by iRobot Corporation and comes from remanufactured Roomba vacuum cleaners. With this version of the Roomba robot, user can manage different movements, produce sounds, trigger the lighting of the LED display but also receive outputs from all the on-board sensors. iRobot Create's Open Interface (OI) allows user to send to the robot any kind of commands without necessarily knowing low-level code. To send instructions to the robot a serial cable is provided by manufactured and it is connected to the notebook that directs the architecture.

iRobot is equipped with two differential driven wheels and optical encoders in each wheel: reading encoder outputs, user can know in real time the distance travelled by the robot since the last sent instruction (if the command was a translational one) or the rotation of the robot since the last sent instruction (if the command was a rotational one).

The two wheels can rotate in an independent way up to  $500 \frac{mm}{s}$  and, using basic math formulas, both geometric center speed and angular velocity can be easily deduced:

$$v = \frac{w_l + w_r}{2} r \quad \left( \frac{m}{s} \right) \quad (1.1)$$

$$w = \frac{w_l - w_r}{2d}r \quad \left(\frac{rad}{s}\right) \quad (1.2)$$

where  $v$  is the geometric center speed,  $w_l$  and  $w_r$  are the angular speed of both left and right wheels,  $r$  is wheel radius and  $d$  is the distance between each wheel and the geometric center of the robot.

For the provided optical encoders the manufacturer declares in the datasheet an accuracy of 1mm for the translational movements and of  $1^\circ$  per the rotational ones.

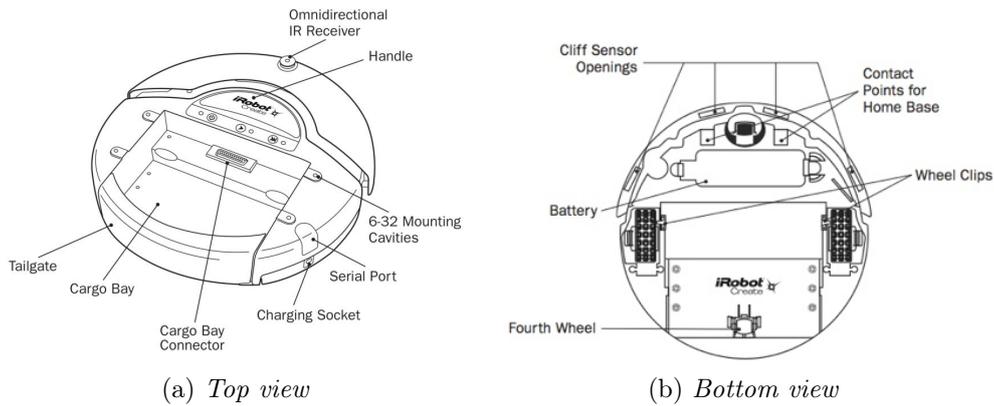


Figure 1.5: iRobot Create2 schematic components view

## Webcam Logitech Orbit



Figure 1.6: Logitech webcam

Logitech Quickcam Orbit is the cam mounted on the Home Robot infrastructure. It is a 2 MP camera with autofocus and a built-in microphone (including noise

cancellation). It provides a High-Speed USB 2.0 port to connect it to the notebook. It is the component that allows physician/caregiver to check patient health status when the robot reaches him. This camera ensures remote user to have a 102° view of the surrounding environment without the necessity to rotate the robot.

### MIMO monitor um 720s



Figure 1.7: MIMO monitor

Mimo Um-720s is a foldable 800x480 monitor. It can bend 90° and it can be stood upright using integrated stand and cover. Like the webcam, also this monitor provides a High-Speed USB 2.0 port to connect it to the notebook. Display is 7-inches large and includes a highly responsive resistive LCD touch screen. Drivers for Windows XP, Windows Vista, Windows 7, Windows 8 and Windows 10 are included. In the Home Robot platform, it is the component that allows patient to have a clear view of the physician/caregiver that is talking with him.

### 1.3.2 Software

Notebook used in the Home Robot project has the task to send movement commands to the robot using a serial connection and it must control the operation of both webcam and monitor. Furthermore, when the sensor reports a fall, the notebook has to receive information from localization system (that will be described in the third chapter) about both robot and patient position and, consequently, it has to direct the robot towards the last known position of the patient. In the meanwhile, it has also to start the videoconference with the physician/caregiver and to enable remote control, allowing the operator to move both the webcam and the iRobot itself exploiting a simple interface.

Software interface is build using Node.js, an open-source JavaScript run-time environment that can run JavaScript code also outside of a browser. Node.js presents an event-driven infrastructure that allows users to implement asynchronous communications: this is the most important feature to create real time web applications.

Unlike traditional programming, where the code execution is sequential and so if there is a blocking function all the execution will stuck, Node.js uses callback functions concept: a callback function is a part of the code that will be run only if results from other previous functions are available, allowing in the meanwhile the other parts of the code to continue working. This is efficient in real time web applications like the Home Robot project: operator, through his interface, can manage different features at the same time without the risk that an interruption in one of them could cause, for example, the loss of data coming from on-board sensors or localization system.

### Software architecture

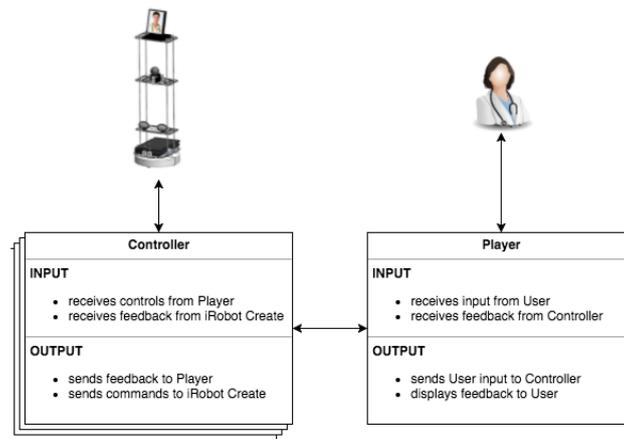


Figure 1.8: Home Robot software architecture

Software implementation includes the division in two main parts. The *Controller*, that has to manage robot hardware components like iRobot, webcam or monitor, and the *Player*, that manage the web server that implements the videoconference, the remote control and that manages user interface for the physician/caregiver side. In this application, both the Controller and the Player run in the same PC but thinking about a bigger platform that could manage more than one robot at the same time relocating the software in an external server should be the preferred option.

The communication from the Player to the Controller takes place using Publisher-Subscriber protocol: in this way a Controller can manage the communications with several Players in an asynchronous way. On the other side, communications from the Controller to the Player takes place through a Producer-Consumer protocol: the Controller is the producer and it sends information in the communication channel, the Player is the consumer and it reads messages in the communication channel. In this last architecture. communication channel works as a FIFO buffer memory.

# Chapter 2

## State of the art

### 2.1 Overview of different localization technologies

During the past years, there has been an increasing request of wireless localization systems. Different localization systems have been developed for both indoor and outdoor necessities. In an outdoor context, for example, they have become increasingly useful to save missing people during natural disasters, like earthquakes, floods or avalanches, or even just to find your car in a parking lot when you do not remember the precise location. In an indoor context, applications are even more numerous: from food tracking inside a supermarket to products tracking in an industrial environment, from patient's localization in a hospital to autonomous navigation inside a factory. Also for the Home Robot project prospective having a reliable indoor navigation system is the main goal: without it, the robot would not be able to reach the patient, making it impossible to mitigate the problem of false positives and thus making all its architecture totally useless.

For outdoor environments, Global Positioning System (GPS) is by far the most used and most reliable technology. GPS is a satellite-based technology that provides continuously both 3-dimensional position and time information. Main applications are, for instance, planes/trains/ships navigation, people localization, wireless communications. However, GPS is not a suitable technology for the Home Robot project: emitted signals are relatively weak, and they face serious difficulties in penetrating a dense environment to allow urban tracking and, consequently, issues to penetrate buildings are even bigger. So, for indoor scenarios, other kind of technologies have been developed [4].

## 2.1.1 Indoor localization

### Infrared-based systems

Infrared (IR) -based systems are widely used in localization because it is really simple to embed an IR transceiver in common devices: mobile phones, TV, cameras are nowadays all equipped with an IR source. Furthermore, an IR architecture is easy to implement, and both installation and maintenance are relatively cheap. However, IR localization systems require Line-Of-Sight (LOS) condition because the majority of indoor environment equipment are IR-opaque and, moreover, this technology is subject to interferences from other IR sources. This means that for complex indoor environment, such as a home environment (that is the environment in which Home Robot project is meant to work), IR infrastructures cannot be applied [4]. An example of an IR system is the Vicon system that will be described later.

### Ultrasound-based systems

Ultrasound (US) techniques are the cheapest technologies used in indoor environment localization systems. However, also in this case, to work properly they require LOS conditions and moreover, considering performance under the best conditions, the overall precision of an US architecture is lower than the one that can be achieved with an IR system, since ultrasounds are more subject to reflections respect to infrared signals, introducing more noise in the final estimation. Furthermore, US systems often face a synchronization problem: to overcome this issue, external Radio Frequency components are added, increasing final cost of the infrastructure [4].

### Radio Frequency-based systems

Radio Frequency (RF) -based systems use electromagnetic waves to establish a communication between transmitter and receiver. RF signals can penetrate obstacles, accepting to have an error on the final estimation depending on the interaction between signals and the obstruction. Furthermore, RF based technologies can be divided into two main categories: narrowband technologies (Bluetooth, RFID, WLAN) and wideband technologies (UWB) [4]. An UWB based system is the one used in this thesis and its working principle will be described in depth in the next chapter. The recognition and mitigation of the NLOS error is one of the major challenges in UWB ranging systems and it is the topic of this thesis.

## 2.2 Definition of the NLOS problem

No-Line-Of-Sight (NLOS) effect is one the biggest issues in localization systems. This concept is defined regardless of the used technology. A measurement is defined to be in NLOS condition if there is an obstacle between the transmitter and the receiver. If the used technology cannot penetrate the obstacle, the final location will not be available; if signals can pass through obstruction (as in Radio Frequency case), at the end a positioning will be available, but it will be affected by a certain error.

In this thesis, the localization system will be implemented using an UWB kit. Being an example of radio frequency signals, also UWB can penetrate the majority of obstacles (this is why it has been chosen), so at the end a measurement is always available. But an error is introduced in the final positioning: more precisely, a positive bias is introduced in the ranging estimation and so, at the end of the trilateration (the process that leads to the position estimation starting from a range measurement, it will be described in the next chapter), also the final Tag location will be wrong.

NLOS problem is always associated with multipath phenomenon. By definition, multipath effect is the result of the electromagnetic waves propagation: when a radio signals is sent by a transmitter, it will not reach receiver only once, but there will be several possible paths and radio signals will run across each of them. At the end, signals that will arrive at the receiver-side will be much more than only one even if the sent signal was only one. The signal that follows the direct path between transmitter and receiver is named Line-Of-Sight (LOS) signal (Figure 2.1).

If a NLOS condition occurs, the LOS signal will be obstructed: its time of arrival will be delayed, its amplitude will be decreased and its direction will be changed. In this kind of situations, independently from the used infrastructure, multipath components will become predominant and errors will be introduced in the final estimation.

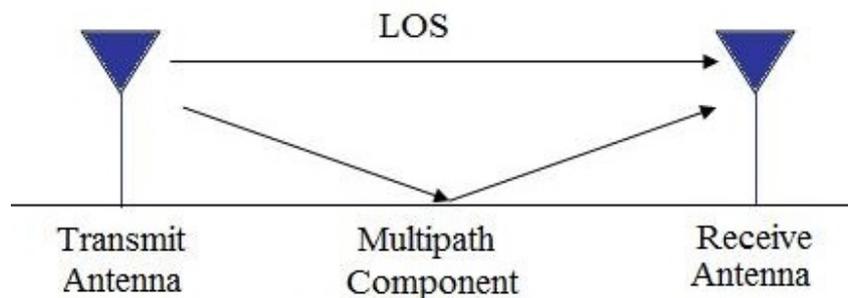


Figure 2.1: Multipath concept explanation.

## 2.3 Related studies about NLOS identification and mitigation in UWB systems

In the literature, NLOS issue has been studied a lot to find a correct way to identify it and, subsequently, to mitigate its effect in UWB systems. Several NLOS identification and mitigations techniques have been proposed: here the most significant ones to the development of this thesis will be reported.

### 2.3.1 NLOS identification

- *M.P. Wylie and J. Holtzman* approached the problem analyzing ranging measurements. As it was explained before, the introduction of a NLOS effect cause a positive bias in the range estimation. So, by using the time history of the range measurements and implementing a simple hypothesis test, it can be determined if the actual measurement corresponds to a LOS or NLOS environment. Basically, the standard deviation of the measurements is compared with an established threshold to decide if NLOS condition occurred.

There are some disadvantages that deny the use of this approach in Home Robot scenario: first, the measurement noise variance should be assumed as known and, most important, this methodology works well only if a dynamic NLOS is present, not identifying at all situations that present only NLOS measurements [5].

- *J. Borrás at all.* proposed a decision theoretic framework for NLOS identification. Their approach is based on the fact that the noise in range measurement is described by different statistical distributions depending on if LOS or NLOS situation is being faced. The core of the algorithm is a modelling phase of both NLOS and LOS error in which authors associate to each one different probability density functions (PDFs).

This type of approach could be useful only in those cases where error models are well known and well characterized. In Home Robot project no assumptions about NLOS error distribution has been done not to be lacking in generality [6].

- *A. Rabbachin at all.* approached the problem considering the energy point of view. Integrating signal energy on a relatively long integration time, an overview of total power of the signal can be available. By comparing the normalized strongest component in the received signal (that should be the direct path or LOS path) with a fixed threshold, a brief identification can be addressed.

Also in this case, the approach works properly only if optimal selection of key parameter (in this case the power threshold) is achieved. Considering that

signals power is strongly environment dependent, a robust selection criterion is difficult to assure [7].

- *I. Guvenc at all.* proposed a NLOS/LOS identification in UWB systems using the channel statistics. For each message exchange, UWB channel is the ensemble of all the signals received by receiver coming from the same signal sent by transmitter (i.e. the LOS path signal plus all the multipath components). The proposed technique analyzes both amplitude and delay statistics extracted by the UWB channel (such as kurtosis, mean excess delay and root mean square delay spread) developing a joint likelihood ratio test to state the presence or not of NLOS condition.

The approach to exploit information from UWB channel will be the one followed also in this thesis, but instead of using statistical features, that involve the extraction and the processing of all the UWB channel, punctual representative features will be extracted from UWB channel and subsequently processed, ensuring a short time consuming algorithm suitable for a real time application like Home Robot project [8].

### 2.3.2 NLOS mitigation

- *P.C. Chen* proposed a residual weighting algorithm to mitigate NLOS error. The paper assumes that the number of available fixed anchors is bigger than three: in this way, the final positioning estimation can be computed considering, in turn, a different combination of three anchors among the available set (knowing that to perform a trilateration operation at least three measurements should be processed). After each trilateration, residual ranging error is calculated. The correct position estimation is chosen keeping the one with smaller residual errors, weighting different estimations with the inverse of the associated residual errors.

The approach relies on the fact that the infrastructure counts more than three fixed anchors: this is a lack of generality considering that, for this thesis, only the basic kit with three fixed anchors has been used [9].

- *S. Venkatraman at all.* approached the problem adjusting the measured NLOS ranges to their closer true LOS values. The range adjustments are obtained by scaling the NLOS ranges with scale factors that are estimated from a constrained nonlinear optimization problem. This method does not require knowledge of NLOS error statistics or of a time history of range measurements. In this algorithm the scale factors are estimated exploiting a constrained nonlinear optimization problem derived from the geometrical topology of the system and so creating a bond too tight with the geometry of the anchors placement (that patient could move around his house everytime). Furthermore, the exploited constrained optimization procedure is the Sequential Quadratic

Programming (SQP) algorithm, whose computational complexity does not make it suitable for real time applications [10].

- *K. Ahmed et al.* proposed the use of a Kalman filter to smooth and mitigate NLOS effect. Here, to face NLOS measurements, a biased version of the Kalman filter is implemented. The working principle is intuitive: the positive range bias error introduced by NLOS condition is canceled by increasing the diagonal elements of noise covariance matrix of the filter. It means that if the filter is processing a previously NLOS classified measurement, it will increase the uncertainty related to that measure, giving more importance to the modelling part.

The use of a Kalman filter could lead surely to a great improvement in NLOS mitigation, but the difficulty in modeling the motion of the robot point by point makes it really challenging to implement efficiently for the Home Robot application [11].

# Chapter 3

## Materials and methods

UWB technology is widely used in indoor localization systems. In this thesis TREK1000 Evaluation Kit, a commercial kit by Decawave, has been used. In this chapter, first an overview on UWB technology is given, then a detail explanation of TREK1000 working principle is provided, both from the hardware and software point of view. At the end, an anticipation about Channel Impulse Response (CIR) characteristics is described: it will be, in the next chapter, the source from which features used in NLOS identification will be extracted.

### 3.1 UWB Technology

UWB is a kind of Radio Frequency technology that presents a large bandwidth and low energy level transmission protocol. As defined by the Federal Communication Commission (FCC), an UWB transmission has a 10-dB bandwidth greater than 500-MHz in the unlicensed frequency range of 3.1–10.6 GHz and the frequency in which the system has the maximum power density  $f_{max}$  has to be bigger than 2.5 GHz.

If  $f_{max}$  is lower than 2.5 GHz,  $B_{app}$  has to be bigger than 0.2, where:

$$B_{app} = \frac{B}{f_{max}} \quad (3.1)$$

where  $B$  is the bandwidth of the transmission.

To compute  $f_{max}$ , if  $f_H$  and  $f_L$  are the frequencies in which the power spectral density is 10-dB below the one in  $f_{max}$ , a simple formula can be used:

$$f_{max} = \frac{f_H + f_L}{2} \quad (3.2)$$

And, consequently,  $B_{app}$  can be derived from  $f_H$  and  $f_L$  easily:

$$B_{app} = \frac{2(f_H - f_L)}{f_H + f_L} \quad (3.3)$$

where  $B$  is considered as  $f_H - f_L$ .

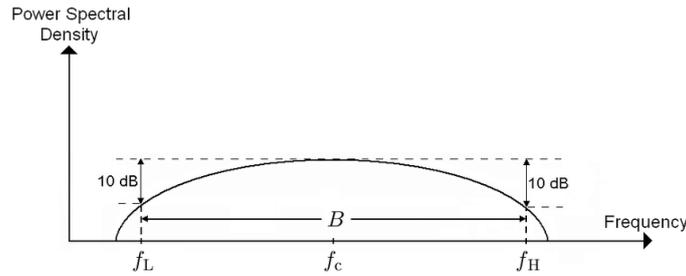


Figure 3.1: Typical bandwidth of an UWB signal.

### Shannon–Hartley theorem

For each Radio Frequency signal, Shannon–Hartley theorem describes the capacity of the signal channel. It declares that there is a direct relationship between capacity and bandwidth and, at the same time, there is also an inverse relationship between bandwidth and power consumption:

$$C = B * \log_2(1 + SNR) \quad (3.4)$$

where  $C$  is the channel capacity,  $B$  the bandwidth and  $SNR$  is the signal to noise ratio.

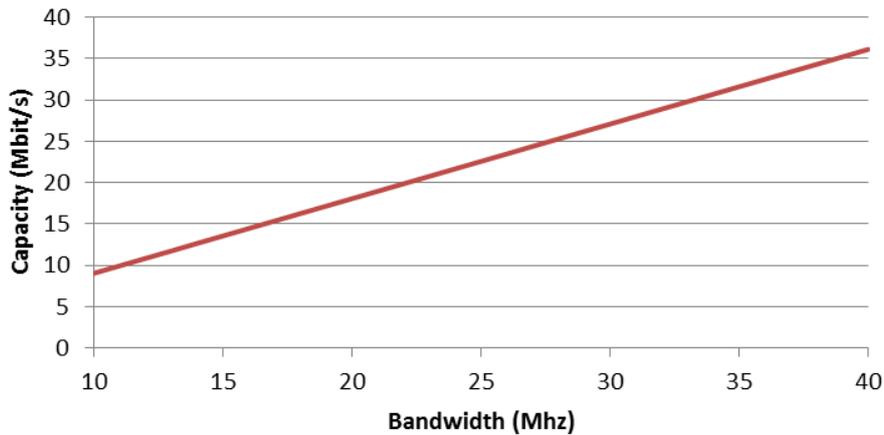


Figure 3.2: Capacity-Bandwidth relationship given a fixed SNR.

The most important result of this theorem is that, for a specific capacity, is possible to consume less power if a larger bandwidth is used: this is one of the most significant features of the UWB and therefore an indoor localization system with UWB devices is feasible.

### 3.1.1 UWB features and characteristics

UWB technology presents a lot of important features that make it suitable for ranging applications and, consequently, for the construction of positioning systems. The most important feature is the large bandwidth respect to other Radio Frequency technologies that are narrowband. Considering the reverse relationship between frequency domain and time domain, the direct result of having a large bandwidth is the short time duration of an UWB signal. This leads to an extremely accurate time resolution and this is a key point for ranging applications since makes the architecture robust against multipath phenomenon.

Another important feature is that UWB technology enables long-range communications since the smallest frequencies in UWB signals suffer of small attenuation and they can produce a long propagating electromagnetic wave.

Furthermore, the transmitted energy spectral density in an UWB signal is much lower than the one in a narrowband technology, cause the total energy is spread over a huge range of frequencies. This aspect permits to have few interference phenomena with other co-existing radio frequency systems.

Finally, UWB systems need simple and relatively cheap hardware: no up-conversion circuit is needed for the transmitter, neither an intermediate Frequency-conversion stage is needed on the receiver side. Moreover, also the use of frequency synthesizers or oscillators both on transmitter and receiver side is avoided [12].

### 3.1.2 Impulse Radio Ultra-Wide Band (IR UWB)

There are in literature a lot of different ways to transmit an UWB signal. All of them are part of two big families: one is based on a continuous transmission of multi-carrier signals, the other is based on the transmission of only short baseband pulses. The most used method for UWB transmission is the second: it can be called Impulse Radio Ultra-Wide Band (IR UWB) technique. In this case, the information that you want to insert in the signal is contained both in the polarity and in the time position of the signal inside the transmission frame.

To use IR UWB technique, the first thing to do is create UWB pulses. According with most of the related papers in the literature, an UWB pulse can be generated starting from a Gaussian pulse:

$$pulse(t) = \pm \frac{\sqrt{2}}{\alpha} e^{-\frac{2\pi t^2}{\alpha^2}} \quad (3.5)$$

with:

$$\alpha^2 = 4\pi\sigma^2 \quad (3.6)$$

where  $\alpha$  is called pulse factor and  $\sigma$  is the variance.

Looking at equation (3.5),  $\alpha$  is the parameter that adjusts the pulse width and that can give life to an UWB pulse: if  $\alpha$  is big, the related width of the pulse will be small and vice versa. To create an UWB signal, pulses should have a time duration width of fractions of nanoseconds [13].

### Modulation Techniques

To insert information in every kind of electromagnetic signal, a modulation technique is needed. Concerning UWB technology, to avoid potential interferences with other electromagnetic systems, randomizing techniques are applied to signals. The two main randomizing modulations used for UWB signals are Time-Hopping (TH) and Direct-Sequence (DS) modulation. TH works randomizing time position of the transmitted UWB pulses: each data package is encoded in the signal transmitting multiple time-shifted radio impulses. On the contrary, DS works transmitting continuous pulses to create a single data bit: the final signal will be a continuous train of UWB pulses, whose number depends on the specific bit rate of the system in which the UWB device is working.

An example of TH transmission is in the figure below: in this case, IR UWB signal appears in one of the chip intervals ( $T_c$ ) inside the related frame ( $T_f$ ) [14].

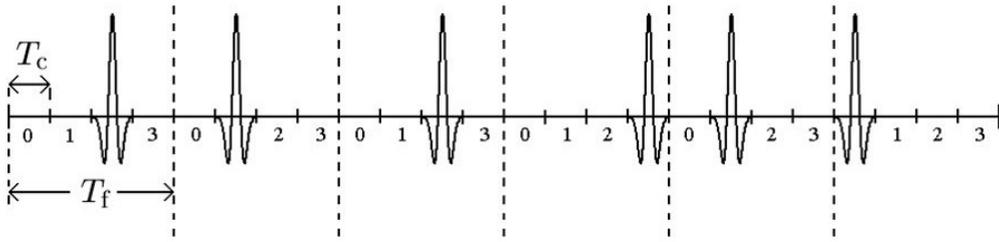


Figure 3.3: IR UWB signal with TH modulation.

### 3.1.3 UWB ranging techniques

Once signals have been created, techniques must be implemented to exploit them first to obtain distance information and then, by combining different distance information coming from different anchors, to estimate the position in the space of the transmitter.

The three most used ranging techniques that exploit UWB signals are Angle of Arrival (AoA), Received Signal Strength (RSS) and time-based techniques (ToA, TDOA, TWR).

- In the AoA technique, anchors compute and process the angle of arrival of the UWB signal coming from the tag to determine its distance. This approach requires omni-directional antennas or arrays of antennas (that are expensive) and, in most of the cases, the final accuracy is too low for Home Robot application. Often this method is used together with time-based measurements, in order to obtain an acceptable precision.
- Methods that use RSS technique rely on the assumption that every electromagnetic wave decays as the space traveled by the wave increases. It means that, knowing the original signal strength when the signal left the transmitter and measuring the one that the signal has when it reaches the receiver, the distance between transmitter and receiver can be deduced. These techniques are very subject to multipath effect and NLOS environments, that makes them not very reliable for the Home Robot application.
- Time-based techniques exploit the time the signal takes to travel from the transmitter to the receiver. Depending on the degree of synchronization of the nodes that make up the system, the used techniques are different.  
Time of Arrival (ToA) requires clock synchronization among all the nodes, usually achieved using wired connections, and simply computes time of flight of the signal comparing the timestamp of the receiver when the signal arrives and the timestamp of the transmitter when the signal has been sent.  
When synchronization is achieved only among anchors and no information are available about the clock of the tag, Time Difference of Arrival (TDOA) is the used technique. In this case, signal time of flight is computed comparing timestamps from different anchors when they receive the same signal.  
When synchronization is not possible at all, the used technique is called Asynchronous ToA, also known as Two Way Ranging (TWR). TWR is the technique exploited by the commercial kit used in this thesis and it is going to be better explained in the next section.

### Time-based ranging techniques

In time-based ranging techniques the distance between two devices (anchor and tag) is computed measuring the propagation time of the signal and assuming that it is travelling at a known speed. The principle is intuitive: knowing the travelled time  $t$  and the wave speed  $c$ , the distance  $d$  between two nodes can be deduced with a simple formula:

$$d = t * c$$

where  $t$  is usually computed as difference of timestamps.

For example, sending a UWB signal from the transmitter and recording the timestamp of emission  $t_1$ , it will travel until the receiver and, once arrived, receiver will send a response backward to the transmitter that can record the timestamp of reception  $t_2$ . The travelled time will be consequently computed as:

$$t = \frac{t_2 - t_1}{2}$$

The speed of an electromagnetic signal is comparable with the speed of light ( $300000\frac{Km}{s}$ ). Considering the orders of magnitude of the variables, the measure of  $t$  has to be very accurate: even a small error, multiplied by the speed of light, could lead to big disasters in the range estimation.

In order to have the maximum possible accuracy, ToA requires to have all the nodes synchronized with each other. In this situation, a single UWB signal between the transmitter and the receiver is enough to implement ranging application. As explained before, the receiver, once it gets the message, is able to estimate the range by subtracting the timestamp of the transmission, recorded by the transmitter and encapsulated in the message, from its own timestamp when the signal has been received. This approach is very basic but, since the nodes have to be synchronized and usually this is achieved by wiring all of them together, it is both expensive and not suitable for a home environment monitoring application.

If a global synchronization between nodes cannot be achieved at all, the ranging technique is called Two Way Ranging. Since it does not require the synchronization, at least the exchange of two messages is required. Calling  $t_{replyT}$  and  $t_{replyR}$  the times that transmitter and receiver take to process the incoming message and to prepare the response to be sent and setting them to constant values, when transmitter receives back the response message from receiver, it can compute the travelled time  $t$  by subtracting its transmission timestamp from its reception timestamp (together with  $t_{replyR}$ ) and then dividing by two:

$$t = \frac{t_{roundT} - t_{replyR}}{2}$$

where  $t_{roundT}$  is:

$$t_{roundT} = 2t + t_{replyR}$$

A more robust approach can be implemented including one more message: in this way the computed propagation time is less affected by errors introduced by clock reference crystal.

This approach is called *Symmetrical Double Sided Two Way Ranging* (SDS-TWR) (Figure 3.4). In this case, the travelled time  $t$  is computed as:

$$t = \frac{t_{roundR} - t_{replyR} + t_{roundT} - t_{replyT}}{4}$$

The commercial kit used in this thesis exploits the SDS-TWR approach, since its anchors are not synchronized and to achieve the maximum possible accuracy. As explained before, the fact that the nodes are not synchronized and wired together is a point of strength for this application: it makes the system more convenient and easier to be installed in a home environment.

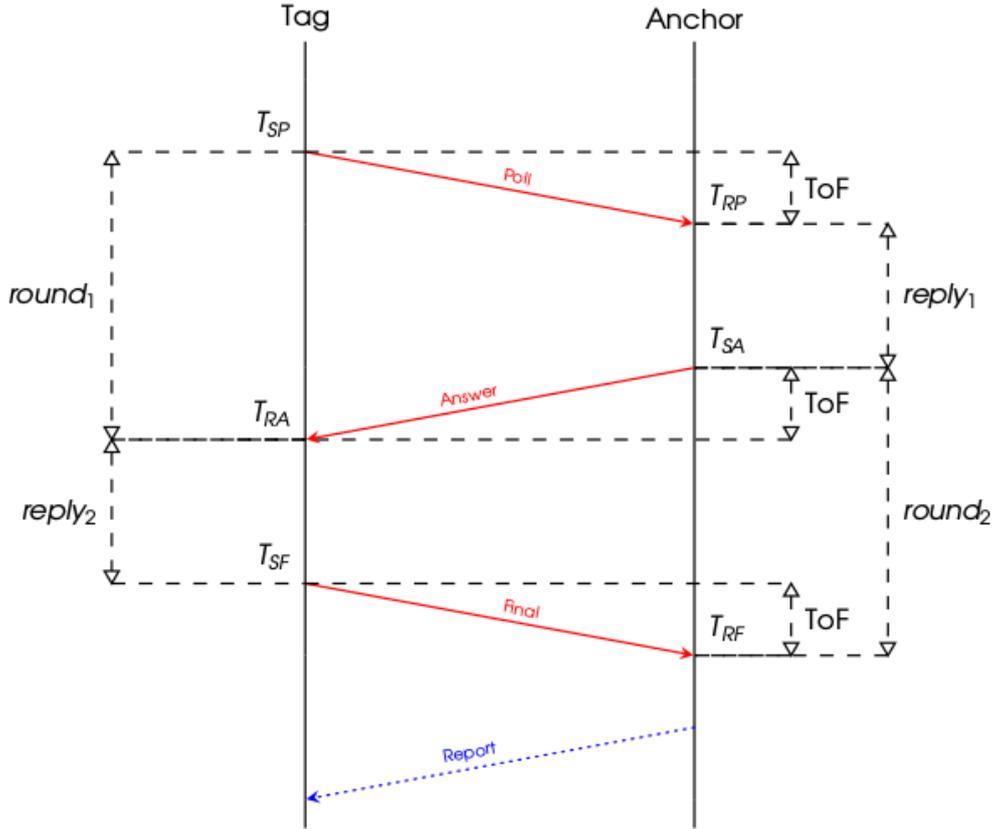


Figure 3.4: SDS-TWR working principle [15].

### 3.1.4 UWB trilateration process

Once each of the fixed anchors is able to provide its distance from the tag, the information has to be processed to find the position of the tag in the space. The number of fixed nodes that is necessary to compute the position of a tag depends on the dimensionality of the problem: to estimate a position in a 3-D space, at least four anchors are necessary, while in a 2-D space just three anchors are enough. In a 3-D space the solution can be obtained exploiting the intersection of four spheres while in a 2-D space the solution can be achieved computing the intersection of three circumferences.

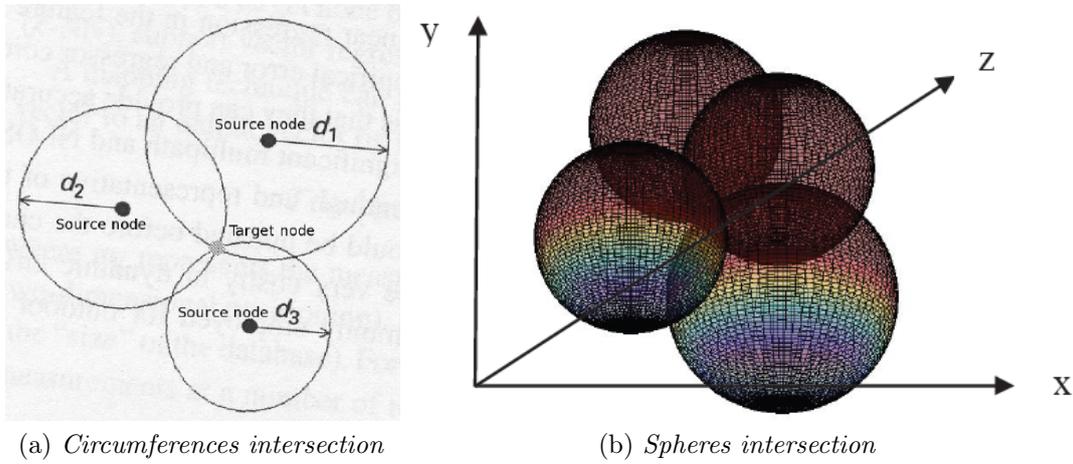


Figure 3.5: Position estimation in 2-D and 3-D spaces.

The mathematical process that allow to find the solution is called *trilateration*. The further explanation is given considering a 2-D problem for simplicity. The process exploits ranges coming from three anchors to determine the tag 2-D position and, if present, also the one coming from a fourth anchor to choose the right solution along the z-axis. If only three ranges are available, the solution is taken as the one below the anchors' plane (the z values will be chosen as negative), considering the assumption that the tag is moving on the ground floor while the anchors are fixed near the ceiling. The inputs of the trilateration algorithm are the 3-D coordinates of the anchors (that have to be known *a priori*) and the measured ranges obtained exploiting the TWR process. The tag position will be found considering this geometric problem as a problem of spheres intersection and solving it analytically for the unknown variables x, y and z (Figure 3.6).

Taking as reference Figure 3.6, equations that represent the three spheres can be written as:

$$r_1^2 = x^2 + y^2 + z^2 \quad (3.7)$$

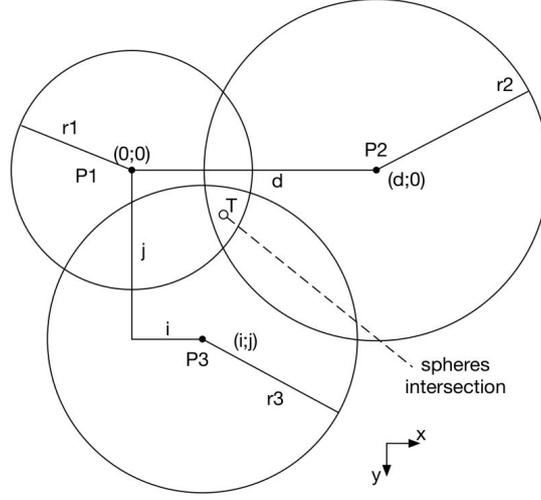


Figure 3.6: Intersection of three spheres in a 2-D space.

$$r_2^2 = (x - d)^2 + y^2 + z^2 \quad (3.8)$$

$$r_3^2 = (x - i)^2 + (y - j)^2 + z^2 \quad (3.9)$$

where  $d, i$  and  $j$  are the distances between the centers of second and third sphere and the center of the first one, located for simplicity in  $(0,0)$ .

The intersection between the three spheres is described by the triplet  $(x, y, z)$ . These coordinates can be obtained solving together the three equations (3.7, 3.8, 3.9).

First, equation 3.8 is subtracted from equation 3.7:

$$r_1^2 - r_2^2 = x^2 - (x - d)^2 \quad (3.10)$$

$$r_1^2 - r_2^2 = 2dx - d^2 \quad (3.11)$$

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d} \quad (3.12)$$

Since equation 3.12 is composed only of known values ( $r_1$  and  $r_2$  are the recorded ranges and  $d$  is known from the set up), tag position along x-axis has been found.

Writing equation 3.7 as:

$$z^2 = r_1^2 - x^2 - y^2 \quad (3.13)$$

and replacing equation 3.13 in equation 3.9 with few mathematical passages:

$$r_3^2 = (x - i)^2 + (y - j)^2 + r_1^2 - x^2 - y^2 \quad (3.14)$$

$$y = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x \quad (3.15)$$

Since  $x$  value is known from equation 3.12, also tag position along  $y$ -axis has been determined.

Theoretically, last step is the determination of the tag position along  $z$ -axis. From equation 3.7:

$$z = \pm \sqrt{r_1^2 - x^2 - y^2} \quad (3.16)$$

As explained above, equation 3.16 admits two possible solutions: one above the anchors' plane and one below. To choose between the two possible solutions or a fourth range (if available) will be used or the solution that places the tag below the plane containing the anchors will be chosen by default.

Ideally, the solution of the intersection of three spheres will be a univocal point. However, ranges coming from ranging processes described in the previous section are always affected by errors. This means that, during the trilateration process, the intersection of both spheres and circumferences will not provide a single specific point, but it will result in an area that describes all the possible positions in which the tag could be located (Figure 3.7).

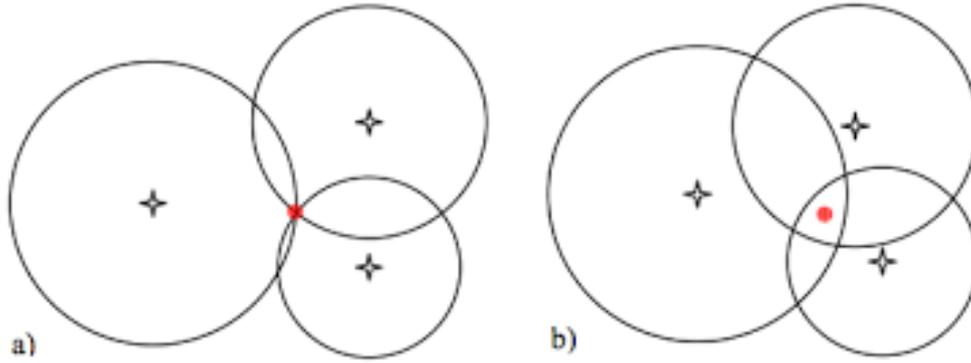


Figure 3.7: Trilateration with ideal measurements (a) and with noisy measurements (b).

Geometric approach described above is not able to solve this problem in presence of noisy ranges. In this situation, a lot of different statistical approaches have been developed in literature. Two main families are the most used: iterative algorithms and closed-form algorithms. Iterative algorithms must be performed several times

to converge to a satisfactory result, usually until a stopping criterion is achieved, while closed-form algorithms lead to a result immediately, often in a less precise way.

An example of Least-Squares iterative algorithm will be described in the next chapter to perform the NLOS mitigation application.

### 3.1.5 UWB regulations

UWB technology is a license-free technology: anyone can implement an UWB system without having a pending patent. However, since UWB technology covers a huge band of frequencies, the necessity of a regulation to avoid interferences among different kind of electromagnetic waves has been felt in recent years.

In 2002 Federal Communication Commission (FCC), a federal independent agency that regulates communications, allowed unlicensed operation between 3.1 GHz and 10.6 GHz for UWB technology. It regulated the use of an appropriate wideband signal format: the allowable output level for UWB transmission (described by the Equivalent Isotropically Radiated Power, EIRP) has been set at  $-41.3 \frac{dBm}{MHz}$  and the spectrum from 3.1 to 10.6 GHz has been divided in 14 different channels, each 500 MHz wide. To compare this power level with the one of other technologies, it is the same level allowed for the noise emission of all the other electronic devices that use different kind of electromagnetic waves (Figure 3.8).

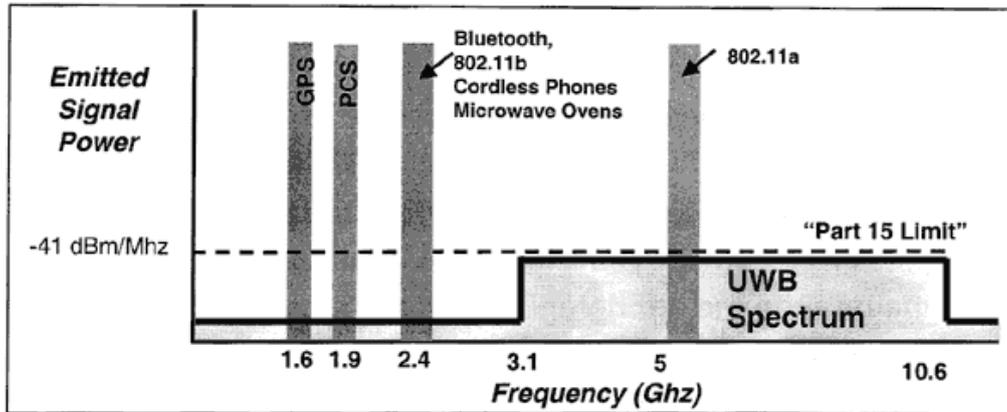


Figure 3.8: FCC regulation for emitted signal power in UWB systems.

However, the spectral mask that identifies allowed frequency bandwidths depends on regions: while in USA regulations are more relaxed, in European and Asian countries they tend to be more strict. As explained above, the UWB spectrum is divided in 14 different channels and many countries do not allow the use of all of them. In some channels, for instance, detect-and-avoid strategies in the UWB transmitter are required. However, the band from 7.25 GHz to 8.5 GHz is the only allowed common part in the spectrum (Figure 3.9) [16].

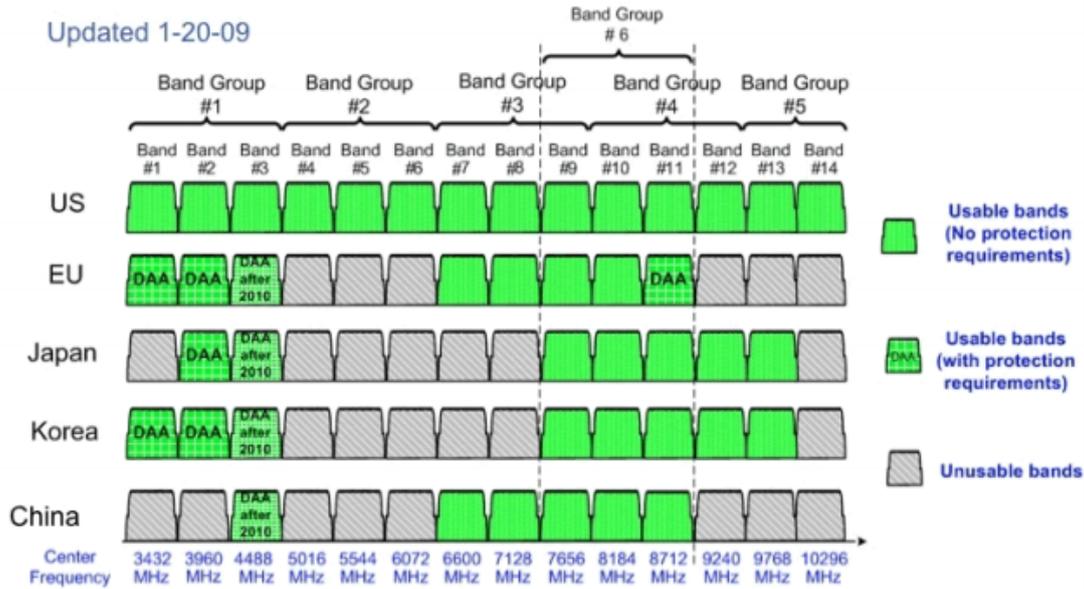


Figure 3.9: Allowed UWB channels in different countries.

In Figure 3.9 "DAA" stands for *Detect and avoid*: it is the technique that has to be implemented to avoid interferences between used UWB transmitter and other electromagnetic waves present in the environment. It is necessary to let UWB technologies use the designed channel without interfering with other systems that exploit same frequencies.

### 3.1.6 Commercial Ultra-Wide Band infrastructures

Nowadays a lot of different commercial UWB systems can be found on the market. They are different from the point of view of cost, dimension, range, accuracy. Here a quick overview about the most used commercial infrastructures is given.

- **Zebra UWB Technology** (Figure 3.10) is designed for accurate, precise, high-update-rate real time positioning systems. This system is compliant with the international UWB standard IEEE 802.15.4.f. Its datasheet declares exceptional performance in strong multipath environments, an overall location accuracy of 30cm in LOS condition, a long battery life up to 7 years with a sampling frequency of 1 Hz, a long-range localization up to 200 m, a programmable sampling frequency up to 200 Hz, waterproof antennas and a fast and intuitive set up of the system. The most important limitation is the overall price of the kit: 12000 USD [17].
- **Pozyx UWB system** (Figure 3.11) provides motion and positioning information with an accuracy less than 1m, integrating UWB technology with particle



Figure 3.10: Zebra Ultra-Wide Band Technology.

filters and machine learning techniques. The system supports up to 8 anchors with an update rate up to 125 Hz (considering a platform with a single tag). Both anchors and tag are compliant with Raspberry Pi and Arduino and the system is equipped also with other external sensors (accelerometer, gyroscope, magnetometer and pressure sensor). Integration with external software is achieved exploiting MQTT protocol and web-based applications. The declared accuracy is 10cm, while the maximum range is typically around 30m. The overall price for the kit is about 740 USD [18].

- **Decawave DW1000** is an integrated UWB transceiver for indoor real time localization applications. It is IEEE 802.15.4-2011 compliant and supports six different radio frequency bands (from 3.5 GHz to 6.5 GHz). Datasheet declares a precision of 10cm. Available data rates are 110 kbps, 850 kbps, 6.8 Mbps. Communication range up to 290m at 110 kbps and high multipath robustness. Declared low power consumption ( $1\mu\text{A}$  sleep mode current) makes this component particularly suitable for monitoring application such as Home Robot project. TREK1000 is the Decawave Evaluation Kit that mounts the DW1000 transceiver and it will be the one selected in this thesis (Figure 3.12). The cost for the overall kit is 900 USD [19].

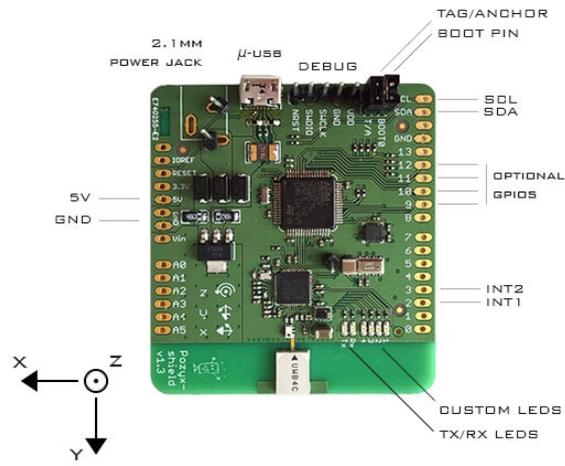


Figure 3.11: Pozyx UWB board.

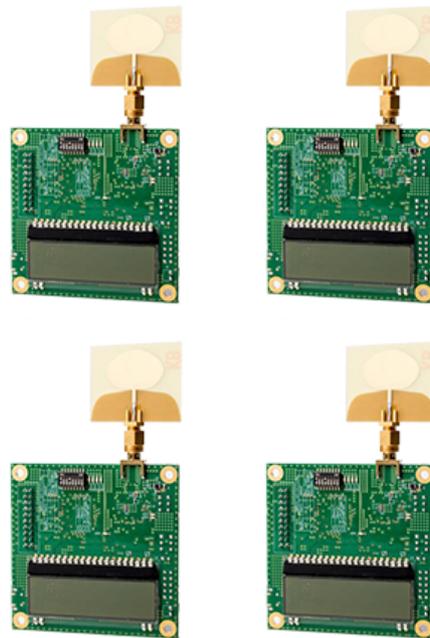


Figure 3.12: Decawave TREK1000 Evaluation Kit.

## 3.2 Decawave TREK1000 Evaluation Kit

Decawave Real Time Localization System (RTL5) is the Wireless Sensor Network (WSN) application used in this thesis to localize both the Home Robot and the patient. It is a core part of Home Robot project considering that it allows the robot to reach the patient and to implement its rescue activities exploiting the software embedded in the notebook. For this thesis, TREK1000 Evaluation Kit has been chosen and its features will be explained in the next sections both from the hardware and software point of view.

### 3.2.1 Hardware features

TREK1000 Evaluation Kit is composed of four EVB1000 (electronic boards that have to manage the DW1000 UWB transceiver), four UWB antennas, stands and USB cables to power PCBs and to allow the connection with the notebook. The kit provide a real-time indoor localization solution exploiting UWB technology. The used ranging technique is the SDS-TWR, since it ensures high performance without the necessity of a wired synchronization. TREK1000 output is the tag position in the 3-D space: it provides  $(x, y, z)$  coordinates. Considering that to achieve a 3-D localization four anchors are necessary but only four boards are contained in the kit (three anchors and one tag), Decawave trilateration software considers the solution of the spheres intersection the one with the negative z-coordinate: the assumption, as explained above, is that the tag is moving below the anchors plane. However, there is the possibility to expand the original kit adding more EVB1000 boards, up to have a system with four different anchors and eight different tags simultaneously. The introduction of the fourth anchors is useful both to increase the overall localization accuracy and to identify uniquely if the tag is above or below the anchors' plane.

TREK1000 can work in different application use cases:

- **Tracking use case** Three EVB1000 are used as anchors, one EVB1000 is used as tag. The task is determining the location of the Tag respect to fixed Anchors. Examples of implementation include asset-tracking or factory automation. The notebook is connected to one of the three fixed anchors to retrieve ranging and position information. This is the set up used during the thesis to implement all the simulations that will be explained in the fourth chapter (Figure 3.14).
- **Geo-Fencing use case** One EVB1000 is used as anchor, three EVB1000 are used as tags. The task is determining when tags enter or leave specific areas/perimeters close to a fixed anchor. Examples of implementations include child monitoring, security and personal safety. Also in this case the notebook is connected with the reference anchor. This set up has never been used during the thesis (Figure 3.15).

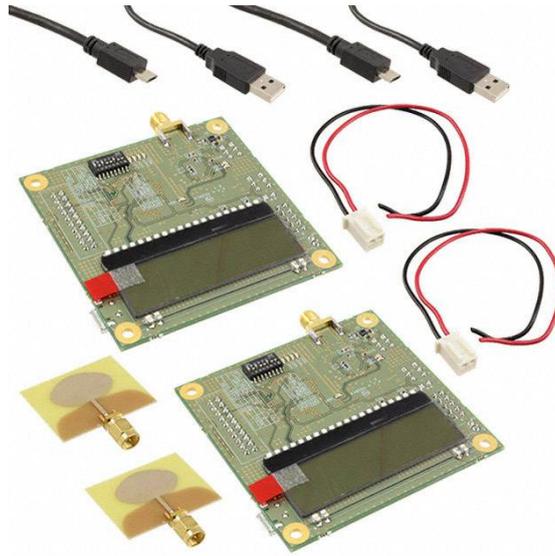


Figure 3.13: Decawave TREK1000 Evaluation Kit

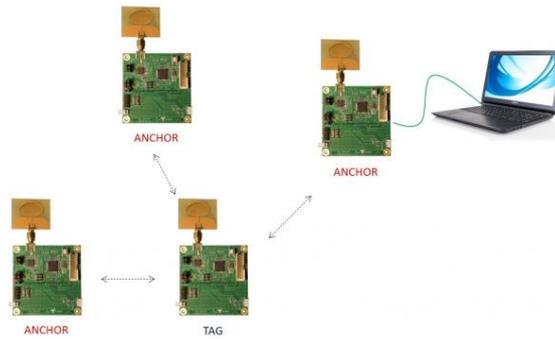


Figure 3.14: TREK1000 Tracking Use Case

- Navigation use case** Three EVB1000 are used as anchors, one EVB1000 is used as tag. The task is tracking the 2-D or 3-D location of the tag respect to the fixed anchors. Examples of implementations include human and robotic navigation. The notebook is connected to the tag to retrieve ranging and position information. This is the set up that will be used in the final Home Robot application: the robot will be equipped with one tag connected to the notebook that has to guide the mobile unit, another tag will be placed on the patient and the fixed anchors will be set all around the house (Figure 3.16).

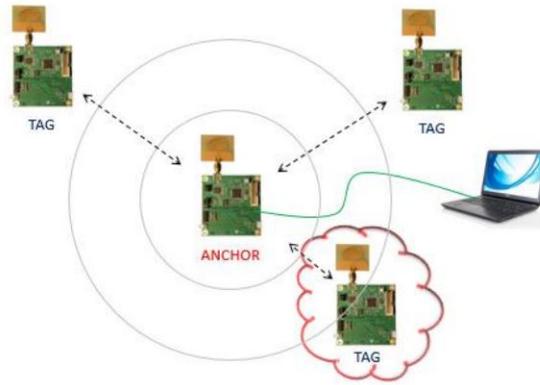


Figure 3.15: TREK1000 Geo-Fencing Use Case

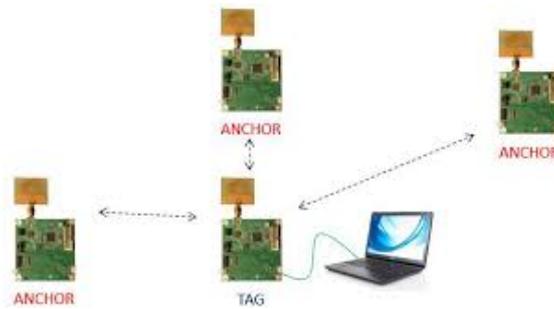


Figure 3.16: TREK1000 Navigation Use Case

## EVB1000

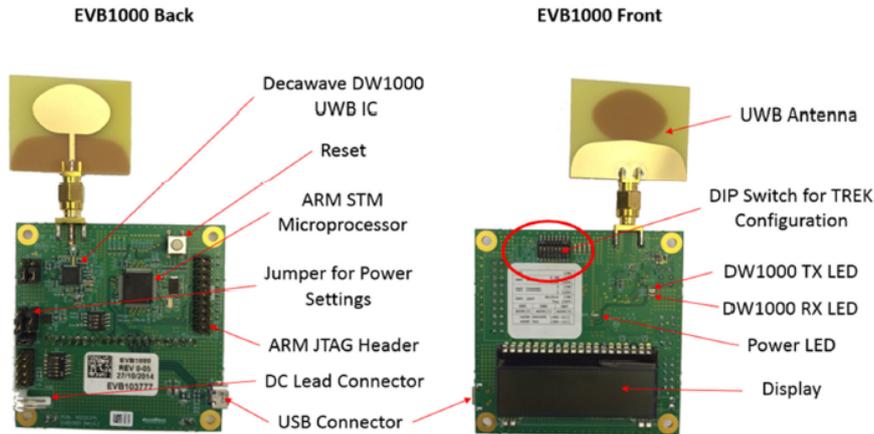


Figure 3.17: EVB1000 front and back view

EVB1000 is the 7cm x 7cm electronic board containing the DW1000 UWB transceiver that is the core of TREK1000. The PCB operation is coordinated with a STM32F105 ARM Cortex M3 processor that manages the DW1000 work through a SPI interface. The microcontroller can be programmed using a JTAG connector. The board can be powered in two ways: using a micro USB connector or directly with a DC 3.3 V power input. The omni-directional UWB antenna is connected to the board with a SMA connector.

From Figure 3.17, it can be seen that, in the front side, the board presents both a LCD display and an eight-pin DIP switch. On the LCD the last measured range is displayed, while the switch is necessary to configure the PCB. Each pin in the switch selects a different configuration:

- First pin is reserved and it is always set in the ON position
- Second pin is necessary to select the needed Data-Rate: 6.8 Mbps (ON position) or 110 kbps (OFF position)
- Third pin is necessary to select the UWB channel that you want to use: channel 5 - 6.5 GHz (ON position) or channel 2 - 4 GHz (OFF position)
- Fourth pin is necessary to select if the board will work as an anchor or as a tag: Anchor (ON position) or Tag (OFF position)
- Fifth-seventh pins are necessary to represent the binary number that distinguishes all the components: Anchor (0-3) or Tag (0-7)
- Eighth pin is reserved and it is always set in the ON position

## DW1000

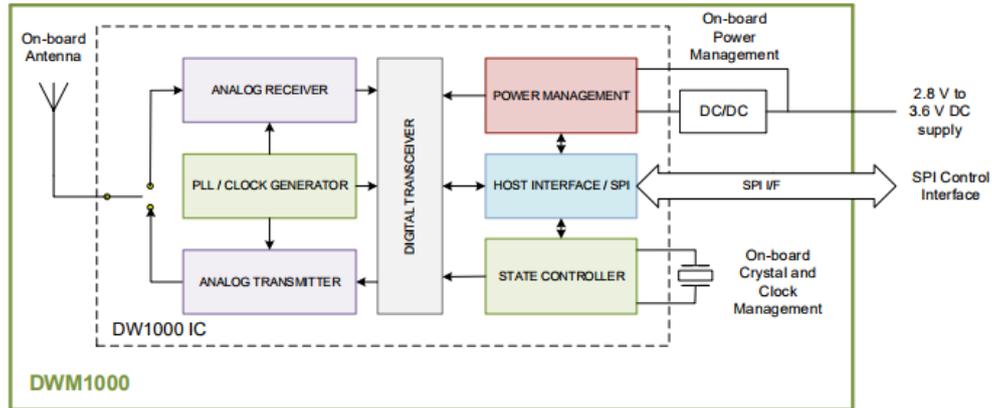


Figure 3.18: DW1000 UWB transceiver schematic view

DW1000 is the Integrated-Circuit UWB transceiver manufactured by Decawave. It manages the Radio-Frequency circuit, power supply, antenna operation and clock temporization. There is an on-board integrated antenna but, in the EVB1000 board, the transceiver is connected with an external one to exploit all the improvements coming from the use of a bigger antenna. It is compliant with the IEEE 802.15.4-2011 standard: it means that DW1000 can work on four different Radio-Frequency bands, from 3 GHz to 6.5 GHz. The transceiver presents also a SPI interface to allow communications with an external microcontroller.

Since all the key components are integrated, no external Radio-Frequency circuit is necessary. Other advantages are the low power consumption and the relatively low cost (about 20 USD).

The clock temporization is entrusted to a 38.4 MHz reference crystal: its declared error in the datasheet is 2 ppm. The precision of the clock is a key point for a RTLS application because no synchronization is implemented between nodes.

Furthermore, the integrated circuit presents also an OTP (One Time Programmable) memory: usually it is used to save information about antenna calibration.

### 3.2.2 Software features

In this section TREK1000 software will be analyzed from both Message Exchange Scheme (MES) and TWR technique point of view.

#### Message Exchange Scheme

The fact that anchors and tags (both the one on the robot and the one on the patient) exchange UWB messages each other is the key point to provide the localization

process [20].

The Message Exchange Scheme follows a Time Division Multiplexing (TDM) organization: each TMD cycle is called *superframe* and it is made of a fixed number of equally long time slots. In each of these time slots, only one tag is allowed to send messages on the selected UWB channel (while other ones, if present, are sleeping). A complete localization process is implemented during each time slot (Figure 3.19). Anchor0 (the one with the fifth-seventh pin of the DIP switch set on "0-0-0") is the manager of the communication: it is the component that has the task to make each tag remaining within its assigned time slot to perform a correct trilateration without interferences.

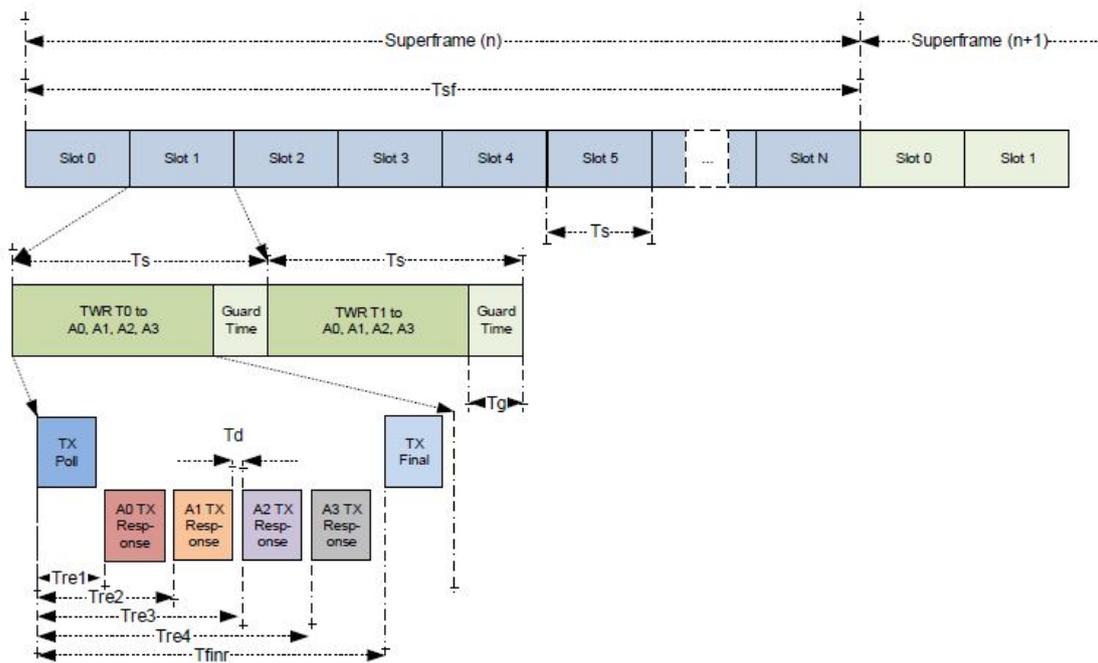


Figure 3.19: Superframe organization in the Decawave RTLS

In each time slot, a TWR process is implemented. To exploit TWR, UWB signals are obviously necessary. When there are more than two nodes that are communicating to exploit TWR (in this case, for example, there are at least three different TWR applications, one for each couple Tag-Anchor), simple signals are not enough: real messages are needed to univocally clarify both the sender and the receiver of each specific UWB signal, to avoid misunderstanding that would lead to wrong final TWR results. Therefore, a standard message format is needed. The format followed by Decawave UWB messages is the IEEE 802.15.4 (Figure 3.20).

- The two Frame Control (FC) octets are constant, as well as the two PAN ID octets.

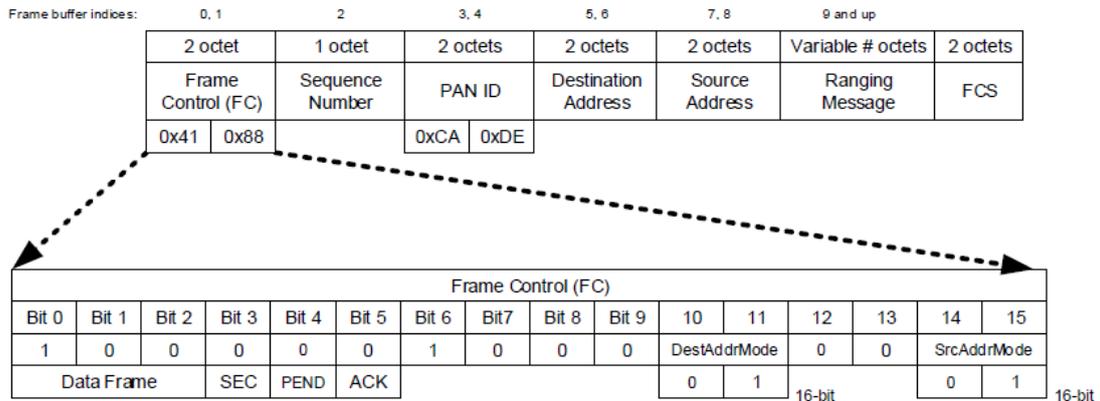


Figure 3.20: Decawave message format

- The Sequence Number octet is incremented modulo-256 every time a message is sent.
- The Source Address and Destination Address octets depend on the DIP switch configuration for each EVB1000 board.
- The two FCS octets are control bytes generated by DW1000 and attached at the end of each message
- The variable Ranging Message octets depend on the kind of message that is sent

Three different kinds of message are used in the Decawave TWR application and Ranging Message octets change for each of them (Figure 3.21).

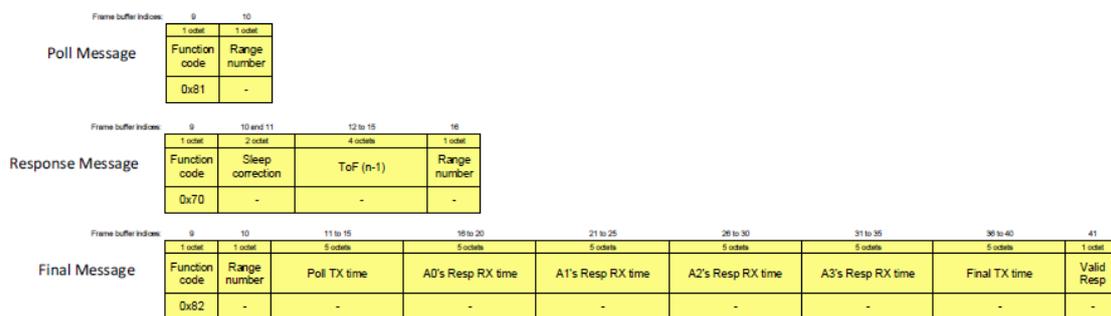


Figure 3.21: Different type of messages exploited during TWR application

- **Poll Message** The Poll Message is sent by the tag and it has the task to start the TWR application. Its function code is 0x81, that identifies univocally the

Poll Message. The range number depends on the sequence: it is incremented every time that specific tag sends a Poll Message.

- **Response Message** The Response Message is sent by each anchor, triggered by the reception of a Poll Message. It is made of eight octets. The first octet contains the specific function code that identifies Response Message (0x70). The second and the third are the Sleep correction: it is a parameter that adjusts Tag's sleep duration not to interfere with the activities of other tags. As explained before, this task is managed by Anchor0: Anchor0, called *gateway anchor*, is the only one that will set these bytes, all other anchors will set this field to 0 in their Response Message. From the fourth to the seventh octets, there is the field where ToF of previous message exchange is inserted. The last octet contains the sequence range number corresponding to the ToF reported before.
- **Final Message** The Final Message is sent by the tag after receiving all the Response Messages and to end the current message exchange. It is made of 44 octets. The first octet contains the specific function code that identifies Final Message (0x82). The second octet is the range sequence number (corresponding to the one sent in the Poll Message). Then there are 25 octets that summarize all the timestamps (5 octets each): the timestamp for the tag's Poll Message, the timestamp for the response from Anchor0, the timestamp for the response from Anchor1, the timestamp for the response from Anchor2 and eventually also the timestamp for the response from Anchor3 (if present). Following 5 octets contain also the timestamp for the actual tag's Final Message (pre-calculated before the message was sent). The last octet specifies which of the reported timestamps is considered enough reliable by the tag.

## TWR application

The final localization application is achieved applying first the TWR technique (as described in 3.1.3) and then the positioning algorithm (as described in 3.1.4). While the trilateration process has been described in depth before, the TWR has been only mentioned. In this chapter, some more details are given.

The overall process is intuitive. The Tag, that is the component that starts the TWR process, sends the Poll Message, expecting to receive back three (or four) responses by the fixed anchors. Then, after sending the Final Message, it will enter in sleeping mode to save energy. After waiting for the start of the next superframe, when it will be again the turn of its associated time slot in the current superframe, it will wake up and it will try again to exploit the TWR process.

In Figure 3.22 Decawave TWR architecture is shown. The tag sends the Poll Message, that is received by three (or four) anchors. Then, all the alive anchors

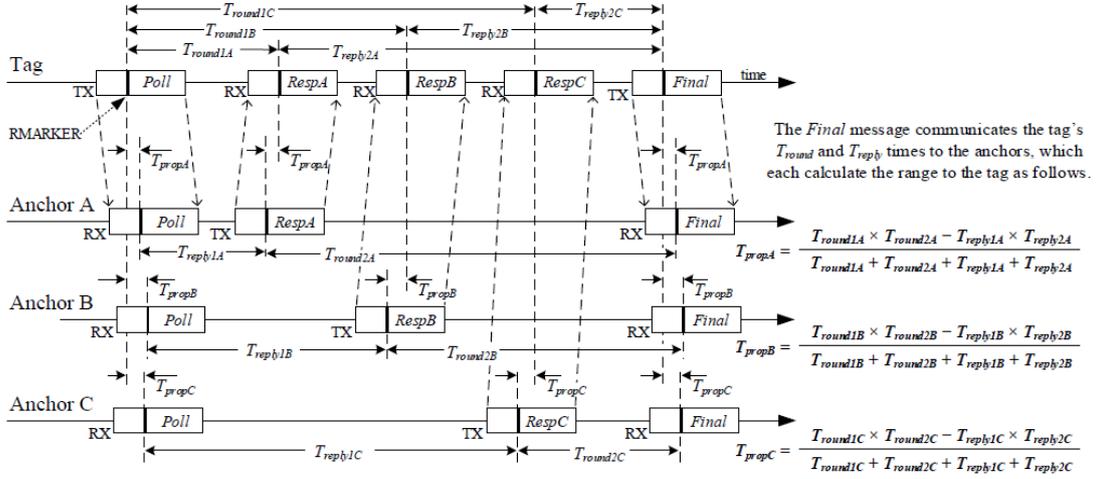


Figure 3.22: Decawave TWR algorithm overview

reply sequentially: with RespA, RespB and RespC the three Response Messages are indicated. To conclude the ranging process, the Tag sends its Final Message, that will be received by all the alive anchors in the infrastructure. This implementation allows the TREK1000 Evaluation Kit to locate the Tag using only 5 UWB signals: two are sent by the Tag, three are sent by the three different anchors. The low number of exchanged messages produces a great saving in terms of battery consumption.

An important feature of this technique is the role played by delays called  $T_{reply}$ .  $T_{reply}$  are known constant delays, inserted in the firmware and used by the software to synchronize Response Messages and to avoid overlapping responses. From the point of view of the tag, knowing that  $T_{reply1A} < T_{reply1B} < T_{reply1C}$ , it is easy and quick to be aware of the temporal sequence of the Response Messages: errors about the identity of the sender of each Response Message are completely avoided. Furthermore, the introduction of these delays is essential to manage the processing time of the signal and to solve potentially related issues: the time that passes from when the signal is processed by the microcontroller to when it is sent by the antenna is not negligible and it could give problems if it is not taken into consideration.

This architecture described above is the starting point from which this thesis will start to extract the features to implement NLOS identification. All the changes to the just described structure will be explained in the next chapter, together with the explanation of the proposed NLOS identification and mitigation algorithm.

### 3.3 Channel Impulse Response

In this section an introduction about Channel Impulse Response (CIR) is given. It is the core tool for the implementation of the NLOS identification algorithm.

In this thesis, NLOS identification is based on channel information: it will be proposed a low-complexity channel condition classifier to estimate the condition of the channel. The fact that Decawave UWB technology is compliant with 802.15.4a standard allows the capability to access the Channel Impulse Response (CIR) [21]. CIR is a diagnostic tool present in DW1000 that shows the LOS path and all the other multipath components (described in 2.2) for each message exchange: in the CIR amplitudes of all signals that arrive at the receiver side can be seen, both the one coming from the direct LOS path and the ones coming from reflections and refractions (Figure 3.23).

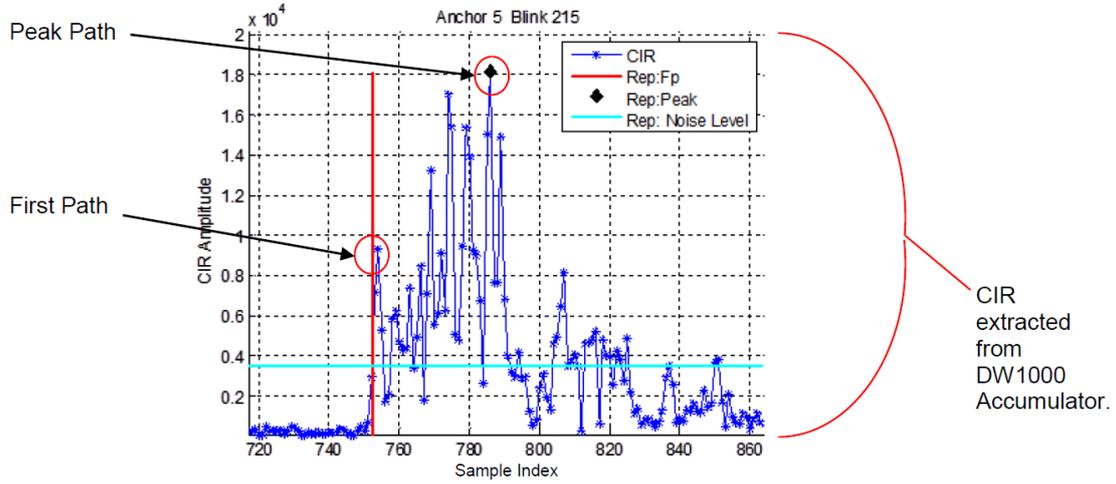


Figure 3.23: Decawave Channel Impulse Response

The idea is to exploit diagnostic channel information contained in the CIR to extract relevant features to implement a binary classification: at the end, the algorithm output will state if the current under analysis recording is a LOS or a NLOS recording.

In NLOS conditions, direct UWB signal usually cannot go through the obstacle without being strongly attenuated. The issue related to this situation is that the LOS signal could fall below the noise threshold set in the DW1000: in this case, the LOS signal will not be considered as a ranging signal. Signals that will take its place as valid ranging signals will be those coming from reflections and refractions. This phenomenon occurs also in LOS condition, but, in that case, the LOS signal is clearly perceived by the receiver and so all the other following ones are discarded and labelled as multipath components. If one of the multipath components is

considered as the valid ranging signal instead of the LOS one, negative consequences are intuitive: since refracted/reflected signal takes more time to arrive at the receiver side respect than the LOS one, the final computed Time of Flight will be greater and so, according with 3.1.3, also the computed distance between the two interested nodes will be greater than the actual one. This is why, as explained in 2.2, NLOS environment introduces a positive bias in the measured ranges and consequently it leads to a wrong final trilateration result. It is important to underline the involved orders of magnitude: if it is true that multipath components usually differ from the LOS signal of a few nanoseconds, it is also true that, since electromagnetic waves propagate at the speed of light, a delay of only one nanosecond leads to a ranging error of 30cm.

The NLOS identification is a critical issue: it is important to discard (if possible) or to correct NLOS ranges before implementing the trilateration process, to avoid the introduction of a significant noise in the localization algorithm. To achieve this identification, in this thesis, a quantitative characterization of the CIR is implemented, to extract information that can be informative about actual channel condition.

It must be remembered that the final Home Robot application is a real-time application. Since the extraction of the CIR from Decawave boards is relatively time consuming (if the TREK1000 Evaluation Kit is working with a sampling frequency of 10 Hz, it means that CIR must be extracted ten times per second, one time for each ranging process), a set of three features have been chosen as relevant and informative about CIR content, according with Decawave DW1000 datasheet [22]. Comparing LOS CIR and NLOS CIR, selected features can be better justified (Figure 3.24).

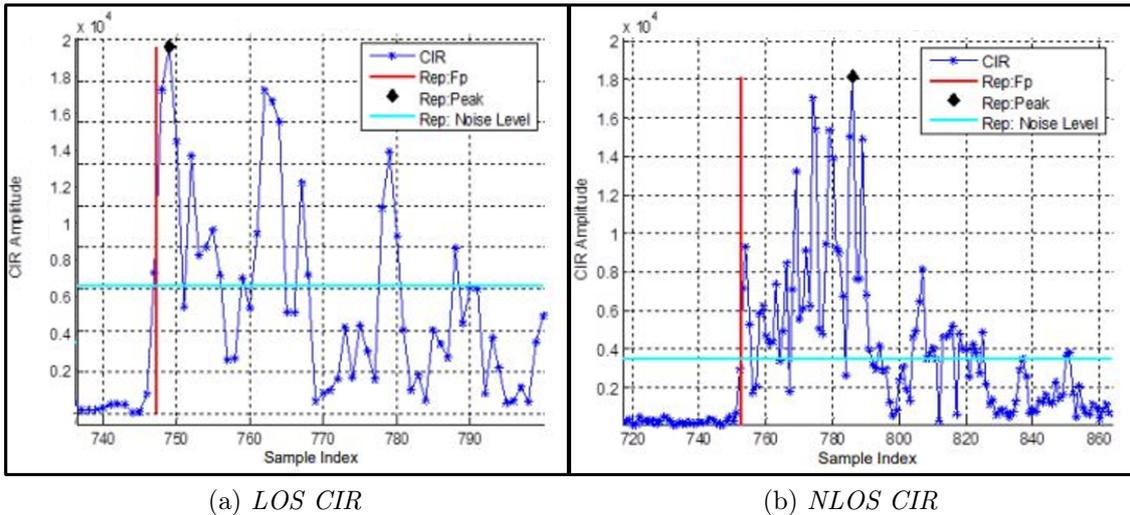


Figure 3.24: Comparison between CIR content in LOS and NLOS condition

- **Receive Signal power Level (RSL)** It is an estimation of the total power of all the signals that appear in the CIR. It is possible to compute an approximation of the receive power level (in dB) using the following simple formula:

$$RSL (dB) = 10 * \log_{10} \frac{C * 2^{17}}{N^2} - A \quad (3.17)$$

where:  $C$  is the Channel Impulse Response Power,  $N$  is the Preamble Accumulation Count and  $A$  is a predefined constant (115.72 for a Pulse Repetition Frequency of 16 MHz or 121.74 for a Pulse Repetition Frequency of 64 MHz).

Both  $C$  and  $N$  are 16-bit variables extracted from ARM microcontroller registers (further explanations will be given in the next chapter), while the *PulseRepetitionFrequency* (PRF) is the frequency at which UWB pulses are repeated (higher PRF gives more accuracy, but at the same time leads to additional power consumption).

Intuitively, the anchor that is experiencing a NLOS situation will present a lower (more negative) RSL. On the other side, it is also true that RSL decreases with the distance: not only the introduction of an obstacle, but also more time in the air, even if in a LOS condition, impacts the RSL value. Furthermore, from previous studies [23], it has been shown that RSL measurement is not totally reliable taken alone. Some LOS measurements can present NLOS RSL values: these sporadic outliers underline as RSL is not enough to perform a reliable NLOS identification.

- **Paths Time Difference** Also called *Rise Time* (RT) for simplicity, it is a parameter that takes into account the different temporal arrivals between the first path (that is the LOS one, considering that the direct LOS path is always the shortest and so it is covered in less time by the electromagnetic wave) and the peak path (defined as the signal in the CIR with the biggest amplitude). The Paths Time Difference is the time difference (in nanoseconds) between the first path and the peak path indexes in the CIR accumulator. Looking at Figure 3.24, it can be seen that the first path and the peak path in the LOS case are much closer than in the NLOS one: as it is easy to understand, in an ideal LOS condition the first path is expected to be also the peak path, so this parameter tends to be equal to 0. Typically, the larger is the RT value, the more probable is the NLOS situation. The simple formula that describes this parameter is:

$$RT (ns) = | PeakPathPosition - FirstPathPosition | \quad (3.18)$$

- **Paths Amplitude Difference** Also called *First Path Gain* (FPGain) for simplicity, it is a parameter that takes into account the different amplitudes

between the first path and the peak path. Specifically, the Paths Amplitude Difference is the ratio between the amplitude of the first path and the one of the peak path:

$$FPGain = \frac{FirstPathAmplitude}{PeakPathAmplitude} \quad (3.19)$$

where, to have a more robust estimation, the first path amplitude is considered as:

$$FirstPathAmplitude = \max(FPAmp1, FPAmp2, FPAmp3)$$

with  $FPAmp1$ ,  $FPAmp2$ ,  $FPAmp3$  that are the amplitudes of the first three arriving signals.

Looking at Figure 3.24, it can be seen that, in LOS condition, usually the peak path is one of the first three arriving signals: it means that FPGain is around 1 in LOS condition. On the contrary, in NLOS situations, due to the obstacle attenuation, the first path amplitude decreases and the FPGain takes a value lower than 1.

To summarize, the observation of the behaviour of these three parameters in LOS and NLOS condition allows to make two main assumptions:

1. In NLOS situations, signals are more attenuated and they have an overall smaller energy amplitude due to reflections and refractions, resulting in a smaller RSL value
2. In LOS situations, the strongest path typically corresponds with the first path, while in NLOS situations other weak components come before the strongest path, resulting in a higher RT value and in a smaller FPGain value

To compute the values of these three parameters described above, 16-bit variables from EVB1000 registers are necessary. The proposed extraction method will be described in the next chapter, together with the implementation of the classifier and the further NLOS mitigation algorithm.

## Chapter 4

# NLOS identification and mitigation algorithm

In this chapter the overall NLOS identification and mitigation algorithm will be described in depth.

The first section presents a high-level block diagram that synthesizes the algorithm working principle and some quick explanations are given. The second section is about the extraction from Decawave firmware of the three features described above (3.3). The third section will explain the construction of the classifier, exploiting also the related data collection. The last section will describe the proposed mitigation algorithm that, starting from the classification output, has the task to mitigate range error in those estimations affected by NLOS positive bias.

### 4.1 Overall algorithm working principle

The overall algorithm structure is described in the block diagram in Figure 4.1.

This algorithm must be considered as something that is repeated every time a location estimation is achieved. The described steps are implemented in each time slot (Figure 3.19), it means ten times per second since the TREK1000 sampling frequency, in this thesis, is set to 10 Hz. The algorithm has been tested with offline simulations, following the navigation set up (3.2.1). So, first of all, all the data have been collected with a MATLAB script that read USB port in which Anchor0 sent information, then the algorithm has been run on the stored data. However, a real-time application is not difficult to be achieved: it is just a matter of converting the MATLAB code used in this thesis in a JavaScript code that allows to process real-time data coming from USB. It must be underlined that it was an author's concern to check that the execution time of the algorithm is in line with the sampling frequency of the system in order not to create delays that could put the real-time factor at risk.

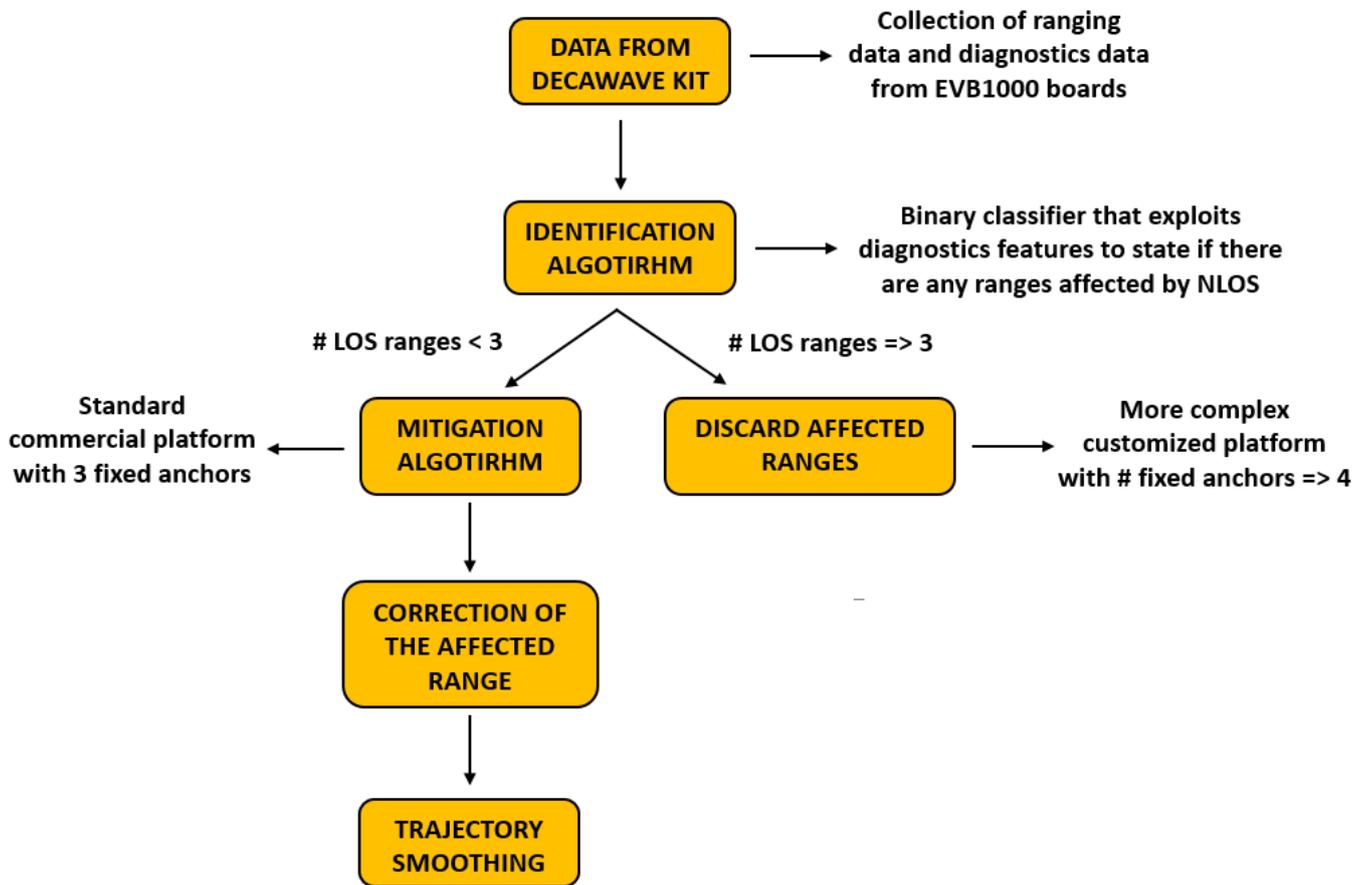


Figure 4.1: Overall algorithm block diagram

The first step is the data retrieving. The notebook is connected to Anchor0 with an USB cable. Anchor0 sends via USB both the ranging data (that is a default output of the Decawave original firmware) and the diagnostics data (basically the value of the variables that are necessary to compute the three features described in 3.3). Diagnostics data are not a standard output of the original TREK1000 firmware: to extract needed variables the firmware has been modified and all the editing will be explained in the next section.

Once both diagnostics and ranging data are available, the identification algorithm can be run. It consists in binary classifier, specifically a Support Vector Machine (SVM) classifier, previously trained on a dataset. Details about the construction of the classifier will be given further in this chapter. SVM is fed with diagnostic data coming from all the couple Tag-Anchor: each couple has its own features values that describe its UWB channel. The output is a binary result: each recorded range (it means each Tag-Anchor channel) is classified as LOS or NLOS.

Having classification results, the number of LOS ranges is evaluated. If, in the infrastructure, there is a number of LOS ranges equal or greater than three (that could happen if we have an infrastructure with three fixed anchors and all of them are in LOS condition or if we have an infrastructure with more fixed anchors but, anyway, at least three LOS anchors are present), the trilateration process could be implemented exploiting only those ones. So, the easiest way to mitigate NLOS effect is just discarding the affected ranges. If the number of LOS anchors is lower than three, the trilateration algorithm could not be implemented exploiting only those ones.

In this case, the mitigation algorithm is necessary. To correct final trilateration error coming from positive biased ranges an optimization algorithm has been chosen. A Taylor Series-based Weighted Least-Squares correction (TS-WLS) has been implemented. The algorithm acts on the final trilateration result: the Tag position is iteratively recalculated, correcting it at each iteration to minimize the difference between the reported original ranges from the anchors and the new ranges computed exploiting the new recalculated tag position. Unlike the original trilateration, this algorithm implements a weight process: the ranges that are previously classified as NLOS by SVM are not taken into account to compute correction factors (otherwise no improvement would be obtained). The algorithm runs until a stopping criterion is achieved: in this thesis, the stopping criterion is a predefined number of iterations, to preserve a reasonable execution time.

Once the new corrected trilateration is achieved, another two smoothing tools to avoid spurious behaviours in the final estimated trajectory are introduced. The first tool faces some problems that could happen in the transition phase between LOS and NLOS condition. It was seen by experimental data that CIR features have some weird behaviours when the tag passes from a LOS condition to a NLOS one and vice versa. This leads to misclassifications by SVM in those zones: sometimes last NLOS recordings are classified as LOS, other times last LOS recordings are classified as NLOS. To work around this problem, a simple method has been thought out: for a window of ten samples, centered on the recording that states the change of condition, the TS-WLS is run two times, considering in turn affected range both as LOS and as NLOS. Then is taken as final trilateration result the average of the two corrections. The second smoothing tool is a moving average along all the time series: each position is replaced with the average of the positions belonging to a window of five samples centered in the current one under analysis. Both smoothing tools introduce a delay: the transition phase correction introduce a delay of 10 samples (1 s) only during the transition phase, while the moving average introduce a delay of 5 samples (0.5 s) during all the trajectory. According to the author, these delays do not affect the real-time nature of the Home Robot application. All the mitigation algorithm will be described in detail in the last section of this chapter.

## 4.2 Features extraction method

This section has the aim to describe original firmware editing, introduced to make available the necessary variables for the calculation of the three features for the classifier construction.

All the needed variables have been extracted from DW1000 registers. To access them, a deep knowledge of how the original firmware is structured and how devices implement the TWR algorithm is necessary: a clear understanding of TREK1000 working principle is not trivial and challenging to achieve.

- For the computation of the Receive Signal power Level, the needed variables are  $C$  and  $N$ .  $C$  is available in the "RX Frame Quality Information" (RX\_FQUAL) register (Figure 4.2) with the name CIR\_PWR, while  $N$  is available in the "RX Frame Information" (RX\_FINFO) register (Figure 4.3) with the name RXPACC.

REG:12:00 – RX_FQUAL – Rx Frame Quality Information (Octets 0 to 3, 2x16-bit values)																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FP_AMPL2																STD_NOISE															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

REG:12:04 – RX_FQUAL – Rx Frame Quality Information (Octets 4 to 7, 2x16-bit values)																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CIR_PWR																PP_AMPL3															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.2: CIR\_PWR variable in the RX Frame Quality Information register

REG:10:00 – RX_FINFO – RX Frame Information																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXPACC								RXPSR	RXPRF	RNG	RXBR	RXNSPL	-	RXFLE	RXFLEN																
0								0	0	0	0	0	0	0	0																

Figure 4.3: RXPACC variable in the RX Frame Information register

- For the computation of the Paths Time Difference, the needed variables are *Peak Path Index* and *First Path Index*. *First Path Index* is available in the "Receive Time Stamp" (RX\_TIME) register (Figure 4.4) with the name FP\_INDEX, while the *Peak Path Index* is available in the "Leading Edge Detection Interface" register (Figure 4.5) with the name LDE\_PPINDX.

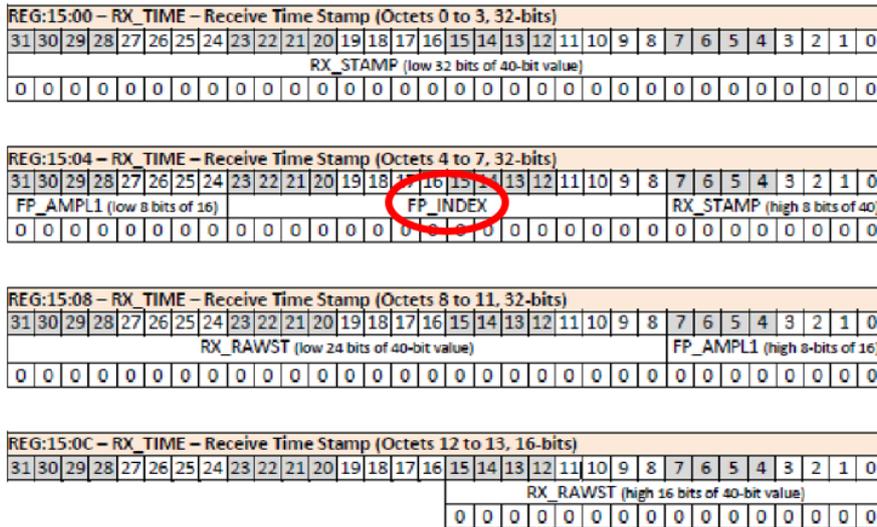


Figure 4.4: FP\_INDEX variable in the Receive Time Stamp register

OFFSET in Register 0x2E	Mnemonic	Description
0x0000	LDE_THRESH	LDE Threshold report
0x0806	LDE_CFG1	LDE Configuration Register 1
0x1000	LDE_PPINDX	LDE Peak Path Index
0x1002	LDE_PPAMPL	LDE Peak Path Amplitude
0x1804	LDE_RXANTD	LDE Receive Antenna Delay configuration
0x1806	LDE_CFG2	LDE Configuration Register 2
0x2804	LDE_REPC	LDE Replica Coefficient configuration

Figure 4.5: LDE\_PPINDX variable in the Leading Edge Detection Interface register

- For the computation of the Paths Amplitude Difference, the needed variables are  $FP_{Amp1}$ ,  $FP_{Amp2}$ ,  $FP_{Amp3}$  and  $PeakPathAmplitude$ .  $FP_{Amp1}$  is available in the "Receive Time Stamp" (RX\_TIME) register (Figure 4.6) with the name FP\_AMPL1,  $FP_{Amp2}$ ,  $FP_{Amp3}$  are available in the "RX Frame Quality Information" (RX\_FQUAL) register (Figure 4.7) with the name FP\_AMPL2 and FP\_AMPL3, while the  $PeakPathAmplitude$  is available in the "Leading Edge Detection Interface" register (Figure 4.8) with the name LDE\_PPAMPL.

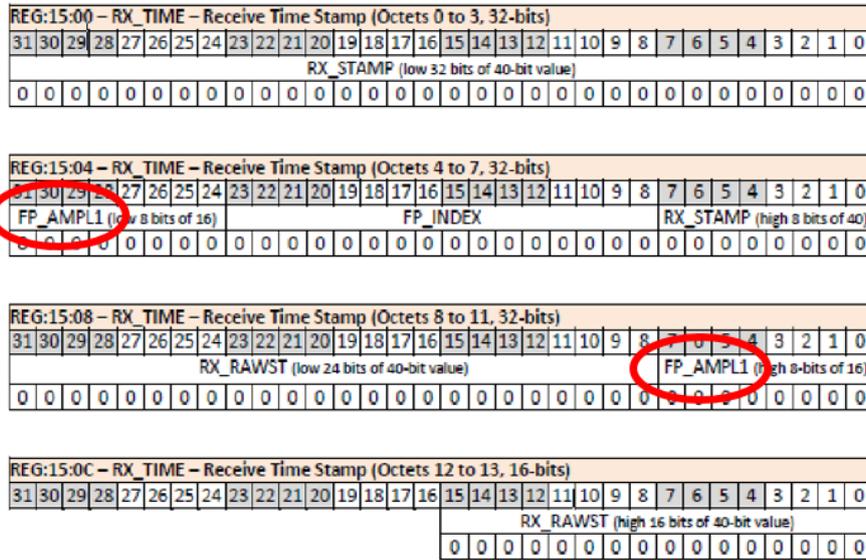


Figure 4.6: FP\_AMPL1 variable in the Receive Time Stamp register

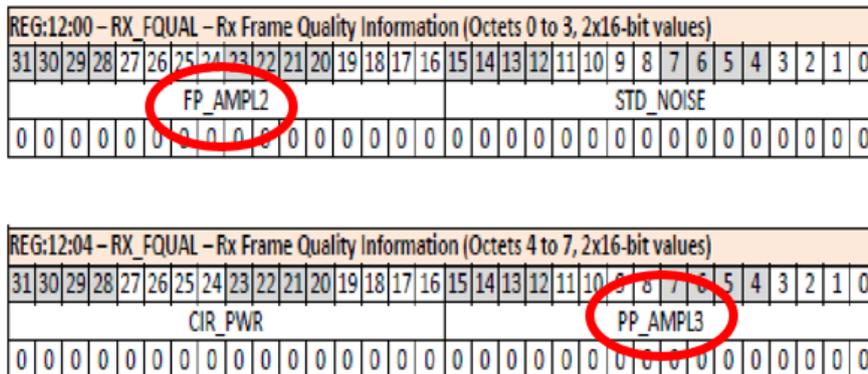


Figure 4.7: FP\_AMPL2 and FP\_AMPL3 variables in the RX Frame Quality Information register

OFFSET in Register 0x2E	Mnemonic	Description
0x0000	LDE_THRESH	LDE Threshold report
0x0806	LDE_CFG1	LDE Configuration Register 1
0x1000	LDE_PATHIDX	LDE Peak Path Index
0x1002	LDE_PPAMPL	LDE Peak Path Amplitude
0x1804	LDE_RXANTD	LDE Receive Antenna Delay configuration
0x1806	LDE_CFG2	LDE Configuration Register 2
0x2804	LDE_REPC	LDE Replica Coefficient configuration

Figure 4.8: LDE\_PPAMPL variable in the Leading Edge Detection Interface register

These necessary variables are retrieved always by the Tag: every time Tag receives a Response Message from an Anchor present in the architecture it reads bytes described above from appropriate registers. This process happens  $n$  times during a trilateration process, with  $n$  equal to the number of fixed anchors. Depending on who is the sender of the current Response Message (i.e. Anchor0, Anchor1 or Anchor2 if the architecture is the standard one), Tag saves variables values in different structs, to avoid mixing information coming from different channels. The used function to recover  $C$ ,  $N$ ,  $FP\text{Amp}1$ ,  $FP\text{Amp}2$ ,  $FP\text{Amp}3$  and  $First\ Path\ Index$  is:

$$\text{void } dwt\_readdiagnostics(dwt\_rxdiag\_t *diagnostics) \quad (4.1)$$

while the one used to retrieve  $PeakPathAmplitude$  and  $Peak\ Path\ Index$  is:

$$\text{uint16 } dwt\_read16bitoffsetreg(\text{int } regFileID, \text{int } regOffset) \quad (4.2)$$

Both of them are included in the Decawave ARM API. It is important to stress that these features are an overall summary indication of CIR information content: the advantage to use them is that a whole CIR reading is very time consuming and it is not necessary for this application.

Once the Tag have retrieved needed information, the following step is to make them available outside TREK1000 system (specifically, to the notebook). Being Anchor0 the only actor connected to the notebook (3.2.1) and being the data available only to the tag, a method to transfer information is necessary. The idea is to encapsulate the diagnostics data (i.e. the variable values) inside the ranging part of the Final Message (Figure 3.20). In this way, when Anchor0 receives the Final Message from the Tag, it can extract both ranging and diagnostic information and they can be sent via USB to the notebook to be processed (Figure 4.11).

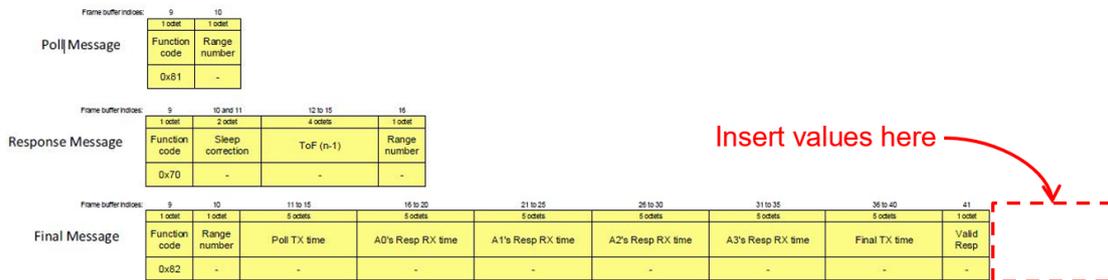


Figure 4.9: Diagnostic data are inserted at the end of the ranging part of the Final Message

Funciton Code	Poll TX Time	Resp TX Time	Final TX Time	F1	F2	F3	FP IND	N	C	PP AMP	PP IND
1 Byte	5 Bytes	5 Bytes	5 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes

Repeated for each couple Anchor-Tag

Figure 4.10: Disposition of diagnostic data inside Final Message

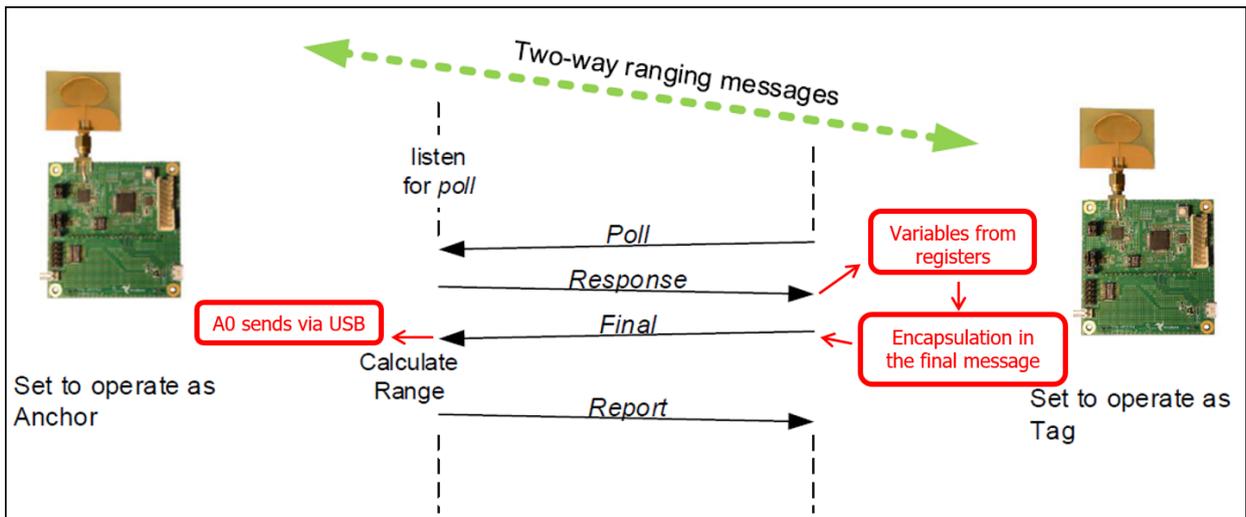


Figure 4.11: Introduced editing in the original TREK1000 firmware

To evaluate if the firmware editing gives expected results, Teraterm software has been used to sniff notebook USB incoming data: from Figure 4.12 it can be seen that, in addition to default ranging strings, now also the values of the parameters coming from all the combinations Tag-Anchor have been set as Decawave firmware output: every time a localization process happens, Anchor0 sends via USB both ranging measurements registered by all the anchors and diagnostic information regarding all the channels.

```

VT COM3 - Tera Term VT
File Edit Setup Control Window Help
ma 06 00000000 0000048e 000004bc 00000000 bd59 4a 0032bd4c a0:0
mc 05 0000034f 00000000 0000002a 00000000 bd5b d1 0032bd96 a1:0
mr 05 0000027d 00000000 ffffff45 00000000 bd5b d1 40474047 a1:0
FP1_AMP_HU: 17664, FP2_AMP_HU: 17570, FP3_AMP_HU: 10261, FP_IND_A0: 745.859375, N_A0: 958, C_A0: 24906, PP_AMP_A0: 17663, PP_IND_A0: 748
FP1_AMP_A1: 17335, FP2_AMP_A1: 17296, FP3_AMP_A1: 16660, FP_IND_A1: 746.156250, N_A1: 868, C_A1: 21569, PP_AMP_A1: 17334, PP_IND_A1: 749
FP1_AMP_A2: 17154, FP2_AMP_A2: 16860, FP3_AMP_A2: 15806, FP_IND_A2: 750.281250, N_A2: 1043, C_A2: 21410, PP_AMP_A2: 17153, PP_IND_A2: 753
ma 06 00000000 00000480 00000497 00000000 bd5d 4b 0032be64 a0:0
mc 05 00000341 00000000 0000002f 00000000 bd5f d2 0032beb0 a1:0
mr 05 0000026f 00000000 ffffff4a 00000000 bd5f d2 40474047 a1:0
FP1_AMP_A0: 11823, FP2_AMP_A0: 15092, FP3_AMP_A0: 16335, FP_IND_A0: 751.171875, N_A0: 1033, C_A0: 21538, PP_AMP_A0: 16334, PP_IND_A0: 752
FP1_AMP_A1: 16941, FP2_AMP_A1: 17207, FP3_AMP_A1: 15242, FP_IND_A1: 748.343750, N_A1: 1043, C_A1: 21954, PP_AMP_A1: 17206, PP_IND_A1: 750
FP1_AMP_A2: 18322, FP2_AMP_A2: 18415, FP3_AMP_A2: 18288, FP_IND_A2: 746.000000, N_A2: 943, C_A2: 22475, PP_AMP_A2: 18414, PP_IND_A2: 748
ma 06 00000000 00000497 000004d8 00000000 bd61 4c 0032bf7c a0:0
mc 05 0000035d 00000000 00000033 00000000 bd63 d3 0032bfc6 a1:0
mr 05 0000028b 00000000 ffffff4e 00000000 bd63 d3 40474047 a1:0
FP1_AMP_A0: 9659, FP2_AMP_A0: 14089, FP3_AMP_A0: 16735, FP_IND_A0: 745.203125, N_A0: 962, C_A0: 24184, PP_AMP_A0: 18605, PP_IND_A0: 749
FP1_AMP_A1: 17374, FP2_AMP_A1: 17280, FP3_AMP_A1: 15788, FP_IND_A1: 745.343750, N_A1: 864, C_A1: 21624, PP_AMP_A1: 17373, PP_IND_A1: 748
FP1_AMP_A2: 17783, FP2_AMP_A2: 17595, FP3_AMP_A2: 16413, FP_IND_A2: 745.203125, N_A2: 1043, C_A2: 22047, PP_AMP_A2: 17782, PP_IND_A2: 748
ma 06 00000000 0000048f 000004a0 00000000 bd6a 4e 0032c1ac a0:0
mc 05 00000367 00000000 00000033 00000000 bd68 d4 0032c0e0 a1:0
mr 05 00000295 00000000 ffffff4e 00000000 bd68 d4 40474047 a1:0
FP1_AMP_A0: 7900, FP2_AMP_A0: 14819, FP3_AMP_A0: 14951, FP_IND_A0: 747.296875, N_A0: 966, C_A0: 23077, PP_AMP_A0: 18340, PP_IND_A0: 751
FP1_AMP_A1: 17006, FP2_AMP_A1: 17074, FP3_AMP_A1: 17001, FP_IND_A1: 746.046875, N_A1: 1043, C_A1: 23109, PP_AMP_A1: 17073, PP_IND_A1: 748
FP1_AMP_A2: 17752, FP2_AMP_A2: 17708, FP3_AMP_A2: 14206, FP_IND_A2: 746.531250, N_A2: 902, C_A2: 21136, PP_AMP_A2: 17751, PP_IND_A2: 749
ma 06 00000000 00000489 000004a0 00000000 bd6a 4e 0032c1ac a0:0
mc 05 0000034b 00000000 00000041 00000000 bd6c d5 0032c1f6 a1:0
mr 05 00000279 00000000 ffffff5c 00000000 bd6c d5 40474047 a1:0
FP1_AMP_A0: 12751, FP2_AMP_A0: 16526, FP3_AMP_A0: 11732, FP_IND_A0: 743.671875, N_A0: 1033, C_A0: 23253, PP_AMP_A0: 17376, PP_IND_A0: 747
FP1_AMP_A1: 16728, FP2_AMP_A1: 16954, FP3_AMP_A1: 16000, FP_IND_A1: 745.296875, N_A1: 1043, C_A1: 23134, PP_AMP_A1: 17264, PP_IND_A1: 751
FP1_AMP_A2: 17835, FP2_AMP_A2: 17666, FP3_AMP_A2: 13660, FP_IND_A2: 748.578125, N_A2: 900, C_A2: 20805, PP_AMP_A2: 17834, PP_IND_A2: 751

```

Figure 4.12: Teraterm view with ranging information ( $ma$ ,  $mc$  and  $mr$ ) and diagnostic information (all the variables described above).

## 4.3 Identification algorithm

Once variables are available, a MATLAB script is in charge to compute the values of the three features, exploiting equations 3.17, 3.18, 3.19. Now these features can be stored and analyzed.

### 4.3.1 Features robustness evaluation

The first elaboration is a study about the robustness of these three features. The purpose is to understand how general they are and if they are always appropriate to distinguish NLOS and LOS situations. To do so, a set of data collections have been made to test the features efficiency when some conditions change. In this section, data collections are described. Refer to the next chapter for results.

#### Simple static and dynamic NLOS discrimination

The first experimental test has been set up to demonstrate the ability of the extracted parameters to qualitatively show the difference between the LOS condition and the NLOS one in a static situation. The intent is purely to confirm the goodness of the theory behind the CIR study that is followed by this thesis. The used materials are just one anchor, one tag and one obstacle, placed as described in Table 4.1

Device	x position ( $m$ )	y position ( $m$ )
Anchor	0	0
Tag	0	1.2
Obstacle	0	0.6

Table 4.1: Anchor-Tag-Obstacle coordinates in the static and dynamic evaluation

Two different recordings have been made: first a 50 seconds LOS recording and then a 50 seconds NLOS recording with an obstacle placed in between Anchor and Tag. The comparison between features values is available in the next chapter.

The second experimental test has been set up to demonstrate the ability of the extracted parameters to qualitatively show the difference between the LOS condition and the NLOS one in a dynamic situation. The used materials are just one anchor, one tag and one obstacle, placed as described in Table 4.1

A single recording has been made, changing the channel condition dynamically. First there are 20 seconds of LOS condition, then for 10 seconds an obstacle has been put in between and then again 20 seconds of LOS condition. The aim of the setup is to verify if the distinction between the two channel situations is clearly represented by extracted features also in a dynamic environment.

The recorded behaviour of features values is available in the next chapter.

### Acquisition length influence

The purpose of this setup was to demonstrate the stability of the extracted parameters over the time (i.e. collecting different number of acquisitions). Parameters have been computed with an obstacle in a fixed position and varying the number of acquisitions from 5 to 200. The expected result is a distinction between LOS and NLOS that is not influenced by the length of the acquisition. The used materials are just one anchor, one tag and one obstacle, placed as described in Table 4.2

Device	x position ( $m$ )	y position ( $m$ )	Recording time ( $samples$ )
Anchor	0	0	
Tag	0	6	(5,25,50,100,200)
Obstacle	0	1	

Table 4.2: Anchor-Tag-Obstacle coordinates in the acquisition length influence evaluation, with an indication of tested recording times

Five different recordings have been made, changing each time the acquisition length: from 5 recordings (about 0.5s) to 200 recordings (about 20s), making the channel condition remain unchanged (i.e. with the obstacle always in the same position).

The boxplots that show features behaviour in different conditions are available in the next chapter.

### Obstacle position influence

The purpose of this setup was to demonstrate the stability of the extracted parameters respect to different obstacle positions (i.e. collecting data with a different obstacle-board distance). Parameters have been computed for a fixed number of acquisitions and varying obstacle distance from the Anchor from 1m to 5m. The expected result is a distinction between LOS and NLOS that is slightly influenced by the position of the obstacle. The used materials are just one anchor, one tag and one obstacle, placed as described in Table 4.3

Device	x position ( $m$ )	y position ( $m$ )
Anchor	0	0
Tag	0	6
Obstacle	0	(1,2,3,4,5)

Table 4.3: Anchor-Tag-Obstacle coordinates in the obstacle position influence evaluation, with an indication of tested obstacle positions

Five different recordings have been made, changing each time the obstacle position: from 1m to 5m far from Anchor position, making the number of acquisitions remain unchanged (i.e. collecting each time 50 recordings).

The boxplots that show features behaviour in different conditions are available in the next chapter.

### Antenna orientation influence

The purpose of this setup was to demonstrate the stability of the extracted parameters respect to antenna orientations (i.e. collecting data with a different Anchor’s antenna orientation respect to the Tag’s one). Parameters have been computed for a fixed number of acquisitions and for a fixed obstacle position, varying Anchor’s antenna orientation from  $0^\circ$  (the two antennas are parallel) to  $180^\circ$  (the two antennas are parallel but they show each other different sides). The expected result is a distinction between LOS and NLOS that is not influenced by the antenna orientation. The used materials are just one anchor, one tag and one obstacle, placed as described in Table 4.4

Device	x position (m)	y position (m)	Antenna orientation (degree)
Anchor	0	0	(0,45,90,135,180)
Tag	0	6	0
Obstacle	0	1	

Table 4.4: Anchor-Tag-Obstacle coordinates in the antenna orientation influence evaluation, with an indication of tested antenna orientations

Five different recordings have been made, changing each time the antenna orientation: from  $0^\circ$  to  $180^\circ$ , making the number of acquisitions and the obstacle position remain unchanged (i.e. collecting each time 50 recordings with the obstacle at 1m from Anchor).

The boxplots that show features behaviour in different conditions are available in the next chapter.

### 4.3.2 Data collection for the training set

Once features have been considered suitable and robust to distinguish between LOS and NLOS conditions, a classifier has been built to perform the binary identification. First thing is to create a training set. To do this, a data collection has been made. The purpose of this data collection was to retrieve features values to implement an identification. The used materials are three anchors, one tag and one obstacle. The setup is the following: the three anchors are placed in fixed positions, while the Tag is placed in eleven different test points to form an evaluation grid. In Figure

4.13 a graphical representation of the grid is given: fixed anchors are indicated with black diamond (A0,A1 and A2) while different Tag test points are indicated with red diamond.

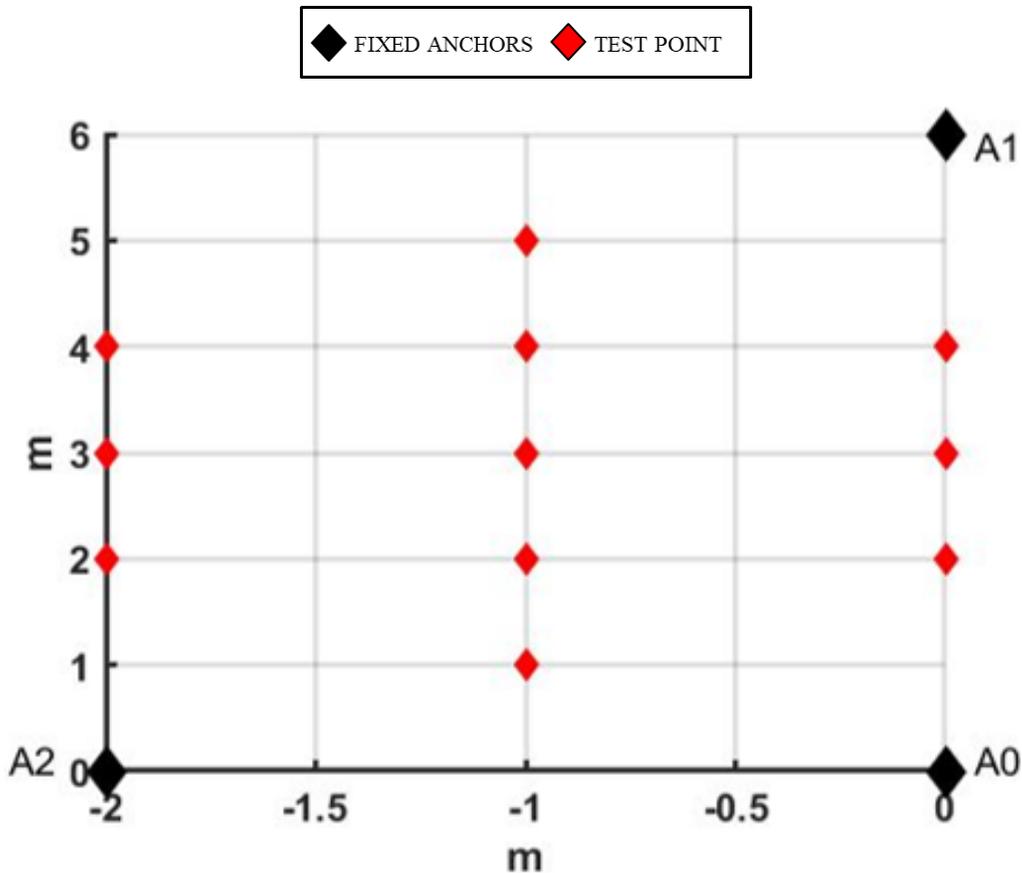


Figure 4.13: Evaluation grid for the training set construction

Features from eleven test points are collected in turn considering four different situations:

- **LOS** Ranging algorithm has been run without obstacle presence in any of the test points. Both ranging and diagnostics information have been collected and labelled as LOS information.
- **NLOS Anchor0** Ranging algorithm has been run with the obstacle in front of Anchor0. Only the channel Tag-Anchor0 has been affected by NLOS presence and only data coming from this channel have been labelled as NLOS information. Data from other two channels have been labelled as LOS information.
- **NLOS Anchor1** Ranging algorithm has been run with the obstacle in front of Anchor1. Only the channel Tag-Anchor1 has been affected by NLOS presence

and only data coming from this channel have been labelled as NLOS information. Data from other two channels have been labelled as LOS information.

- **NLOS Anchor2** Ranging algorithm has been run with the obstacle in front of Anchor2. Only the channel Tag-Anchor2 has been affected by NLOS presence and only data coming from this channel have been labelled as NLOS information. Data from other two channels have been labelled as LOS information.

Data collection has been made in the Main Lab room of the Motion Analysis Lab (MAL) at the Spaulding Rehabilitation Hospital in Boston, Massachusetts.

NLOS condition has been simulated for one anchor in turn using a 1m high aluminum cylinder. In each case, the distance obstacle-anchor has been fixed at 1m. The number of acquisitions for each test point has been fixed at 50 (5 sec). The antenna orientation is not fixed and changes according with the test point (Tag and anchors are set with antenna parallel to their boards). The real position of the tag (for future error evaluations) has been measured with a tape measure, estimating a measurement uncertainty of 2cm.

The total number of recorded sets are 132: eleven test points, each one evaluated in four different conditions, each condition with three sets coming from the three different anchors. Each features set is made of 50 recordings. At the end, 6600 values for each of the three features (divided into 4950 triplets coming from LOS recordings and 1650 triplets coming from NLOS recordings) are the data that have been formed the training set for the classification algorithm.

A 3-D representation of the features space is given by means of a scatterplot graph in the next chapter.

### 4.3.3 Classifier construction

To implement the necessary binary classification a lot of different solutions are widely used in the literature. Decision trees, Bayesian networks, Neural networks are just few examples of commonly used methods. In this thesis a Support Vector Machine (SVM) algorithm has been chosen.

SVM is a robust and computationally efficient algorithm, characterized by a fast training phase and a fast application phase. Its major advantage is the smaller number of parameters to tune in order to make it operational and yielding high accuracy rates. It is the best choice to have an algorithm with low computational cost and easy to implement, that can ensure flexibility and high performance in different conditions [24].

To try to build the most efficient classifier for the described problem, a SVM parameters tuning phase has been faced. It has been chosen to compare three different kernel functions combined with three different values of C-parameter (C is a regularization parameter that controls the trade-off between achieving a low

training error and the ability to generalize your classifier to unseen data). Specifically, SVM with a Linear, Radial Basic Function and Polynomial kernel function and with C values of 0.5, 1, 1.5 have been tested (Table 4.5).

Parameter	Combination 1	Combination 2	Combination 3	Combination 4	Combination 5	Combination 6	Combination 7	Combination 8	Combination 9
Kernel function	Linear	Linear	Linear	RBF	RBF	RBF	Polynomial	Polynomial	Polynomial
C parameter	0.5	1	1.5	0.5	1	1.5	0.5	1	1.5

Table 4.5: Tested combinations of kernel functions and C values

To validate classification results in the training set and choice the best configuration (the couple kernel function/C-value), a 10-fold cross validation has been implemented: in turn, ten test points present in the dataset (4.3.2) are used to train the classifier, while the last one is used to check the performance of the trained algorithm.

Classification performance are evaluated using Accuracy, Specificity and Sensitivity indicators:

- Sensitivity describes the classifier capability to label as NLOS actual NLOS recordings
- Specificity describes the capability to label as LOS actual LOS recordings
- Accuracy describes overall performance of the classifier in both the two classes.

Exploiting these three indicators, SVM with Radial Basic Function kernel function and a C-parameter value equal to 1 has showed the best results, with over 99% in each of the three. More details about classification performance will be given in the next chapter.

#### 4.3.4 Classifier validation phase

Once the best classifier configuration has been found from the training phase, another validation phase over other different datasets has been implemented. This testing phase has been thought to evaluate the behaviour of the classifier in different conditions, different from the ones in which it has been trained, to see the ability of the algorithm to generalize the learned knowledge.

Also in this case NLOS condition has been simulated for one anchor in turn (except the data collection with two anchors in NLOS) using a 1m high aluminum cylinder. In the two NLOS anchors condition two identical aluminum cylinders have been used. In each case, the distance obstacle-anchor has been fixed at 1m. The number of acquisitions for each test point has been fixed at 50 (5 sec). The

antenna orientation is not fixed and changes according with the test point (Tag and anchors are set with antenna parallel to their boards). The real position of the tag (for future error evaluations) has been measured with a tape measure, estimating a measurement uncertainty of 2cm. Three different situations have been tested:

- Architecture with two anchors affected by NLOS conditions, to show that NLOS anchors' number does not affect identification algorithm performance. In Table 4.6, the disposition for this test is given

Device	x position ( $m$ )	y position ( $m$ )	Test Point Number
Anchor0	0	0	
Anchor1	0	6	
Anchor2	-2	0	
Tag	-1	2	1
Tag	-1	3	2
Tag	-1	4	3
Tag	-2	4	4
Tag	-2	3	5
Tag	-2	2	6

Table 4.6: Anchors-Tag coordinates in the data collection with multiple NLOS anchors

- Data collection made in a different recording room, to show that surrounding environment does not affect identification algorithm performance. In Table 4.7, the disposition for this test is given

Device	x position ( $m$ )	y position ( $m$ )	Test Point Number
Anchor0	0	0	
Anchor1	0	3	
Anchor2	-2	0	
Tag	-1	1	1
Tag	-1	2	2
Tag	-2	1.5	3
Tag	0	1.5	4

Table 4.7: Anchors-Tag coordinates in the data collection made in a different recording room

- Architecture with four fixed anchors, to show that platform structure does not affect identification algorithm performance. In Table 4.8, the disposition for

this test is given

Device	x position ( $m$ )	y position ( $m$ )	Test Point Number
Anchor0	0	0	
Anchor1	0	6	
Anchor2	-3	0	
Anchor3	-3	6	
Tag	-1	1	1
Tag	-1	3	2
Tag	-1	5	2
Tag	-2	5	3
Tag	-2	3	3
Tag	-2	1	3

Table 4.8: Anchors-Tag coordinates in the data collection with a platform with a different number of anchors

Once features triplets have been collected, the previously trained classifier has been run on the data. Also in each of these new datasets, more than 99% in each of the three used indicators has been achieved. More details about classification performance will be given in the next chapter.

Evaluating validation results, algorithm robustness can be stated. Collected data show as none of the mentioned and checked factors influence classifier performance. This means that NLOS identification can be achieved in different environments and with different platforms, without decreasing an overall accuracy close to 100%.

## 4.4 Mitigation algorithm

After being able to identify if a range comes from a NLOS channel, the further step is avoiding that this affected range could decrease accuracy of the trilateration process. If in 3.1.4 a positive biased range is introduced, the final position estimation will be wrong, leading to even one meter errors.

The simplest mitigation algorithm is the one that discards the affected range, not considering it during the trilateration process. This method is feasible only if the number of LOS available ranges is equal or greater than three. It means having a platform with at least four fixed anchors or more, to have a greater security that, even if some of them is experience NLOS condition, at least three anchors are usable. Surely, this is the best situation and have a redundant anchor set placed in the home environment is always the most advisable choice. An example of this kind of set up is 4.8 and, exploiting the possibility to rely on four anchors, in the next chapter

will be given also a practical example about the discarding mitigation algorithm. In literature also other approaches have been explored: an example of how LOS ranges can be combined with NLOS ones to improve final position estimation accuracy exploiting linear-programming techniques is given in [25]. By the way all these methods require at least a number of starting LOS ranges equal or greater than three.

This part of the thesis wants to focus on degenerate cases, that can be considered exceptions. Specifically, when the number of available LOS anchors is less than three. In these situations the final solution cannot be a purely geometric one, considering the presence of noisy ranges like the NLOS ones. Therefore, a statistical approach is usually applied, with some mathematical artifacts to mitigate the effect of the wrong ranges in the final trilateration.

In the literature a lot of statistical trilateration algorithms are provided: some of them are closed-form algorithms (that can be evaluated in a finite number of operations) other are iterative algorithms (that need to be performed several times to achieve a reasonable result), some of them are least-squares algorithms (that formulate a cost function and try to minimize it), other are likelihood-based algorithms (that formulate an error function and try to minimize it) [26].

#### 4.4.1 Taylor Series-based Least-Square algorithm

The proposed mitigation algorithm is a simple Taylor Series (TS) based Least-Square (LS) algorithm [27]. Even if it is important to keep the same original performance in terms of sampling frequency, and so a closed-form solution should be preferred as it is generically faster, the developed TS - LS correction presents a reduced complexity due to the Taylor Series approximation and achieves a good localization accuracy. The only prerequisite required by this algorithm is the knowledge of which is/are the NLOS measurement/s: having a trained classifier that can achieve high accuracy rates, this is not a problem for the majority of the collected data.

In this section a description of the TS-LS algorithm working principle is given in depth.

Consider an architecture with  $N$  fixed anchors ( $A_i$ , whose coordinates are known:  $x_i, y_i, z_i$  with  $i = 1, 2, \dots, N$ ) and a tag (T, whose coordinates are to be computed:  $x, y, z$ ). The Euclidean distance between  $A_i$  and T can be expressed as:

$$d_i(x, y, z) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \quad (4.3)$$

with  $i = 1, 2, \dots, N$

The presence in the platform of both LOS and NLOS anchors can be assumed: if there are  $N_{los}$  anchors, there are also  $N - N_{los}$  that are experiencing obstructed channels.

Considering the presence of both measurement noise and NLOS bias (last one only for NLOS anchors), the measured ranges can be generally modeled as:

$$r_i = d_i(x, y, z) + \epsilon_i, \quad \text{with } i = 1, 2, \dots, N \quad (4.4)$$

where:

$$\epsilon_i = \begin{cases} n_i, & \text{with } i = 1, 2, \dots, N_{los} \\ L_i + n_i, & \text{with } i = N_{los} + 1, \dots, N \end{cases} \quad (4.5)$$

$L_i$  is the NLOS positive bias, while  $n_i$  is the measurement error (usually a zero-mean gaussian random variable).

Basically, the working principle of the algorithm is the following. Starting from a first raw tag position estimation, that is the wrong one without any correction and affected by the NLOS positive biased ranges, the distances between the tag and each anchor can be recalculated. It is expected that the new recalculated distances between the tag and the NLOS anchors are more accurate than the original NLOS collected ranges: the fact that, in the trilateration process, also LOS ranges intervene to compute the position estimation acts as a mitigating effect on the NLOS ranges. Therefore, the mitigation step consists in the replacement of the original NLOS collected ranges with recalculated distances. Then the process is repeated iteratively, achieving a more accurate position estimation at each iteration.

In Figure 4.14 a comparison between a correct trilateration coming from three LOS ranges and a wrong trilateration with a NLOS affected range is given.

In Figure 4.15 an example of the correction process is shown. The two blue circumferences come from LOS ranges (from Anchor1 and Anchor2 in this case) while the red circumference comes from NLOS range (from Anchor0). The red dot is the original wrong tag position estimation, while the green dot is the real tag position. Going on with iterations, the estimated position gets closer the real one and the NLOS range effect is iteratively mitigated.

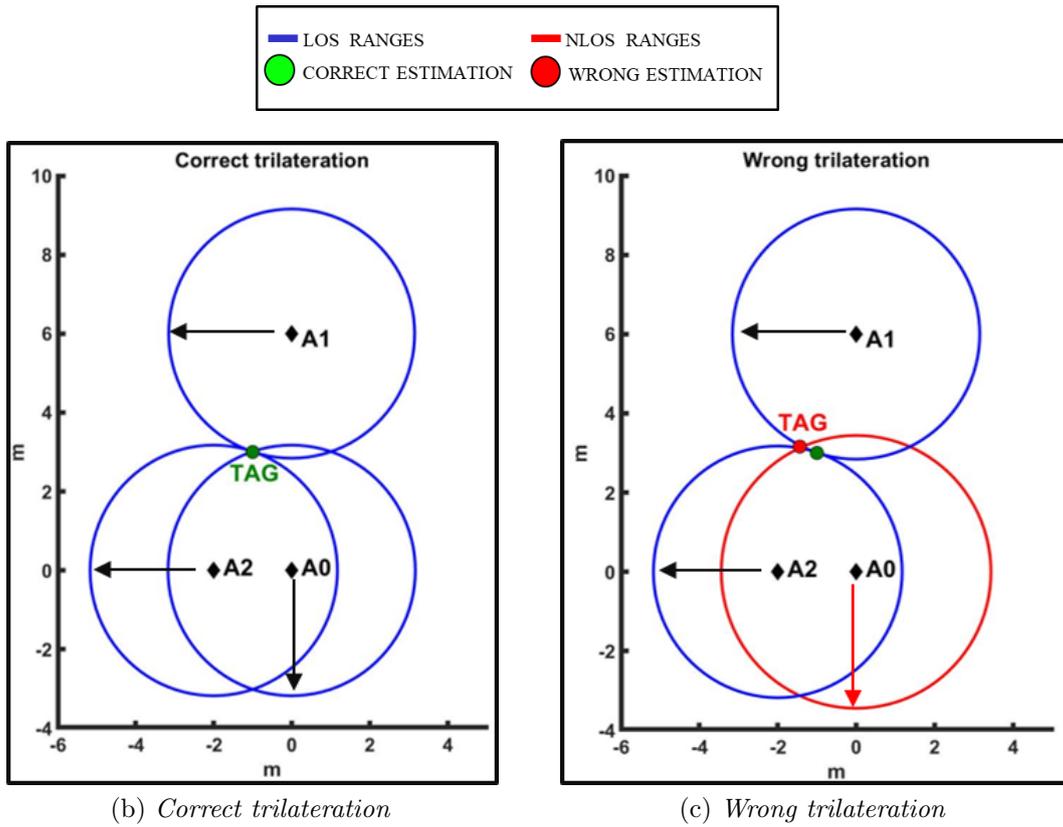


Figure 4.14: Comparison between a correct trilateration and a wrong trilateration

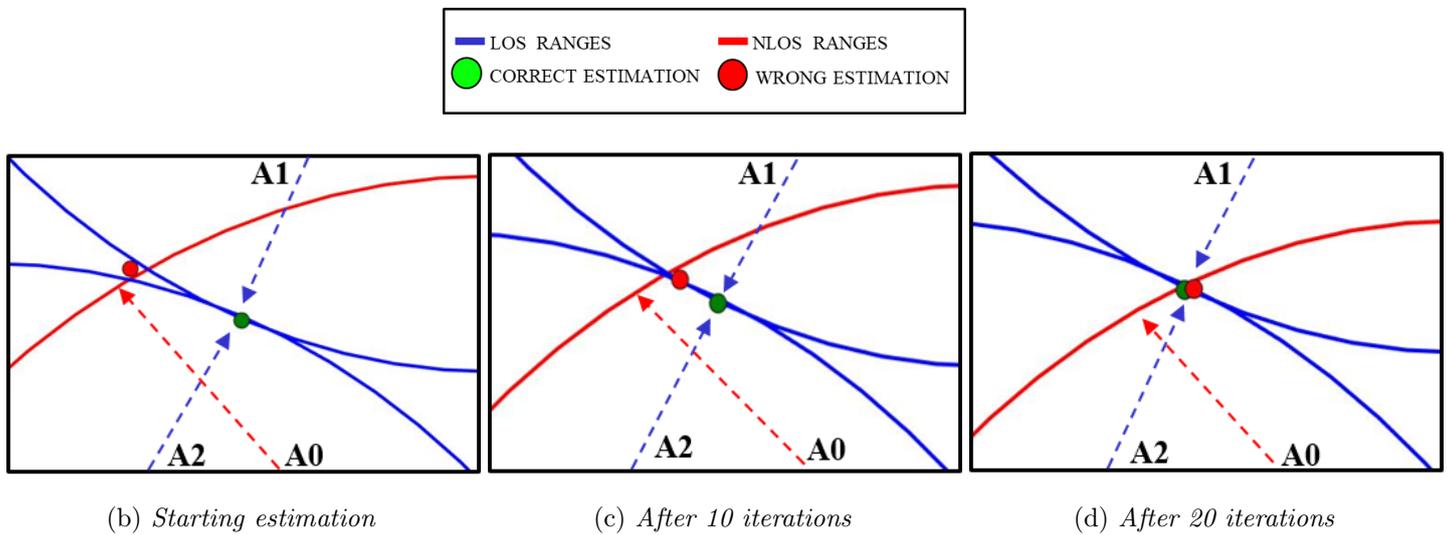


Figure 4.15: Example of TS-LS mitigation effect

To exploit this concept, the Taylor Series-based Least-Squares algorithm is suitable, considering the good trade-off between achieved accuracy and low-complexity. Giving the starting position estimation  $(\hat{x}, \hat{y}, \hat{z})$ , coordinates can be written as:

$$x = \hat{x} + \delta_x, \quad y = \hat{y} + \delta_y, \quad z = \hat{z} + \delta_z \quad (4.6)$$

where  $\delta_x$ ,  $\delta_y$  and  $\delta_z$  are the NLOS errors added to real coordinates that are to be mitigated.

Substituting 4.6 in 4.3, expanding using Taylor series around  $(\hat{x}, \hat{y}, \hat{z})$  and retaining only first two terms, following equations can be obtained:

$$a_{i,1}\delta_x + a_{i,2}\delta_y + a_{i,3}\delta_z \simeq r_i - \hat{d}_i, \quad \text{with } i = 1, 2, \dots, N_{los} \quad (4.7)$$

$$a_{i,1}\delta_x + a_{i,2}\delta_y + a_{i,3}\delta_z \simeq 0, \quad \text{with } i = N_{los}+1, \dots, N \quad (4.8)$$

where:

$$\hat{d}_i = \sqrt{(\hat{x} - x_i)^2 + (\hat{y} - y_i)^2 + (\hat{z} - z_i)^2} \quad (4.9)$$

$$a_{i,1} = \frac{\hat{x} - x_i}{\hat{d}_i}, \quad a_{i,2} = \frac{\hat{y} - y_i}{\hat{d}_i}, \quad a_{i,3} = \frac{\hat{z} - z_i}{\hat{d}_i}, \quad (4.10)$$

It can be noticed how in 4.8 the NLOS affected ranges are no more considered in the cost function that will be subsequently minimized.

Equations 4.7 and 4.8 can be written together in a more compact matrix form:

$$\mathbf{A}\boldsymbol{\delta} \simeq \mathbf{h} \quad (4.11)$$

where:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ \dots & \dots & \dots \\ a_{N,1} & a_{N,2} & a_{N,3} \end{bmatrix} \quad (4.12)$$

$$\boldsymbol{\delta} = [\delta_x \quad \delta_y \quad \delta_z]^T \quad (4.13)$$

$$\mathbf{h} = [r_1 - \hat{d}_1 \quad \dots \quad r_{N_{los}} - \hat{d}_{N_{los}} \quad \mathbf{0}]^T \quad (4.14)$$

Running a weighted least-squares algorithm on 4.11 [28], the residuals  $(\delta_x, \delta_y, \delta_z)$  can be found according with:

$$\boldsymbol{\delta} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{h} \quad (4.15)$$

where  $\mathbf{W}$  is a diagonal weighting matrix that is in charge to weigh more LOS ranges respect to NLOS ones.

Once  $(\delta_x, \delta_y, \delta_z)$  are computed, the position estimation can be updated:

$$\hat{x} = \hat{x} + \delta_x, \hat{y} = \hat{y} + \delta_y, \hat{z} = \hat{z} + \delta_z \quad (4.16)$$

This process is the description of just one iteration. The estimated position is iteratively refined iteration after iteration until a stopping criterion is reached: possible stopping criteria can be the achievement of a predefined threshold on  $\delta$  or simply the number of iterations.

Exploiting this kind of algorithm, a good position estimation accuracy is achieved and, consequently, a huge NLOS mitigation is reached.

To test the algorithm performance, both static and dynamic simulations have been implemented.

#### 4.4.2 Static simulations

The static simulations have been run on the same datasets described in Figure 4.13, Table 4.6 and in Table 4.7. The fourth dataset collected with a four anchors platform is not involved, considering that, in that case, the mitigation algorithm is performed discarding NLOS affected ranges.

TS-LS mitigation results will be shown in depth in the next chapter. To evaluate a trilateration, the used metric is the Euclidean distance between the estimated position and the real one measured with the tap measure:

$$Original\_Error_i = \sqrt{(x\_Original_i - x\_Real_i)^2 + (y\_Original_i - y\_Real_i)^2} \quad (4.17)$$

with  $i = 1, 2, \dots, N_{TestPoints}$

$$Corrected\_Error_i = \sqrt{(x\_Corrected_i - x\_Real_i)^2 + (y\_Corrected_i - y\_Real_i)^2} \quad (4.18)$$

with  $i = 1, 2, \dots, N_{TestPoints}$

where 4.17 represents error of the original trilateration, while 4.18 represents error after the correction algorithm application.

It is useful having a parameter that compares 4.17 and 4.18 to state if the correction introduction leads to a better position estimation accuracy:

$$Improvement_i (\%) = \frac{Original\_Error_i - Corrected\_Error_i}{Original\_Error_i} * 100 \quad (4.19)$$

with  $i = 1, 2, \dots, N_{TestPoints}$

On average, considering all the tested datasets, an improvement (defined by 4.19) of about 50% is achieved: it means that the original trilateration error coming from NLOS positive biased ranges is halved after the algorithm application. Furthermore, in all considered cases, the average final absolute trilateration error (4.18) is less than 20cm that is the considered benchmark: 20cm is the maximum error declared by Decawave in LOS condition. It means that, after the correction, NLOS effect is mitigated until achieving the normal declared behaviour of the used technology in an ideal working condition.

An example of correction result is in Figure 4.16. The situation represented is NLOS condition for Anchor2. Red diamonds represent evaluated test points, blue circles represent original collected trilateration results, green circles represent trilateration results after the algorithm has been run on the dataset.

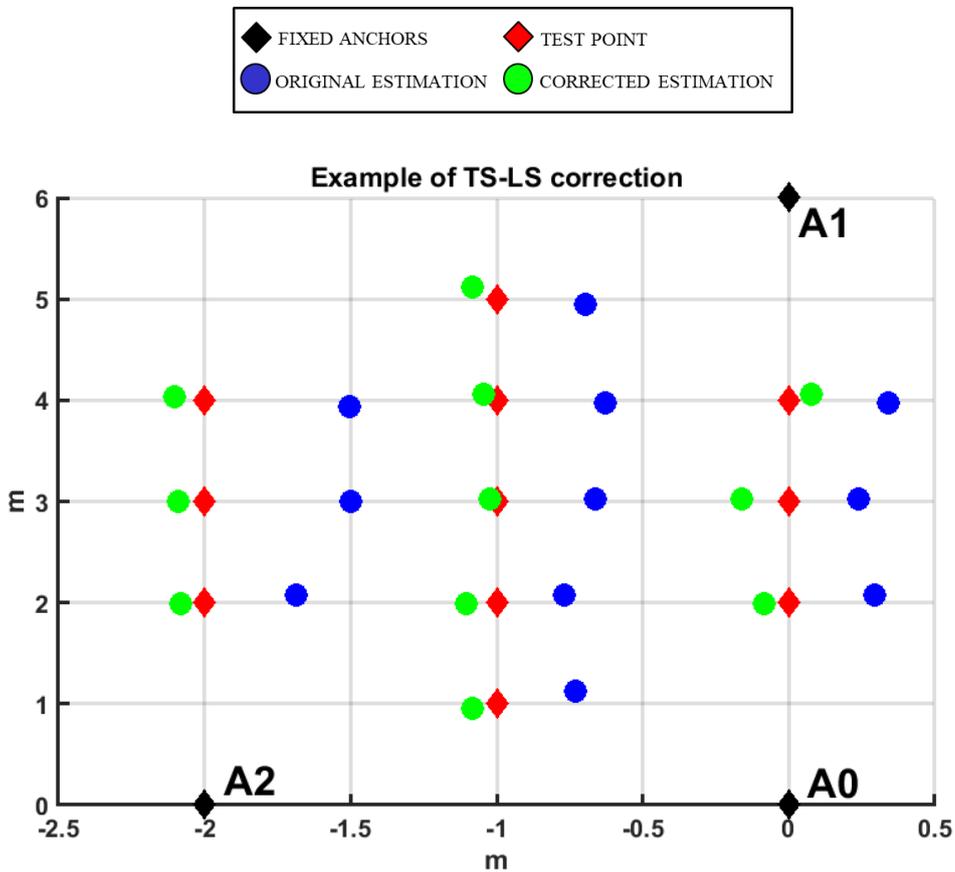


Figure 4.16: Example of TS-LS correction with Anchor2 in NLOS

Algorithm performance decreases when it is applied in a situation with two NLOS anchors. This behaviour is expected: being an optimization algorithm, if the

cost function becomes less representative of the situation (as it happens when LOS ranges number get lower since 4.14 contains less information to be exploited) the final result will be less adherent to real tag position.

### 4.4.3 Dynamic simulations

A second testing phase consists in dynamic simulations.

The purpose of this experimental setup is to have an estimation of which is algorithm behaviour if the data collection is implemented using a tag that moves continuously (instead of one placed in fixed positions as described in the last section).

To simulate this real working scenario, Tag has been put over the Home Robot structure (as represented in Figure 1.3) and the Home Robot has been programmed to do several trajectories with different shapes to simulate different conditions. Each trajectory has been collected for 125 samples, with a Decawave kit sampling frequency of 3.57 Hz (almost 35 s). In each trajectory, always one different anchor out of three is facing NLOS situation (the obstacle is the same used also in static simulations). The characteristic that all the trajectories have in common is that in each one the two phases are present: Home Robot passes from LOS condition to NLOS one (or vice versa) respect to the anchor with the obstacle in front. This kind of set up allows to see algorithm performance in a real working scenario and, specifically, in the transition phase between LOS and NLOS (that is physically the most particular situation, because UWB signals experience weird obstacle border effects that could lead to a decreasing in the identification performance and so, consequently, in the final trajectory estimation accuracy). The cases under analysis are explained in Table 4.9.

To assess the best trilateration accuracy, anchors coordinates have been computed using Vicon system, as well as the Home Robot entire routes, to provide a reliable ground truth with which make the comparison with Decawave system trajectory estimations: exploiting 4.19 with  $x\_Real = x\_Vicon$  and  $y\_Real = y\_Vicon$ , a quantification about proposed correction method efficiency can be given.

Vicon system is the motion capture architecture present in Motion Analysis Lab at Spaulding Rehabilitation Hospital. It consists of twelve infrared cameras placed near the ceiling in the Main Lab room. Exploiting reflective markers placed on the thing whose motion need to be tracked, Vicon system can provide as output  $(x,y,z)$  with a sampling frequency of 120Hz and a millimeter precision. As explained in 2.1.1, IR systems are not suitable for real time indoor tracking systems considering the high opacity of almost all materials to infrared rays (and in the case of Vicon system also the high cost). In this thesis the tested situations are basic and simple, so, using the caution to place two reflective markers on the Home Robot, always at least one of them is visible from Vicon cameras: the result is that all the tested trajectories have a reliable ground truth that approximates the real route with an accuracy of 1mm. Considering that Decawave kit presents an error of the order

Trajectory	NLOS Anchor	Device	x position (m)	y position (m)
1	Anchor0	Anchor0	1.189	-0.693
		Anchor1	1.162	3.037
		Anchor2	-0.954	-0.670
2	Anchor0	Anchor0	1.285	-1.412
		Anchor1	1.321	3.380
		Anchor2	-1.386	-1.377
3	Anchor0	Anchor0	1.285	-1.412
		Anchor1	1.321	3.380
		Anchor2	-1.386	-1.377
4	Anchor1	Anchor0	1.285	-1.412
		Anchor1	1.321	3.380
		Anchor2	-1.386	-1.377
5	Anchor1	Anchor0	1.285	-1.412
		Anchor1	1.321	3.380
		Anchor2	-1.386	-1.377
6	Anchor2	Anchor0	1.285	-1.412
		Anchor1	1.321	3.380
		Anchor2	-1.386	-1.377
7	Anchor2	Anchor0	1.285	-1.412
		Anchor1	1.321	3.380
		Anchor2	-1.386	-1.377

Table 4.9: Anchors disposition for the tested trajectories

of tens of centimeters also in LOS cases, Vicon output error can be neglected and Vicon measurements can be considered as the real Home Robot trajectories.

Considering different sampling frequency between Vicon and Decawave, Vicon output has been down sampled to make the two measures comparable.

An example of Vicon output for Trajectory 1 can be seen in Figure 4.17: black diamonds are fixed anchors, brown rectangle is intended to be the obstacle (in this case in front of Anchor0), black circles are subsequent position estimations from Vicon (putting them together you get the final trajectory).

Taking as example Trajectory 1, in Figure 4.18, on the top, can be seen the comparison between Decawave output and Vicon output. Black diamonds represent fixed anchors, brown rectangle represents the obstacle, black circles represent subsequent position estimations from Vicon, blue circles represent subsequent position estimations from Decawave. It is clear that inside the Anchor0 NLOS zone the distance between the two estimations is bigger than the one outside the NLOS zone: as explained before, this effect is produced by the fact that, in the NLOS area, the

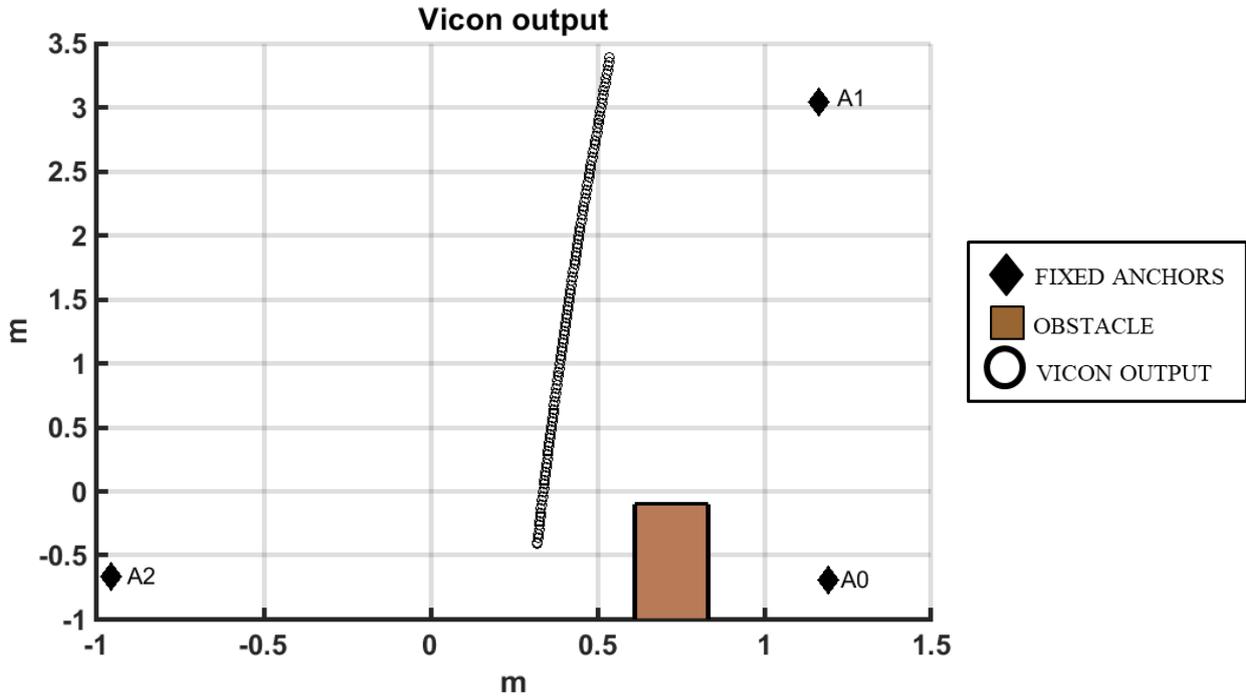
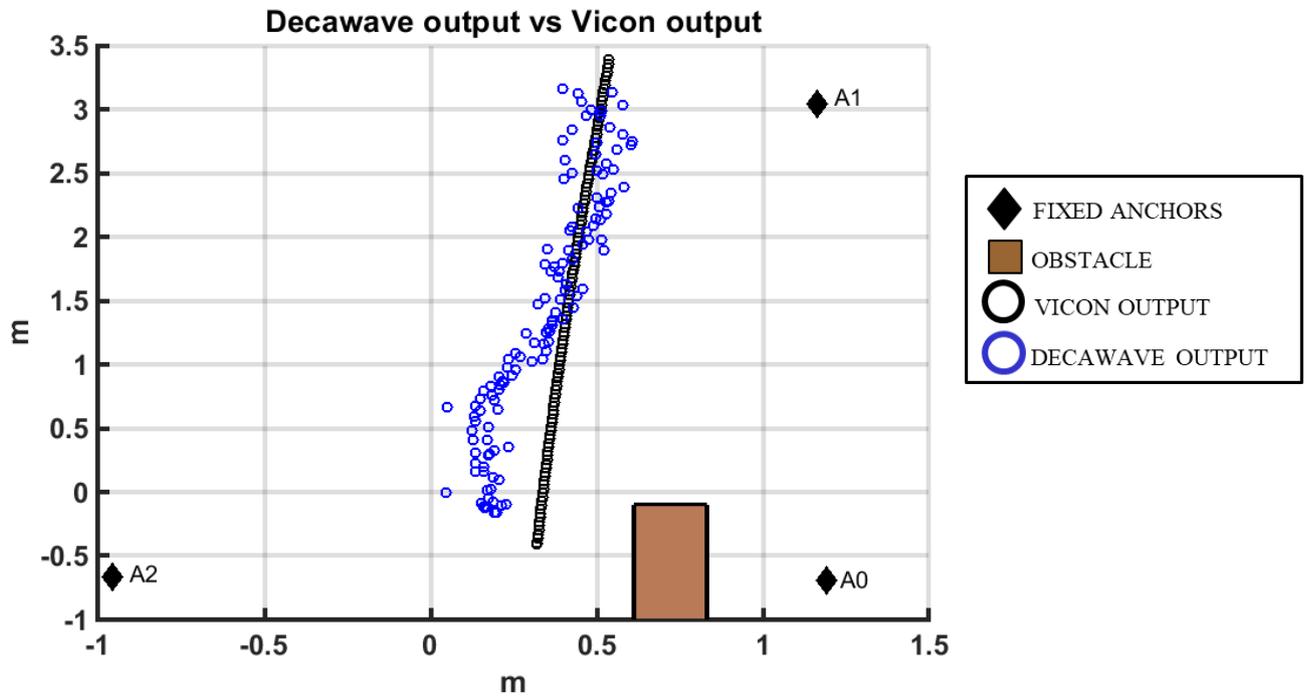


Figure 4.17: Vicon output for Trajectory 1

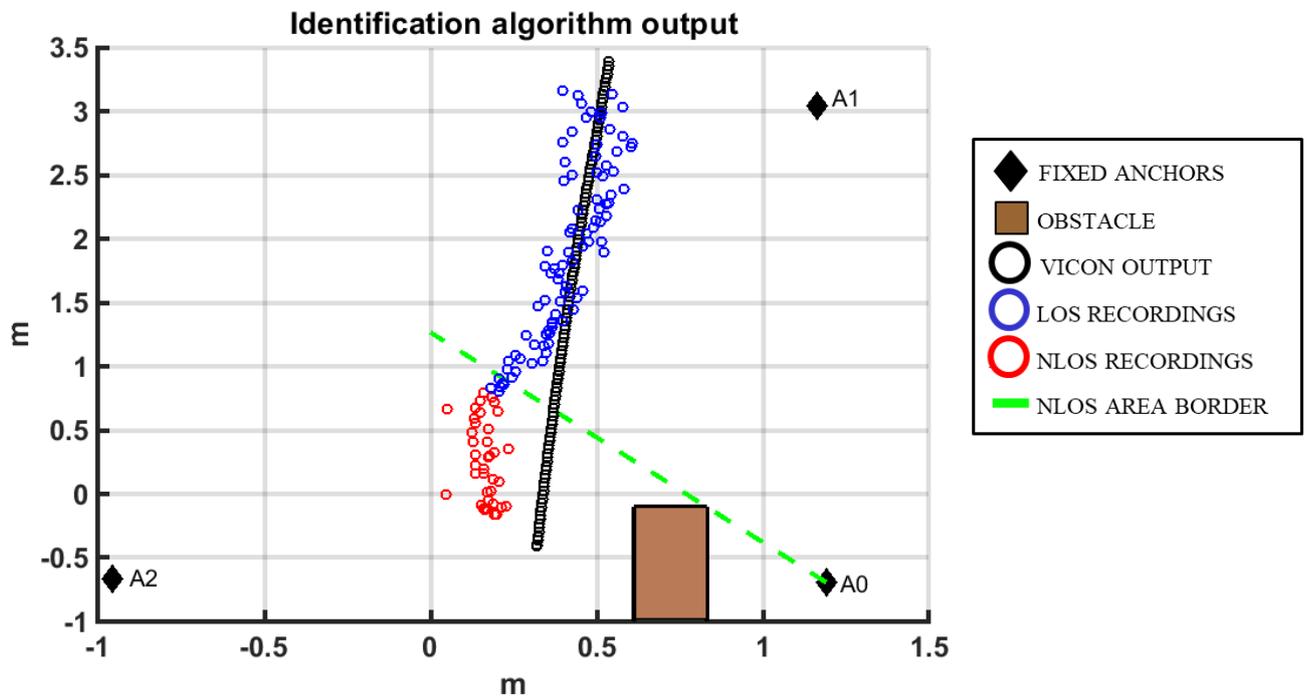
range coming from Anchor0 used in the trilateration algorithm presents a positive bias due to the obstacle presence.

The first step is to run the classifier to understand which recordings are affected by NLOS error. In Figure 4.18, on the bottom, can be seen the identification algorithm output: the overall accuracy can be considered satisfactory, even if a quantitative consideration cannot be given due to the fact that an objective determination of NLOS area is impossible considering the cylindrical shape of the obstacle. Black diamonds represent fixed anchors, brown rectangle represents the obstacle, black circles represent subsequent position estimations from Vicon, blue circles represent position estimations from Decawave classified as LOS recordings, red circles represent position estimations from Decawave classified as NLOS recordings, green dashed line is intended to be a qualitative estimation of NLOS area border.

It can be seen that identification accuracy is excellent in LOS area and hard NLOS area, while in the transition phase recordings that present NLOS error are instead classified as LOS.



(a) Decawave output

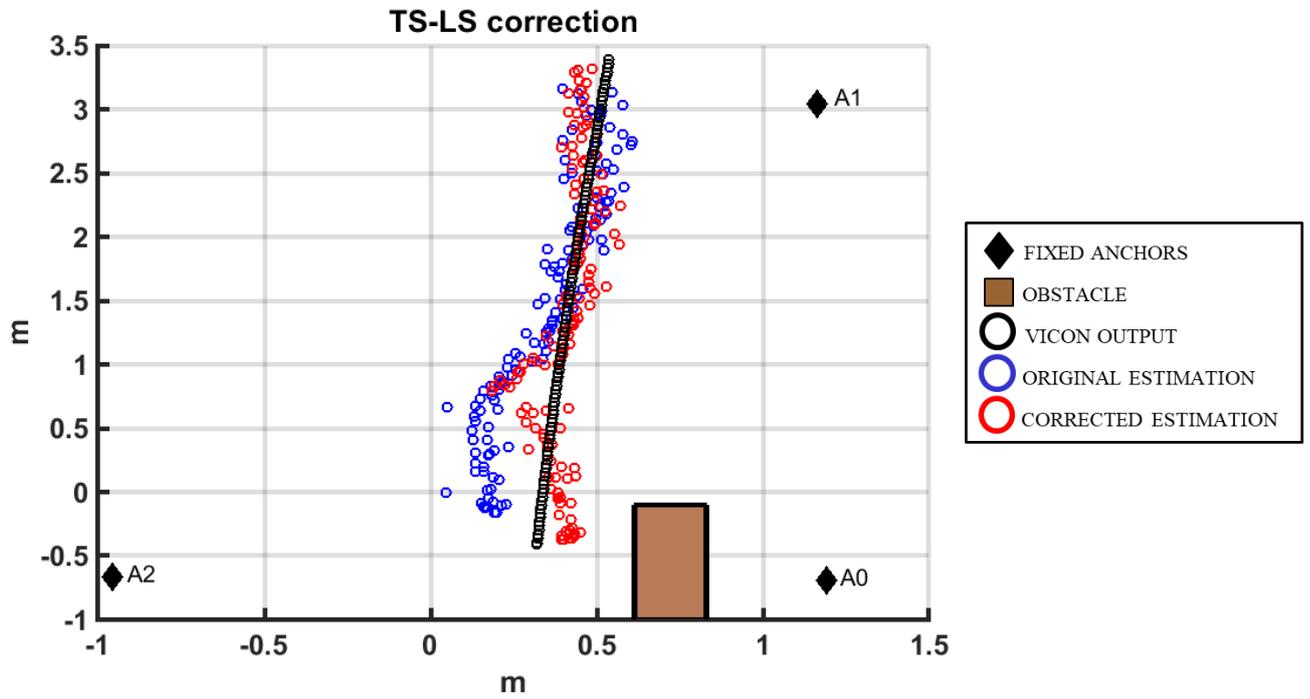


(c) Identification algorithm output

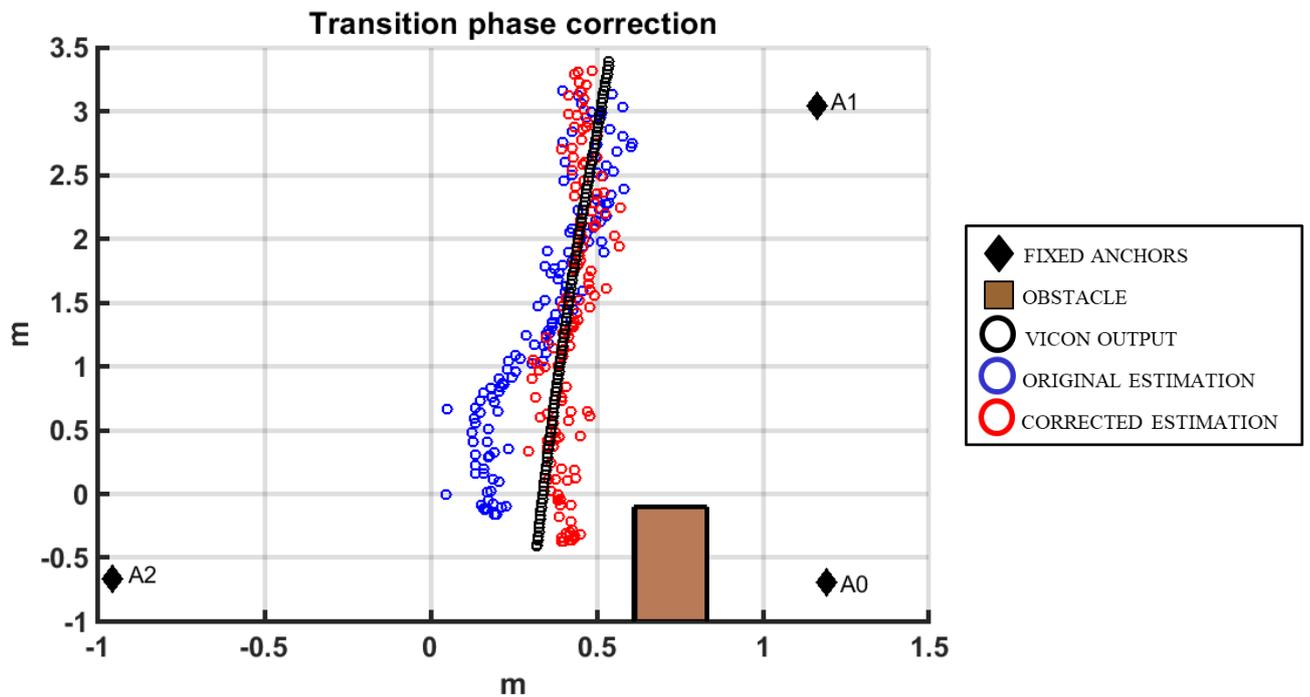
Figure 4.18: Identification algorithm performance for Trajectory 1

The following step is to run the TS-LS algorithm to mitigate NLOS error. It has been decided to apply correction algorithm also if the recording is classified as LOS (obviously considering all three ranges as reliable and consequently giving to all of them the same weight). In Figure 4.19, on the top, can be seen the achieved correction after running the algorithm. Black diamonds represent fixed anchors, brown rectangle represents the obstacle, black circles represent subsequent position estimations from Vicon, blue circles represent subsequent position estimations from Decawave, red circles represent position estimations after TS-LS application. It can be seen how in the transition phase, being those recordings classified as LOS, the correction leads to a wrong final estimation. Looking at the trilateration absolute error (Figure 4.20, top figure) computed respect to the Vicon output, the transition phase is the part of the entire trajectory that presents the biggest error after the correction. Red line represents original absolute error, blue line represents absolute error after the correction, green vertical line represents the NLOS area border according with classifier output.

To overcome this problem, the idea is to create a window of ten samples, centered in the NLOS area border identified by the classifier, and in this window apply two different corrections, one considering that specific anchor in NLOS condition and the other considering that specific anchor in LOS condition, and then the final result is taken as the average of the two. In Figure 4.19, on the bottom, can be seen the achieved correction after this kind of post processing, while in Figure 4.20, on the bottom, can be seen how the absolute trilateration error in the transition phase becomes similar to the one of the rest of the trajectory (the two vertical lines are intended to be the considered transition phase borders).

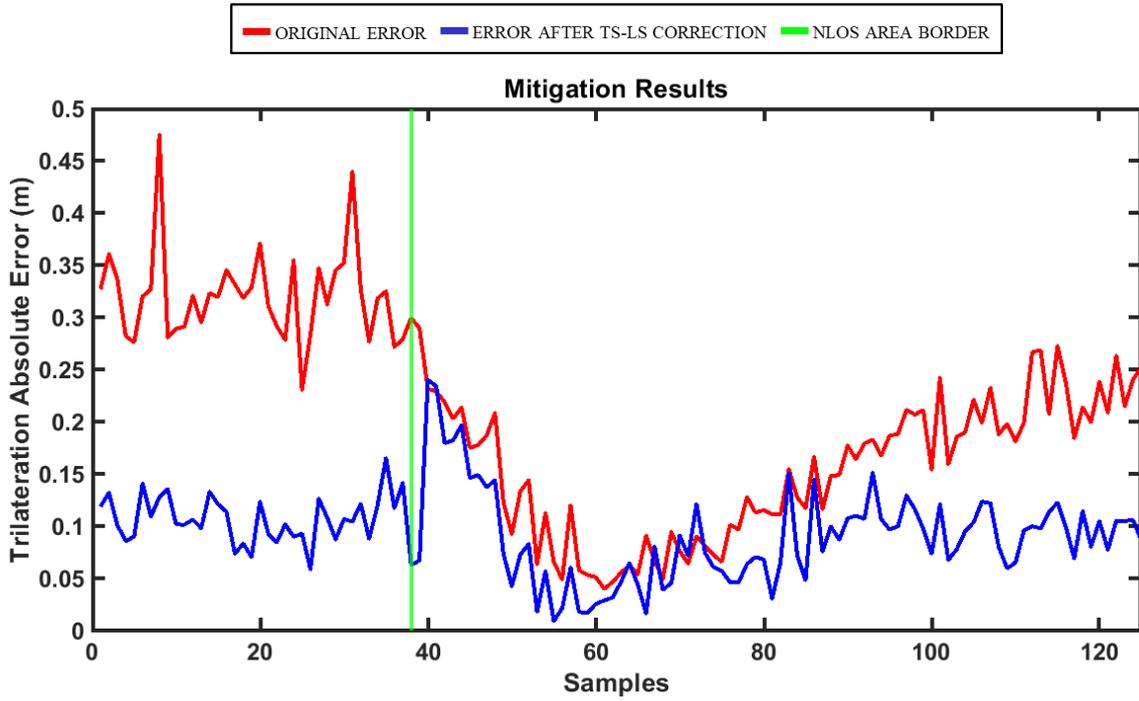


(a) *TS-LS correction*

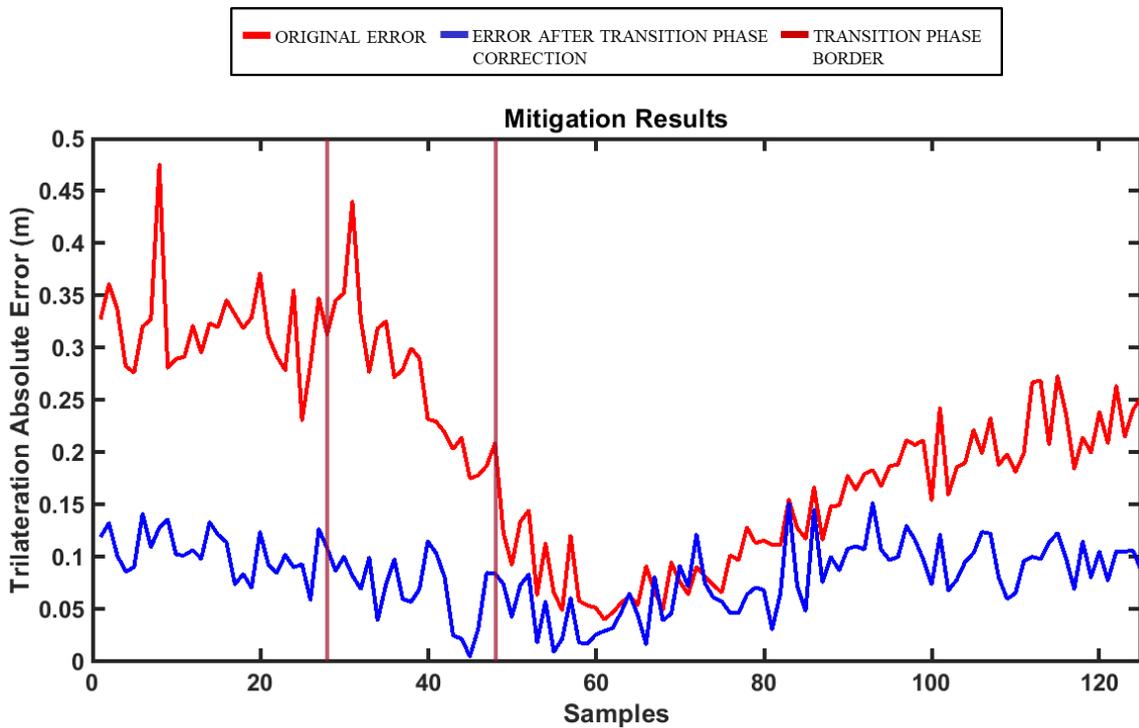


(c) *TS-LS correction after transition phase adjustment*

Figure 4.19: Mitigation algorithm performance for Trajectory 1 before and after transition phase correction



(b) *Trilateration absolute error after TS-LS correction*



(d) *Trilateration absolute error after transition phase adjustment*

Figure 4.20: Trilateration absolute error for Trajectory 1 before and after transition phase correction

### Smoothing tool

The last part of the algorithm consists in a smoothing application to have a cleaner and more homogeneous final result.

The implementation has been done applying a simple moving average along the collected time series. A window of five samples has been defined and each corrected point has been subsequently replaced with the average x-value and the average y-value of the points being part of the window (centered in the actual point under analysis).

Both the moving average and the transition phase correction are two kinds of post-processing that introduce a certain delay in the position estimation: the transition phase correction introduce a delay of ten samples (1 s) only during the transition phase, while the moving average introduce a delay of five samples (0.5 s) in all the trajectory. According to the author, these delays do not affect the real-time nature of the Home Robot application.

In Figure 4.21, the effect of the moving average introduction can be seen: black circles represent subsequent position estimations from Vicon, blue circles represent subsequent position estimations from Decawave, red circles represent position estimations after TS-LS application, green circles represent final position estimations after Moving-Average algorithm introduction.

The overall algorithm has been tested on seven different trajectories. In all of these, after correction application, on average an accuracy of 20cm has been achieved: that means that at the end, also in NLOS conditions, the LOS accuracy declared by Decawave is achieved. The average improvement is almost 66%, respect to original estimations by TREK1000. In NLOS conditions, improvements are bigger considering that the error starting point is always between 30cm and 50cm, while in LOS conditions the algorithm application leads to smaller improvements, but also in these situation original Decawave algorithm has been still perfected in terms of reduced standard deviation.

More details about mitigation performance will be given in the next chapter.

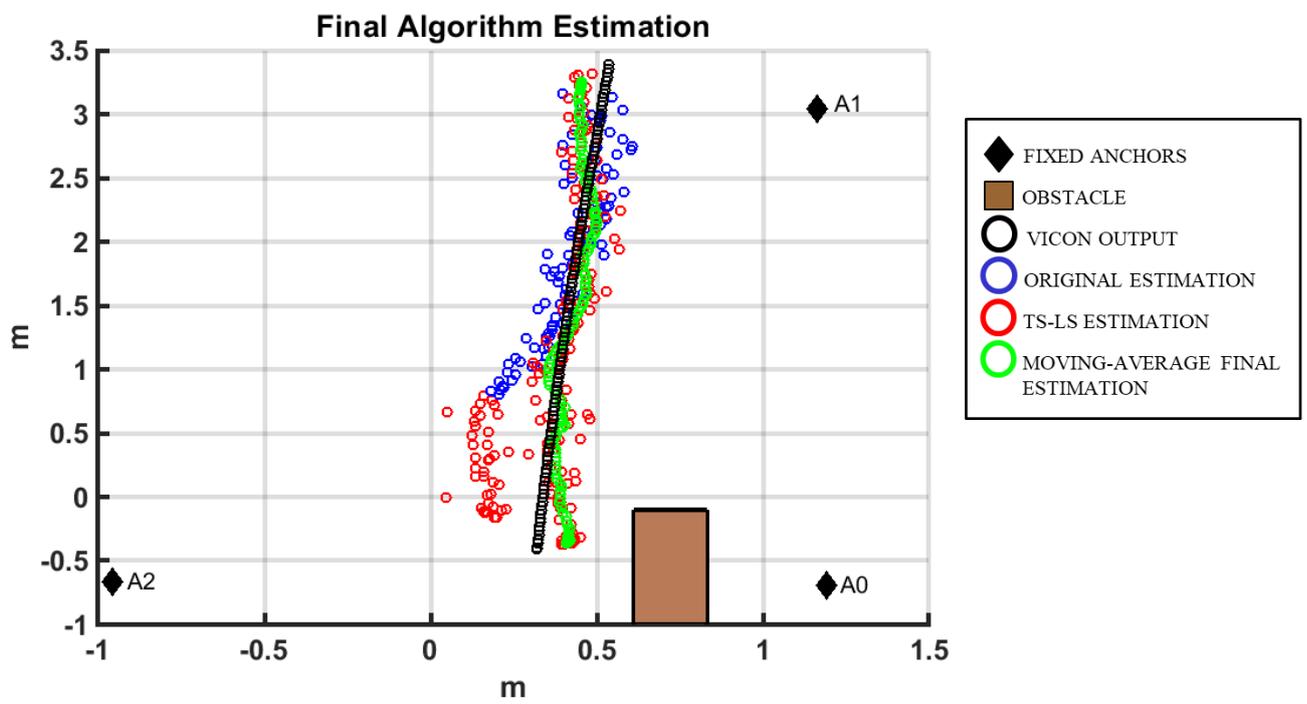


Figure 4.21: Final estimation for Trajectory 1

# Chapter 5

## Measurements and results

This chapter presents the collection of the results coming from algorithm application on the previously explained datasets.

First section is about features robustness evaluation, second section show identification algorithm performance on static datasets, last section show mitigation algorithm performance on both static and dynamic conditions, highlighting achieved improvement in both LOS and NLOS parts of the studied trajectories.

### 5.1 Features robustness evaluation

As explained in 4.3.1, different data collections have been made to assess how general are extracted features and if they are always appropriate to distinguish NLOS and LOS conditions.

#### Static and dynamic NLOS discrimination

The first experimental test has been set up to qualitatively demonstrate the ability of the extracted parameters to highlight the difference between the LOS condition and the NLOS one in a static situation. Two different recordings have been made: first a 50 seconds LOS recording and then a 50 seconds NLOS recording with an obstacle placed in between Anchor and Tag. Figure 5.1 shows how features seem to assume totally different values depending on which is the UWB channel situation. Red line represents NLOS recording, blue line represents LOS recording. As expected, range values are bigger in NLOS condition and features behaviours agree with what would be expected from the theory.

The second experimental test has been set up to qualitatively demonstrate the ability of the extracted parameters to highlight the difference between the LOS condition and the NLOS one in a dynamic situation. A single recording has been made, changing the channel condition dynamically. First there are 20 seconds of

LOS condition, then for 10 seconds an obstacle has been put in between and then again 20 seconds of LOS condition. Figure 5.2 shows how features seem to assume totally different values depending on which is the UWB channel situation also in a dynamic environment. As expected, during NLOS condition, range values are bigger than LOS condition and features behaviours agree with what would be expected.

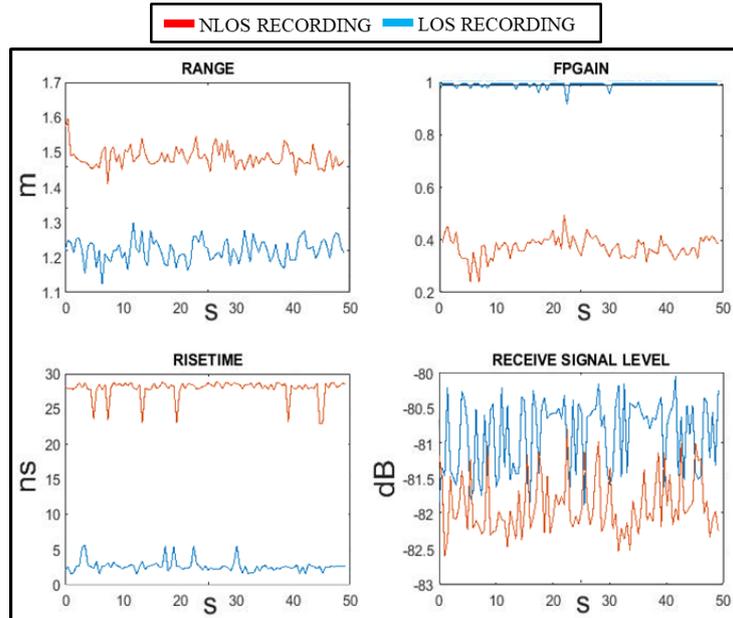


Figure 5.1: Comparison between features behaviour in LOS and NLOS condition

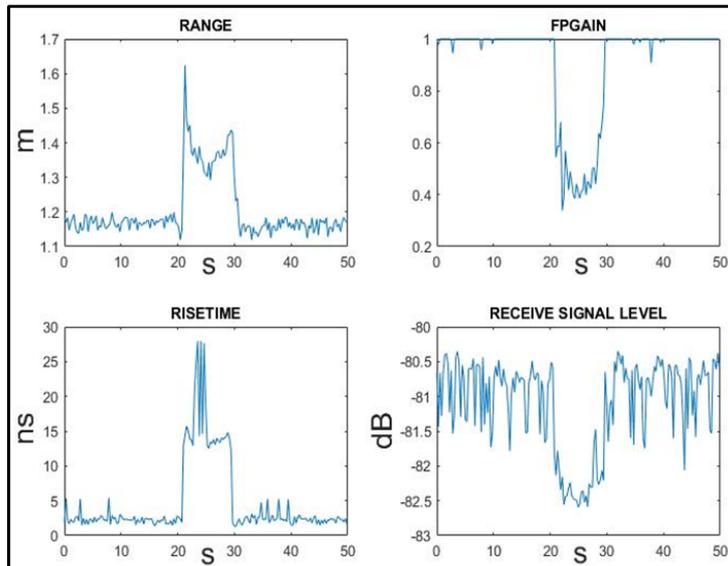


Figure 5.2: Features behaviour in a dynamic NLOS condition

### **Acquisition length influence**

This test has been set up to demonstrate the stability of the extracted features over the time (i.e. collecting different number of acquisitions). Features have been computed with an obstacle in a fixed position and varying the number of acquired samples from 5 to 200. As expected, Figure 5.3 shows how the distinction between LOS and NLOS conditions is not influenced by the length of the acquisition.

### **Obstacle position influence**

This test has been set up to demonstrate the stability of the extracted features respect to different obstacle positions (i.e. collecting data with a different obstacle-boards distance). Features have been computed for a fixed number of acquisitions and varying obstacle distance from the Anchor from 1m to 5m. As expected, Figure 5.4 shows how the distinction between LOS and NLOS conditions is slightly influenced by the position of the obstacle.

### **Antenna orientation influence**

This test has been set up to demonstrate the stability of the extracted features respect to antenna orientations (i.e. collecting data with a different Anchor's antenna orientation respect to the Tag's one). Features have been computed for a fixed number of acquisitions and for a fixed obstacle position, varying Anchor's antenna orientation from  $0^\circ$  (the two antennas are parallel) to  $180^\circ$  (the two antennas are parallel but they show each other different sides). As expected, Figure 5.5 shows how the distinction between LOS and NLOS conditions is not influenced by the antenna orientation.

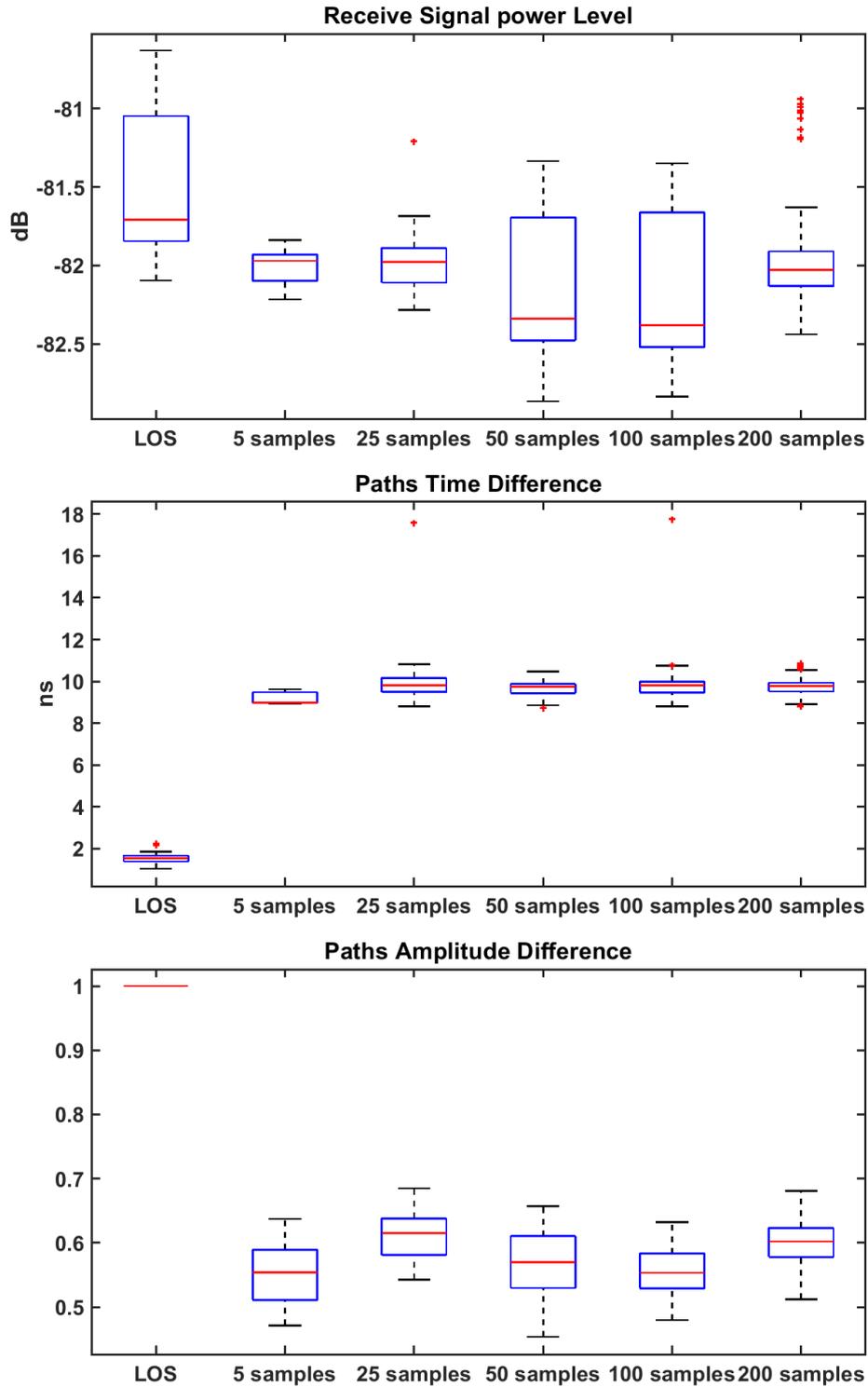


Figure 5.3: Boxplots representing features behaviour varying acquisition length

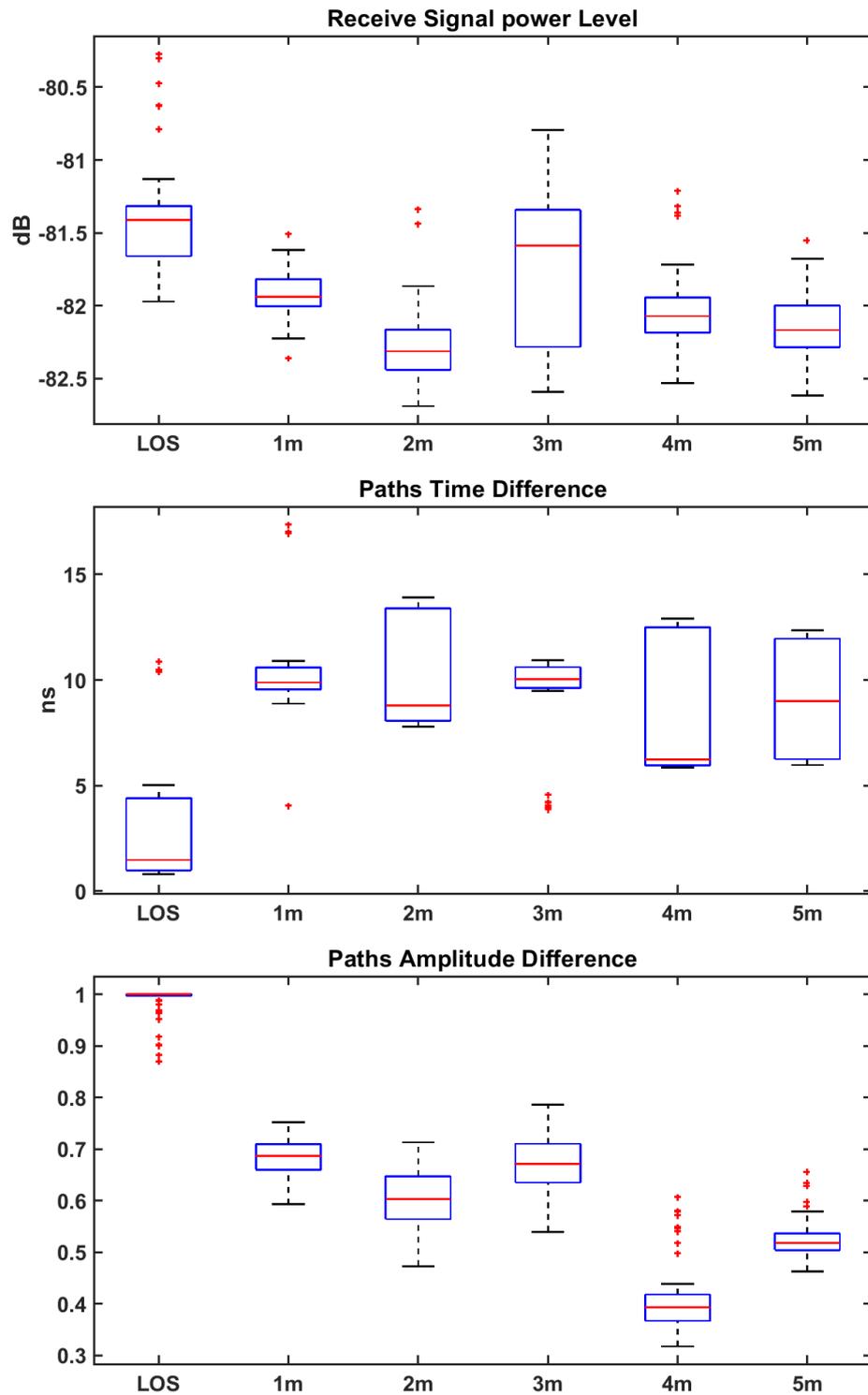


Figure 5.4: Boxplots representing features behaviour varying obstacle position

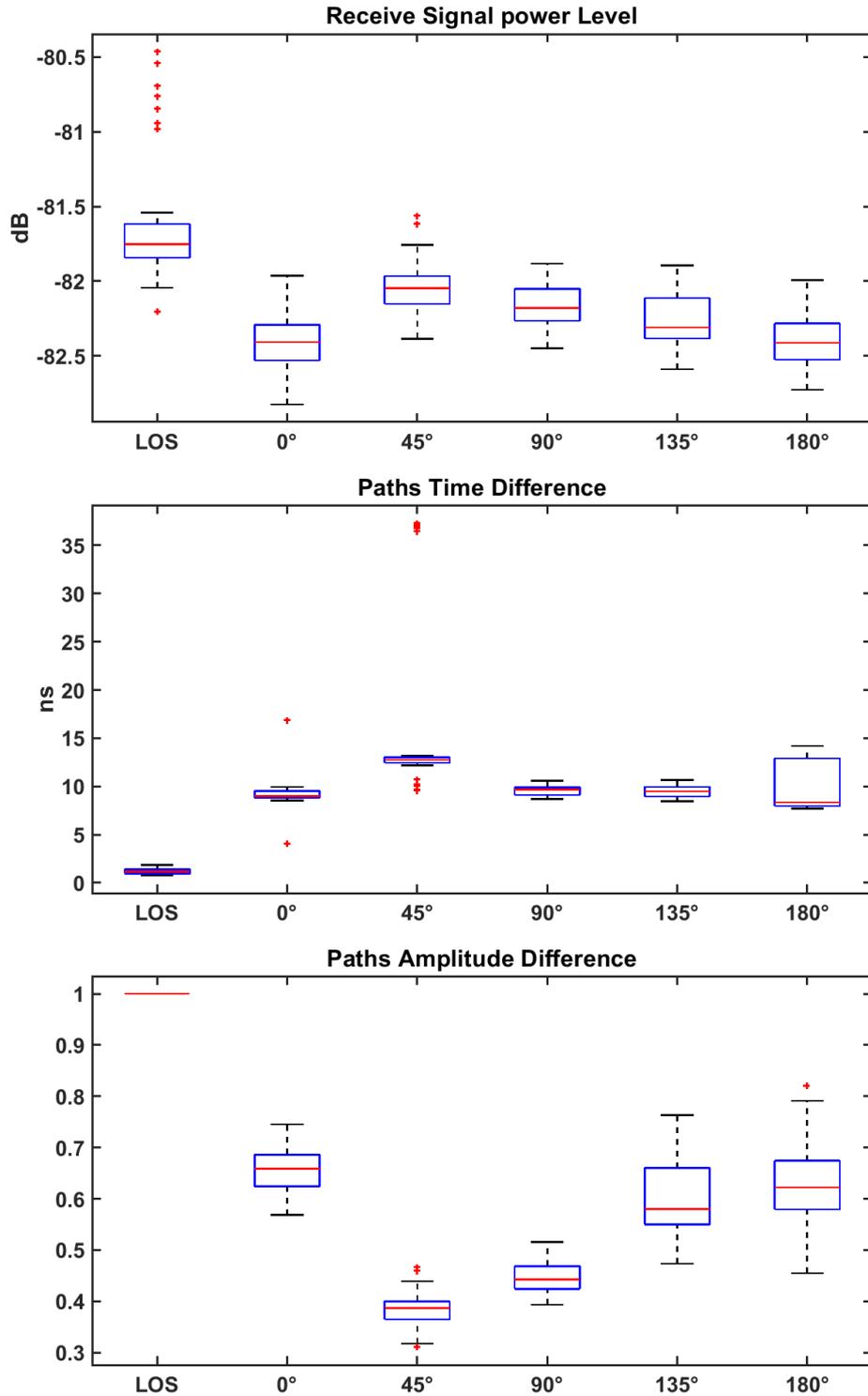


Figure 5.5: Boxplots representing features behaviour varying antenna orientation

## 5.2 Identification algorithm performance

### Data collection for the training set

As explained in 4.3.2, a data collection has been set up to retrieve features values to implement an identification classifier. At the end, 6600 values for each of the three features (split into 4950 triplets coming from LOS recordings and 1650 triplets coming from NLOS recordings) are the data that have been formed the training set for the classification algorithm. Figure 5.6 shows the 3-D representation of the collected features space: blue dots represent LOS recordings, red dots represent NLOS recordings. This dataset represents the training set used to train the classifier to perform the binary identification. The differentiation between the two situations results clear.

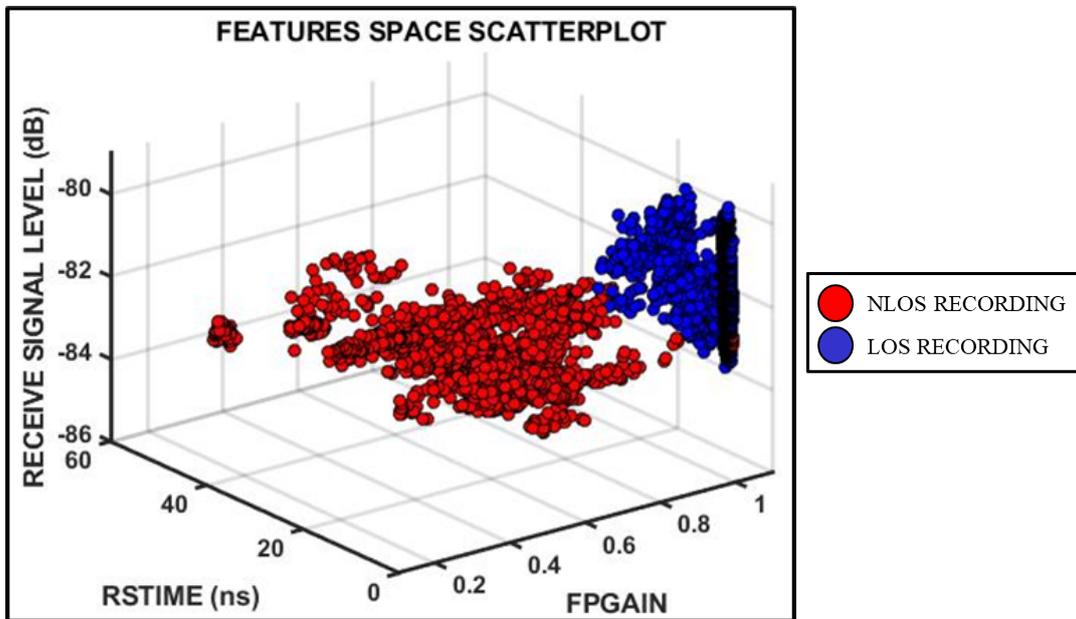
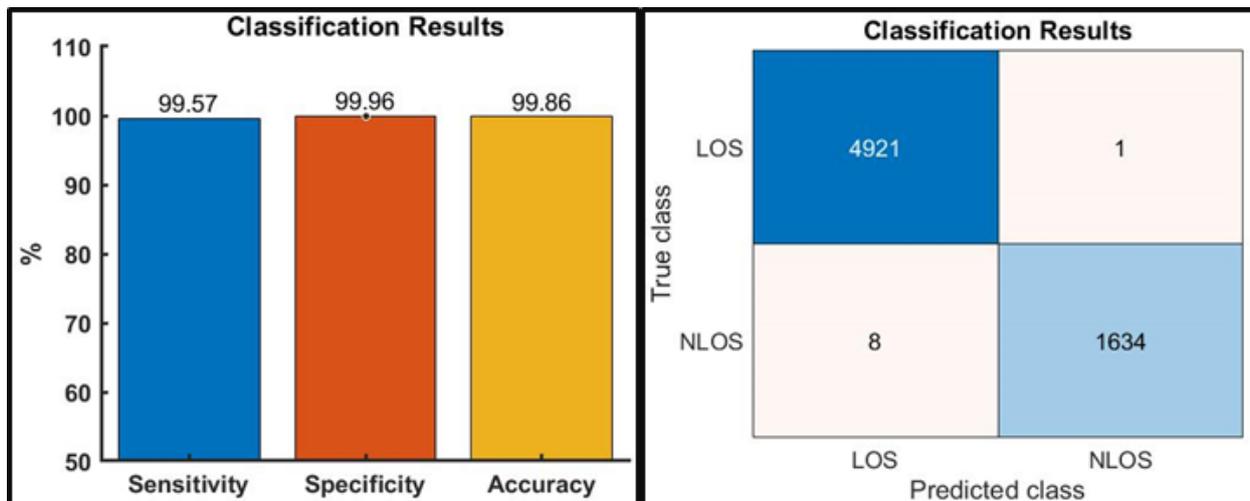


Figure 5.6: Scatterplot graph represents clearly the differentiation between the two situations.

### Classifier construction

As explained in 4.3.3, exploiting Accuracy, Specificity and Sensitivity indicators and the 10-fold cross validation method, SVM with Radial Basic Function kernel function and a C-parameter value equal to 1 has showed the best classification results on training set. Figure 5.7 shows on the left Accuracy, Specificity and Sensitivity achieved values and on the right related confusion matrix from which the indicators have been computed.



(a) Accuracy, Specificity and Sensitivity values

(b) Confusion matrix

Figure 5.7: Identification algorithm performance on training set

### Classifier validation phase

As explained in 4.3.4, a validation phase over other different datasets has been implemented. This testing phase must be thought as a further validation to evaluate the behaviour of the classifier in different conditions and to see the ability of the algorithm to generalize the learned knowledge. Table 5.1 shows classification results (in terms of sensitivity, specificity and accuracy) on validation datasets described in the last chapter. Collected data show as none of the mentioned and checked factors influence classifier performance.

Experimental setup	Sensitivity (%)	Specificity (%)	Accuracy (%)
3 ANCHORS - 2 ANCHORS NLOS MAIN LAB	100	99.94	99.96
3 ANCHORS - 1 ANCHOR NLOS RGR LAB	100	100	100
4 ANCHORS - 1 ANCHOR NLOS MAIN LAB	99.31	98.49	99.15

Table 5.1: Classification results on validation datasets

Summing up all the results obtained by the identification algorithm, it can be stated that NLOS identification can be achieved in different environments and with different working platforms. Table 5.2 shows the overall identification performance, averaging results coming from training set and validation sets and giving also an

idea about performance variability through the standard deviation parameter.

Indicator	Mean (%)	Standard deviation (%)
Sensitivity	99.72	0.34
Specificity	99.60	0.74
Accuracy	99.74	0.40

Table 5.2: Overall NLOS identification performance over training and validation sets

## 5.3 Mitigation algorithm performance

### Static simulations

This section aims to show mitigation results achieved on static simulations (described in 4.4.2).

The simplest mitigation algorithm is the one that discards the affected range, not considering it during trilateration process. An example of a setup where this kind of mitigation is feasible is 4.8, being a platform with four fixed anchors. Exploiting the possibility to rely on four anchors, the discarding mitigation algorithm can be applied. Figure 5.8 show error distribution when, in a platform composed by four anchors, in turn one of them is experiencing NLOS condition and, after being classified as such, it is expelled from the trilateration algorithm. Blue bars represent original trilateration error, red bars represent trilateration error after discarding mitigation algorithm application, yellow bars represent trilateration error when all four anchors are in LOS conditions (it is the reference control). The resulting final trilateration error using only three LOS anchors is similar to what you have in the case of all four anchors in LOS condition, achieving a great improvement respect to the starting situation without any correction.

The proposed Taylor Series-based Least-Square algorithm is useful when the number of available LOS anchors is less than three. The algorithm has been tested on datasets described in Figure 4.13, Table 4.6 and Table 4.7. Table 5.3 shows mitigation results, exploiting indicators described in 4.17, 4.18 and 4.19. In all considered cases, the average final absolute trilateration error is less than 20cm that is the considered benchmark (it is the maximum error declared by Decawave in LOS condition). Reported values have been obtained averaging, for each dataset, performance in all test points and in all combinations of LOS-NLOS anchors.

Figure 5.9 shows correction algorithm results on tested datasets. Red diamonds represent evaluated test points, blue circles represent original collected trilateration results, green circles represent trilateration results after the algorithm has been run on the dataset.

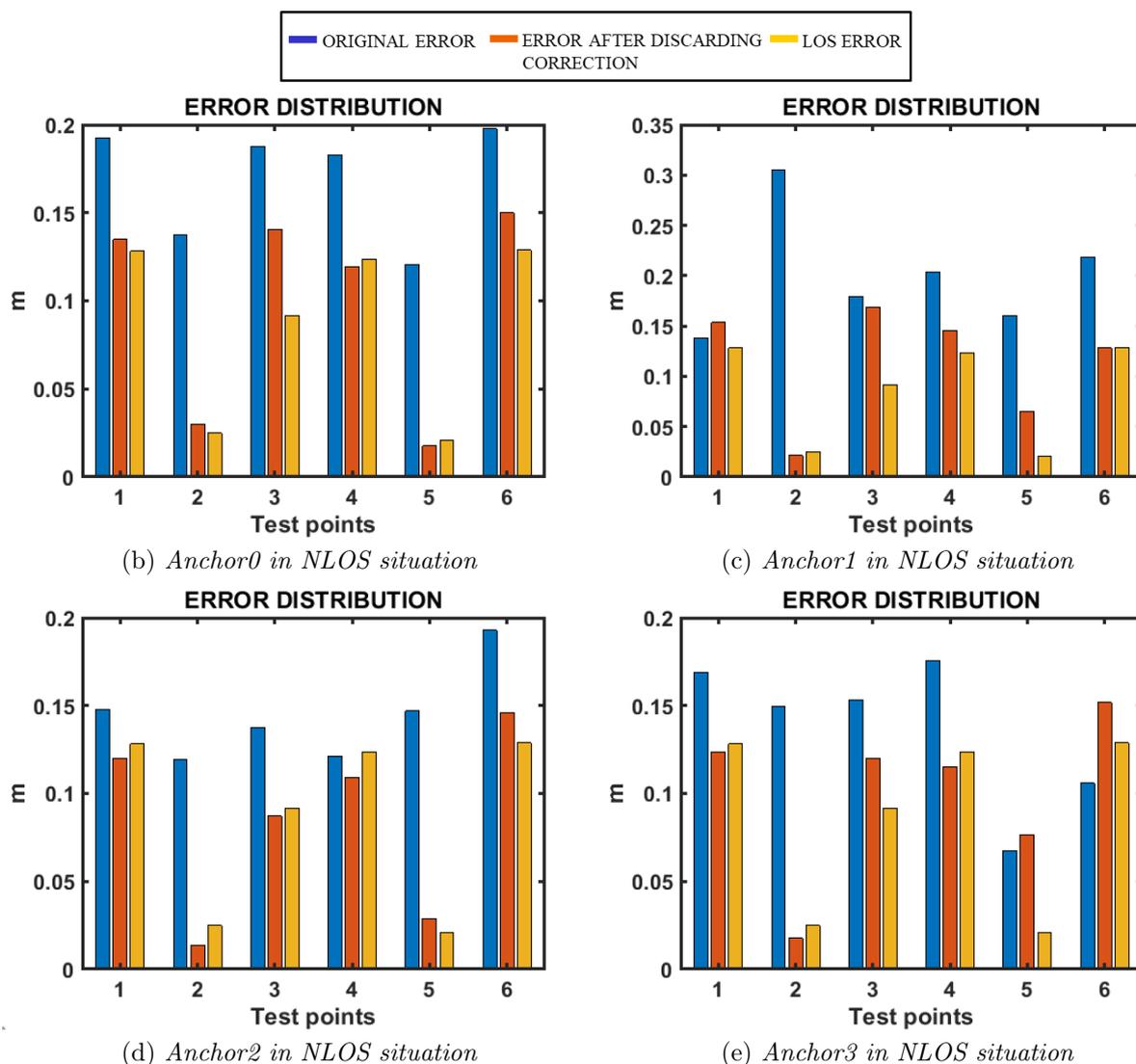
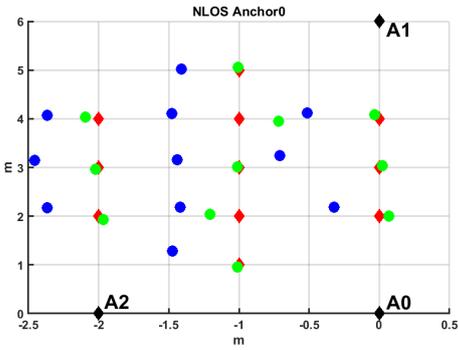
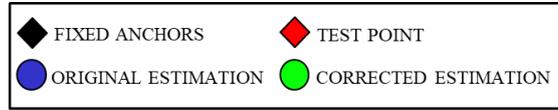


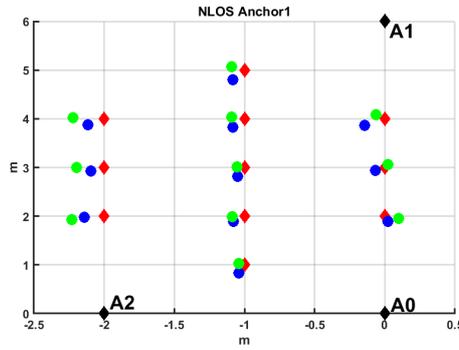
Figure 5.8: Comparison between original error, error after discarding mitigation algorithm application and LOS error

Experimental setup	Mean Improvement (%)	Standard deviation (%)	Original error (m)	Error after correction (m)
3 ANCHORS - 1 ANCHOR NLOS MAIN LAB	55.2	7.4	$0.33 \pm 0.06$	$0.10 \pm 0.03$
3 ANCHORS - 2 ANCHORS NLOS MAIN LAB	33.9	8.3	$0.31 \pm 0.07$	$0.18 \pm 0.08$
3 ANCHORS - 1 ANCHOR NLOS RGR LAB	72.9	12.5	$0.34 \pm 0.07$	$0.07 \pm 0.03$

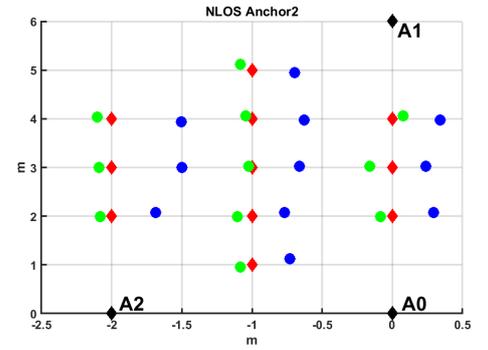
Table 5.3: Mitigation results on tested datasets in terms of improvement and average trilateration error



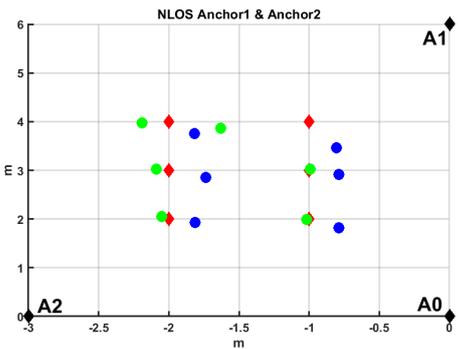
(b)  $A_0$  in NLOS (dataset 4.13)



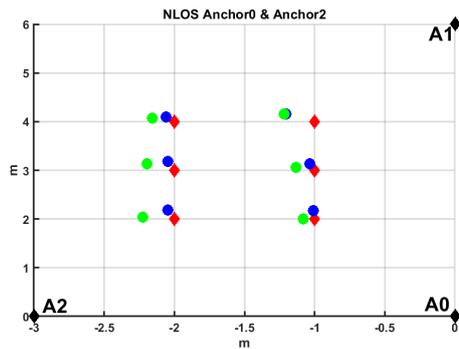
(c)  $A_1$  in NLOS (dataset 4.13)



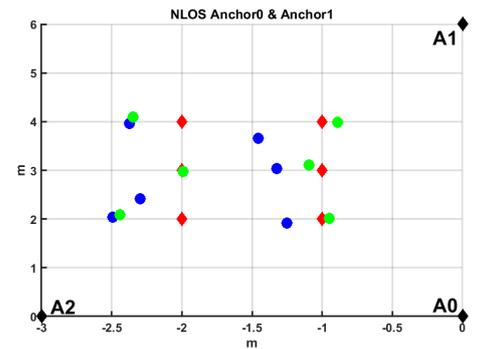
(d)  $A_2$  in NLOS (dataset 4.13)



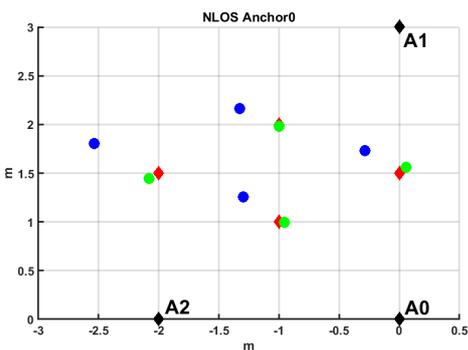
(e)  $A_1$  &  $A_2$  in NLOS (dataset 4.6)



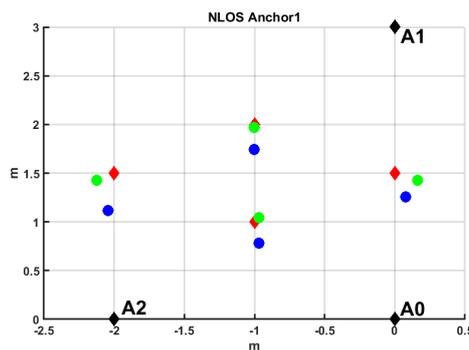
(f)  $A_0$  &  $A_2$  in NLOS (dataset 4.6)



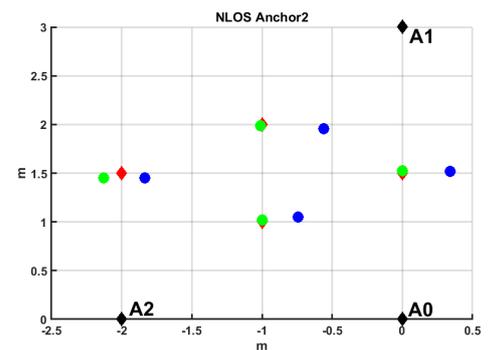
(g)  $A_0$  &  $A_1$  in NLOS (dataset 4.6)



(h)  $A_0$  in NLOS (dataset 4.7)



(i)  $A_1$  in NLOS (dataset 4.7)



(j)  $A_2$  in NLOS (dataset 4.7)

Figure 5.9: Graphic representation of mitigation results in static conditions over different tested datasets

## Dynamic simulations

This section aims to show mitigation results achieved on dynamic simulations (described in 4.4.3).

The purpose of dynamic experimental set-up is to have an estimation of which is algorithm behaviour if the data collection is implemented using a tag that moves continuously (instead of one placed in fixed positions). Home Robot has been programmed to do several trajectories with different shapes to simulate different conditions. The characteristic that all the trajectories have in common is that in each one the two phases are present: Home Robot passes from LOS condition to NLOS one (or vice versa) respect to the anchor with the obstacle in front.

To summarize overall algorithm working principle:

- The first step is to run the classifier to understand which recordings are affected by NLOS error
- The second step is to run the TS-LS algorithm to mitigate NLOS error
- The third step is the application of the transition phase post-processing correction
- The last step is the smoothing process application through a moving average algorithm

The overall algorithm has been tested on seven different trajectories. Figure 5.10 shows, for each tested trajectory, trilateration error before and after mitigation algorithm application, separating LOS (green scale bars) and NLOS (red scale bars) parts of the trajectories. Dark green and dark red bars represent original data, light green and orange bars represent corrected data.

NLOS parts present a great improvement due to the algorithm introduction, while LOS parts are slightly improved. However, after correction application, on average an accuracy of less than 20cm has been achieved: that means that, at the end, also in NLOS conditions the LOS accuracy declared by Decawave has been achieved.

Figure 5.11 and Figure 5.12 show correction algorithm results on tested trajectories (the remaining six in addition to the one shown in Figure 4.21). Black circles represent Vicon output, blue circles represent original collected trilateration estimations, red circles represent trilateration results after TS-LS algorithm has been run, green circles represent final trilateration results after smoothing algorithm application.

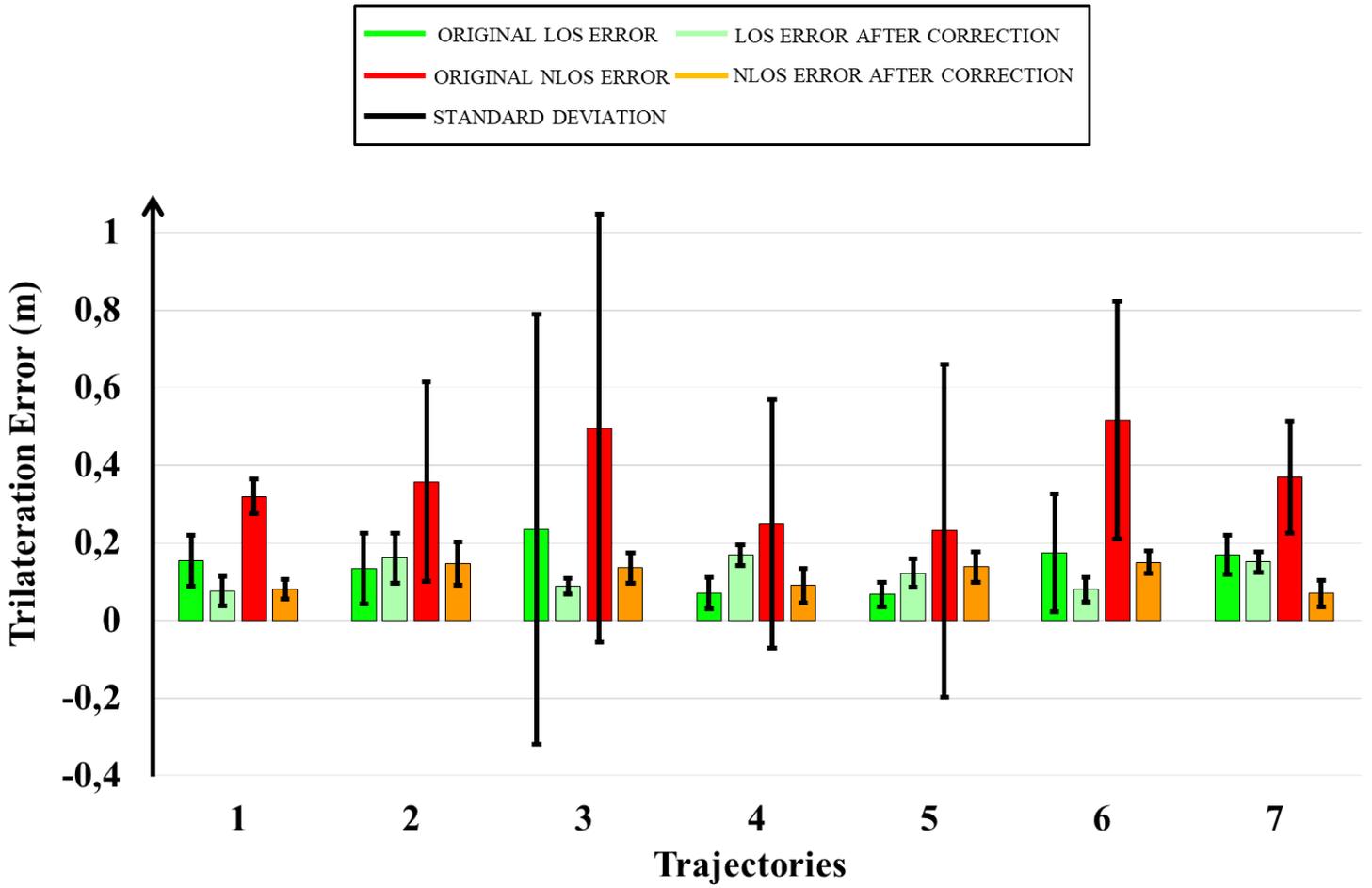


Figure 5.10: Barplot representing achieved mitigation results for analyzed trajectories in terms of absolute trilateration error for both LOS and NLOS parts of each trajectory.

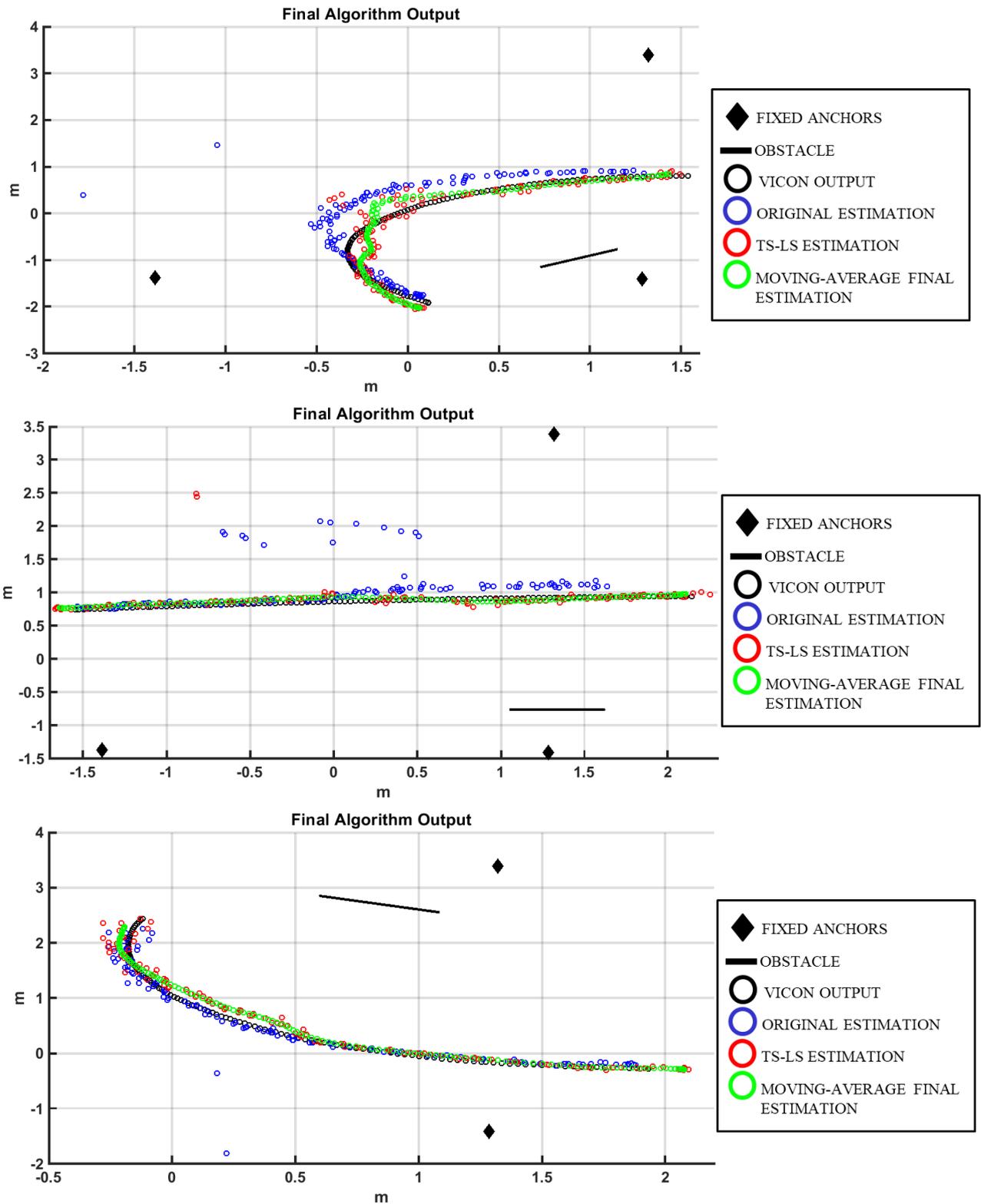


Figure 5.11: Trajectories reconstruction starting from original estimations, with TS-LS algorithm and smoothing algorithm application (Trajectory2, Trajectory3, Trajectory4)

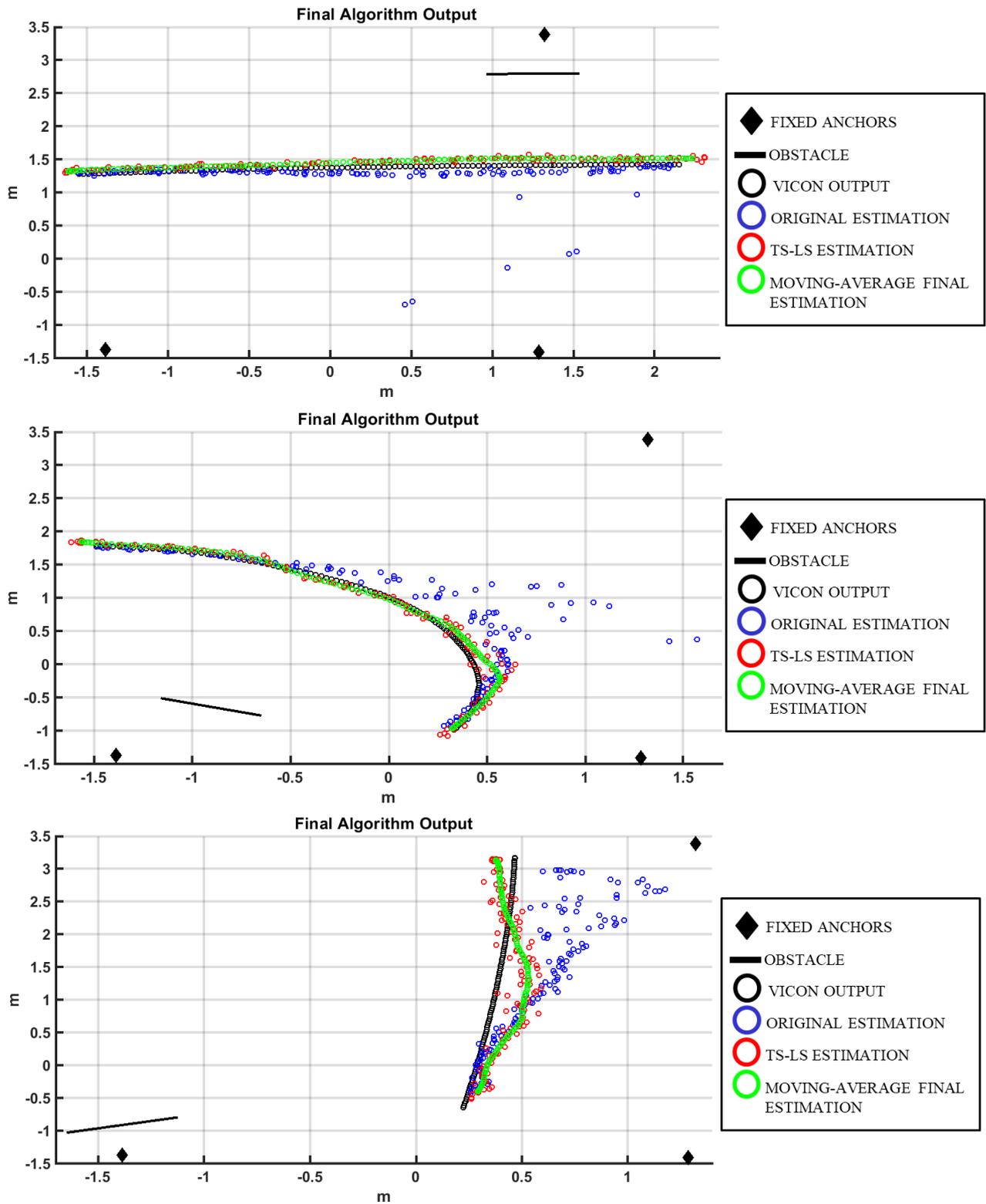


Figure 5.12: Trajectories reconstruction starting from original estimations, with TS-LS algorithm and smoothing algorithm application (Trajectory5, Trajectory6 and Trajectory7)

# Chapter 6

## Conclusions

UWB technology allows the creation of accurate localization systems and provides a great solution for the indoor tracking problem related to the Home Robot's application. With testing, it can be shown that the exploited Decawave TREK1000 Evaluation Kit achieves centimeter accuracy in Tag localization if the LOS condition between Tag and each Anchor present in the infrastructure can be assured. The major challenge faced by the UWB architecture is the presence of obstacles of various kinds in the home environment that could be found between the Tag-Anchor couple during the trilateration process. In these instances, a positive bias is introduced in the range estimation resulting in a wrong final trilateration. Identification of this type of situation is crucial to having a satisfactory Tag localization and subsequently implementing strategies that mitigate NLOS errors in that specific UWB channel. This thesis proposes a low-time consuming and low-complexity algorithm to estimate channel condition and, if the channel is considered affected by NLOS bias, to reduce error introduced by the obstruction.

The channel condition classification algorithm is based on channel parameters extracted from the Channel Impulse Response. Looking at the data collected, three selected features provided a reliable and robust tool to discriminate between LOS and NLOS conditions. Different SVM classifiers fed with these three features have been built: the SVM with Radial Basic Function kernel function and a C-parameter value equal to 1 has showed the best results, with over 99% in sensitivity, specificity and accuracy indicators. Validating the algorithm with three different datasets, more than 99% in each of the three indicators used has been achieved in these cases. Evaluating validation results, algorithm robustness can be stated: NLOS identification can be achieved in different environments and with different platforms, with an overall performance close to 100%. Considering that the proposed algorithm does not require additional hardware and that the involved mathematical complexity is not time-consuming in its implementation, it could be a great choice for the Home Robot application.

The proposed NLOS error mitigation algorithm aims to focus on degenerate cases, that can be considered exceptions. Specifically, when the number of available LOS anchors is less than three, discarding NLOS affected ranges cannot be suitable. The developed Taylor Series (TS) based Least-Square (LS) correction algorithm presents a reduced complexity due to the Taylor Series approximation and at the same time achieves a satisfactory localization accuracy. The proposed algorithm has been tested over different kinds of datasets, from static and dynamic Tag situations. Among static simulations, it has achieved on average an improvement of about 50%. Furthermore, the average final absolute trilateration error is less than 20cm; 20cm can be considered the desired goal as it is the maximum error declared by Decawave in LOS condition. After the correction, NLOS effect is mitigated until achieving the normal declared behaviour of the exploited technology in an ideal working condition. It should be noted that the algorithm's performance decreased when it is applied in a situation with just one LOS anchor (achieving an improvement of 30%), although it is a rare situation in a real scenario since it would be advisable to have a set of redundant anchors. However, this worse performance is expected: exploiting a statistical optimization algorithm, if the cost function becomes less informative (as happens when LOS range numbers gets lower), the final result will be less adherent to the real tag position.

Looking at dynamic simulations, algorithm performance does not particularly suffer if the tag moves continuously. A noticeable problem is the presence of the transition phase (that is physically the most particular situation, because UWB signals experience uncommon obstacle border effects that lead to a decrease in identification performance). Identification accuracy is excellent in LOS areas and hard NLOS areas, while, in the transition phase, recordings that present clear NLOS error are instead classified as LOS or vice versa. This led to a final poor mitigation performance. To overcome this problem a double-correction method has been proposed, followed by a smoothing tool that consists of a moving average along available time series. The absolute trilateration error in the transition phase becomes similar to rest of the trajectory and the smoothing application leads a cleaner more homogeneous final estimation. Both the moving average and the transition phase correction are two kinds of post-processing that introduce a certain delay in the position estimation; according to the author, these delays do not affect the real-time nature of the Home Robot application. Evaluating the correction results among all studied trajectories, it can be noticed that all NLOS paths experience a great error reduction, with an average improvement of 66.1% and an average standard deviation that drops from 29cm to 3cm. There is greater variability among different LOS trajectories. Some experience an improvement, other worsen. The overall average improvement is slightly negative (-8.3%), but also in this case the average standard deviation drops from 14cm to 3cm. This justifies the introduction of the algorithm in these situations.

It must be noted that, also in dynamic conditions, the desired goal is always achieved:

all trajectories present a LOS path and a NLOS path characterized by an absolute trilateration error under the 20cm set threshold.

## 6.1 Future improvements

There is a lot of potential research that could be developed concerning UWB indoor localization systems since this is a relatively new topic.

To start, a first future work could include an extension of the research proposed in this thesis.

Suggested NLOS identification algorithm works with discrete values, analyzing one recording at time. An expansion could be the development of an algorithm that provides a time series analysis, maybe integrating also a more extended data collection and a different features selection to have more informative parameters that could describe better, for example, transition phase condition. About proposed NLOS mitigation algorithm, also in this case the development of an algorithm that exploits extensively time series information is highly recommended. New useful tools of this type could be recursive filters or genetic algorithms. If you want to deal with the first family, probably the most appropriate for this application could be an Extended Kalman Filter, thanks to its high noise cancellation capability, while, if you want to deal with the second family, a great approach could be the introduction of a Particle Filter, thanks to its simplicity since it does not require assumptions about state-space model or state distribution.

Another different researching area related to this thesis could be the integration of different evaluation kits in different rooms to evaluate the architecture capability to handle situations where mobile tag moves from one room to another: the transition between two spaces will be challenging and critical to be managed. A possible solution could be the introduction of a more complex IoT platform to administrate different UWB localization systems, maybe exploiting also the fusion between UWB technology and other type of localization systems (that could work with inertial sensors for example) to improve final accuracy.

Finally, a great area for future research is the improvement of the Home Robot project.

Surely, an important introduction could be an extensive utilization of cameras. Integrating cameras with inertial sensors, Home Robot could be suitable to implement an extensive surrounding environment characterization, allowing the creation of several maps of the home environment in which it is used to work. Being possible this kind of implementation, Home Robot autonomous navigation will be more precise and faster, ensuring, in collaboration with the UWB localization system support, an early intervention when the patient's health could be at risk.

# Bibliography

- [1] Y. Dionyssiotis, “Analyzing the problem of falls among older people”, *International journal of general medicine*, vol. 5, p. 805, 2012.
- [2] J. Fleming and C. Brayne, “Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90”, *Bmj*, vol. 337, a2227, 2008.
- [3] S. Chaudhuri, H. Thompson, and G. Demiris, “Fall detection devices and their use with older adults: a systematic review”, *Journal of geriatric physical therapy (2001)*, vol. 37, no. 4, p. 178, 2014.
- [4] D. Zhang, F. Xia, Z. Yang, L. Yao, and W. Zhao, “Localization technologies for indoor human tracking”, in *2010 5th International Conference on Future Information Technology*, IEEE, 2010, pp. 1–6.
- [5] M. P. Wylie and J. Holtzman, “The non-line of sight problem in mobile location estimation”, in *Proceedings of ICUPC-5th International Conference on Universal Personal Communications*, IEEE, vol. 2, 1996, pp. 827–831.
- [6] J. Borras, P. Hatrack, and N. B. Mandayam, “Decision theoretic framework for NLOS identification”, in *VTC’98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH36151)*, IEEE, vol. 2, 1998, pp. 1583–1587.
- [7] A. Rabbachin, I. Oppermann, and B. Denis, “ML time-of-arrival estimation based on low complexity UWB energy detection”, in *2006 IEEE International Conference on Ultra-Wideband*, IEEE, 2006, pp. 599–604.
- [8] I. Guvenc, C.-C. Chong, and F. Watanabe, “NLOS identification and mitigation for UWB localization systems”, in *2007 IEEE Wireless Communications and Networking Conference*, IEEE, 2007, pp. 1571–1576.
- [9] P.-C. Chen, “A non-line-of-sight error mitigation algorithm in location estimation”, in *WCNC. 1999 IEEE Wireless Communications and Networking Conference (Cat. No. 99TH8466)*, IEEE, vol. 1, 1999, pp. 316–320.

- [10] S. Venkatraman, J. Caffery, and H.-R. You, “Location using LOS range estimation in NLOS environments”, in *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No. 02CH37367)*, IEEE, vol. 2, 2002, pp. 856–860.
- [11] B. L. Le, K. Ahmed, and H. Tsuji, “Mobile location estimator with NLOS mitigation using Kalman filtering”, in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, IEEE, vol. 3, 2003, pp. 1969–1973.
- [12] J. Reed, *Introduction to ultra wideband communication systems*, an. Prentice Hall Press, 2005.
- [13] I. Guvenc and Z. Sahinoglu, “Threshold-based TOA estimation for impulse radio UWB systems”, in *2005 IEEE International Conference on Ultra-Wideband*, IEEE, 2005, pp. 420–425.
- [14] S. M.-S. SADOUGH, “A tutorial on ultra wideband modulation and detection schemes”, *Shahid Beheshti Univ. Fac. Electr. Comput. Eng, Tehran, IR Iran*, no. April, pp. 1–22, 2009.
- [15] F. Despaux, A. van den Bossche, K. Jaffrès-Runser, and T. Val, “N-TWR: An accurate time-of-flight-based N-ary ranging protocol for Ultra-Wide band”, *Ad Hoc Networks*, vol. 79, pp. 1–19, 2018.
- [16] J. R. Fernandes and D. D. Wentzloff, “Recent advances in IR-UWB transceivers: An overview.”, in *ISCAS*, vol. 10, 2010, pp. 3284–3287.
- [17] *Zebra UWB Technology Datasheet*, [Online; accessed 20-February-2019], 2019. [Online]. Available: [https://www.zebra.com/content/dam/zebra\\_new\\_ia/en-us/solutions-verticals/product/location\\_solutions/zebra-ultra-wideband/spec-sheet/uwb-portfolio-spec-sheet-en-us.pdf](https://www.zebra.com/content/dam/zebra_new_ia/en-us/solutions-verticals/product/location_solutions/zebra-ultra-wideband/spec-sheet/uwb-portfolio-spec-sheet-en-us.pdf).
- [18] *Pozyx UWB Technology Datasheet*, [Online; accessed 20-February-2019], 2019. [Online]. Available: <https://www.pozyx.io/shop/product/ready-to-localize-67>.
- [19] *Decawave DW1000 Datasheet*, [Online; accessed 20-February-2019], 2019. [Online]. Available: <https://www.decawave.com/sites/default/files/resources/dw1000-datasheet-v2.09.pdf>.
- [20] *DecarangeRTLS Arm Source Code Guide*, version Version 2.1, 2015.
- [21] J. Kulmer, S. Hinteregger, B. Großwindhager, M. Rath, M. S. Bakr, E. Leitinger, and K. Witrisal, “Using DecaWave UWB transceivers for high-accuracy multipath-assisted indoor positioning”, in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, 2017, pp. 1239–1245.
- [22] *DW1000 Metrics for Estimation of Non Line Of Sight Operating Conditions*, version Version 1.1, 2016.

- [23] B. Tank, *Identification and Mitigation of NLOS based on Channel Information*, [Master Thesis, University of Windsor], 2017.
- [24] T. Jebara, “Multi-task feature and kernel selection for SVMs”, in *Proceedings of the twenty-first international conference on Machine learning*, ACM, 2004, p. 55.
- [25] S. Venkatesh and R. M. Buehrer, “NLOS mitigation using linear programming in ultrawideband location-aware networks”, *IEEE transactions on Vehicular Technology*, vol. 56, no. 5, pp. 3182–3198, 2007.
- [26] M. Capra, *UWB Tracking System for Patient Monitoring in Home Environment*, [Master Thesis, Politecnico di Torino], 2018.
- [27] K. Yu and Y. J. Guo, “NLOS error mitigation for mobile location estimation in wireless networks”, in *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*, IEEE, 2007, pp. 1071–1075.
- [28] W. H. Foy, “Position-location solutions by Taylor-series estimation”, *IEEE Transactions on Aerospace and Electronic Systems*, no. 2, pp. 187–194, 1976.