

The front page is supposed to lie at this place, use of the bruface template required.



# Abstract

This is the abstract blablabla...

*Keywords:* Ultrawide Band, ...

# Acknowledgements

I thank ...

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>State of the art</b>	<b>3</b>
2.1	Ultra-Wideband Technology . . . . .	3
2.2	Real Time Locating Systems . . . . .	4
2.2.1	Symmetric double sided two-way ranging . . . . .	5
2.2.2	Trilateration . . . . .	6
2.3	Implementation of a locating system . . . . .	7
2.3.1	DWM1000 . . . . .	7
2.3.2	Anchor . . . . .	8
2.3.3	Tag . . . . .	9
2.3.4	Android Application . . . . .	10
2.3.5	Precision reached . . . . .	11
2.4	Multi-path aided locating system . . . . .	12
2.4.1	Ray Tracing . . . . .	13
2.4.2	Channel Impulse Response . . . . .	14
2.4.3	Virtual Anchors . . . . .	14
2.4.4	Locating system . . . . .	15
<b>3</b>	<b>Algorithms</b>	<b>16</b>
3.1	Peaks extraction . . . . .	16
3.1.1	Soft case . . . . .	17
3.1.2	Hard case . . . . .	18
3.2	Hard localization algorithm . . . . .	19
3.2.1	Virtual antennas combination . . . . .	19
3.2.2	System solver . . . . .	20
3.3	Soft localization algorithm . . . . .	21
3.3.1	Space solution reduction . . . . .	21
3.3.2	Time MSE . . . . .	22
<b>4</b>	<b>Simulations</b>	<b>24</b>
4.1	Simulation . . . . .	24
4.1.1	Realistic CIR generation . . . . .	24
4.1.2	Code possible outputs ? . . . . .	25
4.2	Hard Simulation . . . . .	26
4.3	Soft Simulation . . . . .	26
<b>5</b>	<b>Experiences</b>	<b>27</b>
5.1	Experimental set-up configuration . . . . .	27
5.1.1	DWM1000 . . . . .	27
5.1.2	PSoC . . . . .	28

5.1.3	Android Application . . . . .	29
5.2	Test Campaign . . . . .	30
<b>6</b>	<b>Conclusion</b>	<b>31</b>
<b>A</b>	<b>Application view</b>	<b>32</b>
<b>B</b>	<b>State Machines</b>	<b>33</b>
	<b>Acronyms</b>	<b>35</b>
	<b>Bibliography</b>	<b>38</b>

# Chapter 1

## Introduction

Nowadays, indoor localization is a growing feature of many smart buildings, either to allow the navigation of autonomous robots, navigate in an unknown building using a indoor version of the Global Positioning System (GPS), etc. The widespread GPS is not suited for indoor localization [2] [18], in this scope, an indoor locating system using the Ultra-Wideband (UWB) technology and based on trilateration has been developed in [16].

This locating system locates a mobile device, called tag using three fixed beacons, named anchors, placed at known position. The schematic of this system is shown in Fig. 1.1. The tag is mainly composed of an UWB antenna and a smartphone with a custom android application developed to display the position and orientation of the tag on a given map.

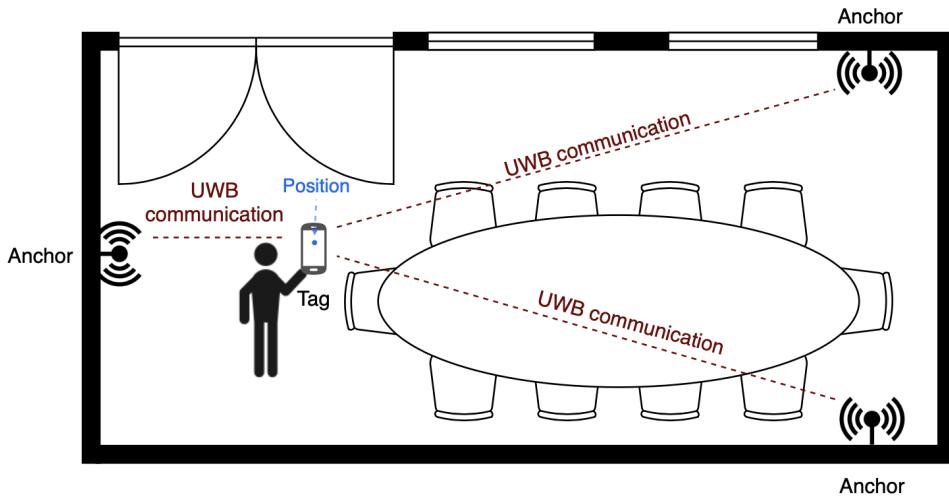


Figure 1.1: Schematic of an indoor locating system, a tag interacting with three anchor to know its position in the room. Taken from [16].

This locating system is based on the assumption that a Line of Sight (LoS) between the tag and the anchor is always clear, no object or body obstructing it. Since this can not be guaranteed for every location at every moment for indoor location, two main possibilities occurs. The first one is to increase the number of available anchors, statistically increasing the probability to always have two LoS. This approach has been studied in [15]. The second approach, is to develop a locating system that may function with less than three antennas available. This thesis focuses on this second approach, studying and qualifying a locating system using only one anchor.

In the second chapter, a state of the art is presented, summarizing the previous work made on UWB, Real-time locating systems (RTLS), indoor locating systems, describing the functioning of the locating system implemented in [16] and detailing the theory behind the some one anchor locating systems. In the third chapter, two different algorithms developed to perform the indoor localization using only one anchor are presented.

Chapter four focuses on the simulation developed to test and study the algorithms developed in chapter 3. The functioning of the simulation is first presented and the simulation results of the two algorithms are then presented. The fifth chapter focuses on the implementation and modification of the experimental set-up in order to test and extract experimental data to test the algorithms in a real environment.

The last chapter comes back on the work carried out and the results obtained during this thesis, outlining the research avenues that should be investigated in further work, concluding this thesis.

### **Message to the reader**

Before reading this thesis, the reader must be warned that the last section will seems rather unfinished. Due to the COVID-19 situation and the resulting lock-down, the experiment part, which was supposed to be the main focus of this thesis, was compromised. From that moment, the scope of the thesis was reoriented towards simulations that should be as realistic as possible in order to conduct the same sort of analysis as would have been done using experimental data.

# Chapter 2

## State of the art

This chapter outlines the state of the art. The first part focuses on the implementation of a locating system, which is introduced after a short explanation of the UWB and RTLS. Afterwards, the concept of virtual anchors and multi-path aided locating systems will be discussed.

### 2.1 Ultra-Wideband Technology

UWB is a communication technology using, as the name states, a large bandwidth. This is not a new technology as it is the one used by Guglielmo Marconi for the first transatlantic communication using radio waves [23]. As defined by the International Telecommunication Union Radiocommunication Sector (ITU-R) it is to be considered as UWB if the bandwidth of communication is at least 20 % of the arithmetic center frequency [24].

One interesting feature of UWB is the possible coexistence with other radio waves already present in the environment such as Wireless Fidelity (Wi-Fi). As it can be seen on Fig. 2.1, the extension of the UWB in the spectral domain is quite large.

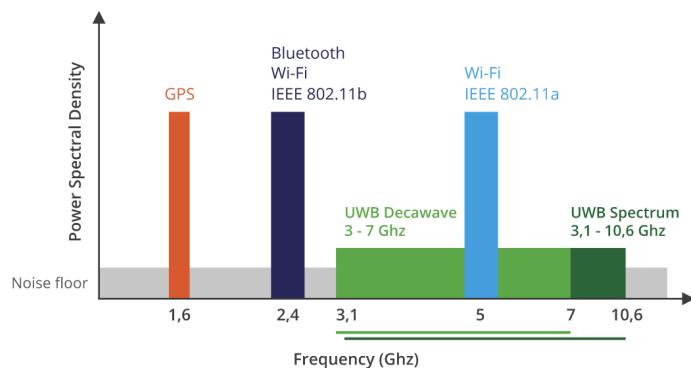


Figure 2.1: UWB spectrum compared to Wi-Fi and other wireless technology. Taken from [24].

Knowing this and based on the time-frequency duality reminded in eq. 2.1, we can see that the extension in the time domain will be quite small compared to other signals type. This is a direct effect of the Fourier transform, the extension of a function and it's extension of it's Fourier transform are inversely proportional. This follows the principle of uncertainty assessing that a trade-off needs to be done between the precision reached in the time domain and that in the frequency domain [10].

$$x(at) \longleftrightarrow \frac{1}{|a|} * X\left(\frac{f}{a}\right) \quad (2.1)$$

The Fig. 2.2 shows the theoretical duration of an impulse of the UWB. We can see that the time extension of the significant part of the pulse is from 0.3 to 0.7 nanoseconds, leading to a pulse duration of 0.4 nanoseconds.

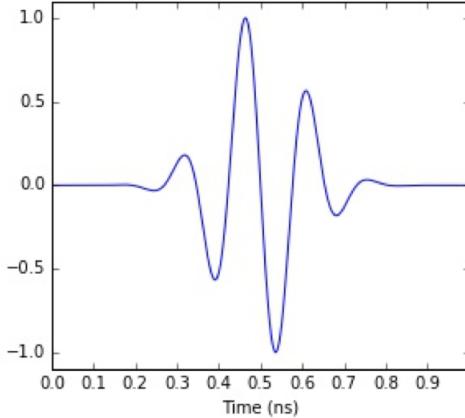


Figure 2.2: Theoretical duration of an UWB pulse. Taken from [9].

An advantage of the UWB is its robustness in regard to the Multipath Channel (MPC). Multipath stands for the propagation phenomena occurring in wireless communication transmission, the signal being able to reach the receiver through several path. The direct one, called LoS, and the indirect one through reflection, diffraction, scattering called Non Line of Sight (NLoS). This can be understood by looking at Fig. 2.3, where several peaks can be distinguished, each corresponding to a different path followed by the wave. Indeed, the probability to have a collision depends on the size of the pulse sent. Therefore, the interest of the UWB in confined area appears as a lot of MPC are present because of the reflections coming from all walls of a room.

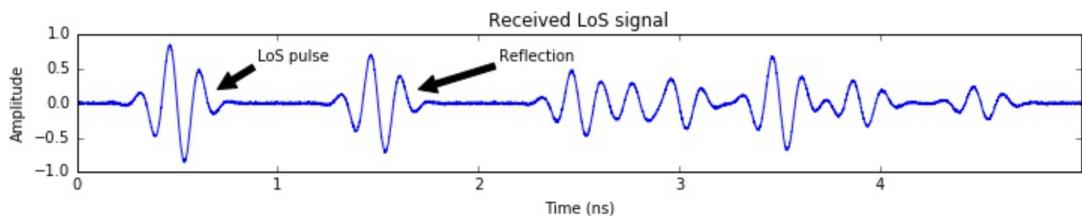


Figure 2.3: Example of an MPC with a LoS and a NLoS. Taken from [9].

## 2.2 Real Time Locating Systems

RTLS are systems used to track and identify the location of objects in real time. This is a rather vague definition since nothing is specified about the means used to achieve the localization. The RTLS that will be shown in this section will all have in common the use of wireless communications, between devices called "anchor" and "tag", the anchor is at a fixed and known location while the tag being associated with the object to locate. Those RTLSs can be split into two categories : "Relative localization" and "Absolute localization".

The relative localization algorithm introduced is the Symmetric double-sided two-way ranging (SDS-TWR), which output a Time of Flight (ToF) representing the needed time for a transmission to transit between the transmitter and the receiver. This method allows to gain some information about the position of the tag relatively to the anchor. This algorithm has been detailed and compared to several other methods in [12], [16].

The absolute localization algorithm performs the localization in a x-y plane<sup>1</sup> based on the three different ToF obtained.

### 2.2.1 Symmetric double sided two-way ranging

SDS-TWR consists in an exchange of three messages between two Ranging-capable DEVICES (RDEVs), respectively ' $RDEV_1$ ' initiating the communication and ' $RDEV_2$ '. Each device needs to save the Time of Emission (ToE) or Time of Arrival (ToA) of every message. Those times being respectively  $t_0, t_1$  for the first message,  $t_2, t_3$  for the second message and  $t_4, t_5$  for the last message.

Each message contains the different timestamps, previously computed, meaning that at the end of this exchange  $RDEV_2$  possess all the information about the timestamps, while  $RDEV_1$  misses the last one. If we want  $RDEV_1$  to be able to compute the ToF, a last message containing  $t_5$  should be exchanged.

A schematic of the exchanges between  $RDEV_1$  and  $RDEV_2$  that occurs in SDS-TWR is shown in Fig. 2.4.

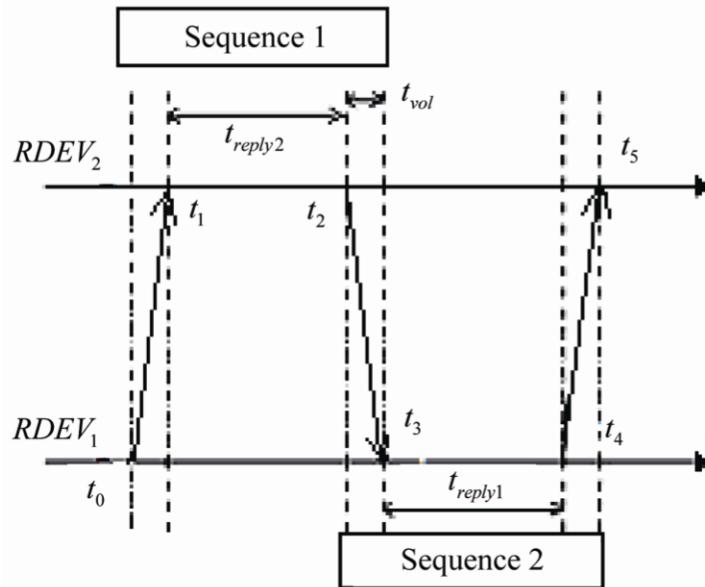


Figure 2.4: Symmetric double-sided two-way ranging. Taken from [4].

Based on those timestamps, the computation of the ToF can be observed in eq. 2.2. This estimation is the arithmetic means of the ToF from ' $RDEV_1$ ' to ' $RDEV_2$ ' and vice-versa.

$$t_{est} = \frac{((t_3 - t_0) - (t_2 - t_1)) + ((t_5 - t_2) - (t_4 - t_3))}{4} \quad (2.2)$$

Since that ToF computed remains an estimation, it is important to know the magnitude of the error as well as its evolution in parallel with the true value of the ToF.

---

<sup>1</sup>Which can be extended in a x-y-z space if needed.

$$t_{true} - t_{est} = \frac{1}{4} * ((t_2 - t_1) - (t_4 - t_3)) * (e_1 - e_2) \quad (2.3)$$

The term  $e_1 - e_2$  being the difference between the internal clocks of both devices. [4]

### 2.2.2 Trilateration

The SDS-TWR allowing us to compute the ToF, the relative distance can be computed using the light speed. If the relative distance between a tag and three different anchors is known, it is possible to compute the intersection of three circles having as center the position of the anchor and radius corresponding to the ToF associated with this anchor. A scheme displaying that solution can be seen on Fig. 2.5.

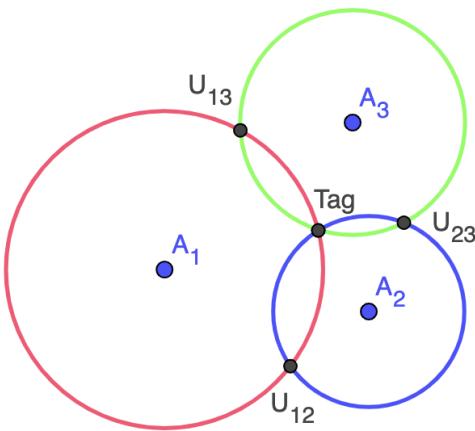


Figure 2.5: Trilateration

As we can deduce, in a two dimensional plan, three anchors are needed to have an intersection of only one point, removing the uncertainty of the solution. If only two anchors were used, we would obtain two possible solutions :  $Tag$  and  $U_{XY}$ . In a three dimensional space, four anchors would be needed. This actually corresponds to the following system of equations :

$$\begin{cases} (x_1 - x_0)^2 + (y_1 - y_0)^2 = d_1^2 \\ (x_2 - x_0)^2 + (y_2 - y_0)^2 = d_2^2 \\ (x_3 - x_0)^2 + (y_3 - y_0)^2 = d_3^2 \end{cases} \quad (2.4)$$

Where the couple :  $p^i = (\vec{x}_i, y_i)$  corresponds to the position of the anchor  $i$  and  $d_i$  corresponds to the distance between this anchor and the tag,  $(x_0, y_0)$  being the position of the tag.

### Uncertainties

Because of the inaccuracy of the computed distances, the system 2.4 can not be solved. There is a lot of probability that there is not a single point  $(x_0, y_0)$  that solves it. To avoid this problem, the estimator  $S(\vec{p})$  developed in [25] is used.

$$S(\vec{p}_0) = \sum_i^N (\|\vec{p}_i - \vec{p}_0\|^2 - d_i^2)^2$$

$$\vec{p}_0 = \underset{\vec{p}}{\operatorname{argmin}} S(\vec{p}) \quad (2.5)$$

Where  $\vec{p}_k = (x_k, y_k) \forall k \in 0, \dots, N$  in the two dimensional cases,  $N$  being the number of anchors<sup>2</sup>. The objective of this estimator is to find the estimate  $\vec{p}_0$  minimizing the value of  $S(\vec{p}_0)$ .

## 2.3 Implementation of a locating system

When using the technology briefly developed in sections 2.1 and 2.2, a locating system has been developed by Quentin Fesler and Cédric Hannotier in [12], [16]. This system is able to find back a localization with an error oscillating between twenty and fifty centimetres inside a building [15].

This locating system is composed of fixed antennas called anchors, based on ESP8266 as micro-controller and a Decawave DWM1000 UWB transceiver[7]. The tag are built using an Android cellphone, a PSoC<sup>3</sup> and also a DWM1000 module.

### 2.3.1 DWM1000

The DWM1000 is the antenna chosen to operate the wireless communication, it will be needed for the tag as well as for the different anchors. The configuration of those antennas and the Serial Protocol Interface (SPI) communication are both explained in this section.



Figure 2.6: DWM1000 module. Taken from [7].

### Configuration

Before using the DWM1000, tests have been done to choose a configuration that minimizes the error rate and the consumption while maximizing the communication speed. This leads to the following choices<sup>4</sup> :

- Channel number : 5
- Bitrate : 6.8 Mbits<sup>-1</sup>
- Pulse Repetition Frequency (PRF) : 16MHz

---

<sup>2</sup>Which can be superior to 3, even in a 2D plan.

<sup>3</sup>The exact model is the : CY8C5888LTI-LP097 [15].

<sup>4</sup>A more detailed discussion on the choice of those parameters can be found in [16].

- Preamble length : 128 bits

The chosen channel number is the number 5 partly because of the European Union (EU) regulations that are more strict in the frequencies bounds from 3.1 to 4.8 GHz than in the frequencies bounds from 6 to 9 GHz[6]. The other channel that stays in those boundaries in the 7<sup>th</sup> one. The difference being a bandwidth being twice as large<sup>5</sup>. The channel 7 also lies within this region of the electromagnetic spectrum, but was not chosen. The difference between those channels lays in the size of the bandwidth.

The choices of the bitrate are restricted between 110 kbit s<sup>-1</sup>, 850 kbit s<sup>-1</sup> or 6800 kbit s<sup>-1</sup>. The reasons behind the choice of the bitrate at 6.8 Mbit s<sup>-1</sup> are detailed in [16]. This main reason is the existence of a "Smart Transmit Power Control" allowing to increase the power of transmission under certain conditions, one of them is that the bitrate must be set at 6.8 Mbit s<sup>-1</sup>.

The PRF can be chosen between 16 MHz and 64 MHz, a higher one increasing the operating range while consuming more power.

The preamble length is used to rate the channel estimation, which describes how a signal propagates from the transmitter to the receiver, in order to perform the equalization to flatten the frequency response. The precision of the estimation increases with the size of the preamble. Unfortunately, a longer preamble means that a larger proportion of the energy consumed will not be used effectively, decreasing the energy efficiency of the setup, and that a shorter time window will be available to transmit actual data. Recommended bitrate in function of the channel is proposed in [8].

## Control

The DWM1000 is driven via an SPI bus, this communication follows a master-slave scheme where the master, which is the micro-controller, controls the communication [3]. On Fig. 2.7, the four needed signals are displayed. Master Input, Slave Output (MISO) and Master Output, Slave Input (MOSI) are the connections used to transmit the data between the master and its slave. The Serial Clock (SCLK), generated by the master defines the speed at which the MISO and MOSI exchange data. Since the SPI allows different slaves for only one master, the Slave Select (SS) is used by the master to select which specific slave it communicates with.

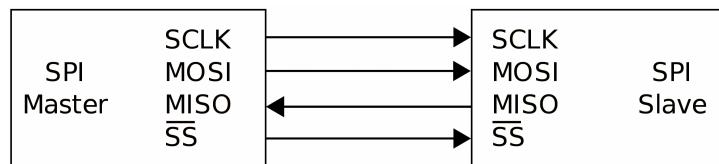


Figure 2.7: SPI Schematic.

### 2.3.2 Anchor

The anchors are permanent antennas composed of a DWM1000 and an ESP8266 [11]. They are placed at known positions in the room and are used to compute the ToF between each and the tag using the SDS-TWR, as explained in section 2.2.1.

---

<sup>5</sup>The bandwidth of the 5<sup>th</sup> one is 499.2MHz while the one from the 7<sup>th</sup> is 1081.6 MHz.



Figure 2.8: ESP8266 mounted on a Feather Huzzah board. Taken from [1]

The micro-controller has been combined with the development board Feather Huzzah from Adafruit [1]. This module is to be seen in Fig. 2.8. This board can be flashed using a USB serie connection, allowing an easy deployment of the code. It also has the advantage of being light and small, a feature which is useful when we want to set several anchors in a room.

### 2.3.3 Tag

The tag is the object of which the localization has to be determined. It is composed of a DWM1000 antenna, a PSoC<sup>6</sup> and an Android application. The Fig. 2.9 shows the PSoC used as well as its custom board made by the electronic BEAMS service of the ULB.

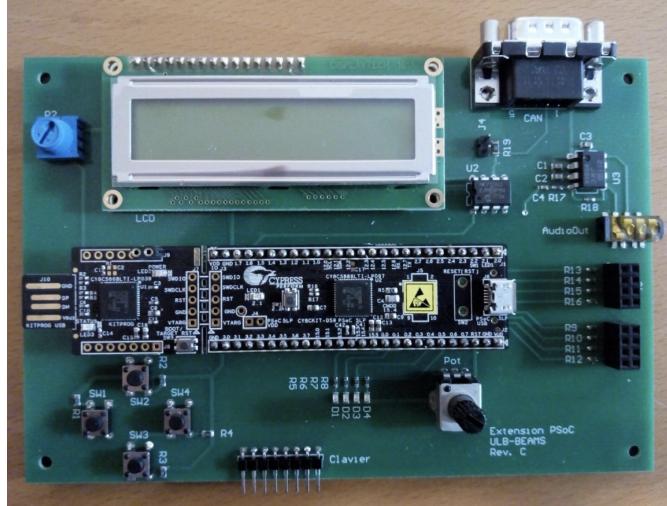


Figure 2.9: PSoC micro-controller put on a custom board made by the BEAMS service.

The communication between the DWM1000 and the PSoC is performed using a SPI bus as for the anchors, the PSoC being the master. As for the ESP8266, the PSoC can be flashed through an USB bus. The micro-controller receives instructions from the application on the cellphone and controls the communications of the DWM1000 with the different anchors. It then transmits the received data from the DWM1000 to the application through an USB bus.

---

<sup>6</sup>The exact model is the CY8C5888LTI-LP097 [15].

### 2.3.4 Android Application

To control the PSoC, an Android application has been developed. A screen-shot of the main window can be seen in the appendix A. It exhibits four buttons, whose functionalities are revealed in what follows.

#### Navigation

The Navigation button opens a map of an environment<sup>7</sup> and an arrow is displayed at the supposed location of the tag, the orientation being the estimation of the orientation of the cellphone. The coordinates are also shown, computed from the bottom left corner of the map. The used map is shown in Fig. 2.10.

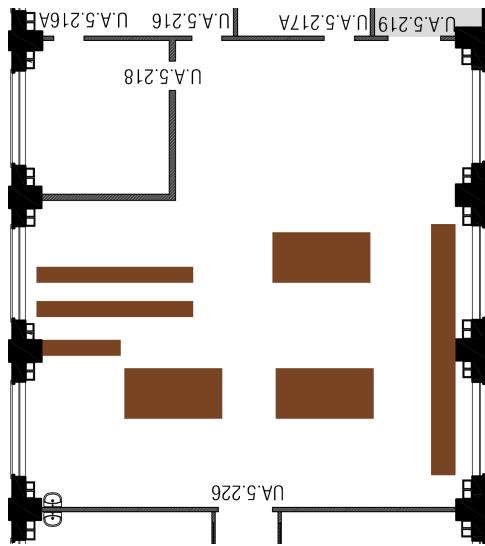


Figure 2.10: Map of the UA5.214

When this function runs, the application enters a loop that continuously requests the PSoC to perform a SDS-TWR with each anchor. The data containing the different timestamps are transmitted towards the application for each anchor separately. From those data, the distance between each anchor and the tag is computed on the cellphone according to the trilateration algorithm described in section 2.2.2, leading to an estimate of the location. The biggest computations are done on the cellphone because of the bigger computational power available in comparison with the PSoC.

A detailed state machine representing the interactions between the tag and an anchor can be found in appendix B. While the anchor only performs one SDS-TWR at a time, the tag needs to keep an history of the anchors contacted to perform the trilateration afterwards.

#### Test USB connection, Test orientation

The Test USB connection allows to test that the USB communication bus used to communicate with the PSoC is fully operational. The test procedure consists of sending a 16 bits long messages to the PSoC, this message being : sta. If the PSoC is well connected, the application is supposed to receive a 32 bits long message : 0x02034637. This feature allows to quickly debug the USB communication.

<sup>7</sup>The room UA5.214 in this case, which is one of the electronic lab at the ULB.

The "Test orientation" has been meant to test the detection of the orientation of the cellphone. The three angles necessary to characterize the orientation of the device are respectively the Azimuth, the Pitch and the Roll.

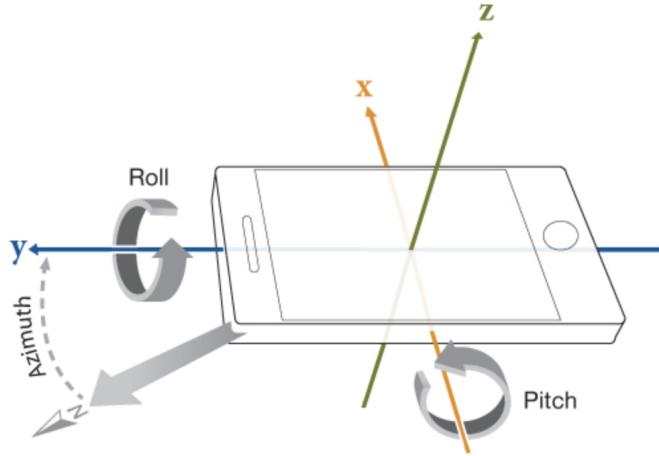


Figure 2.11: Representation of the Roll, Pitch and Azimuth orientation parameters.  
Taken from [20].

## Calibration

The "Calibration" has been meant to enhance the precision on the timestamps. Indeed, there is a difference between the moment when the packet is received at the antenna of the DWM1000 and the moment of its detection, which corresponds to the timestamp. The same phenomenon appears when the UWB transceiver transmits a packet. Those errors on the timestamps are called the transmit/receive antenna delay and must be configured to match the actual antenna delay. This delay originate from the difference in time between the actual reception of the message by the antenna and its actual recognition.

### 2.3.5 Precision reached

The precision obtained with this locating system depends on several factors, some of which being the location of the anchors in the room, the distance of the tag to those anchors, the clutter in the room, etc...

Nonetheless, a statistical study has been made to assess the performances of the locating system. The tag has been placed at several location that can be observed in Fig. 2.12.

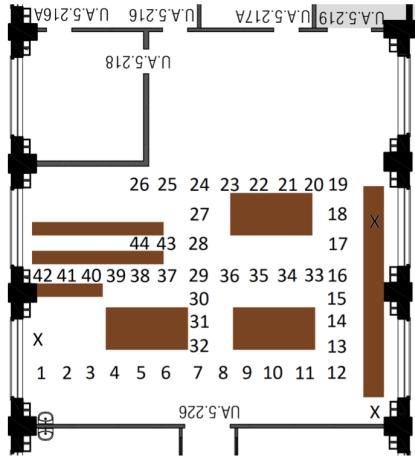


Figure 2.12: Locations corresponding to the measures represented in Fig. 2.13. Taken from [16].

For each location, the measurement was repeated a hundred times. The error remains below 45 cm for 80% of the measurements but reaches up to 85 cm in the worst case. Such deviations can be explained by many different factors, among which the non-trivial geometry of the room, the walls that are not parallel, the presence of windows, ... [16]. The extremity of the vertical segment represents the minimal and maximal errors for each location.

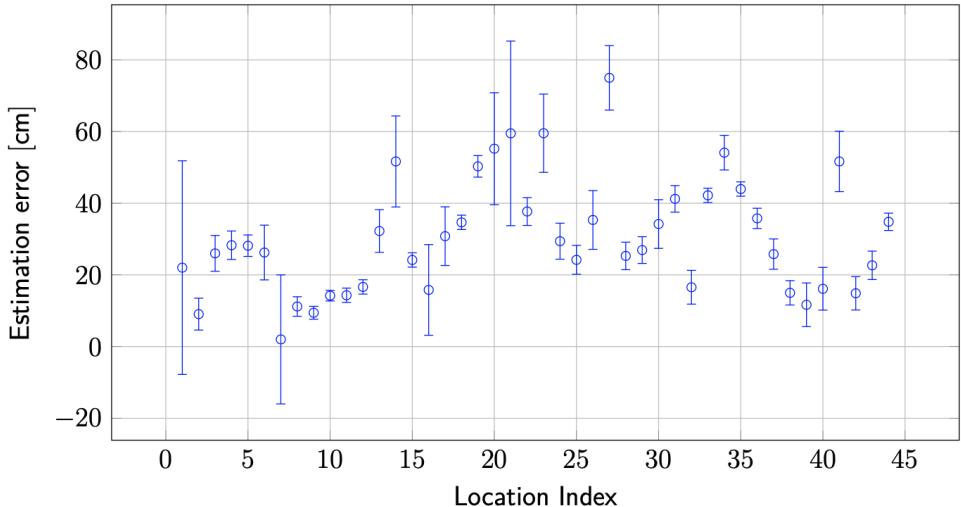


Figure 2.13: Error on the estimated location in function of the position of the tag. Taken from [16].

To achieve a better localization, anchors can be added. It has the advantage to add an equation to the system shown in eq. 2.5, which, if the new anchor gives some coherent measure, improves the estimation of the localization. The drawback is that it will slow down the actualization of the position, since the tag has to communicate with more anchors. Such an algorithm dealing with four anchors is to be seen in [15].

## 2.4 Multi-path aided locating system

This section introduces the crucial notion needed to perform the localization of a tag using a reduced number of anchors. First, the ray tracing is introduced, a method used to

simulate the paths taken by a wireless signal when transiting from the transmitter to the receiver. Based on this ray tracing, the Channel Impulse Response (CIR) will be detailed as well as the concept of virtual anchors. Finally, several locating system based on the MPC and CIR are briefly presented.

### 2.4.1 Ray Tracing

In order to achieve the ray tracing, one must be familiar with the method of mirror images. This method is used to enumerate and deals with the reflection that a wireless communication may occurs in a MPC. An example of this method is shown on Fig. 2.14.

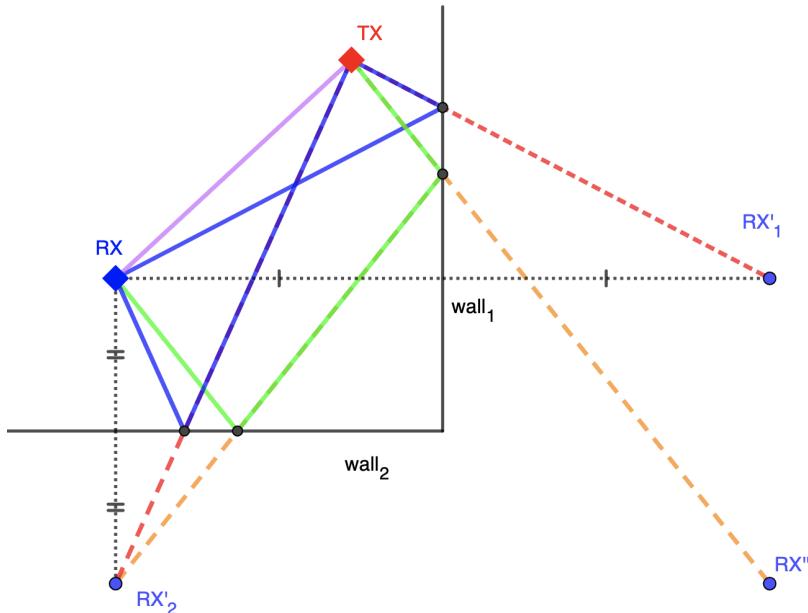


Figure 2.14: Method of mirror images - Example.

In this image, the purple ray is the direct (LoS), the blue rays are the one that incurs one reflection on a wall and the green one is the one incurring two reflections. This can be generalized to as much reflection as one needs. While the generation of the LoS is straightforward, in order to generate the ray that reflects on the wall<sub>1</sub>, the mirror image of the RX through this wall has to be generated. The wall<sub>1</sub> acting as the axial symmetry axis as it can be seen with the dotted line.

To compute the distance travelled following this path, one can compute the distance between TX and RX'<sub>1</sub>, which will be the same. The exact location where the ray will be reflected is the intersection between the current wall and the imaginary ray (red dashed line) between TX and RX'<sub>1</sub>. From this, the two blue segments can be traced, corresponding to the path followed by the electromagnetic wave. Knowing those information, the incidence angle on the wall can be known, which is an important parameter needed in the simulation presented in chapter 4 to compute the suffered attenuation due to the reflection. The same methodology is applied on wall<sub>2</sub> to obtain the reflection due to the second wall.

The double reflected ray presented in green follows the same methodology. From RX'<sub>1</sub> and RX'<sub>2</sub>, RX'' can be found. The total path travelled by the ray is still the distance between TX and RX''. The intersection between wall<sub>1</sub> and the imaginary ray (orange

dashed line) is considered as the origin of the ray going towards RX'₂. This technique can be generalized to  $n$ -different walls.

The ray tracing can be achieved by testing all the different possible combination of walls. This number of possible combination increases with the number of reflections required will the significance of multiple-times reflected peaks decreases in term of absolute amplitude value, mostly since the travelled distance statistically increases with the number of reflection encountered.

### 2.4.2 Channel Impulse Response

The CIR has already been mentioned in the section 2.1. The CIR is not only used in telecommunications systems but also in control theory for example, to characterize the behaviour of a system<sup>8</sup> [14]. As the name states, the goal of this CIR is to characterize the reaction of a system to a stimulus in the form of a pulse.

An example is shown on the Fig. 2.15. A ray tracing has been performed on the left image. The LoS rays and their first reflection can be observed. If a Dirac pulse is sent from the transmitter (TX) to the receiver (RX), since the propagation time will depend on the distance travelled, different peaks will appear, each corresponding to a different ray.

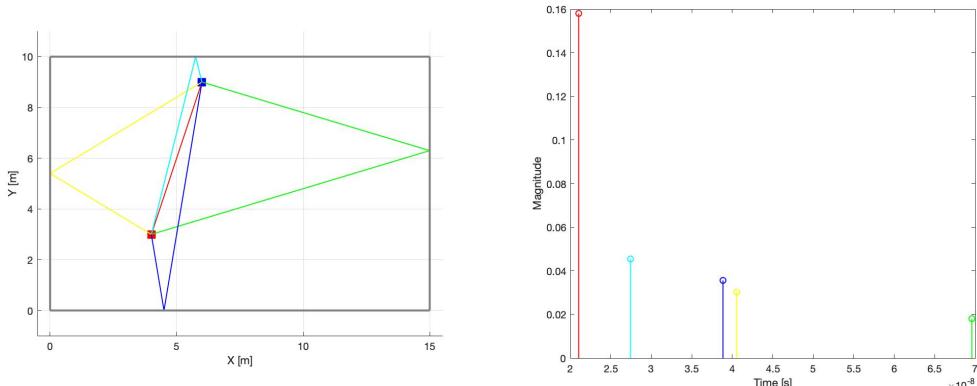


Figure 2.15: (Left) Direct and simple reflected rays between an anchor (4, 3) and a tag (6, 9). (Right) Theoretical CIR associated to each ray shown in the map.

As we can see, those different arrival times for each ray correspond to the different peaks on the right figure of the Fig. 2.15. The heights of the peaks depend on the attenuation due to the reflections on the walls. The equations used to compute those attenuations are detailed in the section *ref{SectionNotWrittenYet}*.

### 2.4.3 Virtual Anchors

The concept of Virtual Anchor (VA) has been introduced in [21]. While the anchors are some actual devices, the VA is not physically implemented. In order to create them, we need to know the location of an anchor in a room as well as the exact geometry of the room. In Fig. 2.16, the VAs have been created using the method of images. To find the location of the VA 1, the left walls act as the symmetry axis for an axial symmetry. From this, it is possible to extend this method to two reflections, two axial symmetries on two different walls would be needed in that case.

<sup>8</sup>In control theory, the transfer function which corresponds to the step response is commonly used.

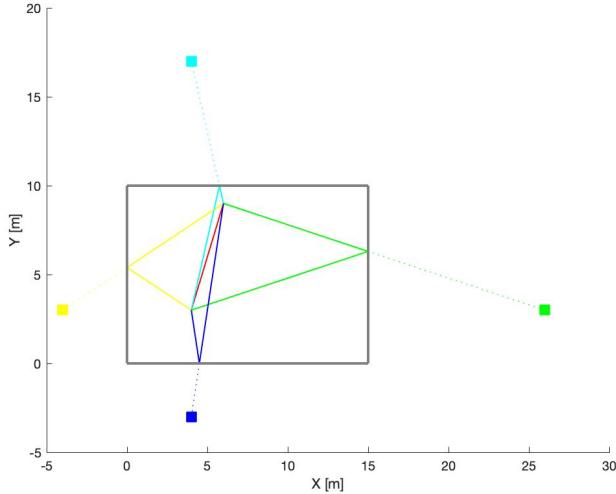


Figure 2.16: VAs associated with its corresponding ray

Using the theoretical CIR from Fig. 2.15, each peak which is associated to a reflected ray can be considered as being the LoS ray from a VA of Fig. 2.16. The colors have been kept for the sake of clarity. The advantages of the methods of images is that the travelled distance by each ray as well as the angular distribution is kept unchanged.

#### 2.4.4 Locating system

This is based on the concept of VAs and CIR: several methods are possible to find the localization of a tag in a room of known geometry using only one anchor.

In [22], the localization of a moving tag is suggested. This method uses the MPC detected as well as the history of the position of the tag which allows to perform a cross-correlation with the newly detected position, therefore maximizing the coherence of the behaviour.

In [13], the Cooperative Multipath-Assisted Indoor Navigation and Tracking (Co-MINT) uses other tags in the room as anchor to perform the location. Each tag, acting as a transceiver and a receiver, communicates with the other tags to compute its distance relatively to those tag. An algorithm is shown that combines those different information to achieve the localization.

In [17], a method providing the tag location without using a history of the previous locations is set. From the CIR, several peaks are extracted, each of this peak being associated to the ToF used in the multiple anchor locating system shown above. The difference being that the ToF extracted from the CIR can not be surely associated with a specific VA<sup>9</sup>.

Therefore, the system presented at eq. 2.5 needs to be solved multiple times in order to find the most suited combination of peak-antenna to locate the tag.

---

<sup>9</sup>An exception is done for the first peak, since the LoS that originates from the physical anchor is assumed to be the shortest path.

# Chapter 3

## Algorithms

This chapter deals with the two locating methods implemented and investigated in the simulation. In order to perform the localization, as seen in 2.4, the CIR obtained needs to be treated in order to extract some useful data. The first section shows such methods. Then, using the extracted peaks, two different localization algorithms are shown, a hard based on the trilateration and a soft based on the comparison of theoretical and realistic CIR.

### 3.1 Peaks extraction

A CIR as shown in 2.15 unfortunately does not exist in the real world, at least, not using the material shown in section 2.3. Firstly, extra peaks will appear, because of the double, triple, etc... reflections on the walls and furniture of the room. Some peaks caused by diffraction may also appear. People walking or standing into the room will also modify the channel and so the CIR. Secondly, as previously stated in 2.3.1, the bandwidth of the DWM1000 is not infinite, occasioning a spatial extension of the different peaks. The evolution from a theoretical "perfect" case to a "real" can be observed in Fig. 3.1.

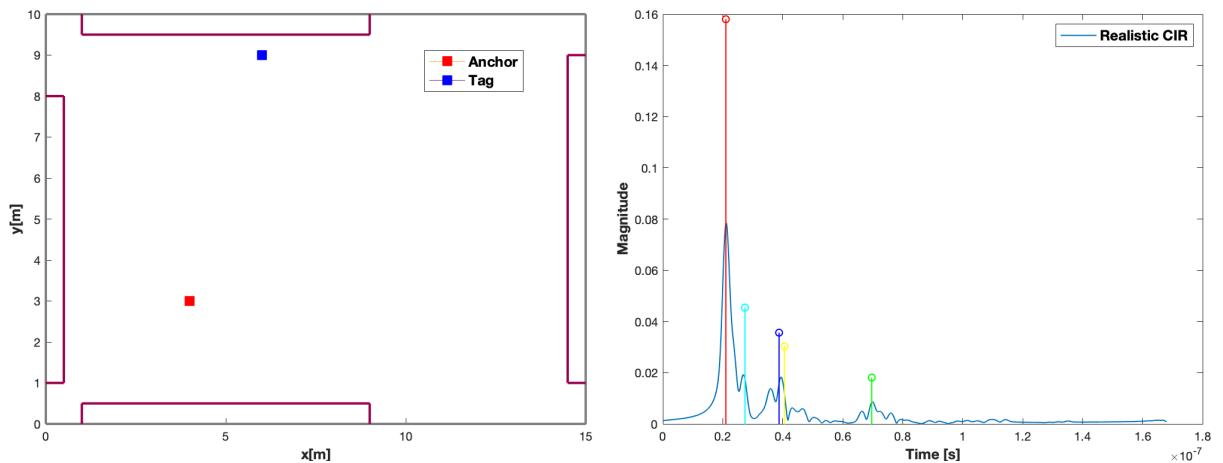


Figure 3.1: (Left) Room used to generate the realistic CIR using up to three reflected rays. (Right) Superposition of the CIR in Fig. 2.3 and the realistic CIR.

The simulation used to generate the realistic CIR is described in chapter 4. As we can observe, the peaks that originates from the theoretical case matches peaks in the realistic. But can also observe that some new peaks arise in the realistic case, such as those around  $0.4 * 10^{-7}s$ .

The goal of the peak extraction function is, as the name states, to extract the peaks matching the needs of the locating algorithm. As it will be seen in section 3.2, 3.3, the two methods do not reach their purpose in the same way, the optimal peaks will be slightly different. Hence the two different peaks extraction methods presented below, a soft and a hard.

### 3.1.1 Soft case

For the soft locating system, the algorithm of the peak extraction is detailed in algo. 1. The algorithm takes as input a vector made of tuples, respectively the time and the amplitude of each point of the CIR. The  $n$  parameter states the number of peaks that the algorithm needs to extract.

---

#### Algorithm 1: Peaks Extraction - Soft case

---

**Inputs:**

$CIR = \{cir\} = \{(t_i, A_i) \text{ s.t. } i \in \mathbb{N}_0\} \in \mathbb{R} \times \mathbb{R}, n \in \mathbb{N}_0 ;$

**Initialize:**

```

Peaks ← {(ti, Ai) s.t. i ∈ {1, ..., n}};
ratio ← 100;
i ← 1;
CIRmax ← local_max(CIR);
CIRmax ← Order(CIRmax, 'Descend', 'Prominence based');
for {cir} in CIRmax do
    if prom({cir}) > max(CIR)/ratio then
        Peaks[i] ← {cir};
        i ← i + 1;
        if i > n then
            break;
```

**Output:**

Peaks;

---

During the initialization phase, first, the output **Peaks** is defined, this variable will be used to store the different extracted peaks. The variable **i** is the counter of the number of peaks found and stored into the **Peaks** vector while **ratio** is used to set a limit to the relative size of the peaks being taken<sup>1</sup>. The local maximum of the **CIR** is then saved in **CIR<sub>max</sub>** and ordered in the decreasing order based on the prominence of each peak. The definition of the prominence is reminded in Fig. 3.2. Since the prominence depends on the local minima between two peaks, a higher amplitude does not always imply a higher prominence.

---

<sup>1</sup>This value of 100 is arbitrary, it avoids the program to take really small peaks produced by the noise.

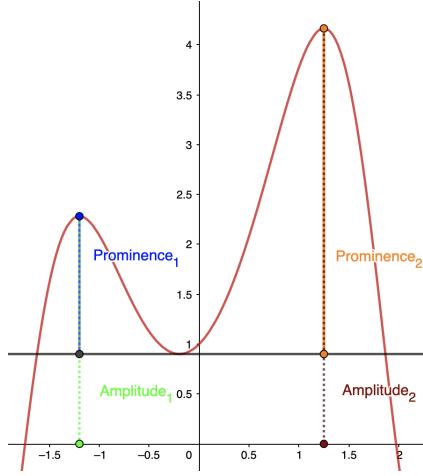


Figure 3.2: Difference between the prominence of a peak and its amplitude

The main loop then gets the  $n$  first values of  $CIR_{max}$  and store it into **Peaks**. The If condition is, as said before, used to remove the small fluctuation in the CIR because of the background, only keeping the most important peaks. If there is less than  $n$  peaks that fulfil the conditions, the algorithm returns as many peaks as possible while still respecting the pre-established conditions.

### 3.1.2 Hard case

For the hard locating system, the same methodology is used. Algorithm 2 slightly changes at several places. Firstly, the input is only CIR, the algorithm always gives back three peaks in **Peaks**. Secondly, the ordering of  $CIR_{max}$  is made based on the amplitude of the peaks. Finally, the amplitude, not the prominence, of the  $\{cir\}$  is compared with  $\max(CIR)/ratio$ .

---

#### Algorithm 2: Peaks Extraction - Hard case

---

**Inputs:**

$$CIR = \{cir\} = \{(t_i, A_i) \text{ s.t. } i \in \mathbb{N}_0\} \in \mathbb{R} \times \mathbb{R};$$

**Initialize:**

```

Peaks ← {(ti, Ai) s.t. i ∈ {1, 2, 3}};
ratio ← 100;
i ← 1;
CIRmax ← local_max(CIR);
CIRmax ← Order(CIRmax, 'Descend', 'Amplitude based');

```

**for** {cir} in  $CIR_{max}$  **do**

```

    if amp({cir}) > max(CIR)/ratio then
        Peaks[i] ← {cir};
        i ← i + 1;
        if i > 3 then
            break;

```

**Output:**

Peaks;

---

## 3.2 Hard localization algorithm

This locating system is based on the idea of trilateration and tries to imitate it. Using three peaks, it tries to match those with the anchor and two VAs to find an intersection point as in the Fig. 2.5. Those three peaks are extracted with the algorithm 2 from the received CIR at the tag. First, the number of systems that need to be solved is shown, then a systematic method to solve them is suggested.

### 3.2.1 Virtual antennas combination

The hypothesis that the greatest peak, which is usually the first one, corresponds to that from the LoS ray is made, this implies that there always exists a LoS in those rooms, meaning that this does not suffer from attenuation. About the second and third peak, since each can not be surely associated with a VA, the only available solution is to try every combination of VAs. The order being important<sup>2</sup>, there are  $P_4^2 = \frac{4!}{2!} = 12$  different systems to solve. This computation has been made for a room with a simple rectangular geometry. Of course, with a more complex geometry, the number of possible combinations would increase.

On Fig. 3.3, a possible problem is shown. The CIR shown on the left side is obtained by computing only the LoS and the first reflections onto the different walls. In theory, five different peaks should be seen, but only four appear. In this particular case, the second peak is formed by the peaks from the blue VA and the green VA, which will be undistinguishable.

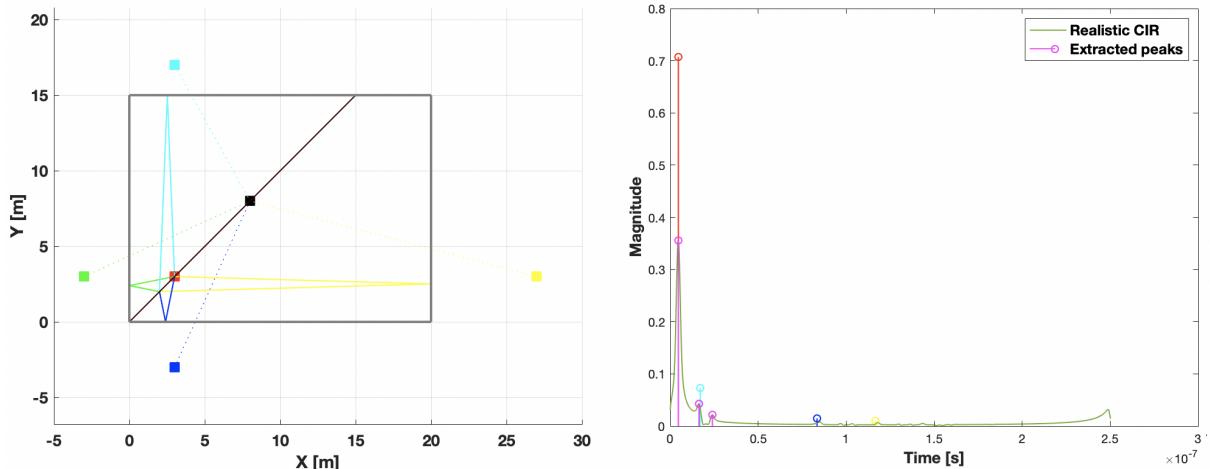


Figure 3.3: (Left) Room with the anchor in red and the tag in black. (Right) CIR corresponding to the room in the left image, the color of the peaks match the color of the associated VAs. The magenta peaks corresponds to those extracted using the algo. 2 from the realistic CIR in green.

A problem that may arise in such situation is that the peaks taken from the CIR could not correspond to some actual theoretical peaks<sup>3</sup>. Such an example can be seen on the right image in Fig. 3.3 where the third magenta peak does not correspond to any theoretical peak. To overcome this problem, the proposed solution is to consider the peaks as mingled. Hence to the twelve possible combinations discussed , we would have

<sup>2</sup>Associating the tuple  $(d_1, d_2)$  to  $(va_1, va_2)$  is not equivalent to associate it to  $(va_2, va_1)$ .

<sup>3</sup>The one that corresponds to one reflection.

to consider that the second magenta peak originates from two different VAs, as the cyan peak on the right image of Fig 3.3.

This method has the disadvantage of requiring much computation since twelve more peaks - VAs need to be checked, but it will solve some symmetry-related problems. This kind of problem can mostly be occurring when the tag is equidistant from two VAs<sup>4</sup>. Since this peak combination due to the tag being equidistant to two VAs only occurs some specific cases, we should first check the original antenna combination before checking those added.

Another source of trouble for this algorithm is the finite bandwidth of the antennas. As it can be seen on Fig. 3.4, peaks that can be distinguished on the theoretical CIR are mingled in the realistic. This phenomenon can be observed with the blue and the cyan peaks for example, which are so close that there are considered as one in the finite band CIR.

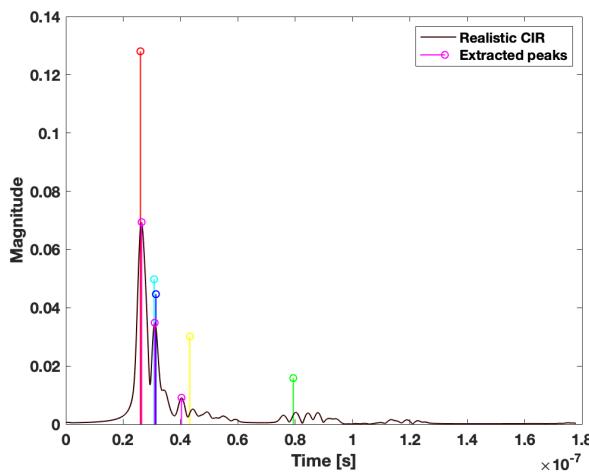


Figure 3.4: Infinite bandwidth CIR generated and the finite bandwidth associated in brown. The extracted peaks are shown in magenta.

To deal with this peak merging issue, the solution proposed is the same that deals with peaks occurring from two VAs equidistant from the tag. The precision achieved on the localization in such case would be reduced in those cases. Such examples will be shown in chapter 4. Of course, the third detected peak could also originate from a VA after a simple reflection, this discussion only make sense when it does not.

### 3.2.2 System solver

Those systems are solved one at a time, starting with the twelve 'basic', going on with the particular cases shown above if no suitable solution has been found. Since there is almost no chance that the system leads into a perfect one point solution, the system 2.4 can not simply be resolved. The three equations are solved two by two, giving six real or complex solutions. If the six computed intersection belongs to the Real, this would correspond to  $U_{12}, U_{13}, U_{23}$  and the Tag<sup>5</sup> in Fig. 2.5. From this point, the algorithm first excludes the solutions lying out of the room, the solutions having a too big imaginary

<sup>4</sup>As on the brown line on the map, being the axial symmetry between the two bottoms and left VAs.

<sup>5</sup>The position of the tag should appears 3 times, one for each combination of anchors.

part are also discarded<sup>6</sup>.

Using the remaining solutions, the algorithm needs to check if those can be combined to find a suitable solution. A solution is considered as suitable if the algorithm can find three solutions being relatively close to each other. This notion of "relatively close" is subjective and is one of the parameter that we can use to tune the algorithm. If no solution is found using all the combinations, then no location is provided, hence the qualification of "hard".

### 3.3 Soft localization algorithm

The locating system shown in this section reaches localization by comparing some theoretical CIR, computed by knowing the geometry of the room with the CIR recovered at the tag. In order to make this comparison, the room is sampled, based on the wished precision we would like to reach. In this thesis, the sample will be made using square of one meter side.

For each peak extracted from the CIR, two informations are available, the time of arrival of this peak and its amplitude. We could use the amplitude of the peaks, this would probably work in a lot of cases in the simulation, but this approach has a major drawback: while the peaks are more likely arrive at the same time in both cases, their amplitude can not be guaranteed.

For example, if a peak suffers from too much attenuation in the real case compared with the simulation, because of losses coming from the transmission, due to reflections on the different materials which are not perfectly simulated, due to a not perfectly isotropic emission of the antenna in the room plane, etc. All those reasons may vary the amplitude of each peak of the CIR, meaning that nothing can ensure that the proportionality between the LoS peak and the other peaks would be conserved as in the simulation.

For those reasons, the comparison will be based on the time-of-arrival of each peak. This method also has some drawbacks, the transition to the finite band does not ensure that the peaks will remain at their exact locations. When two peaks mingle, the new peak remaining in finite band are more likely to have an arrival in the middle of those two peaks. Some precautions have been taken to minimize the errors induced by this phenomena.

#### 3.3.1 Space solution reduction

Since the algorithm needs to compare the theoretical CIR with every possible position of the tag, in order to speed-up the program, we could actually reduce the number of locations to test.

Using the SDS-TWR, the ToF of the signal between the tag and the anchor can be computed<sup>7</sup>. Based on this ToF, a circle drawn be traced with the center on this anchor and the radius being the estimated distance deduced from the ToF. In theory, the tag is supposed to be located on this circle, but because of the discretization and errors on the ToF, a margin is taken to get the set of possible locations. This margin lies in the between of the two orange circles, that can be observed on Fig. 3.5.

---

<sup>6</sup>The notion of "too big" is completely subjective. It appears that when two circles are close to have an intersection, the imaginary part is smaller. **NEED A PROOF**

<sup>7</sup>In the simulation, it will be assumed to be extracted from the CIR

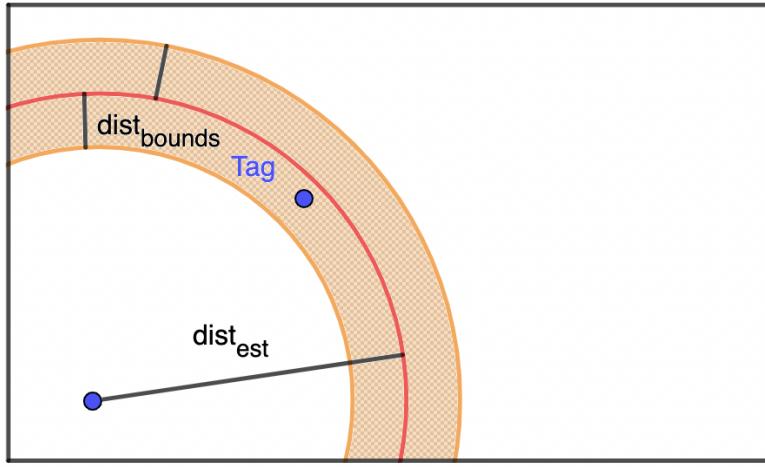


Figure 3.5: Refaire cette image

From the position inside those boundaries, a mask matrix representing all the positions of the room is filled with ones for the position inside of the orange zone. The other positions are left to zero. Later, this mask is used to reduce the computations since only the values associated with a one will be tested.

This will give leads to a matrix, whose size depends on the chosen discretization of the room, filled with one and zero. The meaning of a one in this matrix means that the location associated to this entry of the matrix has been judged as a probable location for the tag. Therefore, the next step of this algorithm is performed only at those location. This results in a map as displayed in Fig. 3.7 where only an arc has been evaluated, not the wall map.

### 3.3.2 Time MSE

For each remaining possible location, a finite band CIR is simulated using only the direct and the single reflected rays. Such example can be found in Fig. 3.6. In order to obtain something similar to the realistic case, the CIR has been passed in the finite band. The peaks are then extracted using algo. 1 from this finite band CIR.

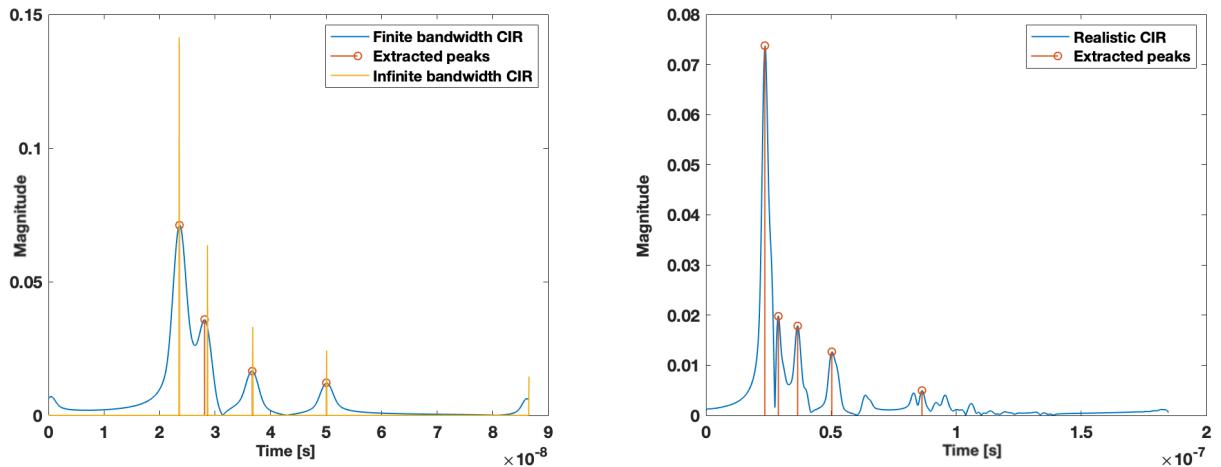


Figure 3.6: (Left) Theoretical CIR obtained using the simulation. The same room has been used as for the right image, without furniture and using only the direct ray and single reflected rays. (Right) Realistic CIR obtained in a room full of furniture using the simulation.

The extraction is also done once on the CIR. By use of the different peaks extracted, the following problem is solved.

$$S(\vec{p}_0) = \sum_i^N (\|\text{Peaks}(\vec{p}_r)_i - \text{Peaks}(\vec{p}_0)_i\|^2)$$

$$\vec{p}_0 = \underset{\vec{p}}{\operatorname{argmin}} S(\vec{p}) \quad (3.1)$$

The peaks extracted are sorted in the chronological order in the peak extraction. Therefore, the sum goes through the peaks of both CIR. The argument N is limited by the smallest list of peaks given by the algo. 1. While the  $\vec{p}_0$  represents all the possible positions tested,  $\vec{p}_r$  represents the real cases.  $\text{Peaks}(\vec{p}_0)_i$  represents the  $i^{th}$  peak extracted from the theoretical CIR computed by placing the tag at the position  $\vec{p}_0$ .

This algorithm will have as output a map of the room with the Mean Square Error (MSE) of the time arrival of the peaks computed at each possible location. An example of such output can be seen in Fig. 3.7, this heat-map has been generated using the MSE computed at each location evaluated. The estimator from eq. 3.1 will provide only one output, being the one with the lowest MSE. A solution will always be provided by this locating system.

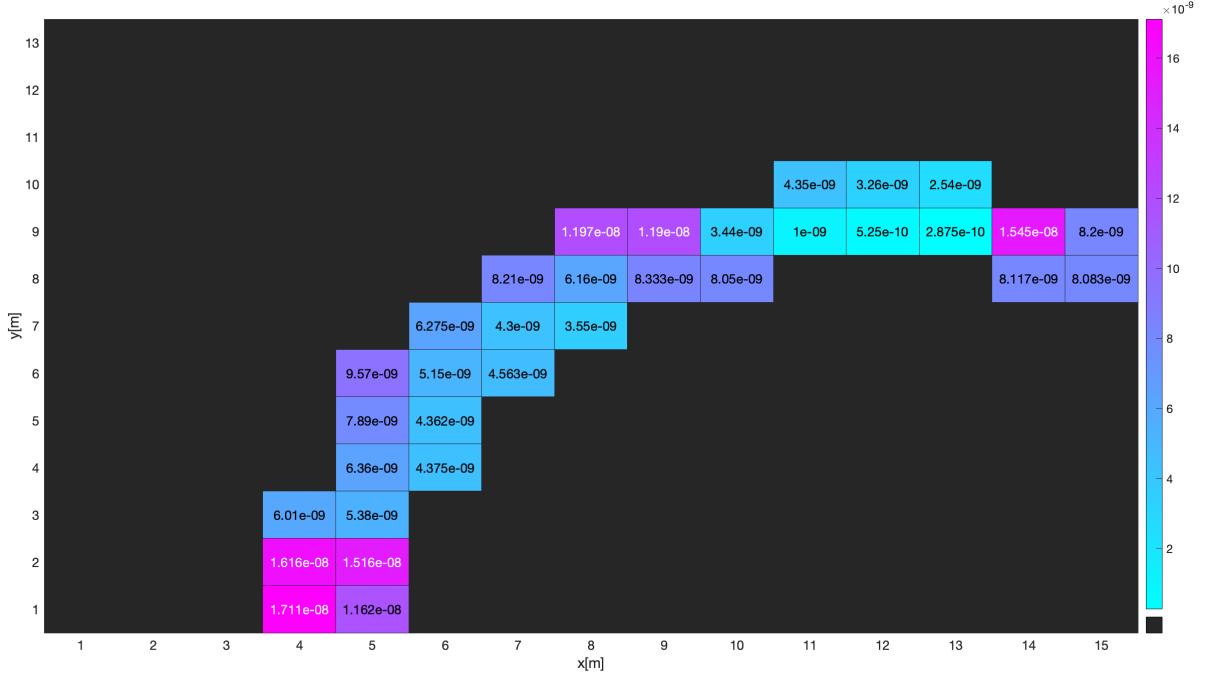


Figure 3.7: Example of an execution of the soft localization algorithm. Anchor at (12,2) and Tag at (13,9).

The locations that were not estimated because of the space solution reduction occurring before the computation of the MSE are colored in black.

# Chapter 4

## Simulations

In order to test and validate the different algorithms proposed in chapter 3, a simulation was implemented. The advantages of such simulations is to get rid of the technical constraints of electronic materials, no configuration of the antennas was needed, no need to implement features in the application to deal with the CIR, no need to modify the state machines presented in appendix B, etc.

However, this simulation needs to be as true as possible compared to the reality. This implementation is detailed in the first section of this chapter. Then, the results obtained using this simulation are presented in the following sections for the locating methods presented in 3.

### 4.1 Simulation

The simulation has been created by adapting a code previously developed for the project of the course of Communication Channel given at the ULB [5]. This simulation is based on the ray tracing, a rendering technique used to compute the trajectory of electromagnetic waves presented in section 2.4.1. Two important features of this program can be separated. The first objective being to simulate a realistic CIR similar to the one produced by the experimental set-up. The second objective is to perform the localization based on this realistic CIR.

#### 4.1.1 Realistic CIR generation

To achieve this objective, first the desired environment has to be simulated in the program. To make it happen, the `room_configuration.m` file has been created. Configuration of the geometry of the room, the type of walls used, the number of different furniture and their location in the room can also be chosen. This file is also used to chose the location of the anchor and the tag in the room. The left figure of the Fig. 3.1 has been produced using this configuration file.

In order to generate a finite-band CIR, the equivalent infinite-band CIR is first generated. This can be done for a direct ray, simple reflected ray, double reflected ray and triple reflected ray the transmission across the different materials is also taken into account. The ray reflected on the ground is also computed but not represented for the sake of clarity. An example of all the other ray can be found in Fig. 4.1. On this figure, furnitures are displayed against the walls, the direct ray is displayed in red, a simple reflected ray on one of the furniture in blue, a double reflected ray in cyan and a triple reflection ray is displayed in green.

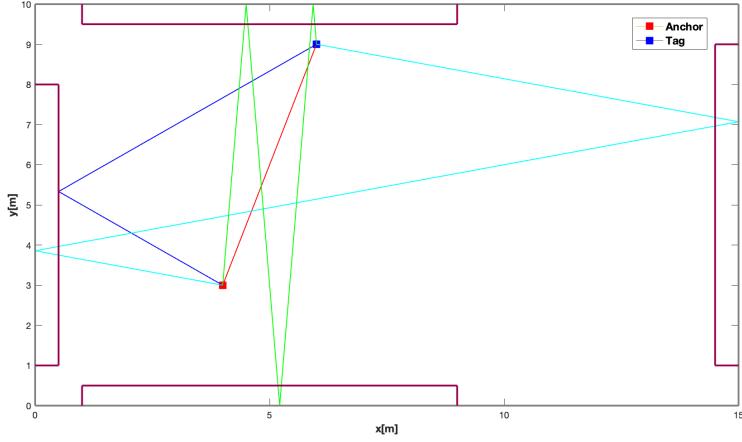


Figure 4.1: Ray Tracing Simulation - Example.

Stating infinite is not completely correct, the real bandwidth being the one of the carrier frequency in this case, however, it presents the characteristics in regard to actual bandwidth used. The different peaks well isolated and punctual. The so-called infinite-band CIR is converted to the frequency domain using the Fourier transform. The built-in Fast Fourier Transform (FFT) function from MATLAB has been used [20]. The frequency spectrum has been reduced from the carrier frequency to 500 MHz, an approximation of the bandwidth used as detailed in section 2.3.1. Applying the Inverse Fast Fourier Transform (IFFT) to return into the time domain results in the red curve on Fig. 4.2.



Figure 4.2: Trouver un nom

As one can observe, this CIR is far from ideal, mostly due to the side lobes appearing next to every peaks. This is a side effect due to the reduction of the bandwidth. In order to counter this effect, a Chebyshev window has been used [19], [20]. The convolution of the finite-band time domain CIR will reduce those side lobes. The orange curve from Fig. 4.2 has been produced using the chebyshev function from MATLAB. The parameter of the Chebyshev window has been set heuristically to 25. A systematic method would have been to use an experimental and a simulated CIR that would correspond to the same situation<sup>1</sup>. The optimal parameter would then be computed using the lsqcurvefit function from MATLAB.

### 4.1.2 Code possible outputs ?

The second objective of the simulation is the locating process. As detailed in the chapter 3, two different methods have been implemented in this scope. In order to compare both methods, JE NE SAIS PAS TROP QUOI DIRE BLABLABLABLA....

---

<sup>1</sup>The same room, antennas configuration, antennas position, ...

## 4.2 Hard Simulation

## 4.3 Soft Simulation

# Chapter 5

## Experiences

This chapter **A finir**

### 5.1 Experimental set-up configuration

#### 5.1.1 DWM1000

One of the feature of the DWM1000 is to store the CIR received if required. To perform such action, the documentation provided by Decawave [8] was used in order to determine which bits were to be activated. It appears that several bits needed to be set in certains configuration, all in the 0x36 : Power Management and System Control (PMSC) register, sub-register 0x00 which is a 32 bits control register. The control register is shown in Fig. 5.1.

- The bits 3,2 : Receiver Clock Selection (RXCKLS) need to be set to 10 to allow the host system to reach the CIR.
- The bit 6 : Force Accumulator Clock Enable (FACE) needs to be set to 1 for the host system to read the accumulator data<sup>1</sup>.
- The bit 15 : Accumulator Memory Clock Enable (AMCE) needs to set to 1 for the same reason as bit 6.

REG:36:00 – PMSC_CTRL0 – PMSC Control Register 0																															
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
SOFTRESET	-	-	-	-	-	KHZCLKEN	-	-	-	GPDNR	GPDCE	GPRN	GPCE	AMCE	-	-	-	-	ADCCE	-	-	-	FACE	TXCLKS	RXCLKS	SYSCLKS					
1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	

Figure 5.1: Register 0x36 - PMSC, sub-register 0x00. Taken from [8].

Those bits being activated, one may now recover the CIR from the DWM1000. Still following the user manual, it appears that the CIR is stored in the 0x25 : Accumulator CIR memory register. This register contains complex values, 16 imaginary bits and 16 real bits for each tap, 992 of them being registered when the PRF is set to 16 MHz. Each tap registered corresponds

<sup>1</sup>Where the CIR is stored.

almost to a frame of 1 ns, which corresponds exactly to a sampling frequency being two times 499.2 MHz, meaning that the whole CIR registered corresponds to almost a  $\mu$ s.

Since the algorithms developed in chapter 3 mostly work based on the direct ray and the simple reflections, one may question the usefulness of extracting the whole CIR stored in the accumulator. Indeed, in a room of dimensions (15x20)m for example, in the longest first reflection that would occurs, the distance traveled would be of  $\sqrt{(20^2+15^2)} = 42.72m$ , which corresponds to a distance time of  $1.42 * 10^{-7}s$ . Which is less than a fifth of the whole CIR stored in the accumulator.

In order to extract partially the CIR, the position of the peak needs to be known. This information is stored in the 0x15 : Receive Time Stamp (RX\\_TIME) register, in the sub-register 0x05 : First Path Index (FP\\_INDEX) . A 16 bits long value representing the index value corresponding to leading edge of the CIR stored, only the 10 Most Significant Bits (MSBs) represent the integer part of this index. By taking a window around this peak, the number of bits to transmit from the DWM1000 to the PSoC can be reduced.

The Fig. 5.2 shows two experimental CIR obtained, one by extracting the whole data set while the other only extracted part of it using the value stored in the FP\\_INDEX sub-register.

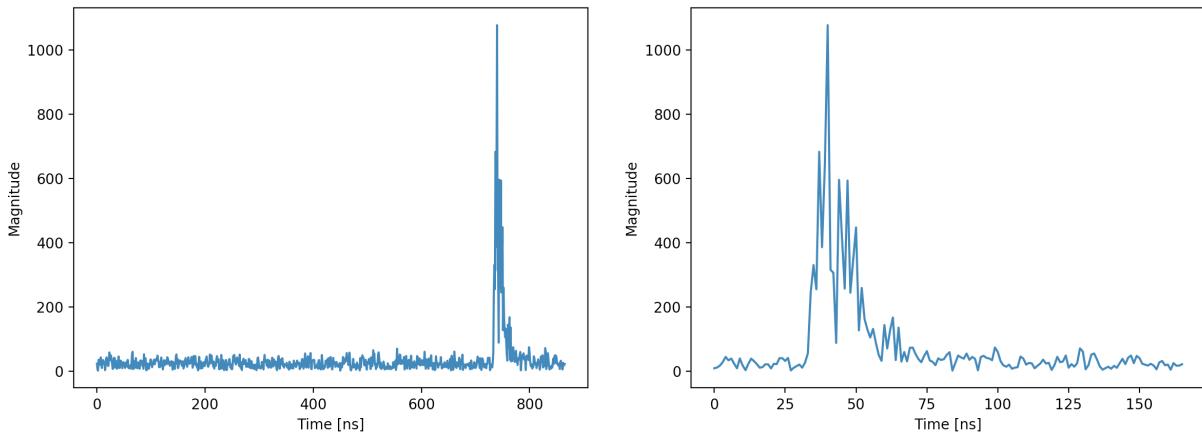


Figure 5.2: Whole CIR extracted from the DWM1000 (Left) vs. Partial CIR extracted from the DWM1000 (Right).

### 5.1.2 PSoC

The PSoC controls the DWM1000 on the tag side. Therefore, it needs to handle the CIR recuperation from register 0x25. In order to avoid a too big modification of the interactions between the tag and the anchors, it was decided to graft this recuperation part to the existing code taking care of the SDS-TWR. At the end of those exchanges, the PSoC checks that the needed bits activating the CIR recuperation are set, gets the FP\\_INDEX and partially extract the CIR around this value. In order to extract the whole CIR, 25 values before and 175 values behind where taken. This corresponds to an extracted CIR which is  $2*10^{-7}$  s long. This choice is arbitrary and may be tuned in function of the room where the localization is performed.

The extraction is made through the SPI port between the DWM1000 and the PSoC, the number of bytes that needs to be extracted is 4 times the number of tap required. This number of bytes that can be transmitted through this bus at once is limited to 256 in this case. The CIR needs to be divided in several part for one to retrieve it all. While extracting those data, one have to be really careful because every time the accumulator

memory is accessed, the first octet output should be discarded. Indeed, it is a dummy one due to memory access. [8].

### 5.1.3 Android Application

Since the Android Application is the master of the PSoC, it also has to be modified. Three different features were added : two activities were added and the Navigation activity was modified to achieve the CIR recuperation.

The objective of the first activity is to validate that the application can access the memory of the smartphone or an external memory in order to store any data on it. In this specific case, the activity takes a string and writes it in a chosen file in the memory. The final objective being to save the extracted CIR in the memory to analyse it later. In order to access the memory, the application needs to ask some special permission to the Operating System (OS). This specific android permission is called: WRITE\_EXTERNAL\_STORAGE, it will create a pop-up the first time the application will try to access the memory, asking the user if the application can access it.

The second activity is made to validate the recuperation and storage of a CIR from the DWM1000 to the memory. This activity actually mimics the Test USB activity, a byte code is sent to the PSoC that triggers the recuperation function of the CIR. This CIR then needs to be transmitted to the application through the USB connection already implemented. As for the PSoC, the CIR is separated in several sections to be fully transmitted through the USB connection.

The third feature actually combines the two new activities and the navigation one. Once again, in order to modify as less as possible the whole localization process, the CIR recuperation has been grafted. Meaning that in addition to the ToF already provided by the PSoC for each anchor, the PSoC will also provide the CIR of each anchor. This whole set of data will be saved in a txt file which is named in function of the time of execution of the program in the following format : yyyy-MM-dd-hh-mm-ss.txt i.e. 2020-05-30-17-05-32.txt.

In order to further analyze the data recuperated, the estimated position of the tag computed by the navigation activity is saved as well. The data structure is shown in Fig. 5.3.

```
ANCHOR_1 : 0x00, 0x11, 0x00, 0x05, 0x00, 0x22, 0x00, 0xff, 0xff, ..., 0x16, 0x00, 0x35, 0x00, 0x03,
0x00, 0xe, 0x00, 0xf8, 0xff, 0xf3, 0xff, 0xeb, 0xff, 0x05, 0x00, 0x1d, 0x00, 0xff, 0xff, 0xf3;
ANCHOR_2 : 0xff, 0x00, 0x00, 0x08, 0x00, 0xfd, 0xff, 0x06, 0x00, ..., 0xd, 0x00, 0xfa, 0xff, 0x14,
0x00, 0x06, 0x00, 0x14, 0x00, 0x1f, 0x00, 0x0c, 0x00, 0x17, 0x00, 0xfb, 0xff, 0x14, 0x00, 0xf8;
ANCHOR_3 : 0xff, 0xf3, 0xff, 0xeb, 0xff, 0x05, 0x00, 0x1d, 0x00, ..., 0xff, 0xff, 0x26, 0x00, 0xf3,
0xff, 0x10, 0x00, 0xf5, 0xff, 0x24, 0x00, 0x22, 0x00, 0xfe, 0xff, 0x11, 0x00, 0x18, 0x00, 0xf;
POSITION : X937.34, Y274.23;
```

Figure 5.3: Data structure example of a measurement in navigation.

For each anchor, the CIR recuperated, which is 800 octets long<sup>2</sup>, has been shortened using the '...' to be more visible. As one can see, each CIR is explicitly associated with an anchor and the position is stored in cm in plain text as well.

---

<sup>2</sup>800 octets long corresponds to 200 values of the CIR as 1 tap is made using 4 octets, 2 real and 2 imaginary.

## 5.2 Test Campaign

The objective behind the test campaign is to validate the simulations previously detailed, testing in a realistic case the locating algorithm using only one anchor. To achieve this objective, an experience was planned. Due to the exceptional situation linked to the COVID-19 and the lock-down, as stated in the introduction, this part is not as complete as expected. The experience procedure is still detailed in this section for information.

In order to perform the needed measurement, a relatively empty<sup>3</sup> room would have been chosen. The three anchors would have been displayed in the room, in three different corner. The exact position of the anchor being hard-coded in the android application to allow the algorithm to perform the trilateration.

Concerning the data recuperation, the objective would have been slowly walk inside of the room during several minutes to extract as much CIR as possible at different localization. Walking slowly would ensure that the three CIR would be extracted at the same location for each anchor, assuring an accurate estimation of the position and a good match between this localization and the CIR extracted.

---

<sup>3</sup>With as less furniture as possible. To reduce the cluttering and the number of MPC.

# Chapter 6

## Conclusion

# Appendix A

## Application view

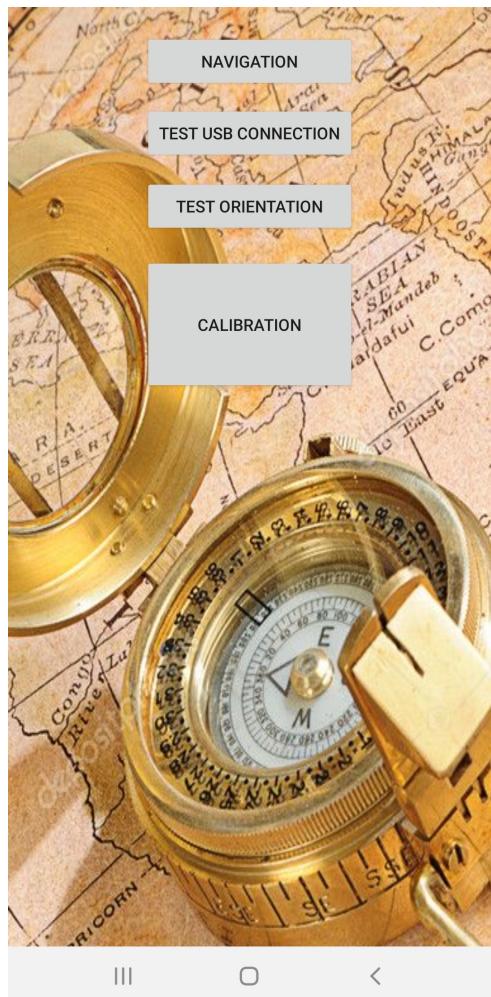


Figure A.1: Main window of the Android application.

# Appendix B

## State Machines

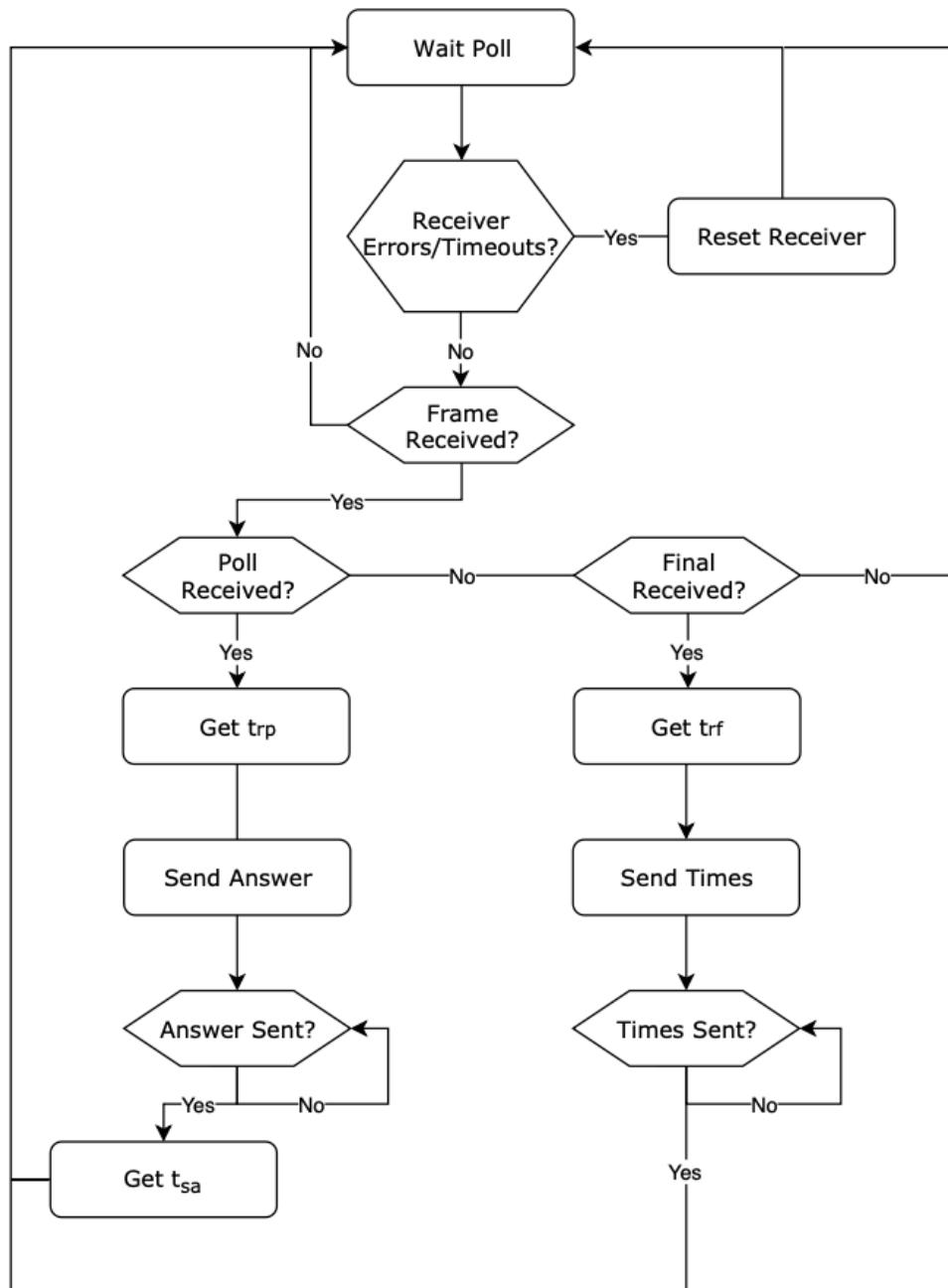


Figure B.1: State machine of the Anchor. Taken from [16].

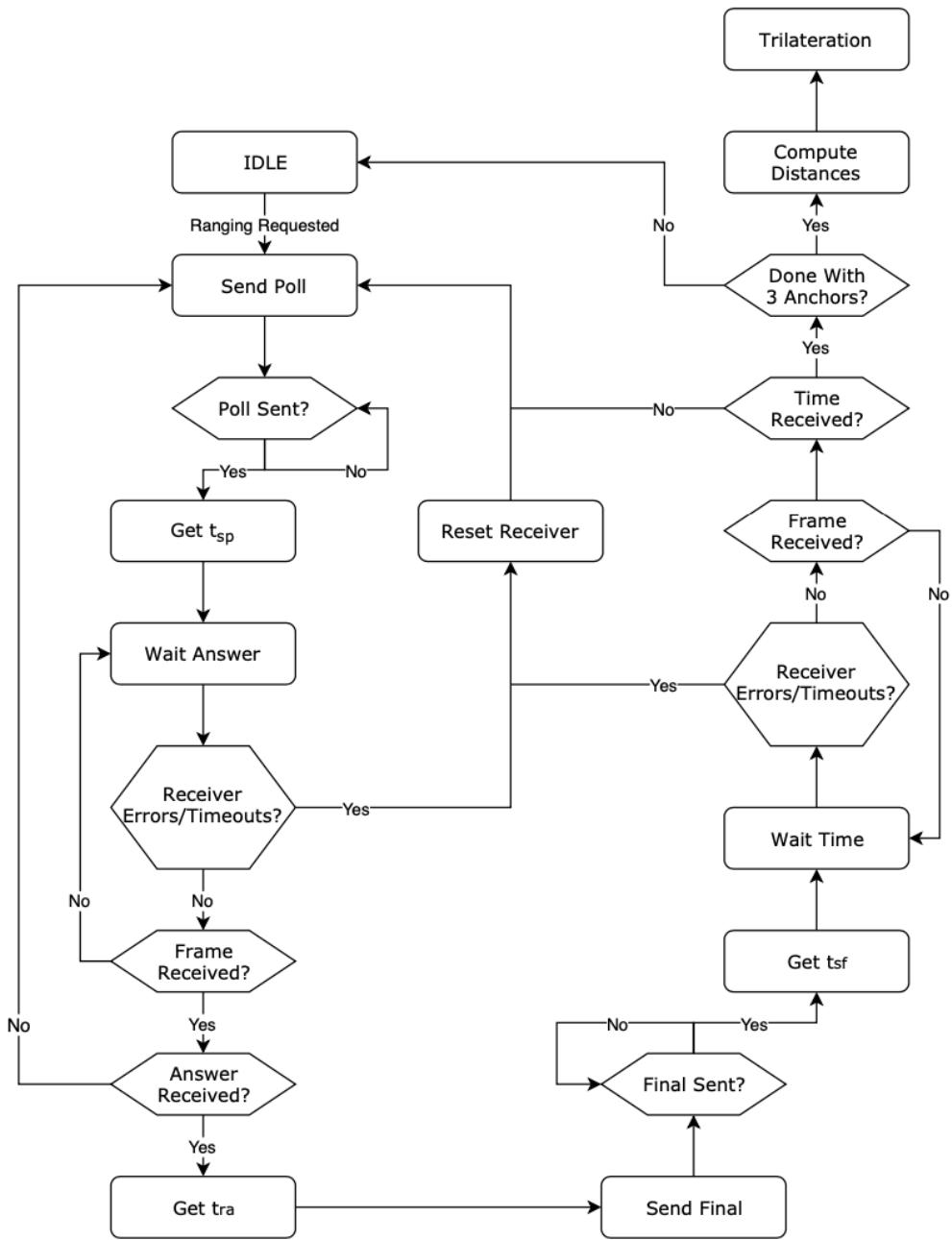


Figure B.2: State machine of the Tag. Taken from [16]

# Acronyms

**AMCE** Accumulator Memory Clock Enable 26

**CIR** Channel Impulse Response 13–28

**Co-MINT** Cooperative Multipath-Assisted Indoor Navigation and Tracking 15

**EU** European Union 7

**FACE** Force Accumulator Clock Enable 26

**FFT** Fast Fourier Transform 25

**FP \_ INDEX** First Path Index 27

**GPS** Global Positioning System 1

**IFFT** Inverse Fast Fourier Transform 25

**ITU-R** International Telecommunication Union Radiocommunication Sector 3

**LoS** Line of Sight 1, 4, 13–15, 19, 21

**MISO** Master Input, Slave Output 8

**MOSI** Master Output, Slave Input 8

**MPC** Multipath Channel 4, 13, 15

**MSB** Most Significant Bit 27

**MSE** Mean Square Error 23

**NLoS** Non Line of Sight 4

**OS** Operating System 28

**PMSC** Power Management and System Control 26

**PRF** Pulse Repetition Frequency 7, 8, 26

**RDEV** Ranging-capable DEVice 5

**RTLS** Real-time locating systems 2–4

**RX \_ TIME** Receive Time Stamp 27

**RXCKLS** Receiver Clock Selection 26

**SCKL** Serial Clock 8

**SDS-TWR** Symmetric double-sided two-way ranging 5, 6, 8, 10, 21, 27

**SPI** Serial Protocol Interface 7–9, 27

**SS** Slave Select 8

**ToA** Time of Arrival 5

**ToE** Time of Emission 5

**ToF** Time of Flight 5, 6, 8, 15, 21, 28

**UWB** Ultra-Wideband 1–4, 7, 11

**VA** Virtual Anchor 14, 15, 19, 20

**Wi-Fi** Wireless Fidelity 3

# Bibliography

- [1] *Adafruit Feather Huzzah datasheet*. <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-feather-huzzah-esp8266.pdf>. 2020.
- [2] Wan Mohd Yaakob Wan Bejuri et al. “Ubiquitous positioning: integrated GPS/Wireless LAN positioning for wheelchair navigation system”. In: *Asian conference on intelligent information and database systems*. Springer. 2013, pp. 394–403.
- [3] *Bus SPI*. [http://projet.eu.org/pedago/sin/term/8-bus\\_SPI.pdf](http://projet.eu.org/pedago/sin/term/8-bus_SPI.pdf).
- [4] Rejane Dalce, Thierry Val, and Adrien V Bossche. “Comparison of indoor localization systems based on wireless communications”. In: (2011).
- [5] Philippe De Doncker. “Course of : Communication Channels”. Belgium, 2019.
- [6] *Decawave Certification Guide Europe*. [https://www.decawave.com/wp-content/uploads/2018/10/APR003\\_Certification-Guide-Europe\\_v1.1.pdf](https://www.decawave.com/wp-content/uploads/2018/10/APR003_Certification-Guide-Europe_v1.1.pdf).
- [7] *Decawave DWM1000 module*. <https://www.decawave.com/product/dwm1000-module/>. Accessed: 05-2020.
- [8] *Decawave DWM1000 User Manual*. [https://www.decawave.com/sites/default/files/resources/dw1000\\_user\\_manual\\_2.11.pdf](https://www.decawave.com/sites/default/files/resources/dw1000_user_manual_2.11.pdf). 2017.
- [9] Jense Defraye. “Determining the position of sporters using ultra-wideband indoor localization”. MA thesis. Belgium: UGent, 2017.
- [10] David L Donoho and Philip B Stark. “Uncertainty principles and signal recovery”. In: *SIAM Journal on Applied Mathematics* 49.3 (1989), pp. 906–931.
- [11] *ESP8266 Datasheet*. [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf). 2020.
- [12] Quentin Fesler. “High-accuracy localization of robotic platforms with ultra-wideband wireless transceivers”. MA thesis. Belgium: ULB-VUB, 2018.
- [13] Markus Froehle et al. “Cooperative multipath-assisted indoor navigation and tracking (Co-MINT) using UWB signals”. In: *2013 IEEE International Conference on Communications Workshops (ICC)*. IEEE. 2013, pp. 16–21.
- [14] Emmanuel Garonne. “Course of : Control System Design”. Belgium, 2017.
- [15] Léon Guyard. “Navigation et localisation en intérieur grâce à l'émission d'ondes à large bande fréquentielle”. Belgium, 2019.
- [16] Cedric Hannotier. “Indoor localization and navigation using ultra-wideband ranging”. MA thesis. Belgium: ULB-VUB, 2019.
- [17] Mads H Jespersen et al. “An indoor multipath-assisted single-anchor UWB localization method”. In: *2018 IEEE MTT-S International Wireless Symposium (IWS)*. IEEE. 2018, pp. 1–3.

- [18] Josef Kulmer et al. “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.
- [19] Peter Lynch. “The Dolph–Chebyshev window: A simple optimal filter”. In: *Monthly weather review* 125.4 (1997), pp. 655–660.
- [20] *Mathworks Capturing Azimuth, Pitch, and Roll Example*. <https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/40876/versions/8/previews/sensorgroup/Examples/html/CapturingAzimuthRollPitchExample.html>. Accessed= 05-2020.
- [21] P. Meissner, C. Steiner, and K. Witrisal. “UWB positioning with virtual anchors and floor plan information”. In: *2010 7th Workshop on Positioning, Navigation and Communication*. 2010, pp. 150–156.
- [22] Paul Meissner, Thomas Gigl, and Klaus Witrisal. “UWB sequential Monte Carlo positioning using virtual anchors”. In: *2010 International Conference on Indoor Positioning and Indoor Navigation*. IEEE. 2010, pp. 1–10.
- [23] Faranak Nekoogar. *Ultra-Wideband Communications: Fundamentals and Applications*. First. USA: Prentice Hall Press, 2005. ISBN: 0131463268.
- [24] ITU-R SM.1755-0. *Characteristics of ultra-wideband technology*. 2006.
- [25] Yu Zhou. “An efficient least-squares trilateration algorithm for mobile robot localization”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, pp. 3474–3479.