Sensor fusion for cooperative driving

A Bachelor's in Science Thesis Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona,

Universitat Politècnica de Catalunya

by

Josep Maria de Castro Catalina

In partial fulfillment of the requirements for the degree of
Bachelor's in Science in Engineering Physics

Advisors:

Univ.Prof. Dr.-Ing. Christoph Mecklenbräuker (TU Wien)

Dipl.-Ing. Mehdi Ashury (TU Wien)

Univ.Prof. Dr.-Ing. Antonio Pascual Iserte (UPC)

Research Unit of Wireless Communications
Institute of Telecommunications
Technische Universität Wien

Wien, 14. Oktober 2022

Acknowledgements

First of all, I would like to thank Prof. Christoph Mecklenbräuker for accepting me to work in his project and for giving me the opportunity to experience working at the TU Wien. In this regard, I would also like to express my gratitude to Mr. Mehdi Ashury, who took the lead from the very beginning of the project, and who has guided and advised me throughout the project, and also to Mr. Francesco d'Apolito and Mr. Christian Eliasch, who helped me in the most difficult moments of the project.

In addition, I also want to mention and thank Prof. Antonio Pascual and Prof. Quim Trullàs, from the UPC, for having helped and supported me at all times, from the distance, and for listening to me and advising me when I needed it.

On the other hand, I would also like to thank Joan, Francesco, Héctor, Eva, Marty, Laura, María and Milu, who have accompanied me during my stay in the city of Vienna, and with whom I have lived one of the best experiences of my life. Thank you, really, you have made me very happy.

Last but not least, I would like to thank my colleagues Paula and Nil for all the support and strength they have given me throughout the four years of my career, they are one of the main people responsible for getting me this far, for which I will always be grateful to them.

Abstract

The aim of this project is to modify, adapt, correct and test two target tracking algorithms to check their feasibility for future implementation in Advanced Driving Assistance Systems (ADAS). These systems, which range from automatic brake action to direct intervention in vehicle steering, require constant real-time monitoring of the environment (other cars, pedestrians, wild animals, etc.) and, in this respect, tracking algorithms have a crucial role to play, as they allow the continuous estimation of a target's trajectory in an accurate and efficient way.

This project is the continuation of a project initiated by the Wireless Communications Research Unit of the Institute of Telecommunications of the TU Wien. As a starting point, two algorithms designed and implemented by the researchers working on the original project have been used. The first of these algorithms is a Particle Filter (PF) implemented in Python, developed to track a single target, while the second consists of a complex algorithm combining a Multiple Hypothesis Tracking (MHT) algorithm coupled to a Particle Filter (PF), also implemented in Python, with the intention of performing multiple target tracking.

The project has been developed as follows. First, a random trajectory of a target was simulated in Matlab, using a random walk. Then, a Frequency Modulated Continuous Wave (FMCW) radar simulator, implemented in Matlab, developed by researchers at TU Wien, was used to perform the corresponding measurements. For the measurement process, a system consisting of four FMCW radars, placed in a square arrangement, was simulated. Finally, all data coming from the four radars was introduced into the two algorithms and combined by means of sensor fusion techniques in order to improve the quality of the trajectory estimates.

Contents

Abstract iii

Contents v

List of Figures vi

List of Tables viii

Acronyms x

1 Introduction and objectives 1

2 State of the art 3

3 The FMCW radar system 5
3.1 Basics of the FMCW radar 5
3.2 Principles of measurement 8
3.3 Introduction to CFAR methods 18

4 Target tracking 21
4.1 Introduction to Multiple Target Tracking (MTT) 22
4.2 Multiple Hypothesis Tracking (MHT) algorithm 26
4.3 The Particle Filter (PF) 33

5 Simulation setup 40
5.1 Measurement generator 40
5.2 Target tracker 43

6 Simulation analysis and results 46
6.1 Measurement preparation 46
6.2 Target tracking algorithms evaluation 49

7 Conclusions and future work 57

Bibliography 59

List of Figures

3.1 Different modulation patterns. 6
3.2 General block diagram of a FMCW radar sensor. 7
3.3 Frequency patterns of both the transmitted and the received signal for a single static target case. 9
3.4 Frequency patterns of both the transmitted and the received signal for a single non-static target case when the object is moving towards the radar. The RX signal frequency pattern shows a vertical displacement due to Doppler effect. 12
3.5 Example of transmitted (TX), received (RX) and intermediate (IF) signal [53]. 13
3.6 Received (RX) and intermediate (IF) signal arising from the second chirp (blue). Comparison with those signals arising out of the first chirp (grey) [53]. 13
3.7 Schematic representation of the meaning of ω_ϕ. 14
3.8 Schematic illustration of the angle estimation system [54]. 16
3.9 Approximation of the azimuth estimation problem (assuming that $d >> l$) [54]. 17
3.10 Example of a threshold calculated with an OS-CFAR algorithm. 19
3.11 Basic functioning of a OS-CFAR algorithm [40]. 20

4.1 Basic elements of a MTT system [10]. 23
4.2 Gating example [10]. 24
4.3 Flow diagram of the Multiple Hypothesis Tracking algorithm [45]. 27
4.4 Example of the 2-to-2 association process and connection formulation. . . 29
4.5 Local hypothesis list for the example set out in Fig. 4.4. 31
4.6 Construction of the node graph with the local hypotheses set out in Fig. 4.5. 32
4.7 Recreation of a non-Gausian pdf from a set of samples (particles) [30]. . . 33
4.8 Measurement model. The sensors estimate range (r) and azimuth (ϕ). Both parameters estimated as noisy observations with variances σ_r^2 and σ_ϕ^2 [48]. 35
4.9 Schematic diagram of basic PF operation. The filter operates over three principal stages, prediction, update and resampling. 38

6.1 Setup topology of the radar system used. 47
6.2 Simulated single-object trajectory, and the corresponding measurements (example with a BW of 1 GHz). 48

6.3 Simulated multiple-object trajectories, and the corresponding measurements (example with a BW of 1 GHz). 48
6.4 PF success probability for $r_m < 1\%$ and different number of sensors. 51
6.5 PF success probability for $r_m = 50\%$ and different number of sensors. . . . 51
6.6 PF estimation error quantiles for for $r_m < 1\%$ and different number of sensors. 52
6.7 PF estimation error quantiles for $r_m = 50\%$ and different number of sensors. 52
6.8 PF estimation error characteristics for $r_m < 1\%$ and different number of sensors. 53
6.9 PF estimation error characteristics for $r_m = 50\%$ and different number of sensors. 53
6.10 MHT-PF success probability for both targets. 55
6.11 MHT-PF estimation error quantiles for both targets. 55
6.12 PF estimation error characteristics for both targets. 56

List of Tables

5.1 Summary of the main features of the FMCW radar system simulator. . . 43
5.2 Particle filter algorithm tuning parameters. 44
5.3 MHT-PF combined algorithm tuning parameters. 45

6.1 Particle filter algorithm tuning parameters used for the simulations. . . . 50
6.2 MHT-PF algorithm tuning parameters used for the simulations. 54

List of Algorithms

4.1 Association 2-to-2. Gating and scoring algorithm. Connection formulation. (example with 2 targets). 30

4.2 Track-to-filter assignment algorithm. 32

4.3 State estimation via Particle Filter [48]. 37

5.1 Random walk generator algorithm. 41

Acronyms

f_B beat frequency. 7–9

ADAS Advanced Driving Assistance System. 1, 2, 5, 57

AIT Austrian Institute of Technology. 2, 4, 44

BW bandwidth. 6, 10, 49, 50

CA-CFAR cell-averaging CFAR. 18, 19

CFAR Constant False Alarm Rate. 7, 18

CW Continuous Wave. 5

DFT Discrete Fourier Transform. 7, 9, 10, 14, 15, 17–19

FFT Fast Fourier Transform. 7

FMCW Frequency Modulated Continuous Wave. 2, 5–7, 11, 12, 18, 21, 40, 42, 46, 47, 49, 55, 57

FOV field of view. 42, 46

GNN Global Nearest Neighbor. 25

IF intermediate frequency. 7–9, 12, 14–17, 42, 43

IR infrared. 21

LNA low-noise amplifier. 7

MHT Multiple Hypothesis Tracking. 2, 4, 22, 25–27, 29, 31, 40, 43–45, 47, 54, 56, 57

MMSE Minimum Mean Squared Error. 36

MTT Multiple Target Tracking. 21, 22

MWIS Maximum Weighted Independent Set. 31

OS-CFAR ordered-statistics CFAR. 18, 43

OS-CFAR Order Statistics Constant False Alarm Rate. 19

pdf probability densisty function. 33, 36

PF Particle Filter. 2–4, 24, 26, 32, 33, 36, 40, 43–47, 49, 50, 52–54, 56, 57

RX received. 7–11, 15–17

SMC Sequential Monte Carlo. 33

STT Single Target Tracking. 3, 21

TU Technischen Universität. 2, 4, 42, 43

TX transmitted. 7, 8, 10–12, 16, 42

TX Transmit. 7

VCO voltage-controlled oscillator. 7

CHAPTER 1

Introduction and objectives

In recent decades, the number of cars on the road has increased exponentially, and more and more people are opting for private transport to get from one place to another. Moreover, according to a study by the Spanish Government's Ministry for Ecological Transition and the Demographic Challenge [33], drivers are making longer and longer daily journeys, which can lead to increased fatigue and lack of concentration at the wheel. This combination of factors, more congested roads and increased driver distractions, therefore, seems to create a scenario conducive to road accidents. According to several studies [49], around 90% of road accidents are due to human error, so, there is an urgent need to find ways to reduce them, as avoiding them could save many lives. Of course, the solution lies in technology, and more and more companies are investing in the research and development of the so-called Advanced Driving Assistance System (ADAS), which are passive and active safety systems designed to eliminate or minimize the human error component in the operation of vehicles of many types. These systems use advanced technologies to assist the driver during driving to improve driving performance. They use a combination of sensor technologies to perceive the world around the vehicle and then provide feedback to the driver or take action when necessary. In summary, the aim is to combine sensors and algorithms to understand the vehicle environment so that the driver can receive assistance or be warned of potential hazards.

A coordinated ADAS then enables safer, more efficient and environmentally friendly traffic between road users. There are several distinct levels of ADAS, from simple backup cameras and blind-spot warning sensors to lane departure warning systems, pedestrian detection and avoidance systems, adaptive cruise control, self-parking, and more. The latest extension of ADAS are semi-autonomous or near-autonomous vehicles, i.e. vehicles that are capable of driving themselves with little or no human intervention. In this regard, systems for detecting pedestrians and nearby vehicles, as well as estimating and tracking their respective trajectories in real time, are key to enable such vehicles to drive themselves properly and without causing any collisions. One of the most important

components of these systems are the automotive radar systems, which typically consist of one or more Frequency Modulated Continuous Wave (FMCW) radars, capable of determining both the distance, the position and the speed of surrounding obstacles. Therefore, the application of a suitable radar data fusion algorithm can significantly increase the quality and robustness of the estimations. The optimal estimation of the neighbouring vehicle's trajectory based on the fused data with the help of Bayesian statistics enables a reliable reaction in a safety-critical situation in road traffic.

Focusing on this area, a group of researchers from the Institute of Telecommunications at the Technischen Universität (TU) Wien and from the Austrian Institute of Technology (AIT) has been working on the design and implementation of several object tracking algorithms, among which are a Particle Filter (PF), capable of estimating the trajectory of individual targets in real time, and a combined algorithm that merges a Multiple Hypothesis Tracking (MHT) algorithm and a PF, suitable for multiple target scenario. Nevertheless, originally only a preliminary version of these algorithms was implemented, so one of the most important tasks of this project has been the completion, improvement and adaptation of both algorithms to make them functional.

Once this has been achieved, the main **objective** of the project has been to test these algorithms in different scenarios, for example by varying the number of targets present or by varying the number of radar sensors that make up the measurement system used, in order to check its behaviour and efficiency in real-time tracking, and to evaluate whether its implementation in ADAS systems for semi-autonomous cars is feasible or not.

To carry out all these analyses, it was first necessary to simulate the radar measurements. For this purpose, a FMCW radar system simulator has been used. This simulator was designed, developed and implemented by the Institute of Telecommunications of the Wien, which has allowed its use for the present project.

The thesis is structured in seven chapters: **Chapter 1** briefly introduces the topic and motivates the Bachelor thesis; **Chapter 2** is dedicated to the study of the state of the art of target tracking algorithms; **Chapter 3** provides a theoretical background and presents the fundamentals of the FMCW radar; **Chapter 4** introduces the basics of target tracking techniques and develops the key ideas and theory underpinning both the and MHT algorithms; **Chapter 5** briefly presents the setup used for this thesis, including the FMCW radar system simulator and the implemented PF and MHT algorithms, and details their main features and operation; **Chapter 6** gathers the experimental results and the different tests carried out with the presented setup, and shows how the implemented target tracking systems perform in different scenarios; and finally, **Chapter 7** exposes the conclusions of the thesis and future work.

CHAPTER 2

State of the art

Firstly, it is important to focus on the state of the art of target tracking techniques. This is fundamental because it allows to learn about new discoveries, the main short- and long-term objectives in this field, the techniques and methods developed by the different authors, among others. That is to say, it provides a basis on which to start working in order to improve what has already been done or to innovate by creating new things, thus contributing to the achievement of new advances in the field.

Tracking is essential for guidance, navigation and control of autonomous systems, so over the years different target tracking techniques and methods have emerged and been developed. Initially, techniques based on closed-loop control systems were widely used for Single Target Tracking (STT) [15, 20]. These techniques basically consisted in using the error signal to adjust the pointing direction of the radar antenna so that it always pointed at the target, and thus be able to track its trajectory. However, these techniques are only useful for tracking a single object, and are ineffective when the number of targets to be tracked is greater than one, so soon Bayesian filters [43], which had much more potential in this area, gained momentum and began to be used for target state estimation. It was at this point, then, that the use of PFs for target tracking began to develop and become popular, along with their great competitor, the Kalman filter [52].

Since their introduction in 1993 [8], PF algorithms have become a very popular class of numerical methods for the solution of optimal estimation problems in nonlinear non-Gaussian scenarios. Compared to standard approximation methods, such as the popular extended Kalman filter, the main advantage of particle methods is that they do not rely on any local linearisation technique or any coarse functional approximation. The price to be paid for this flexibility is computational: these methods are completely expensive. However, thanks to the availability of ever-increasing computational power, these methods are already used in real-time applications appearing in fields as diverse as chemical engineering, computer vision, financial econometrics, target tracking and

robotics. Due to the popularity of particle methods, some tutorials on the subject have already been published [14, 24, 23, 24, 25].

Therefore, one of the starting points of this thesis has been the work of a group of researchers from the Institute of Telecommunications at the TU Wien [48], in which they fused data from different radar sensors by means of a particle filter (PF), thus achieving a better quality of estimations and providing greater robustness to the tracking system.

However, that work only considered the single target racking, as an ordinary particle filter alone does not have the ability to associate the multiple radar detections with each of the moving targets present in the scene and thus make parallel estimates of the state of each of them. In this sense, therefore, one of the goals of the thesis has been to go a step further and combine a data association algorithm with the existing particle filter to see how powerful it was in tracking multiple targets. The main data association methods and techniques are presented and developed in depth in the book by S. Blackman and R. Popoli [10]. One of them, which is also one of the most recent, is the MHT, which was initially proposed in 1979 by D. Reid [4]. Since then, several researchers have decided to follow the same path and have implemented this type of algorithm in various tracking systems [34, 44, 47].

The data association method chosen in this thesis has been, therefore, the MHT, following the work carried out by a group of researchers from the AIT [45], who combined a PF with a MHT algorithm, using sensor fusion techniques, in order to be able to merge data from different radars and provide robustness and quality to the system. This mentioned work has been the other starting point of the thesis, and hence one of the goals has been to adapt, modify, correct and improve the algorithm proposed in [45] and to test it in different scenarios to see its behavior and power when it comes to tracking multiple objectives.

CHAPTER 3

The FMCW radar system

Advanced Driver Assistant Systems (ADAS), like autonomous emergency breaking, take charge of identifying imminent collisions and hitting the brakes before the driver even starts to react. Therefore, this technology requires the simultaneous measurement of range, velocity and angle of objects in the surrounding area of the automobile. There is a wide variety of sensors capable of performing these tasks, e.g. camera or laser, however, the sensor par excellence, and the one most chosen by automotive companies, is the frequency modulated continous wave (FMCW) radar. Compared to other automotive sensors, FMCW radar plays an important role due to its inherent advantages [32], such as its ability to provide stable performance in night and rainy environments or its affordable price, which makes it the ideal candidate for that type of tasks.

3.1 Basics of the FMCW radar

Frequency modulated continuous wave (FMCW) radar is a radar set capable of determining distance and other kinematic magnitudes. Unlike traditional Continuous Wave (CW) radars, FMCW radars can change its operating frequency during the measurement, i.e. the signal transmitted by these type of radars is frequency modulated. Such frequency modulation provides, then, the capability to measure distances as the modulation provides a time reference.

Furthermore, if the frequency modulation is linear, the process of analysing echo signals is considerably simplified, as will be discussed in the following section. In that sense, there are several possible modulation patterns which can be used for different measurement purposes, e.g., sawthooth or triangular. For instance, while the first one is ideal for long range estimation, the second one is very useful to measure velocities. Nevertheless, although it is true that each of them is suitable for a certain situation, the idea behind their functioning is the same. So, in order to make it simple, from now on sawthooth

modulation will be taken as a reference to explain the principles of measurement of FMCW radars.

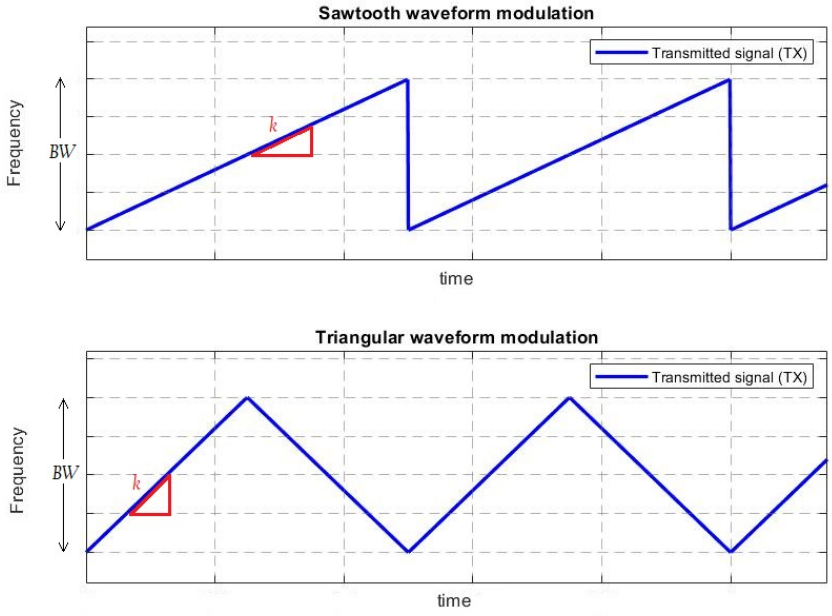

Figure 3.1: Different modulation patterns.

In Figure 3.1 both sawthooth and triangular modulation patterns are presented. Note that in the first pattern, frequency modulation starts at f_0 and then it is increased with constant slope k (also known as chirp rate) during a certain period of time T. The range of frequencies covered by the sawtooth wave in a period is the bandwidth (BW) of the chirp.

Therefore, as can be gathered from Figure 3.1, the expression for the instantaneous transmitted frequency over $(0, T)$ will be:

$$f(t) = f_0 + kt = f_0 + \frac{BW}{T}t \tag{3.1}$$

Then, the phase of the carrier, over $(0, T)$, will be given by:

$$\phi(t) = 2\pi \int_0^t f(\tau)d\tau = 2\pi f_0 t + \pi k t^2 + \phi_0 \tag{3.2}$$

And finally, the real signal transmitted by the radar can be expressed as:

$$s_{TX}(t) = A \cdot cos(2\pi f_0 t + \pi k t^2 + \phi_0) \tag{3.3}$$

This chirp signal, known as Transmit (TX) signal, is generated by a voltage-controlled oscillator (VCO) circuit, which is an oscillator circuit whose output frequency can be controlled or varied through an adjustable control voltage input, i.e., if the input control voltage is increased, the output frequency will increase proportionately, and vice versa. Then, it is amplified and emitted trough the antenna.

If there is any obstacle in the surrounding area, the signal will be reflected, and then an echo of the transmitted wave will be detected at the receiver but at a lower power and delayed temporarily. This echo is then amplified trough a low-noise amplifier (LNA)). The resultant signal of this operation is called received (RX) signal.

Next, both transmitted (TX) and RX signals are introduced into the frequency mixer. The resultant signal is then low-pass filtered giving rise to the intermediate frequency (IF) signal, whose frequency, known as beat frequency (f_B), and whose phase (ϕ_0) are really useful to determine some kinematic magnitudes. The beat frequency and initial phase can be easily found by computing the Discrete Fourier Transform (DFT) of IF signal using a Fast Fourier Transform (FFT) algorithm, which provides the spectrum of the signal from which it is possible to obtain this information.

However, it is often usual to have external interference (e.g. from other radars) or background noise, which hamper spectrum peak detection, so an extra step is required in order to mitigate those interference and select the right DFT peaks. As for interference mitigation, there are several solutions to this problem, among which the application of an adaptive filter, developed and explained in [51], stands out. For its part, Constant False Alarm Rate (CFAR) algorithms, which will be introduced in section 3.3, arise as a great solution to the problem of target detection in the presence of clutter.

In Figure 3.2 the general block diagram of a FMCW radar is shown. It summarizes and illustrates all steps presented before.

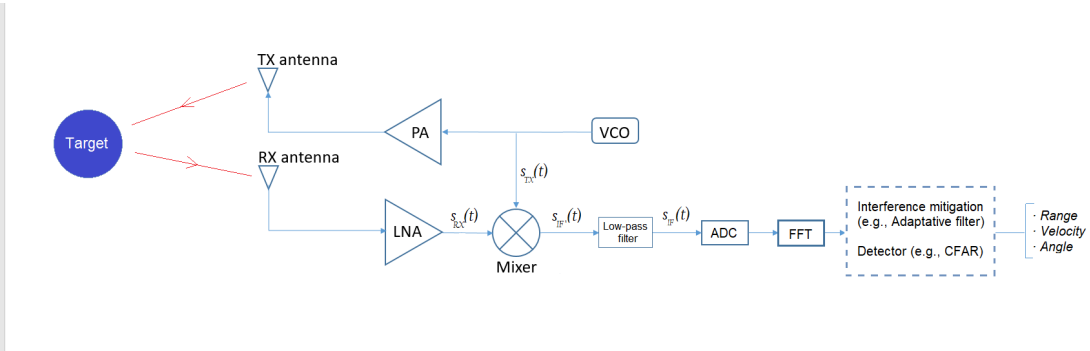

Figure 3.2: General block diagram of a FMCW radar sensor.

3.2 Principles of measurement

As mentioned before, by operating on the transmitted and received signal, it is possible to obtain a third signal, which is called IF signal, whose frequency and initial phase provide kinematic information about the detected object. This process and the mathematical development behind will be detailed next, following the steps of the diagram presented in Figure 3.2. In order to make it simple, though, the last block of the diagram has been omitted along the explanations. It will be discussed in depth in further sections.

3.2.1 Range estimation for static targets

The simplest scenario is that the targets in the surrounding area are static, or at least have zero radial velocity. Then, the received signal has only a temporal delay τ and amplitude attenuation, both due to the round trip the wave makes before reaching the receiver antenna. This time delay is directly proportional to the distance r to the detected object, and it is given by

$$\tau = \frac{2 \cdot r}{c_o} \tag{3.4}$$

where c_o is the speed of light. Furthermore, the temporal delay leads to a constant difference of frequencies during the up-chirp (see Figure 3.3). This frequency difference, which is called f_B, is the key element of this measurement method, as it can be directly related to the range r by using the slope k of the modulation.

$$f_B = \frac{2 \cdot r \cdot k}{c_o} \tag{3.5}$$

Therefore, the aim of this technique is to determine somehow the value of f_B.

As explained in section 3.1, the emitted wave is described as in equation 3.3, so that the expression of the echo signal received at the antenna is given by

$$s_{RX}(t) = A \cdot \alpha \cdot cos\left(2\pi f_0(t-\tau) + \pi k(t-\tau)^2 + \psi_0\right) \tag{3.6}$$

where α is a dumping factor due to path losses.

Then, as shown in Figure 3.2, both the TX and RX signals are mixed, so the output of the mixer is read as follows

$$s_{IF'}(t) = A^2 \cdot \alpha \cdot cos(2\pi f_0 t + \pi k t^2 + \psi_0) \cdot cos\left(2\pi f_0(t-\tau) + \pi k(t-\tau)^2 + \psi_0\right). \tag{3.7}$$

This expression can be easily decomposed into a sum of sinusoids using the appropriate trigonometric identities [1].

[1] It has been used that $cos(\alpha) \cdot cos(\beta) = \frac{1}{2}\left(cos(\alpha+\beta) + cos(\alpha-\beta)\right)$

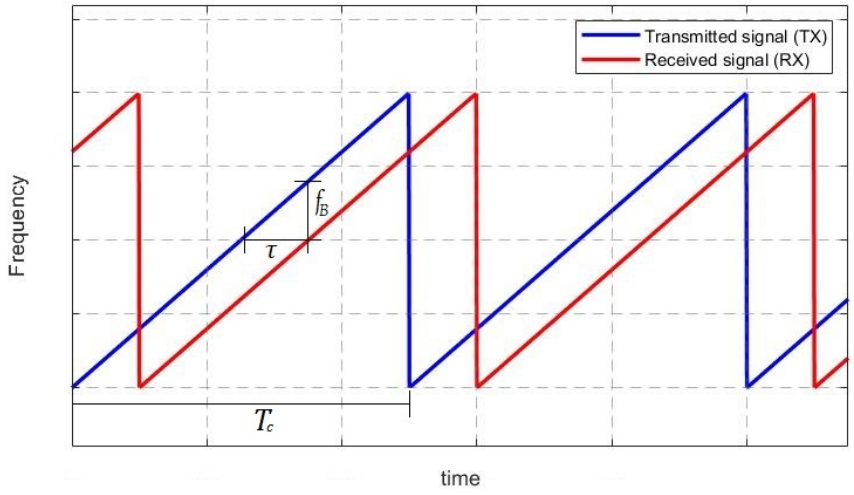

Figure 3.3: Frequency patterns of both the transmitted and the received signal for a single static target case.

$$s_{IF'}(t) = \frac{A^2\alpha}{2}\cdot\Big(cos(2\pi f_0\tau + 2\pi k\tau t - \pi k\tau^2) + cos(4\pi f_0 t + 2\pi k t^2 + 2\psi_0 - 2\pi f_0\tau - 2\pi k\tau t + \pi k\tau^2)\Big)\,. \tag{3.8}$$

As can be seen in the equation above, the frequency of the first sinusoid is already the beat frequency (remind that $f_B = k\tau$). Therefore, in order to get rid of the undesired high frequency components, the signal is low-pass filtered. This gives rise to what is known as IF signal, whose frequency is f_B, and which can be expressed as

$$s_{IF}(t) = s_{IF'}(t) * h_{LP}(t) = \frac{A^2\alpha}{2}\cdot cos\left(2\pi k\tau t + (2\pi f_0\tau - \pi k\tau^2)\right) = \frac{A^2\alpha}{2}\cdot cos\left(2\pi f_B t + (2\pi f_0\tau - \pi k\tau^2)\right). \tag{3.9}$$

Finally, this beat frequency can be obtained by computing the DFT of the IF signal. As commented, f_B is directly proportional to the distance between the radar and the non-moving object. This range can be found, then, with the following relation:

$$r = \frac{c_o}{2k} f_B = \frac{c_o}{2}\frac{T_c}{BW} f_B \quad . \tag{3.10}$$

In a multiple target scenario, the transmitted signal is reflected by each object in the radar environment. Therefore, the resulting RX signal is the superposition of all the RX components, with different time delays, corresponding to the different targets in the surrounding area. Consequently, the IF signal is also a composition of sinusoids with

different beat frequencies, each related to a target. In this case, then, the DFT will show multiple peaks, one for each beat frequency, which will provide information about the distance to each detected obstacle.

If the chirp signal reaches two or more objects that are very close in terms of distance to the radar, the signal at the mixer output will be composed of several very close tones. On this matter, the minimum range difference required between two detected objects for the radar to be able to distinguish them depends only on the bandwidth of the transmitted chirp [36, 50]. This is because, as explained above, the range is estimated from the frequency shift between the TX and RX signal, so the range resolution (Δr) will depend on the frequency resolution (Δf) achieved. Typically, this frequency resolution corresponds to the inverse of the signal observation time, which in this case is

$$\Delta f = \frac{1}{T_c}{}^{2}. \tag{3.11}$$

In turn, this expression for Δf can be easily written in terms of the bandwidth by using the chirp rate (k),

$$\Delta f = \frac{k}{BW}. \tag{3.12}$$

Finally, by introducing this expression into equation 3.2.1, it is found that the radar range resolution is inversely proportional to the bandwidth [36, 50].

$$\Delta r \quad > \quad \frac{c_o}{2 \cdot BW} \tag{3.13}$$

Intuitively, this can be explained as follows. One can realise that there are two possible options for increasing the range resolution. On the one hand, the first, the most trivial one, can be derived directly from equation 3.11, and consists of improving the resolution of the frequency shift estimate by increasing the sweep time (T_c). On the other hand, it is also possible to decrease the time delay for a given frequency shift by increasing the chirp rate (k). However, it should be noted that to do one of these options without sacrificing the other, the only possible solution is to increase the bandwidth (BW), since the relation $BW = k \cdot T_c$ must be mantained.

3.2.2 Range and velocity estimation for non-static targets

If the object to be located is, moreover, moving at a radial speed v, the received echo signal will also show a frequency shift due to the Doppler effect, apart from the time delay τ (Figure 3.4). This frequency shift will be negative or positive depending on whether the object is moving away from or towards the radar, respectively. If speed v is small compared to the speed of the emitted wave this variation, hereafter denoted as as f_D, is directly proportional to v.

[2] Remind that the Fourier Transform of a tone with a rectangular window of size T has nulls at $f - 1/T$ and $f + 1/T$ (being f the tone's frequency).

According to Doppler effect [3] equation, if it is assumed that $c_o >> v$, the frequency shift that the TX signal experiments when it reaches the moving target is described as follows,

$$f_I = \left(1 + \frac{v}{c_o}\right) f_{TX}, \tag{3.14}$$

where f_I refers to the new frequency experienced at the target position when it is hit by the transmitted wave. The signal is then reflected and travels until it reaches the receiver antenna. During this second trip, the signal undergoes another frequency shift, also due to the relative movement between the receiver antenna and the emitting source, which, in this case, is the object with which the TX signal has bounced off. Hence, the Doppler effect of the whole process is described as [39]

$$f_{RX} = \left(1 + \frac{v}{c_o}\right)^2 f_{TX}. \tag{3.15}$$

Assuming, again, that $c_o >> v$, this expression simplifies to [39]

$$f_{RX} = \left(1 + \frac{2v}{c_o}\right) f_{TX}. \tag{3.16}$$

Lastly, the total frequency shift f_D is given by

$$f_D = f_{RX} - f_{TX} = \frac{2v}{c_o} f_{TX} = \frac{2v}{\lambda_0} \tag{3.17}$$

where λ_0 is the wavelength of the TX signal.

Figure 3.4 shows the time behaviour of the received signal frequency when the received wave comes from a target that has non-zero radial velocity. Note that in this case, the frequency shift due to the temporal delay is mixed with that caused by Doppler effect, so that, to determine both the distance and the speed of the moving object it would only be necessary to find the parameters Δf and f_D. However, the Doppler shift introduced by movement of targets is actually several orders of magnitude less than the bandwidth of the signal of interest, making it almost impossible to detect, as it is much smaller than the frequency resolution of the algorithm used to find the beat frequency. For that reason, the radar uses another technique, consisting in sending several pulses in a row, not just a single chirp. The key idea is the following.

The FMCW radar emits two consecutive chirps, separated by a time T_c. The first chirp reaches the target which is located at a certain range r_0, bounces back and gets to the receiving antenna. Due to that round trip, the received signal undergoes a temporal delay denoted by τ, as discussed before. Then, both the TX and RX signals are introduced into

[3] The Doppler effect or Doppler shift is the change in frequency of a wave in relation to an observer who is moving relative to the wave source. It is named after the Austrian physicist Christian Doppler, who described the phenomenon in 1842. Mathematically, this frequency shift is expressed as $f = (\frac{c_o + v_{receiver}}{c_o + v_{source}}) f_0$ [39].

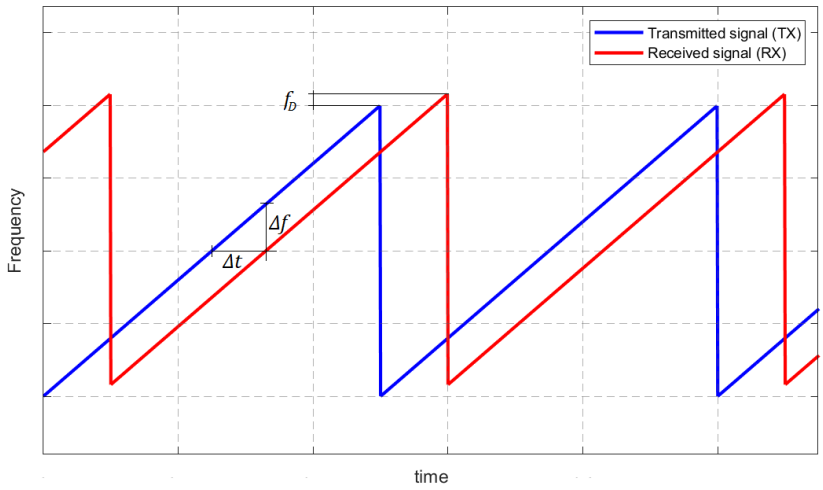

Figure 3.4: Frequency patterns of both the transmitted and the received signal for a single non-static target case when the object is moving towards the radar. The RX signal frequency pattern shows a vertical displacement due to Doppler effect.

the mixer, and the IF signal is obtained. This IF signal, which is a constant frequency sinusoid, has a certain initial phase that depends on the phase difference between the input signals. In fact, the initial phase of the signal at the mixer output is exactly the difference between the initial phases of the two inputs. This situation is illustrated in figure 3.5.

As for the second chirp sent, when it reaches the moving target, the object is not exactly in the same position, as it has moved slightly a certain radial distance Δr. Then, assuming that the motion of the target is linear within the sweep duration (which is a reasonable assumption, since sweep duration in automotive FMCW radar systems is usually in the order of $10 - 1000\mu s$), this Δr can be expressed as $\Delta r = vT_c$.

Therefore, since the distance between the target and the radar is slightly different at the times when the first and the second chirp are transmitted, the time delay between the received signal and the transmitted signal is also different for each chirp. This small variation in the temporal delay, which from now on will be denoted as $\Delta\tau$, represents a not so small variation in the phase difference between the signals at the input of the mixer, which leads, in turn, to a difference in the initial phase of the IF signal ($\Delta\phi_0$), as can be seen in figure 3.6 [4].

Finally, looking at equation 3.9, and assuming that $\Delta\tau << 1$, the term $\pi k\Delta\tau^2$ becomes

[4]Note that the phase difference between the points A and D ($\Delta\phi_{AD}$) of the TX chirp is the same as the phase difference between points C and F ($\Delta\phi_{CF}$) in the third graph, which represents the phase difference between two IF signals arising from two consecutive chirps ($\Delta\phi_0$).

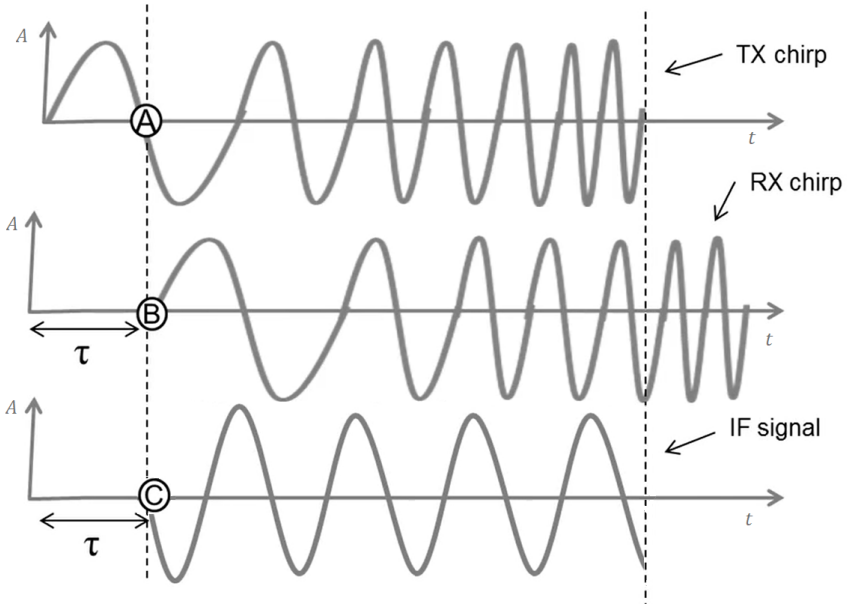

Figure 3.5: Example of transmitted (TX), received (RX) and intermediate (IF) signal [53].

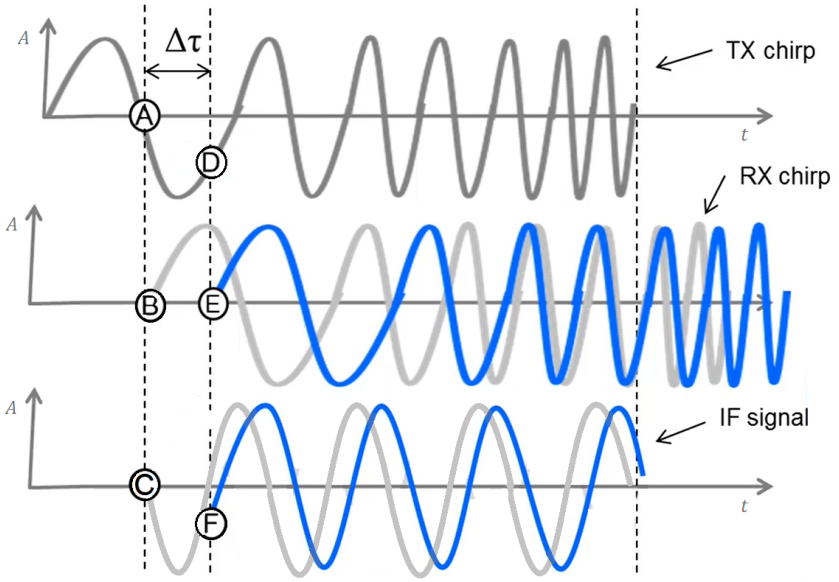

Figure 3.6: Received (RX) and intermediate (IF) signal arising from the second chirp (blue). Comparison with those signals arising out of the first chirp (grey) [53].

negligible and then the expression for the initial phase difference reads as [53]

$$\Delta\phi_0 = 2\pi f_0 \Delta\tau. \tag{3.18}$$

In turn, this expression can be written in terms of the velocity:

$$\Delta\phi_0 = 2\pi f_0 \Delta\tau = \frac{4\pi\Delta r}{\lambda_0} = \frac{4\pi v T_c}{\lambda_0}. \tag{3.19}$$

So, the velocity of the target can be expressed as

$$v = \frac{\lambda_0}{4\pi T_c}\Delta\phi_0. \tag{3.20}$$

In summary, in order to determine the velocity, the phase difference between the IF signals arising from the consecutive chirps sent must be found.

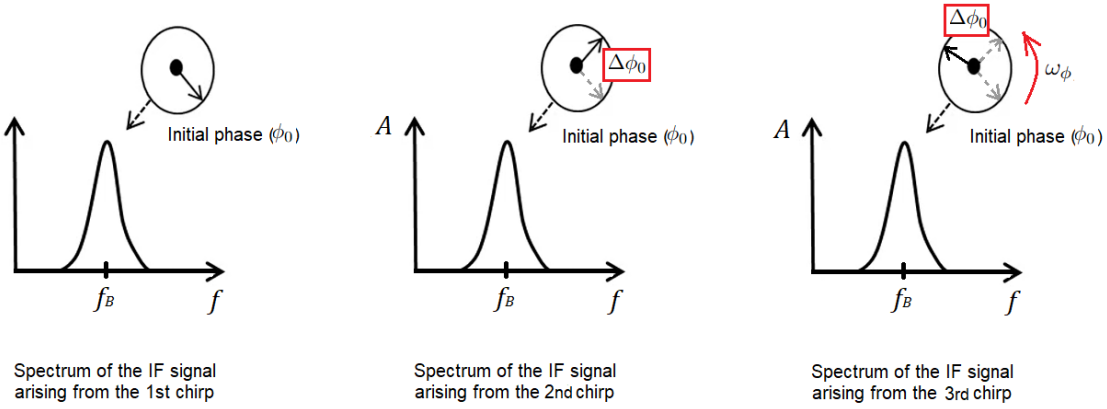

Figure 3.7: Schematic representation of the meaning of ω_ϕ.

Note that this $\Delta\phi_0$ can be understood as the spin rate (ω_ϕ) of the initial phase of the IF signal arising from each chirp, as illustrated in figure 3.7, and that, therefore, equation 3.20 can be rewritten as

$$v = \frac{\lambda_0}{4\pi T_c}\omega_\phi. \tag{3.21}$$

Then, one realises that this ω_ϕ can be easily found by performing a double DFT. First, a DFT of each IF signal, corresponding to each emitted chirp, shall be calculated. Each of these DFTs allows for finding the frequency of the corresponding IF signal and its initial phase. For all calculated DFTs, the spectrum will show peaks at the same place, i.e at the same frequency bin, which means that the frequency of the IF signals coming from each chirp is practically the same. This is because the range variations during the emission of the chirps are so small that the corresponding frequency variations are well

below the frequency resolution[5] of the DFT algorithm. In contrast, as shown before, the phase of the IF signal is very sensitive to small changes in the range of the target, so that significant variations in the phase associated to the dominant frequency in the different spectrums obtained from the different DFTs will be noticeable.

Then, assuming that the radar has emitted M consecutive chirps, there will be M complex solutions, one for each chirp emitted. At this point, another DFT must be calculated, now along the phases associated to the M dominant frequencies found from the M spectrums, which will provide the phase spin rate (ω_ϕ). Then, using equation 3.21, the radial velocity can finally be determined.

Regarding the velocity resolution (Δv), one can deduce from equation 3.21 that it is directly related with the resolution achieved in the estimation of the initial phase spin rate ($\Delta\omega_\phi$).

$$\Delta v = \frac{\lambda_0}{4\pi T_c}\Delta\omega_\phi \tag{3.22}$$

As explained in the previous section, the resolution in frequency (Δf) is given by equation 3.11, so, in this case, $\Delta\omega_\phi$ should be given by [53]

$$\Delta\omega_\phi > \frac{2\pi}{M}, \tag{3.23}$$

where M represents the number of emitted chirps and, hence, the number of initial phase (ϕ_0) samples.

Lastly, by imposing this condition to equation 3.22, the final expression for the velocity resolution (Δv) of this technique is read as [36, 53]

$$\Delta v > \frac{\lambda_0}{2MT_c}. \tag{3.24}$$

3.2.3 Azimuth estimation

Unlike range or radial velocity estimation, azimuth estimation requires at least two receiving antennas (RX), not just one. These antennas should be separated by a small distance l, so that the distance between the detected object and each antenna is slightly different (Δd) (see figure 3.8).

Consequently, the echo signal received by each antenna will have a lightly different time delay $\Delta\tau$. As discussed in section 3.2.2, small variations in range, and the resulting small variations in time delay, do not produce significant changes in the beat frequency of the IF signal, but they do considerably affect its initial phase (figure 3.6). Furthermore, following the same idea as that used to determine the velocity, the initial phase difference

[5] Since the range variations Δr during the emission of the chirps are so small, the variations frequencies f_B of the different IF signals are imperceptible, so the ranges measured based on them are assumed to be equal to each other.

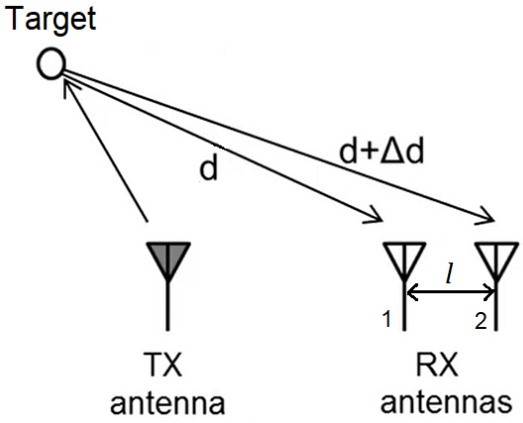

Figure 3.8: Schematic illustration of the angle estimation system [54].

between the IF signals received at each antenna can be directly related to the slight difference in range between the target and each of the receivers:

$$\Delta\phi_0 = 2\pi f_0 \Delta\tau = \frac{2\pi\Delta d}{\lambda_0} \quad [6]. \tag{3.25}$$

As for the small range difference Δd, it can be related to the azimuth angle by making some approximations and using triangulation. Under far-field assumptions, i.e. supposing that $d >> l$, the wavefront is assumed to be flat, so the line of sight from each antenna to the target can be shown to be parallel and the time delays for each consecutive antenna is linearly increasing if the different receivers are evenly spaced forming a linear array.

Therefore, as shown in figure 3.9, Δd can be written in terms of the azimuth angle:

$$\Delta d = d_2 - d_1 = lsin(\theta). \tag{3.26}$$

Equation 3.25 turns then into

$$\Delta\phi_0 = \frac{2\pi lsin(\theta)}{\lambda_0}, \tag{3.27}$$

which means that θ can finally be determined by calculating the initial phase difference between the IF signals generated at each of the antennas:

$$\theta = arcsin\left(\frac{\lambda_0\Delta\phi_0}{2\pi l}\right). \tag{3.28}$$

[6] Note that in this case it has been taken into account that the only wave that travels an extra path is the received wave, not the transmitted one, so the expression has been divided by two with respect to equation 3.19. It has been assumed, then, that the TX signal used as an input to the mixer is exactly the same for all RX antennas, and that it has been emitted by a TX antenna located at the same place as the first RX antenna in the linear array (Fig 3.8).

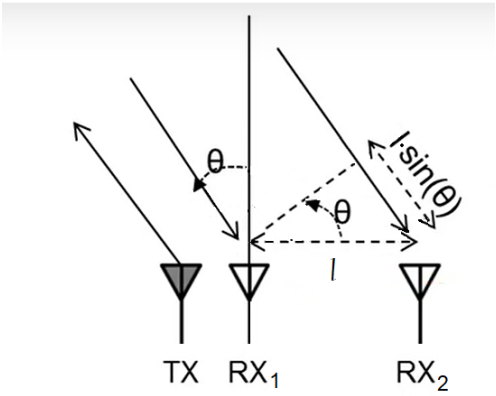

Figure 3.9: Approximation of the azimuth estimation problem (assuming that $d >> l$) [54].

Note again that this initial phase difference ($\Delta\phi_0$) between consecutive chirps can be understood as the initial phase spin rate (ω_ϕ), and therefore equation 3.28 is read as

$$\theta = arcsin\left(\frac{\lambda_0}{2\pi l}\omega_\phi\right). \tag{3.29}$$

Lastly, as it is done to determine the velocity (see section 3.2.2), a double DFT must be performed in order to find the value of ω_ϕ. The procedure is the same. Assuming that the number of antennas in the linear array is L, there will be L IF signals, one for each antenna, and hence a DFT must be performed for each of them. Then, as seen in the previous section, L complex solutions will be obtained, with the same modulus (f_B) and different phases. Last, another DFT must be performed along the phases of these L complex solutions in order to determine the phase spin rate (ω_ϕ), from which we can finally calculate the azimuth angle (θ) at which the detected target is located.

As for azimuth resolution ($\Delta\theta$), this depends on the spin rate resolution ($\Delta\omega_\phi$) achieved. And $\Delta\omega_\phi$, in turn, is obtained similarly to equation 3.23, therefore, it can be written as [54]

$$\Delta\omega_\phi > \frac{2\pi}{L}. \tag{3.30}$$

where L is the number of RX antennas in the linear array, so it is also the number of IF signals obtained, and therefore it is the number of initial phase (ϕ_o) samples.

Then, assuming that two targets are located at azimuths θ and $\theta + \Delta\theta$ relative to the radar, the corresponding initial phase spin rates (ω_ϕ) differ in

$$\Delta\omega_{\phi 1,2} = \omega_{\phi 1} - \omega_{\phi 2} = \frac{2\pi d}{\lambda_0}(sin\,(\theta + \Delta\theta) - sin(\theta)) \approx \frac{2\pi d}{\lambda_0}cos(\theta)\Delta\theta \quad [7]. \tag{3.31}$$

[7] The formal definition of the derivative has been used here: $\frac{d}{dx}[sin(x)] = \frac{sin(x+\Delta x)-sin(x)}{\Delta x} = cos(x)$.

In the end, by applying the condition defined in equation 3.30 to this expression, the final azimuth resolution of this estimation technique is written as [54]

$$\Delta\theta > \frac{\lambda_0}{L \cdot l \cdot cos(\theta)}. \tag{3.32}$$

Note that the resolution is not constant, as it depends on θ. This means that if the azimuth is small, i.e. if the target is in front of the antenna array, the resolution will be much better than if the target is aligned with the antenna array.

3.3 Introduction to CFAR methods

The biggest challenge in object detection is to decide whether a peak in the spectrum corresponds to a potential object or not. Comparing the frequency spectrum with a fixed threshold value might work perfectly well for an ideal spectrum. However, in a real measurement, the presence of noise of unknown power can cause many false alarms if the threshold value is chosen too low. Conversely, if it is set too high, fewer objects will be detected. The so-called constant false alarm rate (CFAR) technique thus provides an output adapted to the background noise and ensures that the number of false alarms does not depend on the power of the noise (see Figure 3.10). The assumed noise model is a zero-mean complex-valued Gaussian random variable, which is independently and identically distributed. To find an adaptive threshold for the given noise model, the noise power has to be estimated. On this matter, there are two main types of CFAR algorithms, which are the two most widely used: the cell-averaging CFAR (CA-CFAR) [37] and ordered-statistics CFAR (OS-CFAR) [29].

This section, though, will focus only on the OS-CFAR method, which is the one implemented in the FMCW radar system generator used for this project. Next, then, the basics of this method will be briefly introduced, just to give the reader a quick overview of the OS-CFAR technique.

3.3.1 Ordered-statistic CFAR (OS-CFAR)

As seen in the previous sections, once a DFT is performed, each frequency bin of the spectrum obtained represents a specific range, azimuth or velocity, depending on the data series over which the DFT is performed. Each frequency bin, hereinafter referred to as cells, must be analysed to determine whether an object is in that cell or not, i.e., to determine whether an actual object is responsible for causing a in the DFT in that particular frequency bin or not.

The basic idea of CFAR algorithms is to create a reference window around each cell under test Y, composed of its neighbouring cells X_1 to X_N, and calculate its threshold value based on the DFT amplitudes of those N neighbouring cells. In this sense, though, it is important to note that since the peaks are not located in a single cell, but are spread

Figure 3.10: Example of a threshold calculated with an OS-CFAR algorithm.

over several cells, the reference window should not be placed directly next to the test cell. This means, therefore, that when constructing it, the cells right next to Y, the so-called guard cells, must be omitted and the next N nearest cells must be selected.

The main difference between the CA-CFAR and Order Statistics Constant False Alarm Rate (OS-CFAR) methods is the way the reference window is used to calculate the threshold value. On the one hand, as for CA-CFAR algorithms, they calculate the threshold by averaging the DFT amplitudes of neighbouring cells. However, the detection procedure of such algorithms is not designed for the detection of multiple objects, as other detected objects within the reference window may distort the noise estimate and consequently increase the threshold value. It is at this point, therefore, that OS-CFAR methods emerge, trying to overcome this issue. The point is that unlike the CA-CFAR procedure, which uses all signal amplitudes in the reference window to determine the threshold, the OS-CFAR algorithm only selects a single amplitude. The general idea of this technique is to sort the cells in the reference window by amplitude in ascending order and then estimate the noise based on the k^{th} value in the list. Lastly, the estimation of the noise floor has to be multiplied by a scaling factor T_{OS} to obtain the threshold value, as shown in Figure 3.11.

Therefore, the first important task is to find a suitable value for k, as it decides which value of the reference window is used for the background noise estimation. According to

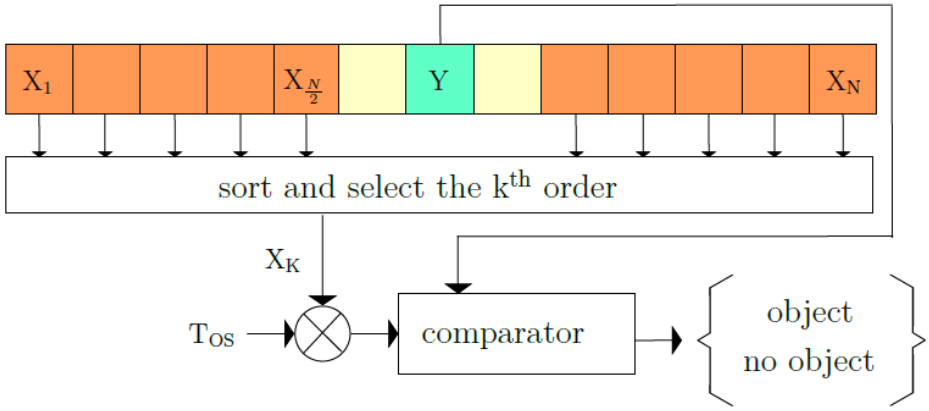

Figure 3.11: Basic functioning of a OS-CFAR algorithm [40].

[27], three quarters of the window size N is a reasonable choice for k. Furthermore, as explained, the amplitude of the k^{th} cell has to be multiplied by a scaling factor in order to achieve a given constant probability of false alarm (P_{FA}). The scaling factor T_{OS} can be calculated by solving [6, 21]

$$P_{FA} = \binom{N}{k} \frac{k!(T_{OS} + N - k)!}{(T_{OS} + N)!} \tag{3.33}$$

for a given probability of false alarm and a given k. The complex process leading to this expression is developed and explained in detail in [6].

CHAPTER 4

Target tracking

Target tracking is an essential requirement for surveillance systems employing one or more sensors, together with computer subsystems, to interpret the environments. Typical sensor systems, such as radar (section 3.1) , infrared (IR), and sonar, report measurements from diverse sources: targets of interest, background noise such as clutter, or internal error sources such as thermal noise. The target tracking objective is to collect sensor data from a FMCW radar containing one or more potential targets of interest and then to partition the sensor data into sets of observations, or tracks, that are produced by the same sources. Once tracks are formed and confirmed (so that the background and other false targets are reduced), the number of targets can be estimated and quantities, such as target velocity, future predicted position, and target classification characteristics, can be computed for each track [10].

In this sense, there are two possible situations that one can encounter in the scope of target tracking. On the one hand, if there is only one object in the field of view of the sensor or if the user's goal is to track the trajectory of a single object, then a STT algorithm is used. On the other hand, though, if the number of targets to track is more than one, then a Multiple Target Tracking (MTT) system will be required. Although the idea behind the algorithms used to solve each of the situations is essentially the same, the fact is that when multiple, co-existing targets are involved, the process becomes more complicated due to the measurement origin uncertainty. For its part, single target tracking corresponds to the simplest case and its solution is summed up to solving a state estimation problem [1][5],which is tackled by employing different so-called Bayesian-filters [3, 19]. For multi-target tracking, the source uncertainty in the observations must be resolved before tackling the state estimation task. Therefore, the extension of STT to MTT first requires a complex data association logic to classify the returning sensor data

[1]The term *state estimation problem* comprehends all those problems that consist of determining the current, or future, state of a complex system given the complete history of observations and measurements that have been taken by sensors so far.

into the general categories of targets of interest, recurring sources that are not of interest (such as background clutter), and spurious signals with little or no correlation in time [10]. In this respect, several data association techniques can be used.

In the following section, the basics of multi target tracking systems will be introduced, and next the data association algorithm and the Bayesian filter implemented in this project will be presented and detailed.

4.1 Introduction to Multiple Target Tracking (MTT)

Multiple target tracking has immense application in areas such as surveillance, air traffic control, defense and computer vision. As mentioned, the aim of a target tracking algorithm is to estimate the target position precisely from the partial noisy observations available. The real challenges of multiple target tracking are to accomplish the same in the presence of measurement origin uncertainty and clutter [35].

As mentioned, in multiple target tracking, the positions of individual targets are estimated in presence of noisy partial observations arising from a number of indistinguishable targets moving about. The random measurements received might belong to any of the targets and an appropriate mapping between the targets and their corresponding measurements is necessary before state estimation. For its part, the measurement originating from any non-target is regarded as clutter which deteriorates the performance of the tracker. In that sense, multiple target tracking seeks solution to two major problems encountered in this scenario: problem of data association and state estimation. On the one hand, suitable data association algorithms are required prior to the filtering in order to overcome the inherent problem of measurement origin uncertainty associated with a multi-target environment [22]. Then, after that data association, state estimation techniques, such as the ones presented in the previous section, are used to track the different targets and predict their following position.

So first, this section will focus on introducing the basics of data association techniques, and then it will centre on presenting and developing the Multiple Hypothesis Tracking (MHT), which is algorithm that has been implemented in this project.

4.1.1 Elements of a conventional MTT system

Figure 4.1 gives a representation of the functional elements of a simple recursive MTT system. However, the truth is that, when recently developed techniques, such as MHT, are used, there is considerable overlap of the functions of these elements and the distinction between these individual elements becomes less apparent. Nevertheless, this representation provides a convenient partitioning that will be used to introduce the typical functions required for an MTT system [10]. The purpose of the figure is to give an overview of the MTT problem and to show how the elements interrelate.

Let us assume recursive processing is being used so that tracks have been formed on the previous scan. Now, input data are received from the sensor and processing loop

described in Figure 4.1 is to be performed. Incoming observations are first considered for the update of existing tracks. Gating tests determine which possible observation-to-track pairings are "reasonable", and a more refined association algorithm is used to determine the final pairings. Observations that are not assigned to existing tracks can initiate new tentative tracks. A tentative track becomes confirmed when the number and quality of observations included in the track satisfy confirmation criteria. Similarly, low-quality tracks, as usually determined by the update history, are deleted. Finally, after inclusion of the new observations, tracks are predicted ahead to the arrival time for the next set of observations. Gates are placed around these predicted positions and the processing cycle repeats [10, 22, 45].

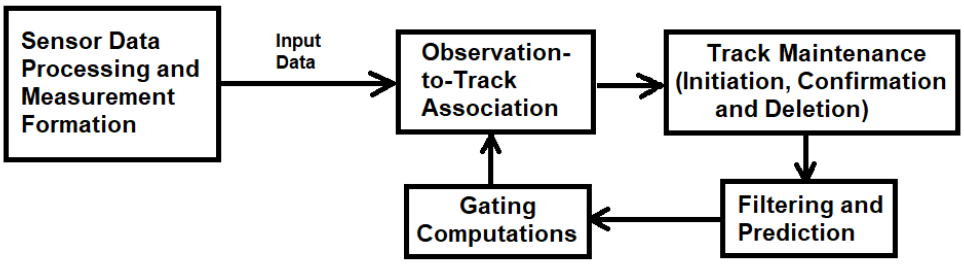

Figure 4.1: Basic elements of a MTT system [10].

4.1.2 Basics of data association

The gating, observation-to-track association and track maintenance functions shown in Figure 4.1 are part of the overall data association function. Each of these will be introduced and explained in more detail in the following sections.

4.1.2.1 Gating

Gating is a technique for eliminating unlikely observation-to-track pairings. It is used as a screening mechanism to determine which observations are valid candidates to update existing tracks and it is performed primarily to reduce unnecessary computations by the association and maintenance functions that follow.

As discussed below, a gate is formed around a predicted measurement and all observations that satisfy the gating relationship (fall within the gate) are considered for track update. The manner in which the observations are actually chosen to update the track depends on the data association method but most data association methods utilize gating in order to reduce later computation.

Figure 4.2 illustrates gating for two closely spaced targets and four observations. Note that the gates may overlap for closely spaced targets. Gates are established and gating is performed in the following general way.

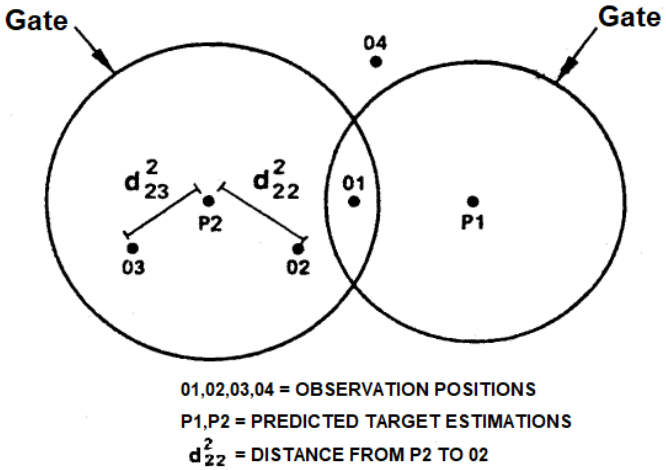

Figure 4.2: Gating example [10].

1. First of all, estimations are made of what the measured quantity should be at the time of the next observation. This process is called prediction, and it is carried out by special filters, such as Kalman-Filter [41] or Particle Filter (PF) (see section 4.10), which require statistics describing the accuracy of these estimates [10].

2. Then, the difference between each measurement and its corresponding estimate is calculated.

3. A maximum error between estimate and measurement is calculated for all measured quantities by using the estimate and measurement accuracy. The computed differences are compared to the computed maximum allowable error, and if the differences are less than the maximum allowable errors, the observation satisfies the gate.

Furthermore, as it is shown in Figure 4.2, it is often useful to form a kind of total distance d^2_{ij} from track i to observation j. Thus, a normalization process is required whereby the differences in each of the component measurements are squared, divided by certain constants, which usually are the variances of the expected differences (if they are known), and summed to form a kind of total normalized distance. For example, if range (R), azimuth (Φ) and elevation (Θ) are measured, then the normalized distance would be

$$d^2 = \frac{(R_p - R_o)^2}{a^2} + \frac{(\Phi_p - \Phi_o)^2}{b^2} + \frac{(\Theta_p - \Theta_o)^2}{c^2} \tag{4.1}$$

where (R_p, Φ_p, Θ_p) is the predicted position and (R_o, Φ_o, Θ_o) is the measured position [10].

Then, as an alternative, the gating can be also carried out by using this normalized distance, as in [22, 45]. Nevertheless, although it is a quite extended praxis, the truth is that computations can usually be saved if the measurement components are first examined individually.

4.1.2.2 Observation-to-track association

Regarding the observation-to-tracking association function, its job is to take the observation-to-track pairings that satisfy gating and determine which observation-to-track assignments will actually be made. That is, new possible tracks are created at this point, with the observations that satisfied gating and, then, they are evaluated, validated or rejected in the next step of the flowchart.

4.1.2.3 Track maintenance

For its part, track maintenance refers to the functions of track initiation, confirmation and deletion. A simple approach to track initiation, used by the Global Nearest Neighbor (GNN) method [17], is to start new tracks on those observations that are not assigned to existing tracks. However, a more preferable method, used with multiple hypothesis tracking (MHT) [22, 45], will start tentative tracks on all observations and use subsequent data to determine which of these newly initiated tracks are valid [10].

Once a tentative track is formed, a confirmation logic is usually required because the probability of a single observation being from an extraneous source is too high for immediate confirmation. Thus, it is usually required that at least one other observation is assigned to a tentative track before the track is considered to be confirmed. However, a much better approach is to define a track score function and compare this score with an appropriately chosen track confirmation threshold [10, 31].

A track that is not updated becomes degraded, and it must be deleted if not updated within some reasonable interval. If a sufficiently long time elapses without detection, the target will probably no longer be within the scan volume. A typical simple rule is to delete a track after N_D consecutive scans have produced no updating observations. Again, however, the use of a track score function is more general and more readily applied to varying detection capabilities. Also, the track score reflects the quality of the update so that updates that barely satisfy the gate may actually decrease the score [10, 31]. The computation of this score function depends on the implemented method and requires complex probabilistic expressions that include all aspects of the data association problem, as discussed in [10].

Further in the text, in section 4.2, the focus will be put on the multiple hypothesis tracking (MHT) algorithm, which is one of the most widely used data association methods, and it is the one that has been evaluated in this project.

4.1.3 Basics of state estimation

State estimation problems, also designated as nonstationary inverse problems [19], are of great interest in innumerable practical applications. In such kind of problems, the available measured data is used together with prior knowledge about the physical phenomena and the measuring devices in order to sequentially produce estimates of the desired dynamic variables. This is accomplished in such a manner that the error is minimized statistically [3].

As mentioned before. state estimation problems are solved with the so-called Bayesian filters [2] [3, 16, 19]. In the Bayesian approach to statistics, an attempt is made to utilize all available information in order to reduce the amount of uncertainty present in an inferential or decision-making problem. As new information is obtained, it is combined with previous information to form the basis for statistical procedures. The formal mechanism used to combine the new information with the previously available information is known as Bayes' theorem [16, 19, 26].

The most widely known Bayesian filter method is the Kalman filter [1, 3, 14, 19]. However, the application of the Kalman filter is limited to linear models with additive Gaussian noises. Extensions of the Kalman filter were developed in the past for less restrictive cases by using linearization techniques [2, 9, 14, 19]. Similarly, Monte Carlo [3] methods have been developed in order to represent the posterior density in terms of random samples and associated weights. Such Monte Carlo methods, usually denoted as Particle Filters (PF) among other designations found in the literature, do not require the restrictive hypotheses of the Kalman filter. Hence, particle filters can be applied to non-linear models with non-Gaussian errors [3, 11, 12, 14]. For this reason, the filter selected to perform the state estimation tasks in this project has been the latter, the particle filter, which will be presented and described in detail in section 4.3.

4.2 Multiple Hypothesis Tracking (MHT) algorithm

As stated before, multiple hypotheses tracking (MHT) is one of the earliest successful algorithms for multi-target tracking and has become especially popular in the radar target tracking community. It was originally proposed in 1979 by Reid [4], and the key idea is to collect all observations taken at a certain time step and pose all possible associations between them, then score each of the associations according to the distance between the associated observations by calculating the likelihood, and finally take those non-conflicting association hypotheses with the highest score [34].

[2] Bayes Filtering is the general term used to discuss the method of using a predict and update cycle to estimate the state of a dynamical system from sensor measurements.

[3] A Monte Carlo simulation is a model used to predict the probability of different outcomes when the intervention of random variables is present. It is a technique in which a large quantity of randomly generated numbers are studied using a probabilistic model to find an approximate solution to a numerical problem that would be difficult to solve by other methods.

In the following sections, the key elements of the MHT algorithm are introduced and detailed. However, a brief clarification of the concepts and the notation that will be used throughout the explanations is appropriate first.

4.2.1 Previous concepts and definitions

As commented, in order to illustrate the working basics of the MHT algorithm, it is first necessary to introduce some concepts that will be used recurrently during the explanation. On the one hand, the term *connection* is defined as the 2-to-2 association between two measurements or observations obtained at the same time step, or as the association between an observation and one of the state estimates made between the previous time step. On the other hand, the term *track* refers to a local hypothesis, i.e. each of the possible links between the multiple connections made. Finally, the term *global hypothesis* will be used to refer to a set of independent tracks, i.e. a set of tracks that are compatible with each other since they do not have any observations in common [45].

As for the observations, from now on they will be denoted as $z^s_{k,t}$, where s refers to the sensor to which the observation belongs (if the total number of sensors is S, then $s \in [1, S]$), k is the id of the detection (assuming that the radar has made K detections at the same time step, then $k \in [1, K]$), and t represents the time instant at which the measurements are made. For its part, connections are denoted as A_n, tracks as h_n and, finally, global hypothesis as H_n.

4.2.2 Main stages of the MHT algorithm

Figure 4.3 shows the flowchart of the conventional multiple hypothesis tracking algorithm. As can be seen, the MHT algorithm operates in three main stages (since the prediction phase is performed by an external state-estimation algorithm).

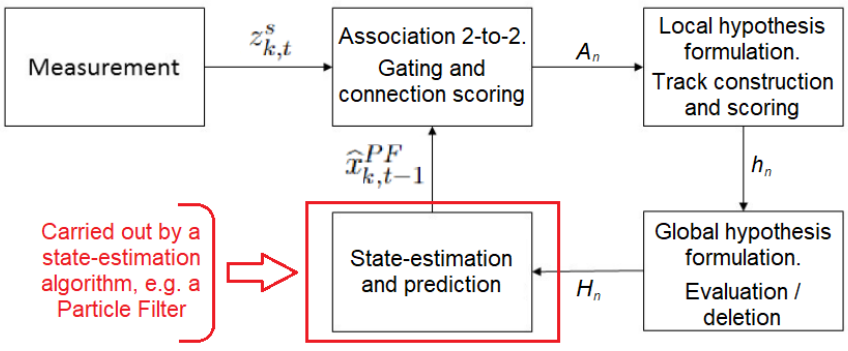

Figure 4.3: Flow diagram of the Multiple Hypothesis Tracking algorithm [45].

4.2.2.1 Association 2-to-2. Gating and connection scoring

The first of these stages, the 2-to-2 association, consists in taking all the observations obtained at a given time instant and comparing them all with each other, using the gating technique, to assess whether they are two potential measurements of the same target or whether, on the contrary, they are independent measurements coming from two different targets. If the gating process is successful, a connection (A_n) between the two observations is established, scored and added to the list of connections. To connect the current measurements with the trajectories estimated so far ($\widehat{x}^{PF}_{k,1:t-1}$), the last state predictions ($\widehat{x}^{PF}_{k,t-1}$), made at the previous time instant, are taken and added to the list of observations, so that they also take part in the 2-to-2 association process, and are connected with the received observations. In addition, the possibility for each measurement to be independent and not associated with any other measurement is also added to the list of connections. In this case, the connection is denoted as $A_n = [z^s_{k,t} \sim 0]$.

This whole process described above is exemplified and illustrated in the figure 4.4.

Gating

As explained in section 4.1.2.1, the gating process is based on the distance between the target position predictions and the observations (in black in Fig. 4.4), or between two different observations (in blue in Fig. 4.4), weighted by the maximum allowable error. Taking into account one prediction as $\widehat{x}^{PF}_{k,t-1} = (R_p, \Phi_p, \Theta_p)$ and a new arriving observation as $z^s_{k,t} = (R_o, \Phi_o, \Theta_o)$, and R as distance, Φ as azimuth and Θ as elevation, then the weighted distance d_w is defined as in 4.1:

$$d_w = \frac{(R_p - R_o)^2}{a^2} + \frac{(\Phi_p - \Phi_o)^2}{b^2} + \frac{(\Theta_p - \Theta_o)^2}{c^2} \tag{4.2}$$

The variables a, b and c are the maximum allowed differences between the values of distance, azimuth and elevation, and they are defined arbitrarily. Note that the equation above describes an ellipsoid. With differences of azimuth, elevation and distance within this ellipsoid, the distance is within 0 or 1. For this reason, the gating procedure is considered passed when the value d_w is less than or equal to 1. If the gating process is passed, the connection is added to the list of connections (see Fig. 4.1). As can be guessed, depending on the magnitude of these values (a, b, c), the gates built around the predicted positions and the arriving observations will be larger or smaller, and hence the amount of connections done will also be larger or smaller, respectively. Therefore, they should be adjusted according to the accuracy of the measurements.

Connection scoring

Finally, as explained in section 4.3 a score based on its likelihood function is given to each new connection. In the model implemented in this project, the score of this connections has been computed based on d_w with the following relation [45]:

$$a = \alpha \cdot e^{-\lambda \cdot d_w} \tag{4.3}$$

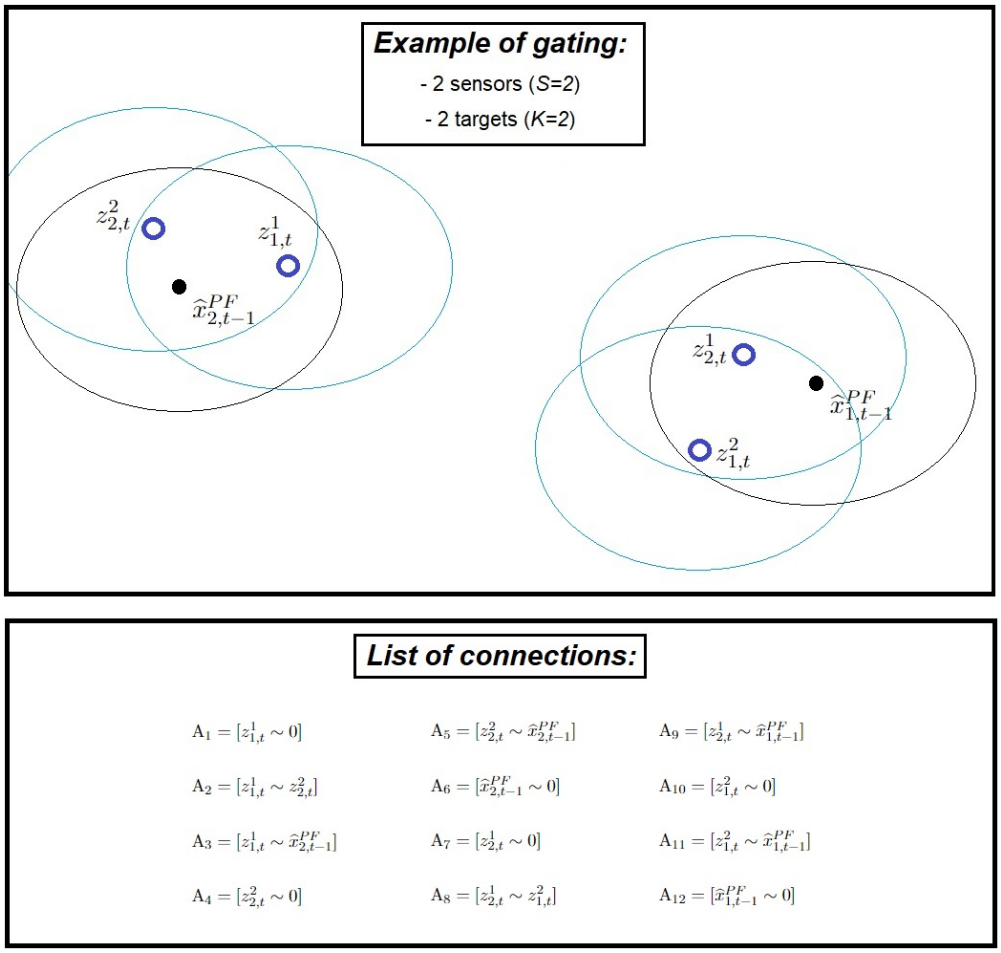

Figure 4.4: Example of the 2-to-2 association process and connection formulation.

The α variable takes into account the confidence parameters of both measurements with λ, a user-defined parameter.

On the other hand, the possibility that one of the observations or predictions is independent and not associated with any other observation or prediction ($A_n = [z^s_{k,t} \sim 0]$) is scored with a fixed low score of 0.05 [45].

All the steps presented and described above are shown and implemented in algorithm 4.1, which forms the first phase (a key one) of the complex MHT algorithm.

4.2.2.2 Local hypothesis formulation

Once all possible connections have been noted, the next step of the MHT algorithm is to formulate all possible local hypotheses. For this, all the connections made are taken into account and, in addition, the possible mergers and combinations between these connections are also added, thus creating associations of more than two observations.

Algorithm 4.1: Association 2-to-2. Gating and scoring algorithm. Connection formulation. (example with 2 targets).

Data: Measurement vectors z_t^s for all sensors s and all time-steps t. Target state predictions $\widehat{x}_{k,t-1}^{PF}$

Result: Connection list (A_{list}) containing all possible 2-to-2 associations.

Observation 1: the measurement vector z_t^s of each sensor contains 2 detections at each time step (remind that there are 2 targets in the scene), which from now on will be denoted as z_{1t}^s and z_{2t}^s.

Observation 2: the estimate of the position of the targets in the previous iteration $\widehat{x}_{1,t-1}^{PF}, \widehat{x}_{2,t-1}^{PF}$ is added to the list of measurements as z_t^{S+1} (where S is the total number of sensors), in order to perform the gating between the arriving measurements and the previous estimates.

for $t \in [0, N-1]$ **do**
  Associations 2 by 2: *these associates will be denoted by* A_h.
  $n = 0$
  for $s \in [1, S+1]$ **do**
    for $r \in [s+1, S+1]$ **do**
      for $k \in [1, 2]$ **do**
        for $j \in [1, 2]$ **do**
          `Connection hypothesis:` $A_n = [z_{k,t}^s \sim z_{j,t}^r]$
          Gating and scoring:
          $\Delta\Phi = \Phi_{k,t}^s - \Phi_{j,t}^r$
          $\Delta\Theta = \Theta_{k,t}^s - \Theta_{j,t}^r$
          $\Delta R = R_{k,t}^s - R_{j,t}^r$
          $d_w = \frac{\Delta R^2}{a^2} + \frac{\Delta\Phi^2}{b^2} + \frac{\Delta\Theta^2}{c^2}$
          if $d_w < 1$ **then**
            `gate = TRUE`
            `Score` $= \alpha \cdot e^{-\lambda \cdot d_w}$
            $A_{list}(n) = A_n$ (*A_n is added to the connection list.*)
            $n = n + 1$
          else
            `gate = FALSE`
          end
        end
        Independent measurements: *In addition, the possibility for each measurement to be independent and not associated with any other measurement is added to the list of connections. This is denoted as* $A_n = [z_{k,t}^s \sim 0]$.
        $A_n = [z_{k,t}^s \sim 0]$
        $A_{list}(n) = A_n$
        `Score` $= 0.05$
        $n = n + 1$
      end
    end
  end
end

These local hypotheses or tracks, therefore, are nothing more than the 2 to 2 connections already made, plus all the possible unions between these connections [45].

An illustration of a local hypotheses list is shown in figure 4.5, following the example set out in section 4.2.2.1.

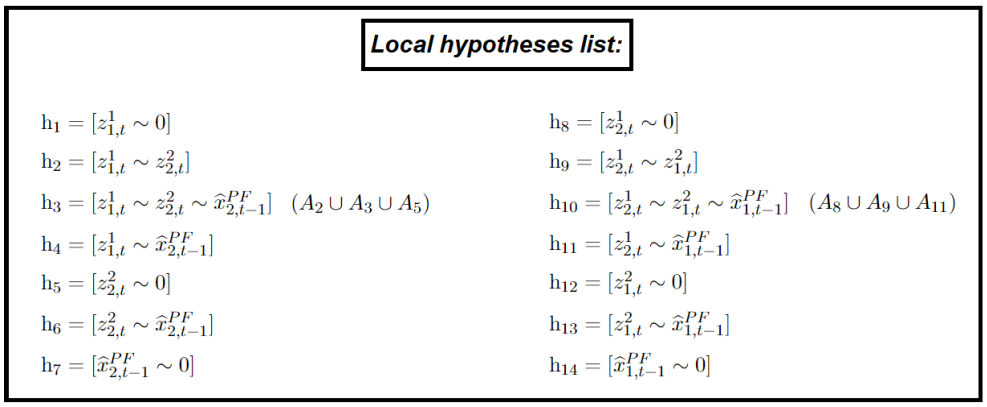

Figure 4.5: Local hypothesis list for the example set out in Fig. 4.4.

As for the local hypotheses score, for those tracks that are the result of the union of several connections, the score will be the sum of the scores of each of the connections that make up the track. Otherwise, if the tracks are directly a connection, the score will be the same as that of the connection [45].

4.2.2.3 Global hypothesis formulation

Finally, in the last stage of the MHT algorithm, the global hypotheses (H_n) are formed by defining sets of independent tracks, i.e. tracks that do not have common measurements. To do this, what is done is to construct a graph in which the nodes represent each of the local hypotheses, and the lines link those nodes that have measurements in common (see figure 4.6). In this way, it is easier to identify which hypotheses are independent of each other and therefore compatible.

Once the graph is constructed, the Maximum Weighted Independent Set (MWIS) [34] method is used to find the set of independent local hypotheses, i.e. the global hypothesis, with the highest score and, therefore, the highest probability. The basic operation of this technique is to select the highest scoring local hypothesis and then search for the next highest scoring local hypothesis that is compatible with the first selected hypothesis. From here, the same recursive approach is followed until there are no observations included in any of the local hypotheses selected during the process. This set of selected independent tracks forms a global hypothesis and its score is the sum of the scores of each of the tracks [34].

Using this procedure, several global hypotheses are formulated and the one with the highest score is selected. Then, each of the tracks that make up the selected global

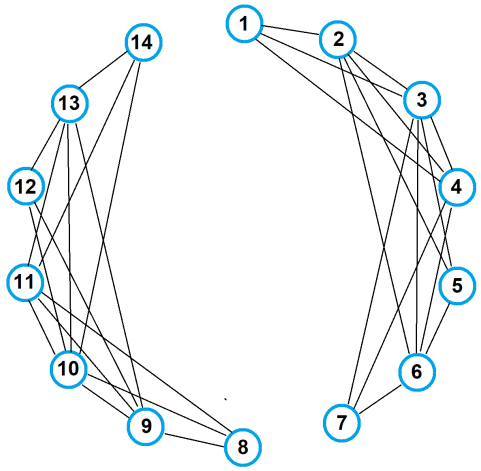

Figure 4.6: Construction of the node graph with the local hypotheses set out in Fig. 4.5.

hypothesis, i.e. each of the different sets of associated measurements, are introduced into a PF, each track to a filter, in order to estimate the state of each of the targets separately. If the track contains one of the target state predictions made in the previous iteration ($\widehat{x}^{PF}_{k,t-1}$), then the track is introduced into the previously created particle filter to continue with the estimation of that target. Conversely, if the track does not contain any of the state predictions, the set of measurements is introduced into a new filter to start the estimation of the state of a new target. This track-to-filter assignment process is shown and implemented next, in algorithm 4.2.

Algorithm 4.2: Track-to-filter assignment algorithm.

Data: Selected global hypothesis $H_n = [h_j, h_i]$, target state predictions $\widehat{x}^{PF}_{k,t-1}$
Result: Correct assignment between the track and the particle filter.

```
for h ∈ H_n do
    if x̂^PF_{1,t-1} ∈ h then
        Assign track h to Particle Filter 1.
        h  →  PF_1
    else if x̂^PF_{2,t-1} ∈ h then
        Assign track h to Particle Filter 2.
        h  →  PF_2
    else
        Assign track h to a new Particle Filter.
        h  →  PF_new
    end
end
```

Finally, local hypotheses that fall outside the global hypothesis are pruned and removed

to save computational cost.

4.3 The Particle Filter (PF)

The Particle Filter (PF) Method [13, 18, 24, 35] is a Monte Carlo technique [55] for the solution of the state estimation problem. This method is based on the estimation of the posterior probability densisty function (pdf) based on all observations and estimates made so far.

The key idea of particle filters is that any pdf can be represented as a set of samples (particles). That is, any pdf can be recreated just by drawing a whole lot of samples from it, so that the density of samples in one area of the state space represents the probability of that region. Each particle has one set of values for the state variables. This method can represent any arbitrary distribution, making it good for non-Gaussian, multi-modal pdfs. Again, the key idea is that one can find an approximate representation of a complex model (any arbitrary pdf) rather than an exact representation of a simplified model (Gaussians).

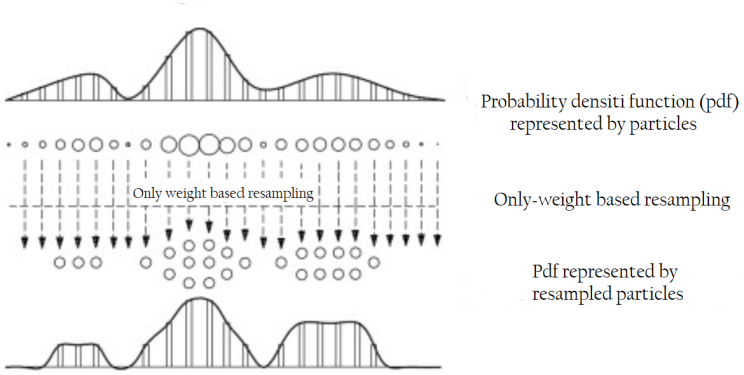

Figure 4.7: Recreation of a non-Gausian pdf from a set of samples (particles) [30].

Particle filters make use of a Bayesian approach towards state estimation in a nonlinear, non-Gaussian scenario [13]. In Bayesian framework, the probability density function of the state is estimated based on all available information. Since this pdf embodies all available statistical information, it may be said to be the complete solution to the estimation problem. PF have this common feature that they approximate the posterior density by means of a set of particles and their associated weights. At least two models, a system model, modeling the evolution of the system with time, and a measurement model are required for efficient estimation. The particles are propagated through the system model and weights are calculated recursively via the noisy observations received. The PF algorithms estimate the posterior density incorporating Monte Carlo sampling and approximation techniques and they are also referred as Sequential Monte Carlo (SMC) filters. As the number of samples becomes very large, this Monte Carlo characterization

becomes an equivalent representation of the posterior probability function and the solution approaches the optimal Bayesian estimate. As stated before, the main advantage of this approach is that it makes no restrictions on the system and measurement models or the distribution of the noise [35].

4.3.1 System model

As explained above, two models are needed for a proper estimation, one that models the evolution of the system over time and one that models the noisy observations.

It is important to underline that the motion model will be the one followed by the particles and that, upon arrival of new observations, i.e. new radar measurements, the probability that each particle is correctly representing the real position of the detected target will be calculated, based on the difference between the observations and the state of each particle, and taking into account the measurement model that models these observations. From there, a weight shall be assigned to each particle and an estimate of the true position shall be calculated taking into account the whole set of particles. However, this will be developed in detail in section 4.3.2.

4.3.1.1 Movement model

In regard to the motion model, there are several alternatives for modelling the behaviour of particles over time. In particular, in this project, the particles move according to a Markov [46] movement, as described next.

First, it has been assumed that a general target is originally located at an unknown position on the considered map according to a uniform distribution, so that the particles are randomly distributed throughout that region. Then, each of the particles follows a movement according to a sequence of random states. The state of a particle at a certain time-stamp is defined by $x \in \mathbb{R}^4 = [x, y, v_x, v_y]^T$ and it evolves according to the following state space equations [48]

$$x_{t+1} = \mathbf{A}x_t + u. \tag{4.4}$$

Here, $\mathbf{A} \in M_{4\times4}(\mathbb{R})$ is the state transition matrix defined as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.5}$$

and u is the driving noise defined as a Normal distributed multivariate variable according to

$$u \sim \mathcal{N}(0^{4\times1}, \mathbf{C}_u). \tag{4.6}$$

For its part, the driving noise covariance matrix $\mathbf{C}_u \in M_{4\times4}(\mathbb{R})$ is defined as

$$\mathbf{C}_u = \sigma_u^2 \mathbf{I}_4, \tag{4.7}$$

where σ_u^2 is the driving noise covariance and $\mathbf{I}_4$ is the 4×4 identity matrix.

4.3.1.2 Measurement model

On the other hand, measurements representing observations of the system are received from time to time. The measurement model indicates how the state x_t is transformed into a measurement z_t by the sensor, depending on its particular characteristics and a noise variable.

Figure 4.8 depicts the basic operation of one radar sensor. As explained in section 3.1, if the radar detects a target, it estimates its range and azimuth (r and ϕ) by comparing both the transmitted and the received signals. However, as expected, these estimates are not fully accurate, due to background noise or the technical limitations of the radar itself, so there is always some error in the measurements. This situation can be modelled by assuming that both measurements follow a normal distribution $\mathcal{N}(\mu_r, \sigma_r)$ and $\mathcal{N}(\mu_\phi, \sigma_\phi)$, whose mean value, μ_r and μ_ϕ, respectively, are the true polar coordinates of the detected target. Then, the noise in the observations, i.e. the error between the true target coordinates and those estimated by the radar, can also be modelled by zero-mean normal distributions with the same variances (σ_r,σ_ϕ): $e_r \sim \mathcal{N}(0, \sigma_r)$ and $e_\phi \sim \mathcal{N}(0, \sigma_\phi)$.

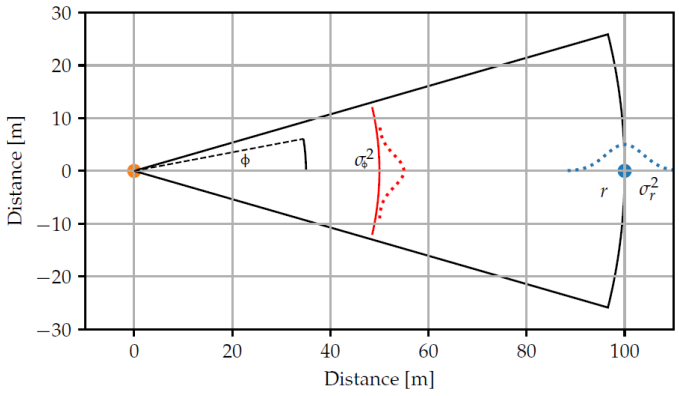

Figure 4.8: Measurement model. The sensors estimate range (r) and azimuth (ϕ). Both parameters estimated as noisy observations with variances σ_r^2 and σ_ϕ^2 [48].

Furthermore, radars may sometimes fail to detect a target due to multiple reasons. Therefore, this must also be taken into account and modelled. For this, the parameter P_d has been defined, which determines the probability of each sensor, at each time step, to detect a surrounding target.

Lastly, each sensor also detects clutter consisting of a random amount M of additional detections. This number is distributed according to the Poisson distribution [48]

$$p(M) = \frac{\lambda^M \mathrm{e}^{-\lambda}}{M!}. \tag{4.8}$$

4.3.2 State estimation via Particle Filter

As explained above, the particle filter uses a particle cloud to approximate the posterior pdf of the target position at time step t given all prior measurements $f(x_t|z_{1:t})$. Then, based on that posterior pdf it is possible to construct the Minimum Mean Squared Error (MMSE) estimate [7]

$$\widehat{x}_t^{MMSE} = \int x_t \, f(x_t|z_{1:t}) \, dx_t. \tag{4.9}$$

Nevertheless, as mentioned, the exact posterior pdf is not known, so what the filter does is to approximate it by distributing a set of J random samples propagating in the state space. Hence, in the PF algorithm, equation 4.9 is approximated by J particles x_t^j and their associated weights w^j as

$$\widehat{x}_t^{MMSE} \approx \widehat{x}_t^{PF} = \sum_{j=1}^{J} x_t^j \, w^j \tag{4.10}$$

The superscript PF refers to the estimate of the particle filter and the weights obey $\sum_j w^j = 1$. This estimation of the target state is performed at every time step. However, state estimation is the last phase of the process, as it requires a prior approximation of the posterior pdf.

4.3.2.1 Main stages of the Particle Filter algorithm

The first step of that process is to initialize the particles by spreading them uniformly around the considered set. Then, the algorithm iterates over three principal stages: prediction, update and resampling (see algorithm 4.3 and figure 4.9).

In the prediction step, the movement model is applied to each particle individually, according to equation 4.4. In this step, the weights are set to be equal.

The update step then computes a global likelihood function for a given particle given the set of all measurements taken at time step t, denoted as $f(z_t|x_t)$, where z_t is a stacked vector comprising all measurements from individual sensors $z_t = [z_t^1, z_t^2, ..., z_t^S]$ (being S the number of sensors). The computation of this global likelihood function is explained in detail later in the section. Finally, the weights are recomputed proportionally to the global likelihood and renormalized [48].

Algorithm 4.3: State estimation via Particle Filter [48].

Data: Measurement vectors z_t^s for all sensors s and all time-steps t
Result: Approximate MMSE estimates $\widehat{x}_t^{MMSE}$ for all time-steps t

`Initialization of the particles:`
for $j \in [1, J]$ **do**
  $w_1^j \leftarrow \frac{1}{J}$;
  $x_{coord}^j \sim \mathcal{U}(x_{min}, x_{max})$;
  $y_{coord}^j \sim \mathcal{U}(y_{min}, y_{max})$;
  $\theta \sim \mathcal{U}(\theta_{min}, \theta_{max})$;
  $v \sim \mathcal{N}(\mu_v, \sigma_v)$;
  $v_x^j = vcos(\theta)$;
  $v_y^j = vsin(\theta)$;
  $x_1^j = [x_{coord}^j, y_{coord}^j, v_x^j, v_y^j]^T$;
end
while $t \leq T$ **do**
  `Prediction:` *Apply the movement model to the set of particles.*
  for $j \in [1, J]$ **do**
    $x_t^j = Ax_{t-1}^j + u$;
    $w_t^j \leftarrow \frac{1}{J}$;
  end
  `Update:` *Compute likelihood function $f(z_t|x_t)$ based on all measurements and update the weights (w_t^j).*
  for $j \in [1, J]$ **do**
    $w_t^j = w_t^j f(z_t|x_t^j)$;
  end
  $w_t^j = \frac{w_t^j}{\sum_j^J w_t^j}$;
  $\widehat{x}_t = \sum_{j=1}^J w_t^j x_t^j$;
  `Resample:` *Draw J new particles (x_t^{j*}) from the old set with replacement. Each old particle is drawn with probability w_t^j:*
  $P(x_t^{j*} = x_t^j) = w_t^j$;
end

$$w_{new}^{j} = \frac{w_{old}^{j} \cdot f(z^{S}|x^{j})}{\sum_{j=1}^{J} w_{old}^{j} \cdot f(z^{S}|x^{j})} \tag{4.11}$$

Is at this point, then, where the current estimate is computed using equation 4.10.

This process has the risk of having very few relevant particles with high weights and all other particles being insignificant, which is called the degeneracy problem [14]. The filter requires a higher number of particles to sample the probability distribution. Hence, resampling is applied. This means that a new set of J particles is generated from the old set by drawing from the old set J times with replacement. Every old particle is assigned its weight as drawing probability. That is, a new set of particles is generated in the following way [48]:

$$\{x^{j*}\}_{j=1}^{N_0} \quad s.t. \quad P(x^{j*} = x^{j}) = w^{j} \quad with \quad N_0 \in [1, J]. \tag{4.12}$$

This process ensures that there is a healthy population of relevant particles.

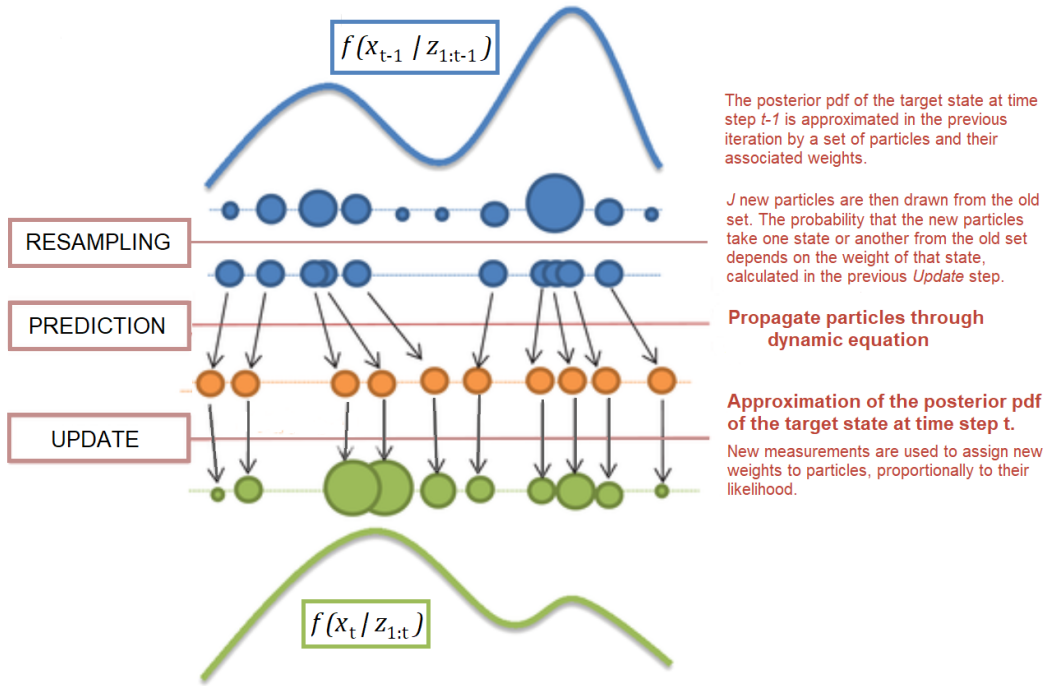

Figure 4.9: Schematic diagram of basic PF operation. The filter operates over three principal stages, prediction, update and resampling.

4.3.2.2 Global likelihood function computation

As mentioned, a global likelihood of a given particle is computed considering all observed measurements. Hence, this means that a global fusion center for all sensors is considered. First of all, the local likelihood function of sensor S having observed measurement $z^{s,m}$ is considered. It is given as [48]

$$f_{local,s}(z^{s,m}|x^j) = f_\phi(z^{s,m}|x^j) \cdot f_r(z^{s,m}|x^j). \tag{4.13}$$

The individual likelihood functions are Normal distributed according to the defined measurement model for range and bearing (see section 4.8):

$$f_\phi(z^{s,m}|x^j) = \frac{1}{2\pi\sigma_\phi^2}\, \mathrm{e}^{\frac{(\phi^j - \phi^{s,m})^2}{2\sigma_\phi^2}}, \qquad f_r(z^{s,m}|x^j) = \frac{1}{2\pi\sigma_r^2}\, \mathrm{e}^{\frac{(r^j - r^{s,m})^2}{2\sigma_r^2}}. \tag{4.14}$$

Then, the overall local likelihood function that takes into account the detection probability P_d as well as the clutter with mean λ can be computed, and its expression ends up at [28, 48]

$$f_s(z^s|x^j) = C\left[(1 - P_d)\lambda + P_d \sum_{m=1}^{M_t^s} \frac{f_{local,s}(z^{s,m}|x^j)}{R2\pi}\right]. \tag{4.15}$$

C is chosen such that $f(z_t^s|x_t^j)$ is a valid pdf, i.e. with an area equal to 1. For its part, the normalization $R2\pi$ derives from the uniformly random placement of the clutter [42]. M_t^s refers to the total number of measurements sensor s has taken at time-step t. Then, if it is assumed that the measurements of different sensors are conditionally independent given x, the global fused likelihood function is computed as [48]

$$f(z^S|x^j) = \prod_{s=1}^{S} f_s(z^s|x^j)\,. \tag{4.16}$$

CHAPTER 5

Simulation setup

As mentioned in the introductory chapter (section 1), the main objective of this project is to test the Particle Filter (PF) and the MHT algorithm, in order to check its behaviour and efficiency in real-time tracking of one or several moving targets, and to evaluate whether its implementation in assisted driving systems for semi-autonomous cars is feasible or not.

In order to carry out these analyses and tests, the setup required for this project must consist of two main blocks, the first one in charge of generating a random trajectory of a target and simulating the corresponding measurements taken by the radar, and the second one responsible for estimating the real trajectory of the target from these noisy observations. Both blocks and their main elements are presented and detailed below [1].

5.1 Measurement generator

To test the designed target tracking algorithms, it is necessary to first generate a random trajectory, emulating a possible real target movement and, then, from this trajectory, simulate the corresponding measurements that would be taken with an FMCW radar. In this sense, therefore, two main elements are required. On the one hand, a Ground Truth generator is required for the first task, while a FMCW radar system simulator is needed for the second one.

[1] For reasons of confidentiality, it has not been possible to explicitly display the Matlab and Python codes used in this project, as they are owned by the Telecommunications Institute of the TU Wien.

5.1.1 Ground Truth generator

As for the simulation of the target trajectory, it is performed by means of a two-dimensional random walk[2], which is implemented in a Matlab code. This Matlab code, which is actually a function, has been designed and developed during the project, as part of it, and requires two main inputs. The first is the number of time steps (N) the user wants the target trajectory to have, while the second corresponds to the initial location of that moving target, in Cartesian coordinates (x_o, y_o). In addition, this ground truth simulator actually requires two additional inputs, which are the variance of the random variable associated with the angle (σ_z^2) and the modulus of the constant velocity of the target (v) (see algorithm 5.1), but these are usually defined within the algorithm itself. In the end, the output of this function is an $N \times 2$ matrix representing a sequence of N random positions, also in Cartesian coordinates, that make up the trajectory of the object.

The main idea of the random walk is that at each iteration a random azimuth (ϕ) angle is generated (see algorithm 5.1), which dictates in which direction the target will move at the next instant of time. This azimuth angle (ϕ) is governed by a random variable z, which follows a zero-mean Gaussian distribution with variance σ_z^2. Next, by applying a linear movement in the vertical and horizontal directions, the coordinates of the new target position are generated. In this case, the target is assumed to have a constant velocity v during this linear displacement. This process is then repeated N times to achieve a sequence of positions that make up the complete trajectory. The following recursive algorithm describes the process in detail.

Algorithm 5.1: Random walk generator algorithm.

Data: Target's initial position $(x0, y0)$, variance of the random variable associated to the angle (σ_z^2), number of time-steps (N) and target's constant velocity (v).

Result: Random trajectory in Cartesian coordinates.

Initialization of the target:

$\phi_0 = 2\pi z$ where $z \sim \mathcal{N}(0, \sigma_z)$;

$x_0 = x0$;

$y_0 = y0$;

Random walk iteration:

for $t \in [0, N-1]$ **do**

  $\phi_{t+1} = \phi_t + \frac{\pi}{3} z$ where $z \sim \mathcal{N}(0, \sigma_z)$;

  $x_{t+1} = x_t + v \cdot cos(\phi_{t+1})$;

  $y_{t+1} = y_t + v \cdot sin(\phi_{t+1})$;

end

As can be seen from algorithm 5.1, the motion model used assumes a constant velocity along the entire trajectory and limits the angle of rotation of the target between two

[2] A random walk is a mathematical object, known as a stochastic or random process, that describes a path that consists of a succession of random steps on some mathematical space such as the integers [38].

consecutive time instants ($\Delta\phi = \phi_{t+1} - \phi_t$) to $\pm\frac{\pi}{3}$. Constant velocity has been assumed because it has been thought to reflect well the movement of objects that a vehicle may encounter while driving, be they other vehicles or pedestrians, all of which tend to show more or less constant movement. As for the rotation angle, the aim was to avoid abrupt changes of direction, as this is not a movement that is typical of vehicles, and so, after several tests, it was concluded that a steering angle of maximum $\pm\frac{\pi}{3}$ was adequate for the scenarios to be studied.

5.1.2 FMCW radar system simulator

Once the target trajectory has been generated, the next step is to perform the corresponding measurements with the FMCW radar. For this purpose, a FMCW radar system simulator, designed and developed entirely by the Institute of Telecommunications of the TU Wien and implemented in Matlab, is available. This simulator works almost in the same way as a real radar system, according to the diagram showed in figure 3.2, and the only things that have to be inputed are the real coordinates of the target, the location of the radar itself and the bandwidth of the TX signal emitted by the radar. In response, the simulator returns noisy measurements of that trajectory, based on the range and azimuth estimation made, in turn, based on the IF signal, as detailed in section 3.2. Both the inputs and outputs of the simulator are in Cartesian coordinates.

However, as in real life, it is possible that during the trajectory the radar may not detect the target at a certain moment, e.g. if the target moves out of the radar's field of view (FOV) or if it moves too far away from the radar, exceeding the maximum measurable range (r_{max}) (see table 5.1). It may also happen that if two objects are very close to each other, the radar detects only one target and not two, due to limitations resulting from the resolution of the radar. In such cases, the radar simulator does not return any value, as no object has been detected. Therefore, in order to identify and localise those missdetections and to convert the data into intelligible material for the tracking algorithms used, a functionality has been added in the post-processing, which consists of filling the output radar detection list with infinities, provided that the length of the output is smaller than the length of the obstacle list input to the measurement simulator. So, at the end of it all, if any missdetection occurs, the data that will be entered to the tracking algorithm will be of the type $[\infty, \infty]$.

The simulated radar system consists of a single transmitting antenna and eight receiving antennas, placed in a row and separated by a distance equal to half the wavelength of the emitted signal ($l = \frac{\lambda_0}{2}$). It is this linear arrangement of the receiving antennas that allows the estimation of the azimuth, as explained in section 3.2.3. The transmitting antenna, for its part, emits a pulse composed of 40 consecutive chirps, in the shape of a sawtooth (see figure 3.1). The carrier frequency (f_0) of each of these chirps is 76 GHz, and their period (T_c) is 10 μs. As commented, the bandwidth of this emitted signal is not fixed, and it can be swept between 75 MHz and 1.5 GHz, to achieve higher or lower range resolution, respectively.

In addition, the radar simulator has an integrated OS-CFAR algorithm, which allows the detection of radar targets embedded in clutter by calculating a threshold for the IF signal spectrum, which reduces detection of ghost targets. The probability of false alarm in this case is set to 10^{-4}.

The FMCW radar system simulator		
Parameter	Value	Units
Number of TX antennas (N_{TX})	1	-
Number of RX antennas (N_{RX})	8	-
Starting frequency of the chirp signal (f_0)	76	GHz
Period of each chirp (T_c)	10	μs
Bandwidth of the chirp (BW)	0.075-1.5	GHz
Number of emitted chirps (N_c)	40	-
Radar's field of view (FOV)	160	º
Maximum measurable range (r_{max})	100	m
Probability of False Alarm OS-CFAR (P_{FA})	10^{-4}	-

Table 5.1: Summary of the main features of the FMCW radar system simulator.

5.2 Target tracker

The second block of the assembly is responsible for real-time target tracking, taking and processing data from the radar, and predicting and estimating the actual target trajectory from these noisy observations, thus achieving a lower error than that present in the radar measurements.

In this sense, there are two possible scenarios. The first one, is when there is only a single target present in the region of interest. In this case, then, the radars will only provide measurements of the same trajectory, so it will be sufficient to apply a particle filter to this data to track the target. However, if the number of targets in the scene is greater than one, the radar system will provide two measurements at every time step, corresponding to two different trajectories, so it will be necessary to use an MHT algorithm, prior to the particle filter, to correctly associate each of the measurements obtained with each of the two trajectories. Therefore, this block consists of two elements, one appropriate for each of these situations.

5.2.1 Particle Filter algorithm

To deal with the individual targets, a PF algorithm implemented in Python is available. A preliminary version of this algorithm was originally designed and implemented by the Institute of Telecommunications at the TU Wien, however, an important part of this project has been to modify, improve and adapt it to the incoming data from the

radar simulator in order to make it functional and suitable for carrying out the required analyses and tests.

The PF algorithm is prepared to process data from several different radars, from one to four, as well as to deal with missdetections and ghost target detection. Furthermore, it is designed only to deal with two-dimensional scenarios, i.e. to track targets in the XY-plane. In order to make it work, the data from each of the sensors, in Cartesian coordinates, and the position of each of the sensors must be input. Then, it is also necessary to set the parameters of the movement and the measurement model (see section 4.3.1), as well as the number of particles the filter will consist of and other parameters that provide statistical reliability to the algorithm such as the detection probability (see section 4.3.2.2). In response, the algorithm outputs an accurate estimate of the target's real position at each time step, allowing real-time tracking of the target.

Particle filter settings	
Parameter	Units
Number of sensors (S)	-
Position of the sensors	m
Number of particles (J)	-
Boundaries of the region of interest ($x_{min}, x_{max}, y_{min}, y_{max}$)	m
Driving noise variance (σ_u^2)	-
State transition matrix (A)	-
Range variance (σ_r^2)	m
Bearing variance (σ_ϕ^2)	rad
Detection probability (P_d)	-

Table 5.2: Particle filter algorithm tuning parameters.

5.2.2 MHT-PF combined algorithm

When more than one target enters the scene, the radars provide multiple detections at each time instant corresponding to each of the moving objects. These measurements are outputed without any order, so an algorithm is needed to sort these detections and associate each of them to each of the targets present in the region of interest. Once the data is associated, the same algorithm is responsible for sending the data belonging to each of the objects separately to a particle filter, in order to be able to perform the state estimation independently.

Therefore, to carry out this entire process, a complex algorithm which combines an MHT algorithm and a PF, has been used. As in the previous case, a draft version of this algorithm was designed and implemented by the Austrian Institute of Technology (AIT) for other projects, but had to be modified, corrected and adapted to make it functional and suitable for the current project. It is also implemented in Python, and consists of a PF with almost the same characteristics as the one used for individual targets, and a

MHT algorithm, placed before the filter, which processes the data coming directly from the radar, associates them, and transmits them in an ordered way to the PF. However, unlike the PF algorithm presented above, this combined MHT-PF algorithm is able to deal with three-dimensional trajectories, although this functionality will not be used as the focus of this project is on two-dimensional tracking.

The algorithm is prepared to work with multiple radars, from one to four, and for it to work, the data from the radars must be entered in spherical coordinates and the positions of the radars must be entered in Cartesian coordinates. In addition, it is also necessary to set the maximum allowable errors (see section 4.2.2.1) for each of the coordinates, in order to be able to perform the gating process, and the characteristic parameters (α, λ) of the score function (see equation 4.3), as well as all the other parameters needed to characterize the PF, as explained in the previous section. As in the previous case, the output of the algorithm at each time step is a set of estimated positions, corresponding to an estimation of the position of each of the targets on scene. In the end, therefore, several sequences of positions will have been constructed, each corresponding to the estimation of the trajectory of each of the moving objects.

MHT-PF settings	
Parameter	Units
Number of sensors (S)	-
Position of the sensors	m
Maximum allowable range error (a)	m
Maximum allowable azimuth error (b)	m
Score factor (α)	-
Score exponential factor (λ)	–
Number of particles (J)	-
Boundaries of the region of interest ($x_{min}, x_{max}, y_{min}, y_{max}$)	m
Driving noise variance (σ_u^2)	-
State transition matrix (A)	-
Range variance (σ_r^2)	m
Bearing variance (σ_ϕ^2)	rad
Detection probability (P_d)	-

Table 5.3: MHT-PF combined algorithm tuning parameters.

CHAPTER 6

Simulation analysis and results

Once the target tracking systems that have been developed throughout this project have been presented and explained, the last step is to test them in different scenarios, analyse the results and evaluate their behaviour in these situations. This chapter will, therefore, present the most important results obtained during this last phase of the project, followed by a detailed analysis of them.

6.1 Measurement preparation

Before proceeding to the analysis of the results, though, it is important to first describe the radar system used to perform the measurements and to show the results of the measurements, as well as to describe the scenarios to be studied.

As for the radar system used, a system consisting of four FMCW radars placed in the shape of a square with a centre at the origin of coordinates, one at each vertex, has been used. Originally, the distance between two consecutive radars, i.e. the length of the sides of the square they form, has been set at 50 m, but it can be changed.

On the other hand, the region of interest through which the target to be tracked will move is precisely the interior of this square, so all radars are oriented towards its centre. This region delimiting the target's movement is also a square centred on the origin of coordinates, but its sides are slightly shorter (40 m). This small margin is left to prevent the simulated trajectory from getting too close to the edges of the radar square, since this is a region where many missdetections occur due to the limitations of the radar's field of view (FOV).

Once the system and the measurement equipment used have been configured and described, the next step is to decide which scenarios will be studied and under which conditions the two designed tracking algorithms will be tested. In this sense, two cases will be studied: the first one, with which the PF algorithm will be tested, will consist of a single target

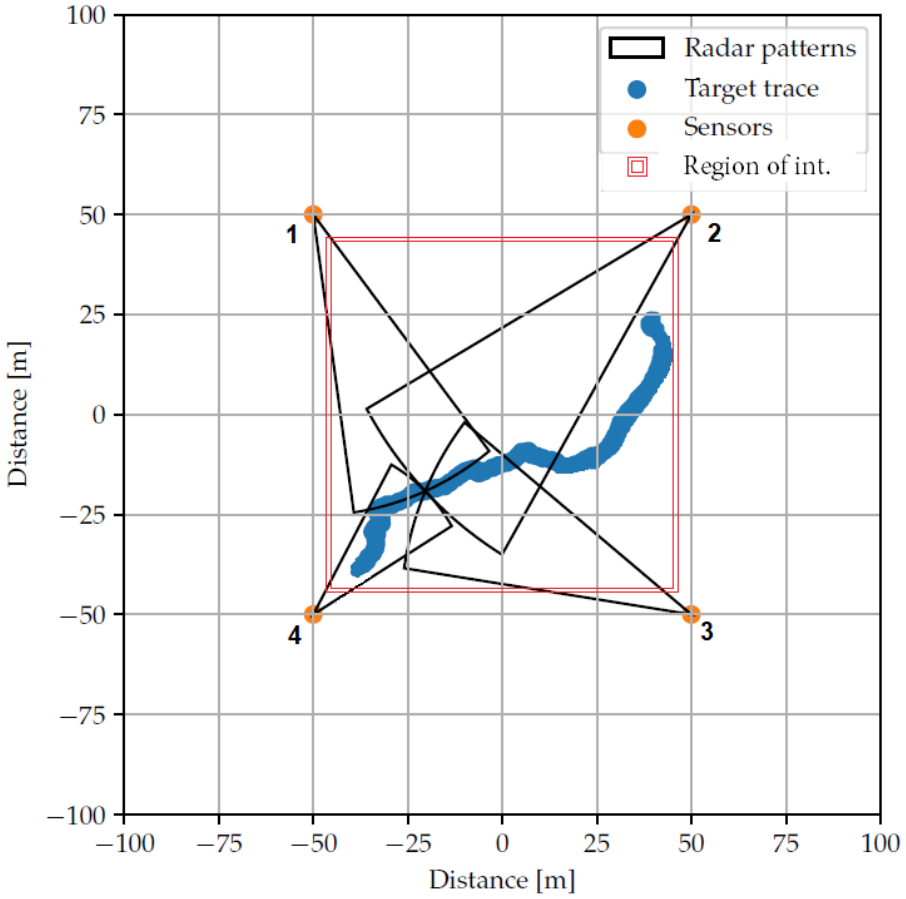

Figure 6.1: Setup topology of the radar system used.

tracking problem; while the second one, for which the combined MHT-PF algorithm will be used, will be a multi-target tracking problem of two simultaneous moving targets. The trajectories of the targets involved in these two scenarios have been simulated with the Ground Truth simulator (section 5.1.1), and the corresponding measurements have been simulated using the FMCW radar simulator, using the setup topology shown in figure 6.1.

In order to have a more representative sample of the results, and with the aim of enriching the analyses and checking how the implemented tracking algorithms behave when the precision and accuracy of the measurements change, a sweep over the bandwidth of the FMCW radar has been performed, and different measurements of the same trajectory have been taken, using the same setup, but varying the bandwidth of the radars that integrate it, from 75 MHz to 1.5 GHz.

Below is an example of the measurements taken of the two simulated scenarios, using a

radar system with a bandwidth of 1 GHz (figures 6.2 and 6.3).

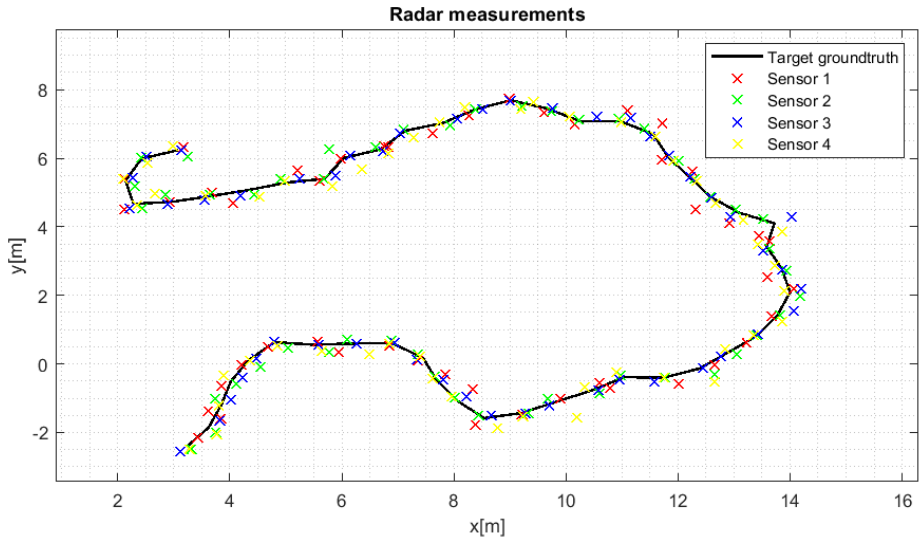

Figure 6.2: Simulated single-object trajectory, and the corresponding measurements (example with a BW of 1 GHz).

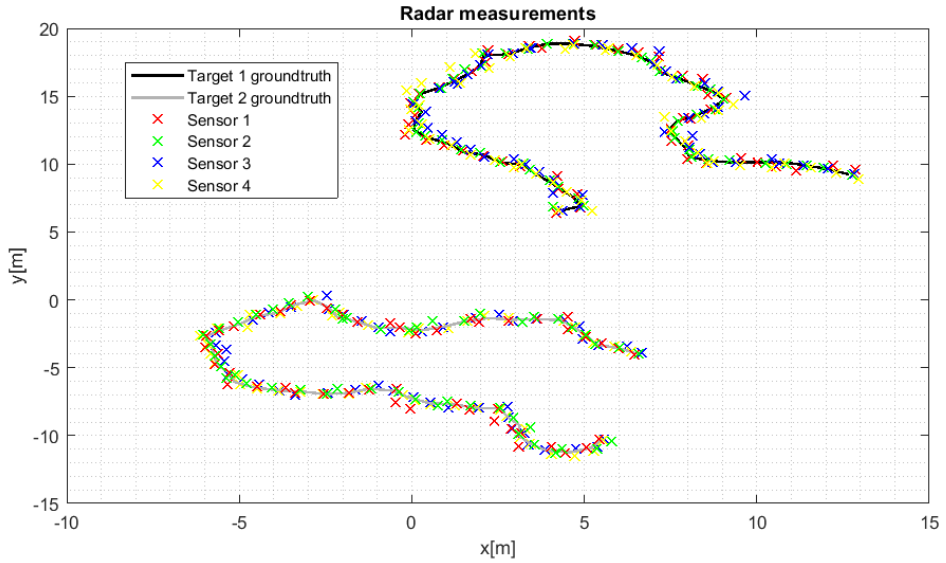

Figure 6.3: Simulated multiple-object trajectories, and the corresponding measurements (example with a BW of 1 GHz).

6.2 Target tracking algorithms evaluation

The following sections will show the results of the tests and analysis performed on the two target tracking systems presented.

As mentioned before, several measurements of the same trajectory have been made using radars with different BW, thus obtaining different accuracies and precisions in the observations, so both algorithms have been tested with all those observations, in order to see their behavior and evolution when working with measurements with different characteristics.

To evaluate these behaviors, some parameters directly related to the estimation error (e_{est}), defined as the Euclidean distance between the estimated position and the real position of the target, will be analyzed and studied.

$$e_{est} = \sqrt{(x_{real} - x_{estimate})^2 + (y_{real} - y_{estimate})^2}. \tag{6.1}$$

On the one hand, one of the objects of analysis will be the probability of success ($P_S(d)$), which is defined as the probability that the estimation error is less than a certain threshold value.

$$P_S(d) = P(e_{est} < d). \tag{6.2}$$

On the other hand, the study will also focus on the quantiles of the estimation error ($Q(d)$), which provide information on the distribution of the errors:

$$Q(x) = e_{est,k} \quad where \quad k = \frac{d}{100} \cdot N, \tag{6.3}$$

where $e_{est,k}$ represents the $k-th$ smallest error, which fixes that the d percent of the total number of errors are below its value, and where N represents that total number of errors.

Finally, to conclude the study, the variance (σ^2_{est}) and mean errors ($\bar{e}_{est}$) of the estimates will be compared with the variance (σ^2_{meas}) and mean errors ($\bar{e}_{meas}$) of the measurements to check the effectiveness of the algorithms. Note that in this case, the term measurement errors refers to the error between the measurements simulated with the FMCW radar simulator and the actual trajectory.

$$e_{meas} = \sqrt{(x_{real} - x_{meas})^2 + (y_{real} - y_{meas})^2}. \tag{6.4}$$

6.2.1 PF algorithm evaluation: Single target tracking

As for the evaluation of the PF algorithm, in addition to repeating the estimation of the same trajectory for different measurements with different characteristics, varying the radar bandwidth, extra elements have been added in the simulations to check the strength of sensor fusion.

What has been done is to add and force missdetections in the measurements [1], thus reducing the information about the environment available to the Particle Filter and making the estimation of the target state more difficult. These false detections are randomly distributed over the measurement vectors provided by the radars ($z^s_{1:t}$), and are regulated by a missdetection ratio (r_m), which sets the percentage of missdetections present in the observations. Furthermore, the number of working sensors has also been modified, in order to check the power of sensor fusion. In such cases, the sensors have been placed according to the following rule. If a single sensor is used, it is placed in the centre of the field. For 2, 3 or 4 sensors, they are placed in a circle with equal angular spacing between them. Therefore, simulations have been carried out with different missdetections ratios (r_m) and different number of sensors (S).

The PF configuration parameters used for these simulations are as follows.

Particle filter settings	
Parameter	Values
Number of sensors (S)	4
Number of particles (J)	1000
Driving noise variance (σ_u^2)	1/9
State transition matrix (A)	*as described in equation 4.5*
Range variance (σ_r^2)	*depending on the measurement characteristics*
Bearing variance (σ_ϕ^2)	*depending on the measurement characteristics*
Detection probability (P_d)	0.95

Table 6.1: Particle filter algorithm tuning parameters used for the simulations.

Finally, the results of the multiple tests performed are summarized in the following figures.

On the one hand, figures 6.4 and 6.5 show the success probability of the Particle Filter for each bandwidth and for different r_m and different number of sensors. On the other hand, figures 6.6 and 6.7 show the quantiles of the estimation errors and, finally, figures 6.8 and 6.9 show a comparison between the measurement errors and the estimation errors.

As can be seen, the results obtained are quite similar for the different BW values. This means that, although the characteristics of the measurements fed into the filter are different (see figures 6.8 and 6.9), the estimates made by the PF are consistent and their quality does not seem to be affected too much by changes in the accuracy and precision of these measurements.

However, there are more noticeable differences when the missdetection ratio and the number of sensors are varied.

[1] The original measurements, i.e. measurements without any modification, have a detection error rate of less than 1%, due to the characteristics of the radars themselves.

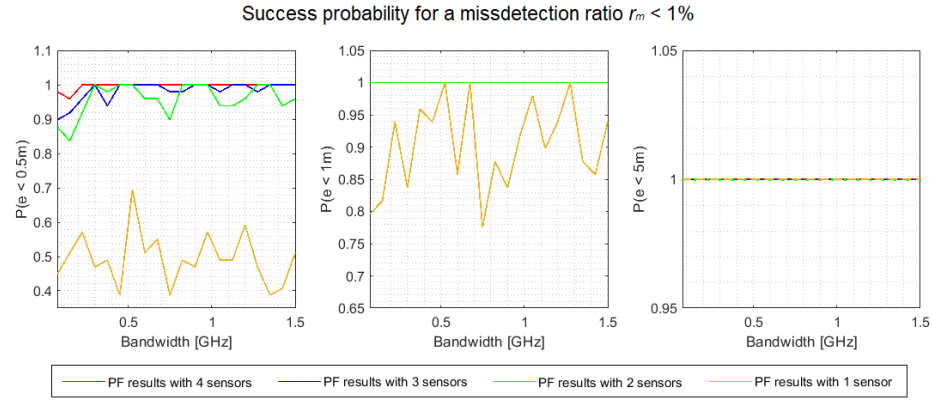

Figure 6.4: PF success probability for $r_m < 1\%$ and different number of sensors.

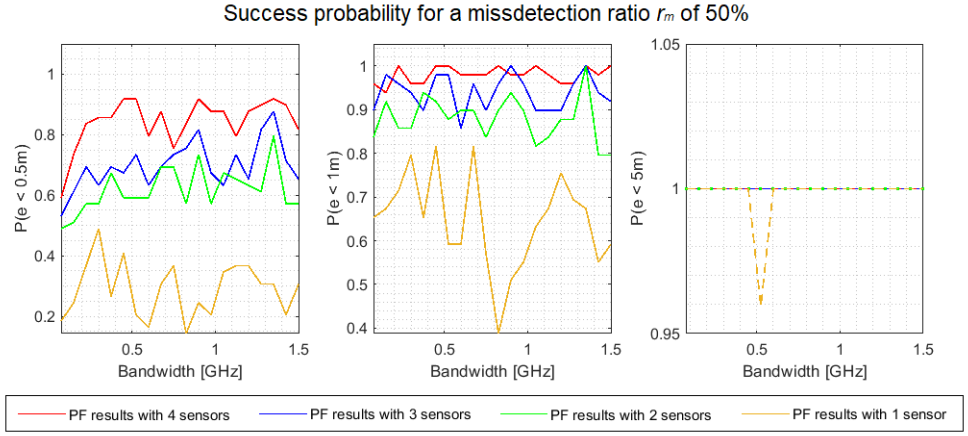

Figure 6.5: PF success probability for $r_m = 50\%$ and different number of sensors.

On the one hand, one can notice that, as is logical, the greater the number of sensors used, the greater the precision and accuracy of the estimates. This happens because the more sensors are active, the greater the amount of information available about the environment and, therefore, the greater the amount of data available to the filter, the higher the quality of its estimates. However, it is observed that the improvement in estimates corresponding to adding an extra sensor becomes smaller and smaller as the number of sensors increases. That is, the difference between using 1 or 2 sensors is much greater than the difference between going from 3 to 4 sensors. In addition, it is observed that under normal conditions, i.e. when $r_m < 1\%$, when a minimum of 2 sensors are used, the estimates made by the particulate filter are more than acceptable, with an average error of less than 30cm (figure 6.8) and 95% of the estimates with an error of less than 70cm (figure 6.6). In contrast, it can be observed that when only 1 radar is available, the quality of the estimates decreases considerably, reaching an average error of 60cm (figure 6.8), and with only a 0.5 probability that the estimation errors are less

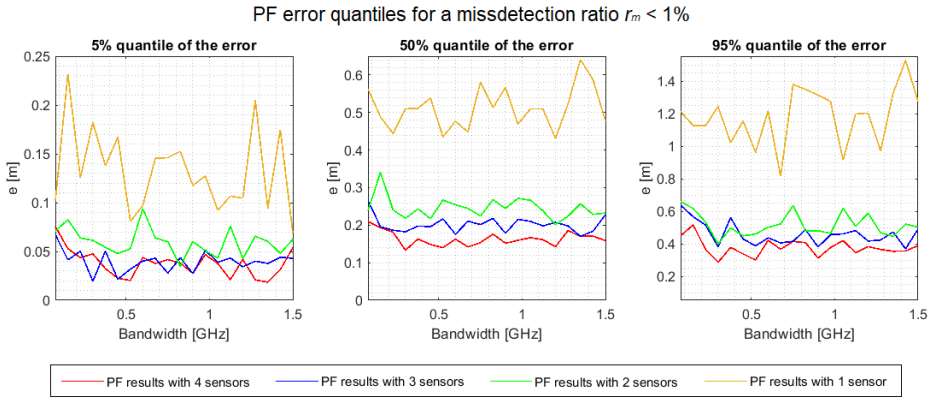

Figure 6.6: PF estimation error quantiles for for $r_m < 1\%$ and different number of sensors.

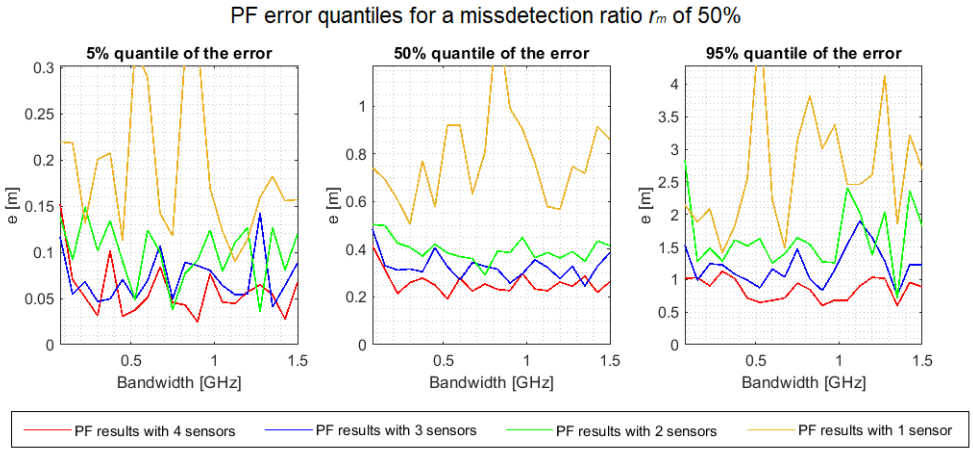

Figure 6.7: PF estimation error quantiles for $r_m = 50\%$ and different number of sensors.

than 50cm (figure 6.4). However, it is found that in no case is the target track lost, as at no time are the estimation errors greater than 5m (figure 6.4).

On the other hand, it is observed that, evidently, when introducing data in the PF with a r_m of 50%, the quality of the estimates decreases. However, it can be seen that, in spite of this high missdetection ratio, when working with data from several radars, the decrease in the quality of the estimates can be controlled quite well, and the results obtained are not much worse than those obtained under normal conditions. In this sense, it is observed, for example, that when working with 4 radars whose measurements have 50% detection errors, the average estimation error remains below 40cm (figure 6.9) and the probability of these errors being less than 1m is practically 1 (figure 6.5). These results are surprisingly good, considering that practically half of the measurements are been ineffective and useless. On the contrary, as the number of sensors decreases, this

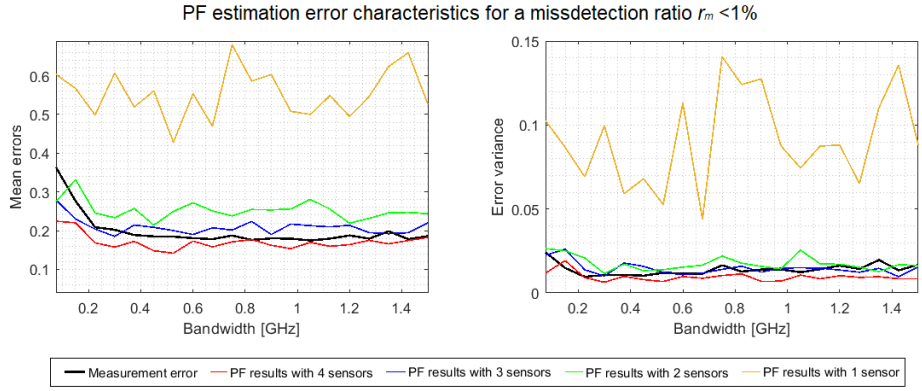

Figure 6.8: PF estimation error characteristics for $r_m < 1\%$ and different number of sensors.

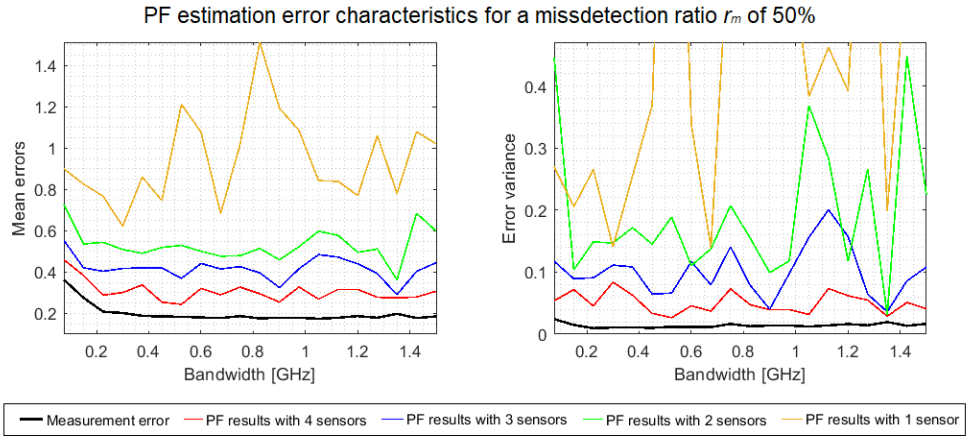

Figure 6.9: PF estimation error characteristics for $r_m = 50\%$ and different number of sensors.

lack of information becomes more and more noticeable, which has a considerable impact on the precision and accuracy of the estimates. For example, if we only have a single radar and its missdetection rate is 50%, the average error of the estimates grows up to 1m (figure 6.9). It is here, then, where we can appreciate the power and effectiveness of sensor fusion techniques. However, although it is evident that increasing the number of radars increases the quality of the estimates, it can be seen that even with a single radar, and with practically half of its observations containing null information of the target, a correct tracking of the target can be performed, with little accuracy, but at no time the track of the object is lost, since the estimation errors in the vast majority of cases remain below 4m (figure 6.7). This fact, then, highlights the great potential and usefulness of PFs as object tracking algorithms.

In any case, it is verified that the particle filter works correctly and that its estimates are more than acceptable. In addition, it is verified that from the fusion of 4 radars, and under normal missdetection conditions, the algorithm already produces more precise (lower error variance) and accurate (lower mean error) estimates than the observations themselves, which was one of the main objectives to be achieved (figure 6.8).

6.2.2 MHT-PF combined algorithm evaluation: Multiple target tracking

Regarding the evaluation of the combined MHT-PF algorithm, as in the previous section, several simulations have been performed using different measurement sets obtained with radar systems with different bandwidths. However, in this case, no forced missdetection[2] has been added to the observation vectors, so the simulations have been performed only with the raw data obtained directly from the radar system, without any manipulation. This is because, given that the particle filter used to perform the state estimation in this case is the same as the one tested in the previous section, the aim of this section is to check the correct functioning of the data association algorithm, the MHT, and to verify that its implementation does not affect the quality of the estimates.

In order to perform these simulations, the MHT-PF combined algorithm has been configured as indicated in table 6.2. It is important to remember that this algorithm uses exactly the same filter as the one tested in the previous section, but in this case an MHT algorithm has been incorporated to perform the above data association.

MHT-PF algorithm settings	
Parameter	Values
Number of sensors (S)	4
Maximum allowable range error (a)	5m
Maximum allowable azimuth error (b)	15^{o}
Score factor (α)	1
Score exponential factor (λ)	0.69
Number of particles (J)	1000
Driving noise variance (σ_u^2)	1/9
State transition matrix (A)	*as described in equation 4.5*
Range variance (σ_r^2)	*depending on the measurement characteristics*
Bearing variance (σ_ϕ^2)	*depending on the measurement characteristics*
Detection probability (P_d)	0.95

Table 6.2: MHT-PF algorithm tuning parameters used for the simulations.

[2]It is important to bear in mind that, as with any radar measurement system, some missdetections have occurred naturally when making observations on the trajectories, specially when multiple there are multiple targets in scene. The missdetection rate of the radar system was, in this case, less than 10%.

The most relevant results obtained from the multiple simulations are summarised in the following figures. For these simulations, a setup consisting of 4 FMCW radars, as shown in figure 6.1, has been used.

As before, figure 6.10 shows the probability of estimation success for the two tracked targets, while figure 6.11 provides information on how the estimation errors are distributed. Finally, figure 6.12 shows a comparison between estimation errors and measurement errors.

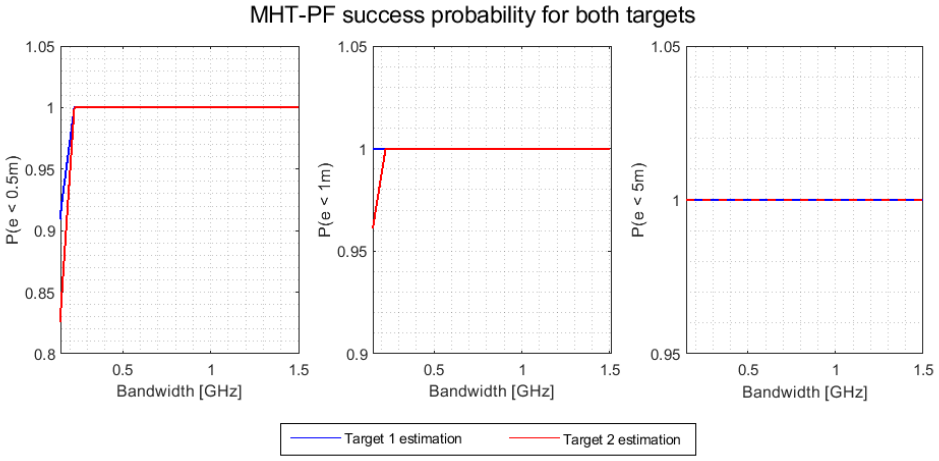

Figure 6.10: MHT-PF success probability for both targets.

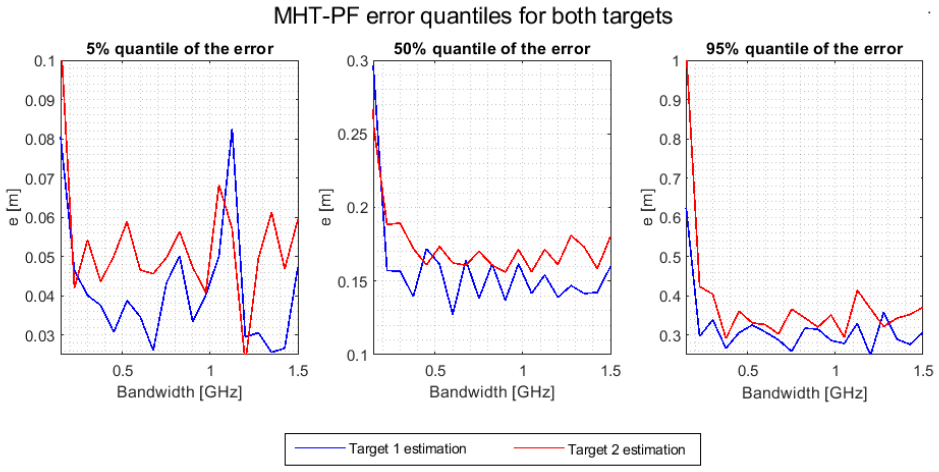

Figure 6.11: MHT-PF estimation error quantiles for both targets.

First of all, it is important to note that, at first sight, the trajectory estimation errors of both targets turn out to be quite similar in terms of accuracy and precision. However, if we look at the quality of the measurements corresponding to each of the targets, we

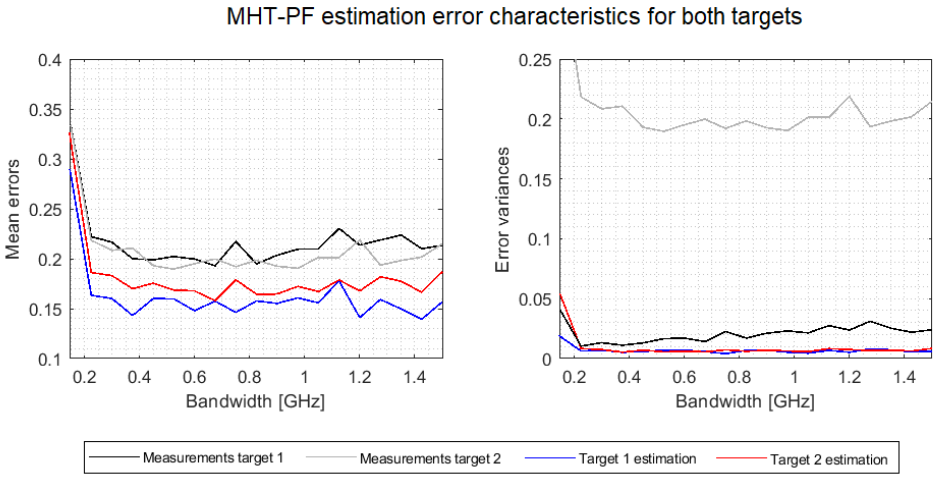

Figure 6.12: PF estimation error characteristics for both targets.

observe that the variance of the measurement errors for the case of target 2 is much higher than that corresponding to target 1 (figure 6.12). This confirms, therefore, the idea that the PF has a very robust and stable behavior, and that it is not too much affected by variations in the characteristics of the measurements used.

As before, the results prove to be very good, and improve the precision and accuracy of the measurements used to make the estimates, as can bee seen in figure 6.12, where the mean and the variance of the estimation errors are quite below the mean and variance of the measurement errors.

In general, therefore, it seems that the results obtained with the MHT-PF combined algorithm are quite similar to the results obtained with the PF algorithm, with almost no loss of accuracy and precision in the estimates. This means, then, that the function of the MHT implemented before the PF fulfills perfectly, since the observations are perfectly associated before passing through the filter, and that this association process does not reduce the quality of the estimates.

CHAPTER 7

Conclusions and future work

From the outset, the aim of this project has been to modify, adapt, improve and evaluate the two tracking algorithms presented. Both algorithms have been correctly implemented and evaluated thanks to the use of a FMCW radar simulator, by means of which real measurements of the trajectory of several targets have been simulated and subsequently introduced into the tracking algorithms to carry out the analyses.

In this sense, the results obtained with both algorithms have been found to be more than acceptable, so they should be taken into consideration as a starting point when developing complex target tracking systems, especially in the automotive field. Obviously, the algorithms shown and tested in this project are a preliminary version with certain limitations, so they are not ready to be directly implemented in any advanced driving assistance system (ADAS). However, it has been proven that they can become a basis on which to start building, researching and developing.

On the other hand, it is clear that both algorithms need to be tested in many more situations than those studied in this project, in order to draw clearer and firmer conclusions, but due to lack of time this has not been possible, so this task will be left to future researchers. For example, it would be interesting to study how both algorithms behave when the errors in the measurements are significantly larger than in the studied situations, and to see what is the limit, if any, beyond which the algorithm is no longer able to track the target. This has not been possible to evaluate in this project as the FMCW radar simulator used did not provide measurements with higher errors.

Another future avenue of research for the MHT-PF combined algorithm would be to increase the number of targets in the scene, and look for the limit. In addition, it would also be interesting to see if, as the number of targets increases, the quality of the estimates decreases.

It would also be of great interest to analyze what happens when there are interferences, of different levels, that alter and hinder the taking of measurements by the radars,

thus causing an increase in missdetections and the appearance of ghost detections, i.e., detections that do not correspond to any real object present in the scene. This scenario is of particular interest because if more and more cars are to be equipped with this type of system to obtain information from the environment, the probability of mutual interference between the radars of the different cars will increase.

Finally, after all these previous studies, the last phase would consist of testing both algorithms in a real situation, i.e., taking measurements of a moving object (e.g., a car), introducing them into the algorithms and estimating the position of the detected objects in real time. The most important thing here is to evaluate whether the processing time of the system (considering both the process of taking measurements and the process of estimating the state of the target) is short enough to be used for real-time tracking functions.

Therefore, as has been seen, there is still a lot of research to be done in this field, so the idea after this project is to continue the research, run more simulations and evaluate many more scenarios, and thus obtain more solid and conclusive results. So, what is in mind is to present all these analyses and more complete results in a future publication to be released soon.

Bibliography

[1] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. ISSN: 0021-9223. DOI: 10.1115/1.3662552. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf. URL: https://doi.org/10.1115/1.3662552.

[2] Harold W. Sorenson. "Least-squares estimation: from Gauss to Kalman". In: *IEEE Spectrum* 7 (1970), pp. 63–68.

[3] Peter S. Maybeck. *Stochastic Models, Estimation and Control: Volume 1*. New York: Academic Press, 1979.

[4] D. Reid. "An algorithm for tracking multiple targets". In: *IEEE Transactions on Automatic Control* 24.6 (1979), pp. 843–854. DOI: 10.1109/TAC.1979.1102177.

[5] Chaw-Bing Chang and JA TABACZYNSKI. "Application of state estimation to target tracking". In: *Automatic Control, IEEE Transactions on* 29 (Mar. 1984), pp. 98–109. DOI: 10.1109/TAC.1984.1103466.

[6] Antoni Elias and Jordi Puga. "Técnicas de detección adaptativa con tasa de falsa alarma constante". In: *Palestra Universitària* 4 (June 1989), pp. 247–303. DOI: http://hdl.handle.net/2117/87253.

[7] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. USA: Prentice-Hall, Inc., 1993. ISBN: 0133457117.

[8] A.F.M. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". English. In: *IEE Proceedings F (Radar and Signal Processing)* 140 (2 Apr. 1993), 107–113(6). ISSN: 0956-375X. URL: https://digital-library.theiet.org/content/journals/10.1049/ip-f-2.1993.0015.

[9] Greg Welch and Gary Bishop. *An Introduction to the Kalman Filter*. 1995.

[10] Samuel S Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Boston: Artech House, 1999. DOI: 10.1036/0071444742. URL: https://searchworks.stanford.edu/view/4239490.

[11] J Carpenter, Peter Clifford, and Paul Fearnhead. "An improved particle filter for non-linear problems". English. In: *IEE Proceedings Radar Sonar and Navigation* 146.1 (Feb. 1999), pp. 2–7. ISSN: 1350-2395. DOI: 10.1049/ip-rsn:19990255.

[12] Arnaud Doucet, N. Freitas, and N. J Gordon. *Sequential Monte-Carlo Methods in Practice*. Vol. 1. Jan. 2001. ISBN: 978-1-4419-2887-0. DOI: 10.1007/978-1-4757-3437-9.

[13] Simon Maskell and N. Gordon. "A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking". In: vol. Workshop. Nov. 2001, 2/1–2/15 vol.2. DOI: 10.1049/ic:20010246.

[14] M. Arulampalam et al. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking". In: *Signal Processing, IEEE Transactions on* 50 (Mar. 2002), pp. 174–188. DOI: 10.1109/78.978374.

[15] Merrill I. Skolnik. "36 - Radar". In: *Reference Data for Engineers (Ninth Edition)*. Ed. by Wendy M. Middleton and Mac E. Van Valkenburg. Ninth Edition. Woburn: Newnes, 2002, pp. 36-1-36–22. ISBN: 978-0-7506-7291-7. DOI: https://doi.org/10.1016/B978-075067291-7/50038-8. URL: https://www.sciencedirect.com/science/article/pii/B9780750672917500388.

[16] Zhe Chen. "Bayesian Filtering: From Kalman Filters to Particle Filters, and Beyond". In: *Statistics* 182 (Jan. 2003). DOI: 10.1080/02331880309257.

[17] Pavlina Konstantinova, Alexander Udvarev, and Tzvetan Semerdjiev. "A study of a target tracking algorithm using global nearest neighbor approach". In: (Jan. 2003). DOI: 10.1145/973620.973668.

[18] Sumakanth Kambhampati et al. "Particle filtering for target tracking". In: (Jan. 2004).

[19] Jari. Kaipio. *Statistical and computational inverse problems / Jari Kaipio and Erkki Somersalo*. eng. Applied mathematical sciences ; v. 160. New York: Springer, 2005. ISBN: 0387220739.

[20] Hsueh-Jyh Li and Yean-Woei Kiang. "10 - Radar and Inverse Scattering". In: *The Electrical Engineering Handbook*. Ed. by WAI-KAI CHEN. Burlington: Academic Press, 2005, pp. 671–690. ISBN: 978-0-12-170960-0. DOI: https://doi.org/10.1016/B978-012170960-0/50047-5. URL: https://www.sciencedirect.com/science/article/pii/B9780121709600500475.

[21] Mark A. Richards. *Fundamentals of Radar Signal Processing*. <country>US</country>: McGraw-Hill Professional, 2005, pp. -1. ISBN: 0071444742. DOI: 10.1036/0071444742. eprint: https://mhebooklibrary.com/doi/pdf/10.1036/0071444742. URL: https://mhebooklibrary.com/doi/book/10.1036/0071444742.

[22] Mohamed Jaward et al. "A data association algorithm for multiple object tracking in video sequences". In: vol. 2006. Feb. 2006, pp. 129–136. ISBN: 0-86341-608-X. DOI: 10.1049/ic:20060565.

[23] Olivier Cappe, S.J. Godsill, and Eric Moulines. "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo". In: *Proceedings of the IEEE* 95 (June 2007), pp. 899–924. DOI: 10.1109/JPROC.2007.893250.

[24] Arnaud Doucet and Adam Johansen. “A Tutorial on Particle Filtering and Smoothing : Fifteen years later”. In: (Jan. 2008).

[25] Paul Fearnhead. “Computational methods for complex stochastic systems: A review of some alternatives to MCMC”. In: *Statistics and Computing* 18 (June 2008), pp. 151–171. DOI: 10.1007/s11222-007-9045-8.

[26] O Fudym et al. “Bayesian approach for thermal diffusivity mapping from infrared images with spatially random heat pulse heating”. In: *Journal of Physics: Conference Series* 135 (Nov. 2008), p. 012042. DOI: 10.1088/1742-6596/135/1/012042. URL: https://doi.org/10.1088/1742-6596/135/1/012042.

[27] Mohamed Adnane Habib et al. “CA-CFAR detection performance of radar targets embedded in "non centered chi-2 gamma" clutter”. In: *Progress in Electromagnetics Research-pier* 88 (2008), pp. 135–148.

[28] Subhash Challa et al. *Fundamentals of Object Tracking*. Cambridge University Press, 2011. DOI: 10.1017/CBO9780511975837.

[29] Hermann Rohling. “Ordered statistic CFAR technique - an overview”. In: *2011 12th International Radar Symposium (IRS)*. 2011, pp. 631–638.

[30] Tiancheng Li, Tariq Pervez Sattar, and Shudong Sun. “Deterministic resampling: Unbiased sampling to avoid sample impoverishment in particle filters”. In: *Signal Processing* 92.7 (2012), pp. 1637–1645. ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2011.12.019. URL: https://www.sciencedirect.com/science/article/pii/S0165168411004634.

[31] Fatih Pektas and Murat Sabuncu. “A score-based track initiation procedure to eliminate false tracks in the presence of sea clutter”. In: *IEEE National Radar Conference - Proceedings* (May 2012). DOI: 10.1109/RADAR.2012.6212252.

[32] H. Meinel and Juergen Dickmann. “Automotive Radar: From Its Origin to Future Directions”. In: *Microwave Journal* vol.56 (Sept. 2013), pp. 24–40.

[33] *¿Cómo nos movemos?* Nov. 2014. URL: https://www.miteco.gob.es/es/ceneam/carpeta-informativa-del-ceneam/novedades/como-nos-movemos.aspx.

[34] Chanho Kim et al. “Multiple Hypothesis Tracking Revisited”. In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 4696–4704.

[35] Rooji Jinan and Tara Raveendran. “Particle Filters for Multiple Target Tracking”. In: *Procedia Technology* 24 (2016). International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015), pp. 980–987. ISSN: 2212-0173. DOI: https://doi.org/10.1016/j.protcy.2016.05.215. URL: https://www.sciencedirect.com/science/article/pii/S2212017316303061.

[36] Yalin Gürcan and Alexander Yarovoy. “Super-resolution algorithm for joint range-azimuth-Doppler estimation in automotive radars”. In: *2017 European Radar Conference (EURAD)*. 2017, pp. 73–76. DOI: 10.23919/EURAD.2017.8249150.

[37] José Raúl Machado-Fernández and Jesús de la Concepción Bacallao-Vida. "Estimación del Multiplicador Óptimo del Umbral CA-CFAR en Clutter Pareto de Parámetros Conocidos". In: *Entramado* 13.1 (ene. 2017), pp. 252–261. DOI: 10.18041/entramado.2017v13n1.25104. URL: https://revistas.unilibre.edu.co/index.php/entramado/article/view/1125.

[38] *Random walk (implementation in python)*. Oct. 2017. URL: https://www.geeksforgeeks.org/random-walk-implementation-python/.

[39] Yunyi Jia, Longxiang Guo, and Xin Wang. "4 - Real-time control systems". In: *Transportation Cyber-Physical Systems*. Ed. by Lipika Deka and Mashrur Chowdhury. Elsevier, 2018, pp. 81–113. ISBN: 978-0-12-814295-0. DOI: https://doi.org/10.1016/B978-0-12-814295-0.00004-6. URL: https://www.sciencedirect.com/science/article/pii/B9780128142950000046.

[40] Christina Katzlberger. "Object Detection with Automotive Radar Sensors using CFAR Algorithms". In: *Bachelor Thesis to obtain the academic degree of Bachelor of Science in the Bachelor's Program Elektroni und Informationstechnik*. 2018, p. 10.

[41] Youngjoo Kim and Hyochoong Bang. "Introduction to Kalman Filter and Its Applications". In: *Introduction and Implementations of the Kalman Filter*. Ed. by Felix Govaers. Rijeka: IntechOpen, 2018. Chap. 2. DOI: 10.5772/intechopen.80600. URL: https://doi.org/10.5772/intechopen.80600.

[42] Giuseppe Papa et al. "Distributed Bernoulli Filtering Using Likelihood Consensus". In: *IEEE Transactions on Signal Processing* PP (Dec. 2018), pp. 1–1. DOI: 10.1109/TSIPN.2018.2881718.

[43] Eduardo Santos DIaz. *Radar target-tracking and measurement-origin uncertainty*. Mar. 2018. URL: https://macsphere.mcmaster.ca/bitstream/11375/23033/2/Santos$%5C%$20Diaz_Eduardo_201804_PhD.pdf.

[44] Kwangjin Yoon, Young-min Song, and Moongu Jeon. "Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views". In: *IET Image Processing* 12.7 (2018), pp. 1175–1184. DOI: https://doi.org/10.1049/iet-ipr.2017.1244. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-ipr.2017.1244. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-ipr.2017.1244.

[45] Francesco d'Apolito, Christoph Sulzbachner, and Felix Bruckmueller. "A Multiple Hypothesis Tracking Approach to Collision Detection for Unmanned Aerial Vehicles". In: *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. 2019, pp. 1–5. DOI: 10.1109/DASC43569.2019.9081705.

[46] Akira Endo, Edwin van Leeuwen, and Marc Baguelin. "Introduction to particle Markov-chain Monte Carlo for disease dynamics modellers". In: *Epidemics* 29 (2019), p. 100363. ISSN: 1755-4365. DOI: https://doi.org/10.1016/j.epidem.2019.100363. URL: https://www.sciencedirect.com/science/article/pii/S1755436519300301.

[47] Ziwei Wang, Jinping Sun, and Naiyu Wang. "A Novel Multiple Hypothesis Tracking Algorithm Integrated with Detection Processing". In: *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*. 2019, pp. 1–5. DOI: 10.1109/ICSIDP47821.2019.9173205.

[48] Mehdi Ashury et al. "Accuracy Requirements For Cooperative Radar With Sensor Fusion". In: Mar. 2020, pp. 1–5. DOI: 10.23919/EuCAP48036.2020.9135378.

[49] Redacción Inese. *La Tecnología de los vehículos salva vidas: El 90% de accidentes es por errores humanos*. Sept. 2021. URL: https://www.inese.es/la-tecnologia-de-los-vehiculos-salva-vidas-el-90-de-accidentes-es-por-errores-humanos/.

[50] *Intro to mmwave sensing : FMCW radars - module 1 : Range estimation*. June 2021. URL: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-1-range-estimation.

[51] Y. Li et al. "An adaptive interference cancellation method for automotive FMCW radar based on waveform optimization". In: Jan. 2021, pp. 666–670. DOI: 10.1049/icp.2021.0741.

[52] Jiankun Ling. "Target Tracking Using Kalman Filter Based Algorithms". In: *Journal of Physics: Conference Series* 2078.1 (Nov. 2021), p. 012020. DOI: 10.1088/1742-6596/2078/1/012020. URL: https://doi.org/10.1088/1742-6596/2078/1/012020.

[53] *Intro to mmwave sensing : FMCW radars - module 3 : Velocity estimation*. June 2022. URL: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-3-velocity-estimation.

[54] *Intro to mmwave sensing : FMCW radars - module 5 : Angle estimation*. June 2022. URL: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-5-angle-estimation.

[55] Will Kenton. *What is a Monte Carlo Simulation?* Feb. 2022. URL: https://www.investopedia.com/terms/m/montecarlosimulation.asp.