Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

AUTOMOTIVE PER UNA MOBILITÀ INTELLIGENTE

Ciclo 34

Settore Concorsuale: 09/F2 - TELECOMUNICAZIONI

Settore Scientifico Disciplinare: ING-INF/03 - TELECOMUNICAZIONI

AUTOMOTIVE RADARS: TOWARDS SMARTER VEHICLES

Presentata da: Alessandro Davoli

Coordinatore Dottorato

Nicolò Cavina

Supervisore

Giorgio Matteo Vitetta

Co-supervisore

Paolo Pavan

Esame finale anno 2022

It's not important how hard you strive along the way,
it's having something inside that
allows you to make it through
M. Jordan

Abstract

Radars are expected to become the main sensors in various civilian applications, especially for autonomous driving. Their success is mainly due to the availability of low cost integrated devices, equipped with compact antenna arrays, and computationally efficient signal processing techniques. This thesis focuses on the study and the development of different deterministic and learning based techniques for colocated *multiple-input multiple-output* (MIMO) radars. In particular, after providing an overview on the architecture of these devices, the problem of detecting and estimating multiple targets in *stepped frequency continuous wave* (SFCW) MIMO radar systems is investigated and different deterministic techniques solving it are proposed. Moreover, novel solutions, based on an approximate maximum likelihood approach, are developed. The accuracy achieved by all the considered algorithms is assessed on the basis of the raw data acquired from low power wideband radar devices. The results demonstrate that the developed algorithms achieve reasonable accuracies, but at the price of different computational efforts. Another important technical problem investigated in this thesis concerns the exploitation of machine learning and deep learning techniques in the field of colocated MIMO radars. In this thesis, after providing a comprehensive overview of the machine learning and deep learning techniques currently being considered for use in MIMO radar systems, their performance in two different applications is assessed on the basis of synthetically generated and experimental datasets acquired through a commercial *frequency modulated continuous wave* (FMCW) MIMO radar. Finally, the application of colocated MIMO radars to autonomous driving in smart agriculture is illustrated.

Acknowledgements

Before going deeper in the detail of this thesis work, I would like to thank all the people that have played a fundamental role in my PhD studies.

In particular, I am very glad to thank my Supervisor Prof. Giorgio Matteo Vitetta and equally my Co-Supervisor Prof. Paolo Pavan for the opportunity they gave me to be involved in the doctorate course on automotive for an intelligent mobility. Especially, I am very grateful to my Supervisor for his constant supervision and help in all my research works, for his precious daily advices, for the possibility he gave me to participate at different PhD schools and for his faith in me.

I would like to thank CNH Industrial Italy and Belgium for the possibility to use their radar sensors and for their investments on this research area.

I would like to thank also Professors Michele Rossi and Francesco Fioranelli for accepting to revise this work. In particular, I am proud to have the opportunity to personally thank Michele Rossi for organising the Summer School of Information Engineering in a wonderful place like Bressanone; this helped me to investigate various machine learning topics and to meet a lot of brilliant students and researchers.

I am very grateful to all the people studying in the Telecommunication Laboratory of the University of Modena and Reggio Emilia and to various colleagues of University of Bologna; I had the opportunity to discuss my research work with them and collaborate with some of them in my research work. Their presence during these three years allowed me to spend very pleasant days in the Telecommunication Laboratory and made my PhD experience unique and unrepeatable.

Finally, I will never finish to thank my family, my sister with her family and my girlfriend Martina, that constantly and patiently supported and encouraged me in achieving this so important result.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ACC	Adaptive Cruise Control
ADAS	Autonomous Driver Assistance System
ADC	Analog to Digital Converter
AE	Auto-Encoder
AI	Artificial Intelligence
AIC	Akaike Information Criterion
AM	Alternating Maximization
AWGN	Additive White Gaussian Noise
Bi	Bipolar
BN	Batch Normalization
BPTT	Back-Propagation Through Time
BSD	Blind Spot Detection
CAE	Convolutional Auto-Encoder
CFAR	Constant False Alarm Rate
CIR	Channel Input Response
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
CS	Compressed Sensing
CT	Computational Time
CVD	Cadence Velocity Diagram
CSDE	Complex Single Delay Estimator
CSDEC	Complex Single Delay Estimation and Cancellation
DBF	Digital Beamforming
DBSCAN	Density Based Spatial Clustering of Applications with Noise
DCNN	Deep Convolutional Neural Network
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DL	Deep Learning
DOA	Direction of Arrival
EM	Expectation and Maximization
ERM	Empirical Risk Minimization
ESPRIT	Estimation Signal Parameters Rotational Invariance Technique
FC	Fully Connected
FCN	Fully Convolutional Network
FDM	Frequency Division Multiplexing
FET	Field Effect Transistor
FFT	Fast Fourier Transform
FM	Frequency Modulation
FMCW	Frequency Modulated Continuous Wave
FOV	Field of View

FPGA	Field Programmable Gate Array
GAN	Generative Adversarial Network
GPS	Global Positioning System
GPU	Graphic Processing Unit
HCI	Human Computer Interface
HCSDEC	Hybrid Complex Single Delay Estimation and cancellation
HGR	Human Gesture Recognition
HMM	Hidden Markov Model
HSTDREC	Hybrid Single Target Detection Range Estimation and Cancellation
HULA	Horizontal Uniform Linear Array
IAA	Iterative Adaptive Approach
IC	Integrated Circuit
IFFT	Inverse Fast Fourier Transform
IDFT	Inverse Discrete Fourier Transform
IOU	Intersection Over Union
KNN	K - Nearest Neighbour
LIDAR	Light Detection and Ranging
LNA	Low Noise Amplifier
LO	Local Oscillator
LPC	Linear Predictive Coding
LRR	Long Range Radar
LS	Least Square
LSTM	Long Short Term Memory
MIMO	Multiple Input Multiple Output
MDL	Minimum Description Length
ML	Machine Learning
MLI	Maximum Likelihood
MLP	Multi-Layer Perceptron
MMIC	Monolithic Microwave Integrated Circuit
MRR	Medium Range Radar
MUSIC	Multiple Signal Classification
MWLA	Modified Wax Leshem Algorithm
NB	Naive Bayes
NN	Neural Network
NRE	Normalized Residual Energy
OFDM	Orthogonal Frequency Division Multiplexing
PA	Power Amplifier
PCA	Principal Component Analysis
PMCW	Phase Modulated Continuous Wave
R	Region
RADAR	Radio Detection And Ranging
RASCA	Range and Angle Serial Cancellation Algorithm
RCS	Radar Cross Section
RF	Radio Frequency
RMSE	Root Mean Square Error
RNN	Recursive Neural Network
ROI	Region of Interest
RPE	Range Profile Estimator
RX	Receive
SAE	Society of Automotive Engineers
SAMME	Stagewise Additive Modelling Multi-class Exponential

SBC	Single Bin Cancellation
SFCW	Stepped Frequency Continuous Wave
SGD	Stochastic Gradient Descent
SIC	Serial Iterative Cancellation
SiGe	Silicon-Germanium
SNR	Signal-to-Noise Ratio
SRR	Short Range Radar
SPE	Spatial Estimator
STDAEC	Single Target Detection Angular Estimation and Cancellation
STDREC	Single Target Detection Range Estimation and Cancellation
SVM	Support Vector Machine
TDM	Time Division Multiplexing
TOF	Time Of Flight
TRP	Target Profile Estimator
TX	Transmit
ULA	Uniform Linear Array
URA	Uniform Rectangular Array
UWB	Ultra-wideband
VA	Virtual Array
VGA	Variable Gain Amplifier
VCO	Voltage Controlled Oscillator
VULA	Vertical Uniform Linear Array
XAI	eXplainable Artificial Intelligence
YOLO	You Look Only Once

1 Introduction

1.1 Autonomous driver assistance systems

In the automotive world, one of the most important and promising research topics emerged during the last few year regards the development of *Advanced Driver Assistance Systems* (ADAS) for autonomous driving. Inside this word, are comprised all the systems to support, assist and eventually substitute the driver for enhancing the safety, the comfort and, more in general, for making feasible the autonomous car of the future. These systems are mainly focused on safety; in fact, the ultimate goal of ADAS is to avoid any kind of accident or collision, even without the driver intervention.

In 2018, the *Society of Automotive Engineers* (SAE) defined six levels of automation (ranging from zero to five), that paved the way for the advancement of the technology employed in autonomous vehicles. In particular, from level zero to two, humans have to actively drive the car, i.e. to steer, brake or accelerate as needed to ensure safety during the driving. Nevertheless, there are some driver support features already implemented, like *Adaptive Cruise Control* (ACC) for modulating the speed of the vehicle according to the distance with respect to the cars or obstacles placed in front of it and *Blind Spot Detection* (BSD) for detecting cars or obstacles near the vehicle (and available starting from Level 1), as well as *line assist* and *Steering Assistance* available starting from Level 2. From Level 3 onwards, the human interventions during driving are even less required, until Level 5, where the automatic features and, in particular, ADAS can drive the vehicle alone, under any weather and environmental conditions. Especially in these last levels, in order to guarantee the safety of the passengers, the car must gain the ability to perfectly reconstruct the environment around it, in order to promptly react in case of dangerous situations. In other words, the car needs to become conscious of the surrounding environment. For this reason, the car of the future does not have to be conceived as an unanimated piece of metal any longer, but as a cognitive system able to sense and react to the environment and lastly humans, as human beings themselves typically do. Moreover, driver-less cars are expected to become systems able to make a critical decision in a very short time and within an extremely fast and potentially dangerous environment; for this reason, it is important that a fully autonomous car is also able to take into consideration its past decisions or past observations and learning from them.

A full perception of the environment can be acquired through a holistic approach, also known as sensor fusion, that consists on putting together data received from multiple sensors like radars, RGB or infrared thermal cameras, LIDAR, and GPS. The first problem when all these sensors are put together is related with mechanical problems and the space occupied by their installation in the vehicle. However, nowadays, hardware advancements have made feasible the realization of devices characterized by a

very compact size; as a matter of fact, they can be certainly installed in a car occupying a very small space. The second (and certainly more difficult) problem is related to the fact that all these sensors return large quantities of data that must be processed and aggregated in real time, as fast as possible. This is still one of the biggest challenges today, since a lot of sophisticated algorithms are required for managing those data.
The third problem is finally related to the coexistence of multiple sensors at the same time: multiple devices of the same type may be installed in a car or in different cars, and they may operate at the same time, so interfering with each other. Moreover, assuming that all the sensors are managed correctly, the car itself should be equipped with a robust decision algorithm for deciding how to react to the information collected by those sensors.

1.2 Radar signal processing for autonomous systems

The main lesson to be learnt from the previous section is related to the fact that a driverless car will be equipped with a lot of sensors and a lot of algorithms. The aim of my PhD project is mainly related to these two aspects, i.e. the sensors and the algorithms; in particular, my PhD research activity aims at providing a contribution to the field of colocated *multiple-input multiple output* (MIMO) radars developed for automotive applications and to the development of new algorithms for radar imaging; these algorithms should achieve a good resolution along all the dimensions in a physical space (range, azimuth and elevation). One of the main reason because I preferred to concentrate on colocated MIMO radar systems instead of other types of sensors is due to the fact that the application of these sensors to civilian applications and, more definitely, in autonomous vehicle is very recent, and it looks at the same time tremendously promising. Even if radar technology is not as mature as the technology of other sensors, in the years of my PhD studies I have had the pleasure to see how the use of these radar systems for sensing the environment has increased in a wide range of applications, not only autonomous driving, but also assisted living, health care monitoring, human machine interface, communication and smart agriculture.

The popularity of colocated MIMO radars in all these fields is due to the availability on the market of new technological solutions, especially for antennas and *radio-frequency* (RF) *monolithic microwave integrated circuits* (MMICs), that allow the development of extremely compact and small boards, containing a lot of antennas operating at 77 GHz or at more than 100 GHz. On the other hand, their popularity is also due to the development of novel algorithms and signal processing techniques for high resolution radar imaging, i.e. for estimating the parameters of multiple detected targets, such as range and *direction of arrival* (DOA). My PhD research activities have focused on this last field; in fact, they have concerned the development of novel and powerful techniques for estimating the parameters of multiple detected targets in bi-dimensional and three dimensional scenarios. Most of the developed techniques are based on a deterministic and approximate maximum likelihood approach, implemented through computationally efficient and iterative deterministic algorithms. These algorithms offer the advantage of achieving good estimation accuracy at the price of a reasonable computational complexity.

In a number of recent automotive applications, however, MIMO radars are forced to work in extremely complex, highly dynamic and time varying scenarios, and such methods may fail, since the theoretical model on which such techniques are based may

not accurately represent the observed dynamics in some real life cases. Moreover, modern radar systems may be not only used for sensing information on the environment, but also for classifying obstacles or gestures or activities. For these reasons, a data-driven approach based on machine learning and deep learning techniques for obstacle detection and radar parameter estimation has also been investigated during my PhD studies. All the proposed algorithms have been tested on the basis of the raw data acquired from low power, wideband and state of the art commercial, automotive radars. The obtained results have evidenced that these algorithms achieve similar accuracies, but at the price of different computational efforts. Moreover, other activities have been accomplished in collaboration with CNH Industrial Italy and Belgium with the aim of assessing the performance of the developed algorithms in different agricultural scenarios.

1.3 Main contributions of my research project

The main activities carried out during my PhD studies have aimed at:

1) Achieving a deep understanding of colocated, multiple input multiple output radars employed in automotive applications, both from a technological and a signal processing point of view. This has allowed me to understand the received radar signal model and the structure of the *virtual array* (VA) available on the commercial radars I have employed. This represents a fundamental step in the development of new algorithms for MIMO radars.

2) Implementing deterministic methods based on an iterative cancellation procedure for *two-dimensional* (2D) detection of multiple targets and joint estimation of their range and azimuth. The developed methods, dubbed CLEAN and *modified* WLA, have been tested on data acquired through a colocated SFWC MIMO radar radiating an ultra wide-bandwidth signal in C-band (at 5 GHz) and operating in *time division multiplexing* (TDM). The proposed results show that these algorithms achieve similar accuracies, but at the price of different computational efforts. Moreover, these methods are able to achieve similar accuracies even respect state-of-the-art algorithms like MUSIC, but employing less computational complexity and time. Such methods have been initially devised for solving a 2D estimation problem only. Then, they have been extended for the first time to a *three-dimensional* (3D) one, to also include the estimation of target elevation. The accuracy and computational complexity of the 3D algorithms has been assessed on the basis of the data acquired through another state-of-the-art colocated SFCW MIMO radar operating in TDM and radiating a wideband signal in the E-band (at 77 GHz).

3) Developing a novel deterministic algorithm, dubbed *complex single delay estimator* (CSDE) (*complex single delay estimation and cancellation* (CSDEC)), based on an approximate maximum likelihood approach for the estimation of a single undampted exponential (multiple superimposed undamped exponentials).
This method has become the core of another method, called *range and angle serial cancellation algorithm* (RASCA) for range and angle (both azimuth and elevation) estimation of multiple targets. The accuracies of these methods have been assessed in the presence of closely spaced targets, through computer simulations and on the basis of data collected through an SFCW MIMO radar operating at 77 GHz. The obtained results confirm that the proposed method is able to estimate the position of multiple targets (up to five) in terms of range, azimuth and elevation with a resolution of few centimeters.

4) Implementing and testing: a) machine learning techniques and a *convolutional neural network* (CNN) in the classification of three different human activities; b) two deep learning techniques for target detection and localization. A custom dataset acquired in an indoor scenario through a colocated *frequency modulated continuous wave* FMCW MIMO radar operating at 77 GHz and in TDM mode has been processed to assess the accuracy of these methods. The obtained results confirm that the first CNN method is able to classify different human activities with an accuracy higher than 90 %, while the second techniques allow to detect, localize and classify a given target in the space with an accuracy higher than 94 %.

5) Accomplishing further research activities in collaboration with CNH Industrial Italy and Belgium. These have allowed me to assess the performance of the developed algorithms in different agricultural scenarios. In particular, a custom regression algorithm and the RASCA method have been used to analyse the raw data collected through one or multiple FMCW radar systems installed on a tractor moving through rows of a plantation.

The organization of the remaining part of this thesis follows the same order as that of the list provided above. Moreover, some conclusions are provided in Chapter 7.

2 | Automotive radars: an overview

This chapter aims at providing an introduction to the world of automotive radar systems and more generally to the colocated MIMO radars, that are commonly employed in the automotive market. After illustrating some basic information about their characteristics and outlining their evolution in the last two decades, their architecture is described. Finally, the received signal model is briefly analysed for both FMCW and SFCW MIMO radars.

2.1 Basic Principles and Classification of radars

A radar (radio detection and ranging) is a system able to exploit electromagnetic waves for measuring the position and the velocity of targets. A *radio-frequency* (RF) signal, radiated by the transmitter of the considered radar system, hits an object reflecting it back; the resulting echo, after a certain delay, is then detected by the radar receiver. The key role of this sensor in a lot of civilian applications and, in particular, in autonomous driving is motivated by its ability of measuring distance and speed at the same time and to its robustness even in bad weather and lighting conditions. In an automotive scenario, the target whose position and velocity need to be estimates is usually represented by a car, a pedestrian, a bicyclist and whatever represents a potential danger for the autonomous vehicle. Therefore, automotive radar systems are required to operate in a polluted environment (from an electromagnetic point of view), where a number of extended and different-sized targets needs to be accurately detected.

The interest in highly-accurate, compact and low-power automotive radar sensors to be installed on modern autonomous vehicles has pushed the frequency of such devices from 5 GHz up to 77 GHz. These devices are also named *mm-wave radars*. Radar systems able to transmits signals at sub-THz frequencies are currently under development [1]. Operating at mm-wave frequencies has several advantages; first of all, increasing the transmitting frequency allows to increase the bandwidth and, as consequence, improve range resolution, i.e. the capability of resolving two adjacent targets. Secondly, velocity estimation becomes easier, because the Doppler shift observed at a certain speed gets larger. Automotive radars can be classified on the basis of the maximum measurable range. According to this classification, these systems are divided in (see Table 2.1, where, for each type of radar, the achievable range, the transmission frequency and the typical applications are listed):

1. *Short range radars* (SRRs) - These are able to measure a maximum range of about 30 m and offer the highest angular resolution.
2. *Medium range radars* (MRRs) - These are characterized by a maximum range of about 100 m, offer a quite large azimuthal FOV and achieve a reasonable angular resolution.

3. *Long range radars* (LRRs) - These are characterized by the largest maximum range (250 m) and the smallest FOV.

Radar type	Max range (m)	Freq. (GHz)	Typical applications
Short range	30	5-77	Park assist, pre-crash
Mid range	100	24-77	Blind spot detection Rear collision avoidance Cross traffic alert
Long range	250	40-77	Adaptive cruise control

TABLE 2.1: Classification of automotive radars on the basis of their maximum measurable distance.

This thesis work is mainly focused on these three classes; medium and short, but also long range radars are expected to detect with very high precision not only range and velocity, but also the angular position of multiple targets in the environment around the vehicle. For this reason, these devices are equipped with antenna arrays at both transmit and receive sides. The initial excitement about the use of antenna arrays at both sides (i.e., briefly, about MIMO) in wireless systems has been sparked by the pioneering work of J. H. Winters [2], G. J. Foschini [3], Foschini and M. J. Gans [4], and E. Telatar [5]; these researchers predicted huge capacity gains in *wireless communications* affected by multipath fading [6]. A few years later, the exploitation of antenna arrays has been also investigated in the radar field for the potential improvements it could provide in terms of *signal-to-noise ratio* (SNR), resolution and detection capability. In fact, in principle, the availability of multiple transmit/receive antennas allows to (e.g., see [7]–[9])

1) increase the SNR characterizing target echoes and make it more stable;

2) implement *spatial filtering* (i.e., *beamforming*) for directional signal transmission/reception and, consequently, achieve a large *field of view* (FOV);

3) increase the overall number of degrees of freedom and, consequently, the maximum number of targets that can be detected at a given range;

4) improve the *angular resolution* with respect to traditional radar systems;

5) exploit *spatial diversity*, so that uncorrelated aspects of a given target can be perceived.

Generally speaking, MIMO radar systems can be divided in *statistical* MIMO radars [10], [11] and *colocated* MIMO radars [12], [9] on the basis of the distance between their transmit and receive arrays. In fact, the transmit and receive antennas of the radar systems belonging to the first class are widely separated; on the contrary, in radar systems of the second class, transmit antennas are close to the receive ones and, in particular, are usually placed on the same shield.

Colocated MIMO radars can be further classified as: a) *mono-static radars*, where transmit and receive arrays share their antenna elements; b) *pseudo-bistatic* radars, where transmit and receive arrays are made of distinct antenna elements, placed at different positions. It is important to keep in mind that, in statistical MIMO radars, spatial diversity originates from the fact that distinct receive antennas, being well separated, can observe uncorrelated parts of the same target.

In colocated MIMO radars, instead, a large spatial aperture is achieved by radiating *orthogonal* waveforms. Based on the way these waveforms are generated, colocated

MIMO radars can be divided in: a) *time division multiplexing*, i.e. TDM radars [13], b) *frequency division multiplexing* (FDM) radars [14] and c) *orthogonal frequency division multiplexing* (OFDM) radars [15]. On the one hand, in TDM (FDM) radars, orthogonality is achieved by transmitting through distinct antennas over disjoint time (frequency) intervals; on the other hand, in OFDM radars, any transmit antenna can be used to radiate multiple orthogonal waveforms at the same time. In the last section of this chapter, the architecture of a pseudo-bistatic colocated MIMO radar operating in TDM mode and largely preferred as automotive radar is described in detail.

2.2 A brief history of MIMO radars

The birth of *radio detection and ranging* (briefly, radar) systems dates back to 1904, when the German inventor *Christian Hulsmayer* built a simple ship detection device for avoiding collisions in fog [16]. However, the first practical radar system was developed by the British physicist Sir *Robert Watson-Watt* in 1935, and was employed by the British army in World War II to detect air and sea aggressors [17]. Another fundamental step in the evolution of radar technology is represented by the early studies on *optimal filtering*; the rigorous formulation of this problem and its solution are due to the American scholar *Norbert Wiener* and date back to the 40' [18]. Since then, many advancements have been made in military and civilian radar systems, thanks to the development of signal processing techniques and to the evolution in electronic technology. The most significant advances in signal processing methods applicable to radar systems equipped with antenna arrays have involved both the transmit side and the receive side, and can be summarised as follows.

As far as the transmit side is concerned, substantial research efforts have been devoted to the study of *analog beamforming* (ABF) and *digital beamforming* (DBF) methods for controlling *phased arrays*; both types of methods allow to obtain electronic *beam steering*, i.e. to *steer* the main lobe of the array radiation pattern without any movement of the antennas forming it. It is worth stressing that phased arrays have been around for more than fifty years [19], and that a radar equipped with a phased array is much simpler than a MIMO radar. In fact, a radar system endowed with a phased array generates a single waveform feeding each transmit antenna with a different phase (or, equivalently, with a different delay); consequently, the waveforms radiated by distinct antennas are highly correlated. Moreover, analog beamforming represents the earliest method for electronic beam steering; in this case, each of the signal feeding a transmit antenna is first amplified and then delayed through a phase shifter in a *radio frequency* (RF) stage; an important drawback of this method is represented by the fact that the shape of the resulting beam is fixed. On the other hand, DBF is based on the idea of implementing beam steering in the (digital) baseband portion of the radar hardware by multiplying each signal by a complex gain [20]. This procedure allows to digitally customize the radiated beam, adapting its direction to channel conditions. This technique, also known as *adaptive beamforming* [21], plays an important role in the presence of severe path loss. However, it should be always kept in mind that any radar transmitter exploiting beamforming requires some time (in practice, multiple dwells) to scan the area of interest. On the contrary, if a MIMO radar is employed, the entire observed area is illuminated in a single dwell and beamforming is obtained through the use of different orthogonal waveforms [22].

Another important research area concerning the transmit side of radar systems endowed with antenna arrays concerns the design of the radiated waveforms [9]. Despite the fact that significant theoretical results have been achieved regarding their optimal design of waveforms (e.g., see [23]), few modulation techniques have been employed in commercial MIMO radars until now. These include the FMCW technique [24] (also known as *chirp signal modulation*) and the *Stepped Frequency Continuous Wave* (SFCW) technique [25]. In the last years, considerable attention has been also paid to the use of the OFDM technique [26], [27] and to the *phase modulated continuous wave* (PMCW) technique [28].

Early research work regarding the receive side of radar systems endowed with antenna arrays has focused on the development of beamforming methods [29]. One of the most important contributions to this area is represented by the so called *Capon beamformer*, which can provide good resolution and interference rejection capability [30], [31]. Other fundamental contributions about the processing of multiple signals acquired by a radar systems through its antenna array concern the estimation of the *direction of arrival* (DOA) of the electromagnetic waves impinging on the array itself. Here, I limit to mention the MUSIC [32] and the *estimation of signal parameters via rotational invariance* (ESPRIT) techniques [33], [34].

The development of signal processing methods for MIMO radars started after the end of 2003; in fact, in that period, the concept of MIMO radar, defined as a device able to probe a wireless channel by transmitting multiple signals and receiving their echoes with similar multiplicity, was proposed for the first time [8]. Since the beginning, it was clear that MIMO technology could have represented an important tool to improve the SNR of received signals and to increase radar aperture [7], [9], [22], [35]. Since then, the exploitation of known DOA estimation strategies, developed in the previous years for antenna arrays (like MUSIC and ESPRIT), has been widely investigated for this new type of radars (e.g., see [36], [37] and [38]). However, the availability of MIMO radars able to radiate wideband signals by a large number of antennas and to acquire their echoes through an even larger number of antennas have raised various problems, whose solution requires substantial research efforts. In fact, on the one hand, these devices allow to acquire a rich set of information about the surrounding propagation environment; on the other hand, they require storing and processing large datasets. This has motivated the investigation of *compressed sensing* (CS) and statistical sparsity-based techniques, since these can be exploited to perform signal detection and parameter estimation on the basis of a much smaller dataset than that available in the case in which the received waveforms undergo Nyquist sampling [39], [40]; various examples of CS-based estimation algorithms can be found in ref. [41].

As far as the advancement in electronic technology is concerned, in the remaining part of this paragraph let us focus on some important results achieved in the development of compact integrated radar devices employed in the automotive field. The first generation of commercial *ultra-wideband* (UWB) automotive radar sensors operating in the 77 GHz band has become available in 1999. These devices were not endowed with antenna arrays and their implementation was based on discrete electronic components (in particular, gallium-arsenide Gunn diodes mounted inside a waveguide cavity were employed in the generation of RF waveforms). However, electronic technology progressed quickly in this field and, after few years, MMICs employing high-performance *silicon-germanium* (SiGe) transistors became available for the implementation of fully integrated radars. Advantages of SiGe include the boosting of the f_{max} of the transistors

up to several hundreds of GHz [42]. Pioneering work in the development and manufacturing of such a technology has been accomplished by the *Infineon* company, that has started its production in 2004 [43]. It is also worth mentioning that, in the same year, a description of the first fully integrated 24-GHz eight-element phased array receiver in SiGe and of the first fully integrated 24-GHz four-element phased array transmitter with integrated power amplifiers in *complementary metal-oxide semiconductor* (CMOS) has appeared [44]; these devices were able to accomplish beamforming and could be used for communication, ranging, positioning, and sensing applications. Other examples of phased arrays operating in X and Ku-band have been described later in ref. [45]. The first FMCW MIMO radar transceiver operating at 77 GHz has been implemented in SiGe technology in 2008 [46], whereas the production of the first MIMO FMCW radar, operating according to a TDM strategy and equipped with an array of colocated antennas, started in 2009 [47], [48]. To the best of our knowledge, the last device represents the first compact MIMO radar system based on a MMIC in SiGe, operating at 77 GHz and radiating ultra-wideband signals. In this system, wide-band and high-frequency patch antennas are built on a RF substrate [49], while the base-band MIMO signal processing is accomplished off-chip by a *field programmable gate array* (FPGA) board. Moreover, the *analog-to-digital converters* (ADCs) at the receive side are implemented in CMOS technology and embedded in the transceiver chip; this has been made possible by the SiGe Bi-CMOS process. In the last decade, radar designers working on the development of new integrated radar devices have investigated the use of the more scalable CMOS RF technology [50], that allows to integrate both RF and digital functions on a single chip and at low cost. An important trend in the technological evolution of MIMO radar systems is also represented by the attempt of exploiting the same hardware for both radar and communications [51]. Some milestones achieved in the evolution of the signal processing methods and of the technology employed in colocated radar systems during the last two decades are summarized in Fig. 2.1.

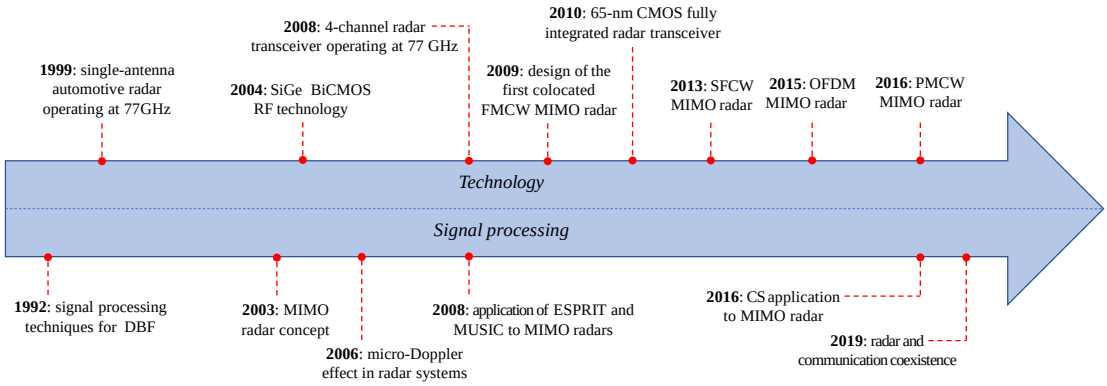


FIGURE 2.1: Milestones in the evolution of automotive radars and radar signal processing techniques.

2.3 MIMO radar basic architecture

In the remaining part of this thesis, colocated and bistatic MIMO radar systems, and in particular FMCW or SFCW radars, are considered; their generic architecture is illustrated in Fig. 2.2. As it can be inferred from this figure, a MIMO radar is equipped with multiple antennas at both transmit and receive sides. Typically, the structure illustrated here, especially for the RF part, can be repeated multiple times for each radiating element or a group of them.

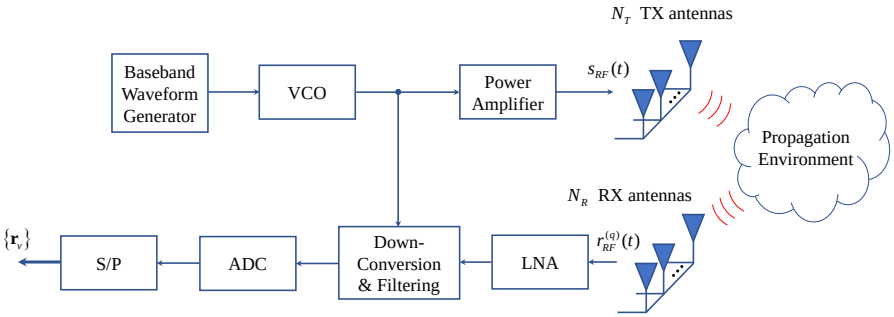


FIGURE 2.2: MIMO radar transmitter (upper part) and receiver (lower part).

However, for simplicity, in the following I focus on the basic building blocks behind a single radiating element. The transmit side associated with each antenna or a small group of them of an FMCW or SFCW radar is characterized by a baseband waveform generator, whose goal is to generate a baseband signal having certain characteristics. In the case of a FMCW system implemented in SiGe Bi-CMOS technology, this block can be implemented through an integrated *direct digital synthesizer* (DDS) and *phase locked loop* (PLL) chip [52], [53]. From their combination it is possible to obtain a sequence of highly linear and ultra wideband frequency sweeps also called *chirps*. This baseband signal enters a fully integrated 77 GHz SiGe *voltage controlled oscillator* (VCO) based on Colpitts architecture and is up-converted to a central frequency of 77 GHz. If this procedure is repeated for all the TX antennas, this produces a chirp FM signal centered at 77 GHz, whose instantaneous frequency evolves periodically, as illustrated in Fig. 2.3.

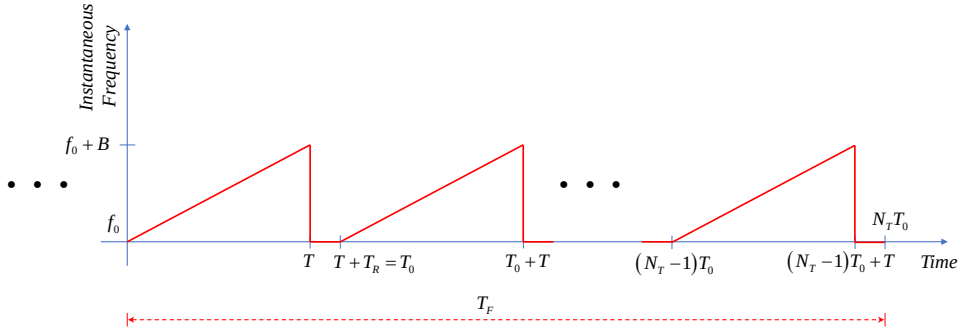


FIGURE 2.3: Representation of the instantaneous frequency of the RF signal generated by the VCO in an FMCW radar system.

In this figure, a TDM strategy is considered for guaranteeing the orthogonality between different transmitting elements and a time slot of T_0 s is assigned to each TX antenna; therefore, transmission from all the TX antennas is accomplished over an interval lasting $T_F \triangleq N_T T_0$ s; this interval represents the duration of a single *transmission frame*. The parameters T, T_R and T_0 represent the *chirp interval*, the *reset time* and the *pulse period* (or *pulse repetition interval*), respectively [54], whereas the parameters f_0 and B are the *start frequency* and the *bandwidth*, respectively, of the transmitted signal. Finally, this signal is amplified by a single (or a cascade) of power amplifiers (PA) and transmitted. It is important to mention that the RF and digital parts of state-of-the-art automotive FMCW radars can be fully integrated in CMOS technology; in this case, the PLL is used for generating both the digital clock (as synchronization signal for the digital part) and the input frequency on which the *local oscillator* (LO) is locked. Moreover, for mitigating

the phase noise of the LO, the VCO generates an oscillation at lower frequency; in particular, more precise oscillators called *Subharmonic Injection-Locked Oscillators* (SH-ILO) are employed for the final conversion at 77 GHz [42]. At the receive side, the first block following the antenna is a two-stage Low Noise Amplifier (LNA); in turn, this is followed by an active Gilbert cell mixer for the downconversion. Then, after a *variable gain amplifier* (VGA) and a bank of *low pass filters* (LPFs), *Analog-to-Digital Converters* (ADC) are employed to digitize the downconverted signal. Even in this case, the ADC ICs are off-chip components in the case of SiGe technology; however, if the CMOS technology is adopted, these two parts can be integrated together on a single chip, saving cost and reducing the whole power consumption. The last issue is also strictly related to the PA; other electronic devices based on the newest GaN (Gallium-Nitride) and FinFET transistors, are still under development for implementing high frequency and efficient PAs. As far as the architecture of a SFCW radar system is concerned, it is very similar to that of a FMCW radar system, except for the baseband synthesizer, that changes the frequency of the oscillations in a step-wise manner, as shown in Fig. 2.4.

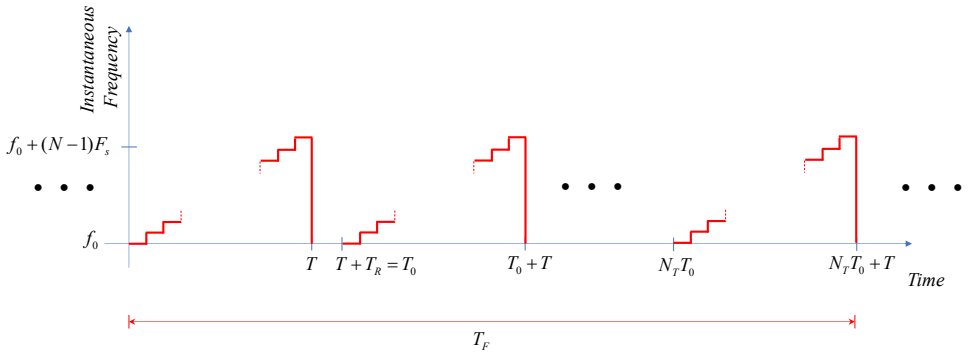


FIGURE 2.4: Representation of the instantaneous frequency of the RF signal generated by the VCO in an SFCW radar system.

In this case, similarly as the SFWC radar system, a TDM strategy is assumed and a time slot of T_0 s is assigned to each TX antenna; therefore, transmission from all the TX antennas is accomplished over an interval lasting $T_F \triangleq N_T T_0$ s; this interval represents the duration of a single *transmission frame*. The parameters T, T_R and T_0 represent the *step duration*, the *reset time* and the *pulse repetition interval*, respectively, whereas the parameter f_0 represents the *start frequency* and $(N-1)F_s$ is the *bandwidth* B of the transmitted signal. A common problem in radar receivers is the so called TX-to-RX spillover; in fact, since the TX and RX antennas are quite close one to one another, even if the transmitter shows a good insulation, some of the radiated power may enter the receiver chain. This unwanted received power manifests itself as a strong target at very small range (roughly at zero distance) and can potentially saturate the receiver. For an FMCW system, the spillover can be removed right after the downconversion (this technique is called DC offset compensation), whereas, in the case of SFCW systems, the removal procedure is more complicated, because the power is spread uniformly over the different pulses and frequencies.

2.3.1 Antenna arrays

In general, the radar front end contains N_T TX antennas and N_R RX antennas. At 77 GHz, each antenna is commonly implemented as a single or a series of patch elements (typically from four to eight), for improving the gain and the directivity of the radiating elements and for reducing the sidelobes of the radiated and received power. This

is important in the case of MRR or LRR radars. The distance between two adjacent patches is $\lambda/2$, in order to maximize the radiated power at the broadside direction [55]. These TX and RX elements made of more than one patches can be combined in different manner, forming a *physical* array of antennas, that, according to its shape, can have different characteristics. The simplest *mono-dimensional* (1D) array is the *uniform linear array* (ULA), where the physical distance between two adjacent antennas is constant for the entire array and typically half a wavelength in order to avoid any ambiguity in angular estimation. An example of ULA physical array is shown in Figure 2.5-a), where the red squares represent the modelled TX antennas and the blue ones the RX antennas. In this figure, the distance between the TX and RX element is $d = \lambda/2$, while d_t (d_r) indicates the distance of the TX (RX) antenna from the array center.

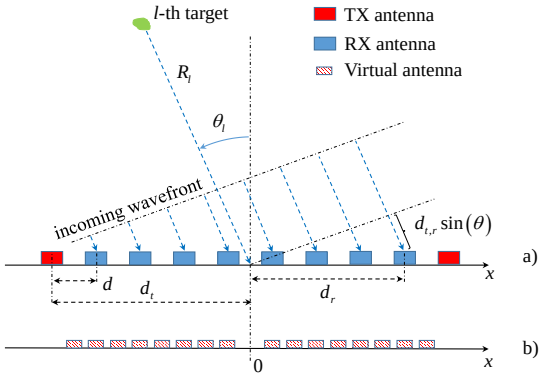


FIGURE 2.5: Representation of: a) an ULA (characterized by $N_T = 2$ and $N_R = 8$), and the relevant geometric parameters referring to the l-th target, the t-th TX antenna and the r- th RX antenna; b) the associated virtual ULA (consisting of $N_T \cdot N_R = 16$ antennas).

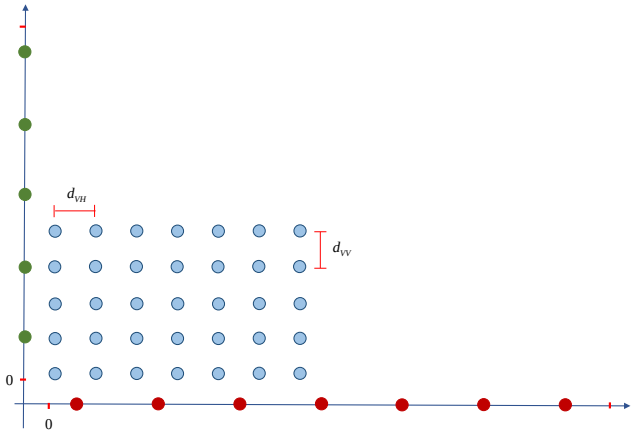


FIGURE 2.6: Representation of an URA characterized by $N_T = 5$ TX antennas (green) and $N_R = 7$ RX antennas (red) and the associated virtual URA (consisting of $N_V = N_T \cdot N_R = 35$ antennas) in light-blue.

In radar signal processing, every pair of *physical* TX and RX antennas (namely, the t-th TX antenna and the r-th RX antenna) is often replaced by a single *virtual antenna* of an equivalent *monostatic radar*. In particular, the abscissa x_v and the ordinate y_v of the v-th *virtual antenna element* associated with the t-th TX antenna and the r-th RX antenna

are computed as[1]

$$x_v = \frac{x_t + x_r}{2} \tag{2.1}$$

and

$$y_v = \frac{y_t + y_r}{2}, \tag{2.2}$$

respectively, with $v = 0, 1, ..., N_V - 1$; here, (x_t, y_t) $((x_r, y_r))$ are the coordinates of the TX (RX antenna) and $N_V \triangleq N_T \cdot N_R$ represents the overall size of the resulting *virtual array* (see the one represented in Fig. 2.5 and composed by sixteen elements). An ULA has the advantage to be a very simple array for range and azimuth estimation, but it does not allow to obtain any estimate of the elevation angle, since the FOV along the vertical direction of each single radiating element is very limited. To this purpose, an *uniform rectangular array* (URA), like the one shown in Fig. 2.6, can be adopted, since multiple antennas become available along both horizontal ($N_R = 7$) and vertical directions ($N_T = 5$). The general rules for deriving the virtual array are still represented by eqs. (2.1) and (2.2); the corresponding rectangular virtual array is shown in Figure 2.6 and consists of $N_V = N_T \cdot N_R = 35$ antennas. Other planar array structures with an irregular distribution of physical antennas are possible and can be found in literature or in commercial devices, like stretched, staircase or diagonal array [56].

2.4 MIMO radar received signal model

After the description of the general architecture of FMCW and SFCW radars, in this section, the general signal model received by a MIMO radar for both FMCW and SFCW is shown.

2.4.1 FMCW MIMO radar

Let us consider the sequence of chirp represented in Fig. 2.3. If we focus on the time interval $\bar{t} \in (0, T)$ and assume that, in this interval, the t-th TX antenna is employed by the considered radar system (with $t \in \{0, 1, ..., N_T - 1\}$), the radiated signal can be expressed as

$$s_{RF}(\bar{t}) = A_{RF} \Re\{s(\bar{t})\}, \tag{2.3}$$

where A_{RF} is its amplitude,

$$s(\bar{t}) \triangleq \exp[j\theta(\bar{t})], \tag{2.4}$$

$$\theta(\bar{t}) \triangleq 2\pi \left(f_0 \bar{t} + \frac{\mu}{2} \bar{t}^2 \right) \tag{2.5}$$

and

$$\mu = \frac{B}{T} \tag{2.6}$$

is the *chirp rate*, i.e. the steepness of the generated frequency chirp. Let $x_{RF}^{(r)}(\bar{t})$ denote the signal available at the output of the r-th receive antenna, with $r = 0, 1, ..., N_R - 1$. If we assume that the radiated signal $s_{RF}(\bar{t})$ (2.3) is reflected by L *static point targets*, as the one shown in Figure 2.4, the useful component of $x_{RF}^{(r)}(\bar{t})$ consists of the superposition of L echoes, each originating from a distinct target. In this case, if the propagation

[1] This is not the only rule adopted in the technical literature to compute the coordinates of the v-th virtual antenna element. For instance, in ref. [22, Par. 4.3.1, pp. 159-161], the abscissa (ordinate) of this element is evaluated as $2x_v$ ($2y_v$), where x_v and y_v are expressed by eqs. (2.1) and (2.2), respectively.

environment undergoes slow variations and the far field condition is satisfied[2], a simple mathematical model can be developed to represent the sequence of samples generated by the ADC in a single chirp interval.

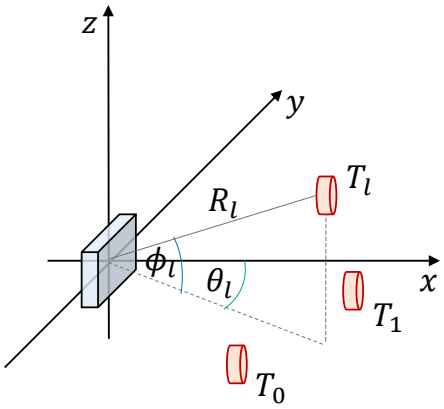


FIGURE 2.7: Representation of a simple three dimensional scenario in which three targets (red cylinders) are present. The azimuth (elevation) angle θ (ϕ) associated to the l-th target is measured respect the centre of the radar (grey cube) along the horizontal (vertical) direction.

Based on these assumptions, the n-th received, base-band signal sample acquired through the v-th virtual antenna element (with $v = 0, 1, ..., N_V - 1$) can be expressed (e.g., see [57, Par. 4.6, eq. (4.27)])

$$x_{r,n}^{(v)} = \sum_{l=0}^{L-1} a_l \cos\left(2\pi n F_l^{(v)} + \psi_l^{(v)}\right) + w_{r,n}^{(v)}, \tag{2.7}$$

with $n = 0, 1, ..., N - 1$; here, the index r indicates that the signal is real, N is the overall number of samples acquired over a chirp period, a_l is the amplitude of the l-th component of the useful signal (this amplitude depends on both the range R_l and the reflectivity of the l-th target, but is assumed to be independent of v for simplicity),

$$F_l^{(v)} \triangleq f_l^{(v)} T_s \tag{2.8}$$

is the normalized version of the frequency

$$f_l^{(v)} \triangleq \mu \, \tau_l^{(v)}, \tag{2.9}$$

characterizing the l-th target detected on the v-th virtual receive antenna and considering T_s the ADC sampling period,

$$\tau_l^{(v)} = \frac{2}{c} \left[R_l + x_v \cos(\phi_l) \sin(\theta_l) + y_v \sin(\phi_l)\right] \tag{2.10}$$

or

$$\tau_l^{(v)} = \frac{2}{c} \left[R_l + x_v \sin(\theta_l)\right] \tag{2.11}$$

is the delay of the echo generated by the l-th target and observed on the v-th virtual channel for a 3D or 2D scenario, respectively, x_v and y_v are given by eq. (2.1) and (2.2), R_l, ϕ_l and θ_l denote the range of the l-th target, its azimuth and its elevation, respectively,

[2] this means that the distance between the radar and the closest object $d \geq \frac{2D^2}{\lambda}$, considering D the aperture of the radar array and λ the wavelength of the radiating signal [55, Par. 2.2.4, pp. 34-36]

as shown in Fig. 2.7,

$$\psi_l^{(v)} \cong 2\pi f_0 \tau_l^{(v)}, \tag{2.12}$$

and $w_{r,n}^{(v)}$ is the n-th sample of the *additive white Gaussian noise* (AWGN) sequence affecting the received signal (the noise variance is denoted by σ^2 in the following and is assumed to be independent of v). This model can be adopted in all the FMCW radar systems that extract only the *in-phase* component of the signal captured by each RX antenna. However, some commercial MIMO radars provide also the *in-quadrature* components of the received RF signals [12, Par. 2.1 eq. (2.2)]. In the last case, the complex model

$$x_{c,n}^{(v)} = \sum_{l=0}^{L-1} A_l^{(v)} \exp\left(j2\pi n\, F_l^{(v)}\right) + w_{c,n}^{(v)}, \tag{2.13}$$

must be adopted in place of its counterpart expressed by eq. (2.7) for any n; here,

$$A_l^{(v)} \triangleq a_l \exp\left(j\psi_l^{(v)}\right) \tag{2.14}$$

for any v and l, where $\psi_l^{(v)}$ is expressed by eq. (2.12). The received signal models (2.7) and (2.13) hold if all the observed targets are static. Let us focus now on a FMCW radar system operating in the presence of L moving point targets and having the following characteristics: a) it is equipped with a single TX antenna and a single RX antenna (i.e., $N_T = N_R = 1$); b) its reset time T_R is equal to 0, so that $T_0 = T$ (see Fig. 2.2); c) its transmission frame consists of N_c chirps, so that the duration T_F of the transmission frame is equal to $N_c T_0 = N_c T$ s; d) N distinct ADC samples are acquired in each chirp interval at the receive side. Under these conditions, it is not difficult to prove that, if the ranges of all the targets are much larger than their displacements observed during the considered transmission frame the n-th sample of the converted signal acquired in the k-th chirp interval (with $k = 0, 1, ..., N_c - 1$), can be expressed as [54]

$$x_{c,n}^{(k)} \triangleq \sum_{l=0}^{L-1} A_l^{(k)} \exp\left(j2\pi n F_{o,l}\right) + w_{c,n}^{(k)} \tag{2.15}$$

where $n \in [0\,,\, N-1]$, $F_{o,l} = f_{o,l} T_s = (f_l + f_{D,l})\, T_s$, f_l is defined by eqs. (2.9)-(2.10) considering $v = 0$, i.e. $x_v = y_v = 0$,

$$f_{D,l} = \frac{2v_l}{\lambda} \tag{2.16}$$

is the *Doppler frequency*

$$A_l^{(k)} = a_l \exp\left(j\psi_l^{(k)}\right) \tag{2.17}$$

$$\psi_l^{(k)} \approx \frac{4\pi}{\lambda} R_l^{(k)} \tag{2.18}$$

with $R_l^{(k)} = R_l + v_l kT$ and $w_{c,n}^{(k)}$ is the AWGN noise factor affecting the n-th sample of the k-th chirp of the received complex channel. Here R_l represents the frontal distance, i.e. the range, between the radar and the l-th target, while v_l is the speed at which the considered target is moving. In case the FMCW system manages only the in-phase component, the expression of $x_{r,n}^{(k)}$ can be defined in a similar way as eq. (2.7), namely as

(e.g., see [54, eq.(5)]):

$$x_{r,n}^{(k)} = \sum_{l=0}^{L-1} a_l \cos(2\pi n \, F_{o,l} \, + \psi_l^{(k)}) + w_{r,n}^{(k)}, \tag{2.19}$$

where $\psi_l^{(k)}$ is defined by eq. (2.18) and $w_{r,n}^{(k)}$ is the n-th sample of the AWGN sequence affecting the received signal in the k-th chirp interval.

The derivation of models (2.7), (2.13) holds in the presence of multiple static targets and considering a rectangular array, while the derivation of the simplified models (2.15) and (2.19) is valid if the targets are moving and the radar is equipped with a single channel. However, let us consider now the case in which targets are moving and the radar sounds the environment through a multitude antennas on both transmit and receive side; in this case, the FMCW radar system has the following characteristics: a) it is equipped with N_T TX antennas and N_R RX antennas; b) its reset time T_R is equal to 0, so that $T_0 = T$; c) its transmission frame consists of $N_c \cdot N_T$ chirps, so that the duration T_F of the transmission frame is equal to $N_c N_T T$ s; d) N distinct ADC samples are acquired in each chirp interval at the receive side. The received signal model for an FMCW system receiving only the in-phase component becomes:

$$x_{r,n}^{(v,k)} = \sum_{l=0}^{L-1} a_l \cos\left(2\pi n \, F_{o,l}^{(v)} \, + \psi_l^{(v,k)}\right) + w_{r,n}^{(v,k)}, \tag{2.20}$$

where in this case, $F_{o,l}^{(v)} = \left(f_l^{(v)} + f_{D_l}\right) T_s$, $f_l^{(v)}$ (f_{D_l}) is defined by (2.9) ((2.16)) and

$$\psi_l^{(v,k)} = \frac{4\pi}{\lambda} R_l^{(v,k)} \tag{2.21}$$

with

$$R_l^{(v,k)} = R_l^{(v)} + v_l k T \tag{2.22}$$

and $R_l^{(v)} = \frac{c}{2}\tau_l^{(v)}$, $\tau_l^{(v)}$ is defined by eq. (2.10) and v_l is the speed of the l-th target. Here, the noise $w_{r,n}^{(v,k)}$ is the AWGN affecting the considered v-th channel in the k-chirp and the n-th sample. On the other hand, for an FMCW system with a received complex signal,

$$x_{c,n}^{(v,k)} \triangleq \sum_{l=0}^{L-1} A_l^{(v,k)} \exp\left(j2\pi n F_{o,l}^{(v)}\right) + w_{c,n}^{(v,k)} \tag{2.23}$$

where, $A_l^{(v,k)}$ can be obtained considering $\psi_l^{(v,k)}$ (see eq. (2.21)) instead of $\psi_l^{(k)}$ in eq. (2.17) and $F_{o,l}^{(v)} = \left(f_l^{(v)} + f_{D_l}\right) T_s$. Here, the noise $w_{c,n}^{(v,k)}$ is the AWGN affecting the considered v-th channel in the k-chirp and n-th sample. The expressions (2.20) and (2.23) represent the two most complete models for FMCW radars. In this case, for each detected target, the parameters to estimate are four: range, azimuth, elevation and velocity; for this reason, in this case, the problem of estimating the parameters of a target is dubbed *four-dimensional* (4D) radar imaging [58].

2.4.2 SFCW MIMO radar

The second considered case corresponds to the signal received by an SFCW MIMO radar. This system provides, for any pair of TX and RX antennas, i.e., for every virtual channel, a set of N measurements in the frequency domain; each measurement represents an estimate of the frequency response of the communication channel between such antennas at a specific frequency. In practice, these measurements are acquired by sounding the communication channel at N equally spaced frequencies; on the basis of the radiated signal represented in Fig. 2.4, the n-th value of the instantaneous frequency can be expressed as

$$f_n = f_0 + n\,\Delta f, \tag{2.24}$$

with $n = 0, 1, ..., N-1$, where f_0 is the minimum radiated frequency, Δf is the *frequency step size*. It is not difficult to prove that, under the same assumptions made in the derivation of eq. (2.13), the measurement acquired through the v-th virtual element at the n-th frequency can be expressed as

$$x_{c,n}^{(v)} = \sum_{l=0}^{L-1} A_l^{(v)} \exp\left(-j2\pi n F_l^{(v)}\right) + w_{c,n}^{(v)}, \tag{2.25}$$

with $v = 0, 1, ..., N_V - 1$; here, the amplitude

$$A_l^{(v)} \triangleq a_l \exp\left(-j\psi_l^{(v)}\right) \tag{2.26}$$

and the phase $\psi_l^{(v)}$ is still expressed by eq. (2.12) (this term can account also for an initial phase shift), and

$$F_l^{(v)} \triangleq \Delta f\, \tau_l^{(v)} \tag{2.27}$$

is the normalised delay characterizing the l-th target and observed on the v-th virtual antenna, the parameters a_l is the RCS, as well as $\tau_l^{(v)}$ and the random variable $w_{c,n}^{(v)}$ that are expressed by the same definition used for the received signal model (2.7).

Similarly as the FMCW counterpart, let us focus next on a SFCW radar system operating in the same dynamic scenario as the one just described for a FMCW radar system and having the following characteristics: a) it is equipped with a single TX antenna and a single RX antenna; b) its reset time T_R is equal to 0, so that $T_0 = T$; c) its transmission frame consists of N_c frequency sweeps; d) in each sweep (lasting T s), N distinct and uniformly spaced frequencies are generated; d) the time between two consecutive frequencies, i.e. the duration of each frequency step is T_p. Then, assuming $f_0 >> N\Delta f$, the measurement acquired at the n-th frequency in the k-th frequency sweep (with $k = 0, 1, ..., N_c - 1$) can be expressed as

$$x_{c,n}^{(k)} = \sum_{l=0}^{L-1} A_l^{(k)} \exp\left[-j\left(2\pi n F_{o,l}\right)\right] + w_{c,n}^{(k)}\,, \tag{2.28}$$

where

$$A_l^{(k)} \triangleq a_l \exp\left(-j\psi_l^{(k)}\right) \tag{2.29}$$

$\psi_l^{(k)}$ is still expressed by eqs. (2.21), $F_{o,l} = (F_l + F_{D,l})$[3], F_l is expressed by eqs. (2.10)-(2.27) for $v = 0$, i.e. $x_v = y_v = 0$, $F_{D,l} = f_{D,l} T_p$ and $f_{D,l}$ is indicated by eq. (2.16). The derivation of the model (2.25) is valid in the presence of multiple static targets and of a

[3] In some cases the expression of $F_{o,l} = (F_l - F_{D,l})$

rectangular array, while the derivation of the simplified model (2.28) is valid if the targets are moving and only a single channel is considered. Therefore, let us consider now the most general case in which targets are moving and all the antennas are employed; in particular, the considered radar has the following characteristics: a) it is equipped with N_T TX antenna and N_R RX antennas; b) its reset time T_R is equal to 0, so that $T_0 = T$; c) its transmission frame consists of $N_c \cdot N_T$ frequency sweeps; d) in each sweep (lasting T s), N distinct and uniformly spaced frequencies are generated; d) the time between two consecutive steps, i.e. the duration of each step is T_p. In this case, the received signal model for an SFCW system becomes:

$$x_{c,n}^{(v,k)} \triangleq \sum_{l=0}^{L-1} A_l^{(v,k)} \exp\left[-j\left(2\pi n F_{o,l}^{(v)}\right)\right] + w_{c,n}^{(v,k)} \tag{2.30}$$

where

$$A_l^{(v,k)} \triangleq a_l \exp\left(-j\psi_l^{(v,k)}\right) \tag{2.31}$$

$\psi_l^{(v,k)}$ is still expressed by eqs. (2.21)-(2.22) and $F_{o,l}^{(v)} = F_l^{(v)} + \bar{F}_{D,l}$ (see eq. (2.16)). Here, the noise $w_{c,n}^{(v,k)}$ is the AWGN affecting the considered v-th channel in the k-chirp and n-th sample. The expression (2.30) represents the most complete model for SFCW radars and it is quite similar to the one expressed by eq. (2.23) for an FMCW radar system, except for the fact that the exponential factors contains a minus sign.

2.5 Basic signal processing considerations: a case study

In this section a simple example concerning with the detection of a single point target, and the estimation of its range R and its azimuth θ in a 2D propagation scenario is proposed. In this case, an FMCW radar system equipped with an ULA, consisting of three antenna elements, is employed (see Fig. 2.8-a). This array is made of a central TX antenna and a couple of antipodal RX antennas (these are identified by a red box and two blue boxes, respectively, in the considered figure), so that $N_T = 1$ and $N_R = 2$; consequently, a *virtual* array, consisting of $N_V = 2 \cdot 1 = 2$ virtual elements, is available. The abscissa x_v of the v-th *virtual antenna element* associated with the TX antenna and the v-th RX antenna is computed as (see eq. (2.1))

$$x_v = \frac{x_t + x_{r,v}}{2} \tag{2.32}$$

with $v = 0$ and 1; here, $x_t = 0$ and $x_{r,0} = -d$ ($x_{r,1} = d$) are the abscissas of the TX and of the first (second) RX antenna, respectively (Note that the origin of the reference system coincides with the center of the array).

If the target is in far field, the wavefront of the electromagnetic echo originating from it is a straight line and is orthogonal to the line connecting the target with the center of the array. In these conditions, the n-th time-domain sample acquired on the v-th virtual antenna in a single snapshot can be expressed as (see eq. (2.7))

$$\begin{aligned} x_{r,n}^{(v)} &= a^{(v)} \cos\left(2\pi n f^{(v)} T_s + \psi^{(v)}\right) + w_{r,n}^{(v)}, \\ &= C^{(v)} \exp\left(j 2\pi n F^{(v)}\right) + C^{(v)^*} \exp\left(-j 2\pi n F^{(v)}\right) + w_{r,n}^{(v)}, \end{aligned} \tag{2.33}$$

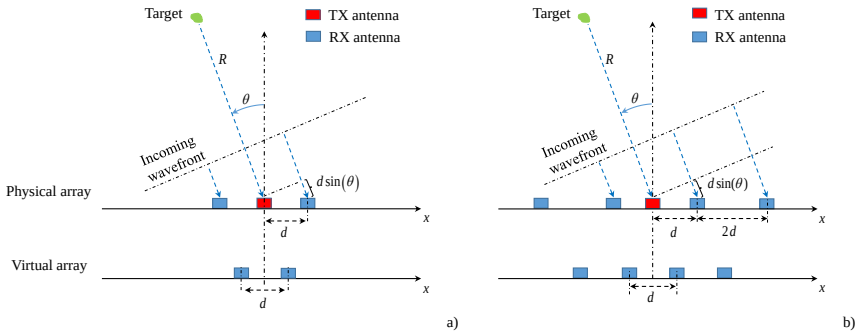


FIGURE 2.8: Physical geometry and virtual array of a colocated FMCW MIMO radar equipped with an ULA composed of a single TX antenna and: a) two RX elements; b) four RX elements.

for $n = 0, 1, ..., N-1$, where (see eqs. (2.8), (2.9) and (2.10))

$$F^{(v)} \triangleq f^{(v)} T_s, \tag{2.34}$$

$$f^{(v)} = \frac{2\mu}{c} \left[R + x_v \sin(\theta) \right], \tag{2.35}$$

$$x_v = (-1)^{(v+1)} \frac{d}{2}, \tag{2.36}$$

d is inter-antenna spacing of the considered ULA[4],

$$C^{(v)} \triangleq \frac{1}{2} a^{(v)} \exp\left(j\psi^{(v)} \right) \tag{2.37}$$

is a complex parameter depending on the target reflectivity $a^{(v)}$ (see eqs. (2.14)), where in this case the amplitude $a^{(v)}$ depends on the considered antenna and (see eq. (2.10) and (2.12))

$$\psi^{(v)} \triangleq \angle C^{(v)} \cong \frac{4\pi}{\lambda} \left[R + x_v \sin(\phi) \right] \tag{2.38}$$

is the phase observed on the considered antenna. It is important to point out that:

a) Relevant information about the target *azimuth* are provided by the frequency difference

$$\Delta f_{0,1} \triangleq f^{(1)} - f^{(0)} \tag{2.39}$$

or by the phase variation

$$\Delta\psi_{0,1} \triangleq \angle C^{(1)} \left(C^{(0)} \right)^*, \tag{2.40}$$

where the quantity $\angle X$ represents the phase of the complex number X (it belongs to interval $[-\pi, \pi)$). In fact, on the one hand, from eqs. (2.35)-(2.36) it can be inferred that (see the definition (2.39))

$$\Delta f_{0,1} = 2\frac{\mu d}{c} \sin(\theta); \tag{2.41}$$

[4]This distance will be also called d_{VH} in sect. 4.2.

on the other hand, based on eqs. (2.36)-(2.38), it is easy to show that (see the definition (2.40))

$$\Delta\psi_{0,1} = \psi^{(1)} - \psi^{(0)} = 4\pi\frac{d}{\lambda}\sin(\theta), \tag{2.42}$$

provided that the inequality

$$4\pi\frac{d}{\lambda}|\sin(\theta)| \leq \pi \tag{2.43}$$

holds for any θ. The last condition is met for any $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2})$ if

$$d \leq \lambda/4. \tag{2.44}$$

b) If the received signal is noiseless, the frequency $f^{(v)}$ is known and N is large, the complex amplitude $C^{(v)}$ can be easily estimated as[5]

$$\hat{C}^{(v)} \cong \frac{1}{N}\bar{X}^{(v)}(f^{(v)}), \tag{2.45}$$

where

$$\bar{X}^{(v)}(f) \triangleq \sum_{n=0}^{N-1} x_{r,n}^{(v)} \exp(-j2\pi n f T_s) \tag{2.46}$$

represents the Fourier transform of the sequence $\{x_{r,n}^{(v)}; n = 0, 1, ..., N-1\}$.

c) Information about the target *range* is provided by the *average frequency* (see eq. (2.35))

$$f_m \triangleq \frac{f_0 + f_1}{2} = \frac{2\mu}{c}R \tag{2.47}$$

Therefore, the estimation of the frequency of the sinusoid contained in the noisy data sequence acquired through each virtual antenna represents a fundamental problem in target detection and estimation. It is well known that the so called *periodogram method* can be employed to solve it in an approximate way [59], [60]. This method is based on the computation of the amplitude spectrum of the zero-padded measurement sequence and on the identification of its peak.

Based on the mathematical results and the considerations illustrated above, a simple deterministic algorithm, consisting of the three steps listed below, can be easily derived for the detection of the target and the estimation of its spatial coordinates (R, θ).

1. DFT *processing* - In this step, the N-dimensional vector

$$\mathbf{x}^{(v)} \triangleq \left[x_{r,0}^{(v)}, x_{r,1}^{(v)}, ..., x_{r,N-1}^{(v)}\right]^T, \tag{2.48}$$

with $v = 0$ and 1, undergoes *zero padding* (ZP); this results in the N_0-dimensional vector

$$\mathbf{x}_{ZP}^{(v)} \triangleq \left[\mathbf{x}^{(v)^T}\ \mathbf{0}_P^T\right]^T \tag{2.49}$$

[5]This result can be easily proved by substituting eq. (2.33) in the *right-hand side* (RHS) of the definition (2.46).

where $N_0 \triangleq M\,N$, $\mathbf{0}_P$ denotes the P-dimensional (column) null vector and M represents the selected *oversampling factor* adopted in *time-domain processing*. Then, the vector $\mathbf{x}_{ZP}^{(v)}$ (2.49) feeds a N_0-th order *discrete Fourier transform* (DFT); this produces the N_0-dimensional vector

$$\mathbf{X}^{(v)} \triangleq \left[X_0^{(v)}, X_1^{(v)}, ..., X_{N_0-1}^{(v)}\right]^T, \tag{2.50}$$

where

$$X_l^{(v)} = \frac{1}{N_0}\bar{X}^{(v)}(\bar{f}_l) \tag{2.51}$$

$\bar{X}^{(v)}(f)$ is defined by eq. (2.46) and

$$\bar{f}_l \triangleq \frac{l}{N_0 T_s}, \tag{2.52}$$

with $l = 0, 1, ..., N_0 - 1$. Finally, the N_0-dimensional vector

$$\mathbf{P} \triangleq [P_0, P_1, ..., P_{N_0-1}]^T, \tag{2.53}$$

where

$$P_l \triangleq \frac{M^2}{2}\left[\left|X_l^{(0)}\right|^2 + \left|X_l^{(1)}\right|^2\right], \tag{2.54}$$

with $l = 0, 1, ..., N_0 - 1$, is computed; note that the quantity P_l (2.54) represents a sort of *average power spectrum* evaluated at the frequency $\bar{f}_l$ (2.52).

2. *Target detection* - The problem

$$\hat{l} = \arg \max_{\tilde{l} \in \{0,1,...,N_0/2\}} P_l \tag{2.55}$$

is solved and a target is detected if the condition

$$P_{\hat{l}} > P_{th} \tag{2.56}$$

is satisfied, where P_{th} is a proper threshold. When this occurs, the next step is executed; otherwise, the algorithm stops.

3. *Estimation of target coordinates* - The estimate

$$\hat{f}_m = \frac{\hat{l}}{N_0 T_s} \tag{2.57}$$

of the frequency f_m (2.47) and the estimate

$$\hat{C}^{(v)} = M\,X_{\hat{l}}^{(v)} \tag{2.58}$$

of the complex amplitude $C^{(v)}$ (2.37) (with $v = 0$ and 1) are computed. Then, the estimate (see eq. (2.47))

$$\hat{R} = \hat{f}_m \frac{c}{2\mu} \tag{2.59}$$

of the target range R and the estimate (see eq. (2.42))

$$\hat{\theta} = \arcsin\left(\frac{\lambda}{4\pi d}\Delta\hat{\psi}_{0,1}\right) \tag{2.60}$$

of the target azimuth θ are evaluated; here,

$$\Delta\hat{\psi}_{0,1} = \angle X_{\hat{l}}^{(1)} \left(X_{\hat{l}}^{(0)}\right)^{*}. \tag{2.61}$$

represents an estimate of $\Delta\psi_{0,1}$ (2.42) and its expression is based on eqs. (2.45), (2.51) and (2.58).
This concludes the description of the proposed detection and estimation algorithm. It is important to point out that:

a) The accuracy achievable in range estimation is influenced by the DFT order N_0 and, consequently, for a given N, by the oversampling factor M. Increasing the value of the parameter M leads to a more refined analysis of the spectrum $\bar{X}^{(v)}(f)$ (2.46) and, consequently, allows to locate the spectral peak originating from the target with higher accuracy; however, this result is achieved at the price of an higher computational cost.

b) The estimate $\hat{\phi}$ (2.60) is *unambiguous* if the condition (2.44) is satisfied or if, for a given $d > \lambda/4$, the azimuth θ belongs to the interval $[-\theta_m, \theta_m)$, where (see eq. (2.43))

$$\theta_m \triangleq \arcsin\left(\frac{\lambda}{4\,d}\right) \tag{2.62}$$

c) Eq. (2.41) has not been exploited to compute an estimate of the target azimuth. This is due to the fact that the quality of this estimate is limited by the accuracy of frequency estimation on each antenna; such an accuracy, in turn, is intrinsically limited by the DFT order N_0.

d) Assuming that the target reflectivity observed on the two antennas is approximately the same (i.e., if $a^{(0)} = a^{(1)} = a$ as assumed in eq. (2.14)-(2.26)), an estimate of it can be computed as (see eqs. (2.37) and (2.58))

$$\hat{a} \triangleq M\left[\left|X_{\hat{l}}^{(0)}\right| + \left|X_{\hat{l}}^{(1)}\right|\right]. \tag{2.63}$$

e) The estimation of the azimuth characterizing the echo from a specific target requires at least two RX antennas, since it is based on computation of the phase variation observed at a specific frequency on at least two receive antennas (see eqs. (2.60) and (2.61)).

f) The maximum number of detectable targets depends on the number of virtual elements of the whole array. It is worth noting that, unlike a phased array system, where a single waveform is transmitted, a MIMO radar system endowed with N_T different TX antennas can radiate N_T independent signals. This leads to the conclusion that the maximum number of targets that be can uniquely identified by a MIMO radar is N_T times larger than that of its counterpart employing a phased array [9], if the first system employs an ULA whose virtual elements do not overlap (like the ULAs shown in Fig. 2.8).

The estimation accuracy achieved by the considered radar system can be improved by increasing the size of its ULA, i.e. the overall number of its antennas, so that a larger number of virtual channels becomes available. For instance, if the ULA shown in Fig. 2.8-a) is replaced by the one represented in Fig. 2.8-b) (and characterized by $N_T = 1$ and $N_R = 4$), $N_V = 4$ virtual channels become available, i.e. the overall number of virtual antennas is doubled with respect to the previous case. Note that this results not only in an increase of the maximum number of detectable targets, but also in an improvement of the *angular resolution* $\Delta\theta$, defined as the minimum angular separation below which the DOAs of two distinct targets cannot be separated. More specifically, if an ULA is used and the bore-sight direction is considered, we have that (e.g., see [61, Par. 4, eq.

(51)])

$$\Delta\theta = \frac{\lambda}{2\,d\;(N_V-1)\cos(\theta)}. \tag{2.64}$$

It is also worth mentioning that the algorithm illustrated above for a pair of virtual channels can be promptly extended to the case of an ULA providing N_V virtual channels. The only relevant modification concerns step 3., since the N_V-dimensional vector

$$\hat{\mathbf{C}} = \left[\hat{C}^{(0)}, \hat{C}^{(1)}, ..., \hat{C}^{(N_V-1)}\right]^T, \tag{2.65}$$

where $\hat{C}^{(v)}$ is still expressed by eq. (2.58) for any v, becomes available and, consequently, (N_V-1) phase variations, referring to the (N_V-1) distinct pairs of adjacent virtual antennas can be evaluated. If we assume that the variations of the target reflectivity over the whole virtual array are negligible and that the SNR on each virtual antenna is high, such variations are approximately constant, being all expressed by the RHS of eq. (2.42). This means that a *phase modulation*, characterized by the *normalised spatial frequency*

$$F = 2\frac{d}{\lambda}\sin(\theta), \tag{2.66}$$

is observed in the sequence $\{\hat{C}^{(v)};\, v = 0, 1, ..., N_V-1\}$. An estimate of the parameter F can be computed by exploiting, once again, the *periodogram method*. In practice, this requires executing the following three steps:

1. DFT *processing* - The vector $\hat{\mathbf{C}}$ is zero padded by appending to it a null vector of size $(M_A-1)N_V$, where M_A represents the *oversampling factor* adopted in *spatial processing*; this produces the $\bar{N}_0$-dimensional vector $\hat{\mathbf{C}}_{ZP}$, where $\bar{N}_0 \triangleq M_A N_V$. The vector $\hat{\mathbf{C}}_{ZP}$ feeds a $\bar{N}_0$-th order DFT, generating the $\bar{N}_0$-dimensional vector

$$\mathbf{s} \triangleq \left[s_0, s_1, ..., s_{\bar{N}_0/2}, s_{-\bar{N}_0/2+1}, s_{-\bar{N}_0/2+2}, ..., s_{-2}, s_{-1}\right]^T. \tag{2.67}$$

2. *Azimuth estimation* - After solving the problem

$$\hat{p} = \arg \max_{\tilde{p}\in\{-\bar{N}_0/2+1, -\bar{N}_0/2+2, ..., \bar{N}_0/2\}} \left|s_{\tilde{p}}\right|, \tag{2.68}$$

the estimate (see eq. (2.60))

$$\hat{\theta} = \arcsin\left(2\frac{\hat{p}}{\bar{N}_0}\right) \tag{2.69}$$

of the target azimuth θ is evaluated. Note that the angular resolution provided by the DFT computed in step 1. improves as $\bar{N}_0$ increases. The deterministic algorithm and its extension illustrated above have the following relevant properties: a) their derivation is based on a well defined mathematical model; b) if they fail detecting a given target, or generate inaccurate estimates of its range and/or azimuth, the causes of such events can be identified; c) they are computational efficient. This algorithm can be extended also for the elevation and velocity estimation, supposing that a rectangular array is employed and the targets are moving. The following chapters will propose different types of deterministic and learning-based algorithms for the detection of multiple targets and the estimation of their parameters using colocated MIMO radars.

2.6 Current trends in the automotive radar technology

A car implementing at least a level three of autonomy (see Fig. 2.9) needs to be equipped with two different types of colocated and pseudo-bistatic radars: a short range, imaging, radar and a long range radar. The former, being equipped by a multitude antennas (few hundreds) on both transmit and receive side, is characterized by a wide antenna aperture and, consequently, a very high angular resolution, while the latter is characterized by a lower angular resolution, but a much higher transmitting power. In particular, short range radars are typically equipped with an URA and may detect an obstacle or a person in the three-dimensional space, estimating the range, azimuth and the elevation of the detected targets. A long range radar, on the other hand, is typically made by an ULA with a small number (few decades) of antennas. Also the waveforms of the signal transmitted by the two radars is different: the short range system is typically a stepped frequency continuous wave radar, while the long range radar typically send an FMCW signal. Both the radars topology, however, operate at very high frequency, typically in the mm-wave band, having a working frequency higher than 24 GHz; the high operating frequency allows these systems to transmit large bandwidth signals, typically higher than 2 GHz. Increasing the signal bandwidth means having a very high range resolution. Short range radar are especially suited for sensing the surrounding environment in close proximity to the vehicle and, as already stated above, they are typically installed for park-assist and blind spot detection, while long range radar, having a much higher radiated power, are useful for sensing the environment at hundred of meters in front or rear the vehicle. Nowadays, it is possible to cascade multiple long range radar front-end in order to improve the horizontal or vertical angular resolution of long range, radars. Another possible solution for increasing the angular resolution of these devices is to use *cognitive*, long range radar that are able to steer the beam transmitted by the device in precise directions of interest, where the obstacle is eventually expected.

FIGURE 2.9: Generic representation of an autonomous car endowed with multiple radar sensors and able to see in its surrounding [62].

3 Deterministic Detection and Estimation techniques

3.1 Introduction

In this chapter, different deterministic detection and estimation algorithms for colocated MIMO SFCW radars operating in TDM mode and radiating ultra-wideband signals are proposed. From a signal processing perspective, the detection and estimation algorithms developed for these radars are mainly based on the use of: a) DFT methods for estimating range and DOA ([63] and paragraph 2.5), or range and Doppler only (see [64], [65]); b) standard beamforming methods [66]. All these methods, being deterministic, offer the important advantage of a complexity substantially smaller than that of well known statistical methods like the MUSIC [32], [67] and ESPRIT techniques [33].

My work basically consisted in implementing the deterministic methods described above for a colocated MIMO radar and in comparing their performance; in particular in this chapter, a comparison between the standard beamformer and other *iterative deterministic algorithms* is proposed, together with an analysis of their accuracy and complexity for 2D and 3D imaging. More specifically, the following iterative algorithms are taken into consideration:

a) the so called CLEAN technique [68], [69];

b) the estimation algorithm proposed by M. Wax and A. Leshem in [70] (dubbed WLA in the following and closely related to the algorithm devised in [71]);

c) a *modified* version of the WLA (dubbed MWLA);

d) two estimation algorithms based on the combination of the CLEAN and the MWLA with *expectation-maximization* (EM) technique [72].

Moreover, the CLEAN and MWLA techniques are also extended for 3D images, i.e., to the estimation of range, azimuth and elevation of a target detected in a 3D scenario. These algorithms may play an important role in a number of applications for various reasons; in fact, they process a single snapshot, in principle do not assume a prior knowledge of the number of targets and, unlike the *maximum likelihood* (MLI) approach, involve one-dimensional or 2D maximizations only. Moreover, the numerical results, based on the experimental data acquired from a low-power radar device, show that they are able to achieve a good accuracy at a reasonable computational cost. It is also important to point out that:

1) the application of the standard beamformer and of the CLEAN technique to SFCW radars has not been investigated previously;

2) the WLA has been originally proposed for range and DOA estimation in a narrowband system and its adaptation to a wideband radar system is analyzed for the first time;

3) the MWLA and the estimation algorithms based on combining the CLEAN algorithm (or the MWLA) with the EM technique are new.

The remaining part of this chapter is organized as follows. The signal models for 2D and 3D imaging associated with the considered radar array and for the measurements acquired through a MIMO SFCW radar are illustrated in Section 3.2. The above mentioned estimation methods for 2D (3D) imaging, together with a detailed study about the computational complexity are described in Section 3.3 (3.4); various numerical results about the accuracy they achieved are discussed in Section 3.5. Finally, some conclusions are offered in Section 3.6.

3.2 Considered signal model

The deterministic methods described in this chapter have been applied to data collected through a colocated MIMO SFCW radar; this device is endowed with: a) a virtual *uniform linear array* for the 2D detection and estimation of L targets; b) a virtual *uniform rectangular array* for the 3D detection and estimation of L targets. Moreover, a static or quasi-static scenario is considered, so that the influence of the speed on the received radar signal can be neglected; as described in Par. 2.4.2, it is assumed that this device is able to provide an *estimate* of the frequency response of the communication channel between each *transmit-receive* antenna pair by sounding it at N equally spaced frequencies, where the n-th frequency can be expressed as $f_n = f_0 + n\Delta f$ (see eq. (2.24)). Considering the general problem of 3D imaging, in which the considered radar is equipped with an URA, the channel estimates between the t-th TX antenna and all the associated receive antennas at the n-th frequency are collected in $N_R[t]$-dimensional vector

$$\tilde{\boldsymbol{H}}_t[n] \triangleq \left[\tilde{H}_{t,r_1}[n], \tilde{H}_{t,r_2}[n], ..., \tilde{H}_{t,r_{N_R[t]}}[n]\right]^T \tag{3.1}$$

$$= \sum_{l=0}^{L-1} h_l \, \mathbf{a}_t(\theta_l, \phi_l, f_n) \, b(\tau_l, f_n) + \boldsymbol{n}_t[n], \tag{3.2}$$

where $\boldsymbol{n}_t[n]$ is a $N_R[t]$-dimensional additive Gaussian noise vector and h_l, θ_l and ϕ_l denote the complex amplitude (accounting for both attenuation and phase shift), the azimuth and the elevation, respectively, of the l-th target (with $l = 0, 1, ..., L-1$). Here,

$$\mathbf{a}_t(\theta_l, \phi_l, f_n) \triangleq [a_{t,r_1}(\theta_l, \phi_l, f_n), a_{t,r_2}(\theta_l, \phi_l, f_n), ..., a_{t,r_{N_R[t]}}(\theta_l, \phi_l, f_n)]^T \tag{3.3}$$

denotes the $N_R[t]$-dimensional *steering vector* associated with the considered TX antenna. The r-th element of the last vector can be expressed as

$$a_{t,r}(\theta_l, \phi_l, f_n) \triangleq \exp(-j2\pi f_n \Delta\tau_{t,r}) \tag{3.4}$$

with $r = 0, 1, ..., N_R - 1$; here,

$$\Delta\tau_{t,r} \triangleq \frac{2}{c}\left[x_{t,r}\,\sin(\theta_l)\cos(\phi_l) + y_{t,r}\,\sin(\phi_l)\right], \tag{3.5}$$

c is the speed of light, θ_l (ϕ_l) is the *azimuth* (*elevation*) of the l-th target, $x_{t,r}$ and $y_{t,r}$ represent the coordinates of the virtual antenna associated with the t-th TX antenna and

the r-th RX antenna (also indicated as x_v and y_v in paragraph 2.3.1 in eq. (2.1) and eq. (2.2)), and

$$b\left(\tau_l, f_n\right) \triangleq \exp\left(-j2\pi f_n \tau_l\right) \tag{3.6}$$

is a function depending on the delay $\tau_l = 2R_l/c$ associated with the range R_l of the l-th target (i.e., with its distance from the center of the URA). It is important to note that the (t,r)-th element $\tilde{H}_{t,r}[n]$ in eq. (3.2) can be seen as another possible representation of $x_{c,n}^{(v)}$ in eq. (2.25); in fact, for the v-th virtual channel of the array associated to the t-th (r-th) transmit (receive) antenna (the index pair (t,r) can be replaced with v),

$$\begin{aligned} a_v\left(\theta_l, \phi_l, f_n\right) b(\tau_l, f_n) &\triangleq \exp\left[-j2\pi f_n \left(\tau_l + \Delta\tau^{(v)}\right)\right] \\ &= \exp\left[-j\left(2\pi\left(f_0 + n\Delta f\right)\tau_l^{(v)}\right)\right] \\ &= \exp\left[-j\left(2\pi n F_l^{(v)} + \psi_l^{(v)}\right)\right] \end{aligned} \tag{3.7}$$

and $h_l = a_l \exp(-j\psi_0)$; assuming that the term ψ_0 is incorporated in the phase term $\psi_l^{(v)}$ (see eq. (2.12)), the following equivalent equation can be derived

$$\tilde{H}^{(v)}[n] = x_{c,n}^{(v)} \triangleq \sum_{l=0}^{L-1} A_l^{(v)} \exp\left(-j2\pi n F_l^{(v)}\right) + n_v[n], \tag{3.8}$$

where $n_v[n]$ corresponds to $w_{c,n}^{(v)}$ of eq. (2.25) and $A_l^{(v)}$ ($F_l^{(v)}$) is still defined by eq. (2.26) (eq. (2.27)). Starting from the general expression of eq. (3.2), it is possible to derive an equivalent model to be employed in the case of a MIMO SFCW radar equipped with a single ULA, i.e.

$$\tilde{\mathbf{H}}_t[n] \triangleq \sum_{l=0}^{L-1} h_l\, \mathbf{a}_t\left(\theta_l, f_n\right)\, b(\tau_l, f_n) + \mathbf{n}_t[n], \tag{3.9}$$

where $b(\tau_l, f_n)$ is given by eq. (3.6) and $\mathbf{a}_t\left(\theta_l, f_n\right)$ is the steering vector associated with the considered TX antenna. Its r-th element can be expressed as

$$a_{t,r}\left(\theta_l, f_n\right) \triangleq \exp\left(-j2\pi f_n \Delta\tau_{t,r}\right) \tag{3.10}$$

with $r = 0, 1, ..., N_R - 1$ and

$$\Delta\tau_{t,r} \triangleq \frac{2}{c}\left[x_{t,r}\, \sin(\theta_l)\right]. \tag{3.11}$$

This last expression and consequently (3.10) differs from (3.5) and (3.4), respectively, because the dependence on the elevation angle is no more considered. It is worth noting that the 2D model expressed by eq. (3.9) is used in paragraph 3.3 for developing the 2D imaging algorithms, while the model (3.2) is used in paragraph 3.4 for 3D imaging.

3.3 Bi-dimensional deterministic estimation methods

In this paragraph, various mathematical details about range and azimuth estimation algorithms for 2D imaging are provided; then, their computational complexity is analysed in detail (see subsection 3.3.5). In the following, these assumptions are made, for simplicity: a) a single transmit antenna (in particular, the t-th TX antenna) and multiple

(namely, $N_R[t]$) receive antennas are exploited for range and DOA estimation and b) the radar received signal is expressed by (3.9).

3.3.1 Standard beamformer

The standard beamforming algorithm employed in this work is based on the *cost function* (e.g., see [69] and [59])

$$J_t\left(\tilde{\theta},\tilde{\tau}\right) \triangleq \left|S_t\left(\tilde{\theta},\tilde{\tau}\right)\right|^2, \tag{3.12}$$

where

$$S_t\left(\tilde{\theta},\tilde{\tau}\right) \triangleq \sum_{n=0}^{N-1}\sum_{k=1}^{N_R[t]} \tilde{H}_{t,r_k}[n]\, a^*_{t,r_k}(\tilde{\theta},f_n) b^*(\tilde{\tau},f_n). \tag{3.13}$$

$\tilde{H}_{t,r_k}[n]$, $a_{t,r_k}(\tilde{\theta},f_n)$ and $b(\tilde{\tau},f_n)$ are defined by eq. (3.9), (3.10) and (3.6), respectively. It is well known that the cost function $J_t(\tilde{\theta},\tilde{\tau})$ (3.12) describes the power density distribution of the received signal versus the azimuth $\tilde{\theta}$ and the delay $\tilde{\tau}$. For this reason, estimates of the target parameters $\{(\theta_l,\tau_l)\}$ (and also, approximately, of the gains $\{h_l\}$) can be computed by identifying the positions of its local maxima; note also that, if a local maximum is found at $(\tilde{\theta},\tilde{\tau}) = (\hat{\theta},\hat{\tau})$, the estimate

$$\hat{h} = \frac{S_t\left(\hat{\theta},\hat{\tau}\right)}{N \cdot N_R[t]} \tag{3.14}$$

of the complex gain h associated with the corresponding target can be promptly evaluated. The most computationally demanding task in the implementation of this algorithm is represented by the identification of the peaks of the function $J_t(\tilde{\theta},\tilde{\tau})$ (3.12); in practice, this requires accomplishing a search over a sufficiently fine rectangular grid in a 2D space. The resolution of this method in both range and azimuth is limited by the fact that the contributions of weak echoes might not be visible in the presence of strong echoes due to spatially close targets; in fact, the local maxima originating from the former echoes might be hidden by the sidelobes associated with the latter ones. These considerations motivate the use of the estimation algorithms described in the following two subsections.

3.3.2 The CLEAN algorithm

In this subsection, the so called CLEAN algorithm, originally proposed by *J. A. Hogbom* for radio astronomy applications [68], is adapted to the measurement model (3.9) and the cost function $J_t(\tilde{\theta},\tilde{\tau})$ (3.12). The resulting algorithm exploits the same cost function as standard beamforming but, unlike it, employs an *iterative beam-removing process*. This means that, within each iteration of this algorithm, the parameters of a new target are estimated and the contribution of this target (together with those due to the targets identified in the previous iterations) are subtracted from the function $S_t(\tilde{\theta},\tilde{\tau})$ (3.13). Then, this resulting residual function is passed to the next iteration, where it is processed to identify a new target. The processing tasks executed by the CLEAN algorithm can be divided in six steps; a detailed description of each of them is provided below.

1) *Initialization* - Set the iteration index l to 0 and define (see eq. (3.9))

$$\tilde{\mathbf{H}}_t^{(0)}[n] \triangleq \left[\tilde{H}^{(0)}_{t,r_1}[n], \tilde{H}^{(0)}_{t,r_2}[n], ..., \tilde{H}^{(0)}_{t,r_{N_R[t]}}[n]\right]^T = \tilde{\mathbf{H}}_t[n]. \tag{3.15}$$

for $n = 0, 1, ..., N-1$.

2) *Computation of the cost function over a rectangular grid* - Compute the cost function (see eq. (3.12))

$$J_t^{(l)}\left(\tilde{\theta},\tilde{\tau}\right) \triangleq \left|\tilde{S}_t^{(l)}\left(\tilde{\theta},\tilde{\tau}\right)\right|^2, \tag{3.16}$$

for $\tilde{\theta} = \theta_k \triangleq \theta_0 + k\,\Delta\theta$ (with $k = 0, 1, ..., N_\theta - 1$) and $\tilde{\tau} = \tau_p \triangleq \tau_0 + p\,\Delta\tau$ (with $p = 0, 1, ..., N_\tau - 1$); here, the function $\tilde{S}_t^{(l)}(\tilde{\theta},\tilde{\tau})$ is defined as (see eq. (3.13))

$$\tilde{S}_t^{(l)}\left(\tilde{\theta},\tilde{\tau}\right) \triangleq \sum_{n=0}^{N-1}\sum_{k=1}^{N_R[t]} \tilde{H}_{t,r_k}^{(l)}[n]\; a_{t,r_k}^{*}(\tilde{\theta},f_n)\, b^{*}(\tilde{\tau},f_n), \tag{3.17}$$

and θ_0 (τ_0), $\Delta\theta$ ($\Delta\tau$) and N_θ (N_τ) represent the lower limit of the search interval considered for azimuth (delay), the step size and the overall number of values selected for $\tilde{\theta}$ ($\tilde{\tau}$). In this work, $\tau_0 = 2R_m/c$, $\tau_{N_\tau-1} = 2R_M/c$ and $\Delta\tau = 2\Delta R/c$ have been selected, where R_m , R_M and ΔR represent the minimum value, the maximum value and the step size, respectively, for the trial value of target range.

3) *Estimation of the parameters of a new target* - Perform an exhaustive search for the global maximum over the set $\{J_t^{(l)}(\tilde{\theta}_k,\tilde{\tau}_p)\}$ (consisting of $N_\theta \cdot N_\tau$ values); the coordinates of the point associated with the global maximum are denoted by $(\hat{\theta}_l,\hat{\tau}_l)$. Then, compute the estimate (see eq. (3.14))

$$\hat{h}_l = \frac{S_t^{(l)}(\hat{\theta}_l,\hat{\tau}_l)}{N \cdot N_R[t]} \tag{3.18}$$

of h_l and store the estimates $(\hat{h}_l,\hat{\theta}_l,\hat{\tau}_l)$.

4) *Threshold test for identifying false targets* - If

$$|\hat{h}_l| < T, \tag{3.19}$$

where T denotes a proper (positive) threshold, go to step 6); otherwise, proceed with the next step.

5) *Cancellation of the last identified target* - Cancel the contribution of the last target in the measured frequency response by computing the *residual frequency response*

$$\tilde{\mathbf{H}}_t^{(l+1)}[n] \triangleq \tilde{\mathbf{H}}_t^{(l)}[n] - \hat{h}_l\,\mathbf{a}_t(\hat{\theta}_l,f_n)\,b\left(\hat{\tau}_l,f_n\right), \tag{3.20}$$

for $n = 0, 1, ..., N-1$. Then, increase the iteration index l by one and go to step 2);

6) *End* - The final output provided by the algorithm is expressed by the set of values $\{(\hat{\theta}_l,\hat{\tau}_l,\hat{h}_l); l = 0, 1, ..., \bar{L}-1\}$, where $\bar{L}$, the estimate of L, is given by the last value taken on by the index l.

It is important to point out that, before executing the CLEAN algorithm, a proper value for the threshold T appearing in eq. (3.19) must be selected. In fact, on the one hand, relevant echoes might be missed if T is too large; on the other hand, false targets might be identified if T is too small. This problem can be circumvented by estimating L before running the CLEAN algorithm (so that step 4) is no more accomplished). In principle, this result can be achieved by exploiting the *minimum description length* (MDL) method [73] or the *Akaike Information Criterion* (AIC) [74]; in practice, however, this entails a significant computational cost. Finally, it is worth mentioning that the cancellation procedure expressed by eq. (3.20) may suffer from error accumulation; this is due to the fact that the effects of estimation errors accumulate over successive iterations. This may result in poor accuracy; in particular, in the presence of multiple and/or closely spaced targets this could even result in the identification of false targets. The scheme of the proposed CLEAN algorithm is described in Algorithm 1.

Algorithm 1: Two dimensional CLEAN algorithm

Input: Measured channel response: $\tilde{\boldsymbol{H}}_t$

Output: Targets position: $\left\{\left(\hat{h}_l, \hat{\theta}_l, \hat{\tau}_l\right)\right\}$, for $l = 0, 1, ..., \bar{L} - 1$.

Initialisation: Set $l \leftarrow 0$ and $\tilde{\boldsymbol{H}}_t^{(0)} = \tilde{\boldsymbol{H}}_t$;

while $(|\hat{h}_l| > T \vee l = 0)$ **do**

a- *Standard Periodogram*:

- Computation of $\tilde{S}_t^{(l)}\left[\tilde{\theta}, \tilde{\tau}\right]$ through eq. (3.16);
- Cost fuction: $J_t^{(l)}\left(\tilde{\theta}, \tilde{\tau}\right)$ through eq. (3.17);

b- *Estimation of the parameters of a new target*:

- Estimation of:$(\hat{\theta}_l, \hat{\tau}_l) = \arg\max_{\tilde{\theta} \in S_\theta, \tilde{\tau} \in S_\tau} J_t^{(l)}\left(\tilde{\theta}, \tilde{\tau}\right)$, where $S_\theta \in [\theta_0\,;\, \theta_0 + N_\theta \Delta\theta]$ and $S_\tau \in [\tau_0\,;\, \tau_0 + N_\tau \Delta\tau]$;
- Estimation of: $\hat{h}_l = \frac{S_t^{(l)}(\hat{\theta}_l, \hat{\tau}_l)}{N \cdot N_R[t]}$ (see eq.(3.18))

c- *Cancellation of the strongest echo*:

for $r \leftarrow 1$ ***to*** $N_R[t]$ **do**

$\tilde{H}_{t,r}^{(l+1)} \leftarrow \tilde{H}_{t,r}^{(l)} - \hat{h}_l\, a_{t,r}\left(\hat{\theta}_l, f_n\right)\, b\left(\hat{\tau}_l, f_n\right)$

end

$l = l + 1$

end

return $\left\{\left(\hat{h}_l, \hat{\theta}_l, \hat{\tau}_l\right)\right\}$ *for* $l = 0, 1, ..., \bar{L} - 1$;

3.3.3 The Wax and Leshem estimation method

In this subsection, a specific instance of the estimation algorithm proposed by *M. Wax* and *A. Leshem* in ref. [70] (this algorithm is dubbed WLA in the following) is described; then, it will be also shown how this method can be modified to solve the numerical problems experienced in its use.

The WLA is an iterative method devised to solve the problem of jointly estimating the DOA and the time delays of multiple reflections occurring in a 2D multi-target scenario, in which a known *narrowband* signal is transmitted; its main feature is represented by the fact that it requires solving *one-dimensional* optimizations only. Despite the differences between the scenario described in [70, Sect. II] and the one considered in my work (in which a *wideband* signal is radiated), this algorithm can be modified to solve the estimation problem since: 1) it operates in the frequency domain (in fact, it processes the output of a DFT fed by the time domain samples of the received signal); b) the structure of the received signal vector in the frequency domain is similar to the one adopted for $\tilde{\mathbf{H}}_t[n]$ (3.2) (the structures become equivalent if $s(\omega_k) = 1$ in eq. (9) of [70, Sect. III]). A detailed derivation of the formulas employed by the WLA can be found in [70, Sect. IV]; in the following, I limit to summarise the steps it consists of and to illustrate the employed formulas. Similarly as the CLEAN algorithm, the WLA, in each of its iterations, estimates the parameters of a new target; however, unlike the CLEAN algorithm, the WLA updates the estimates of the parameters related with the previously identified targets. Let us assume now that, at beginning of the l-th iteration of the WLA

(with $l = 0, 1, ...\bar{L} - 1$, where $\bar{L}$ is an estimate of the number of targets L), the estimates $\{(\hat{h}_k^{(l-1)}, \hat{\theta}_k^{(l-1)}, \hat{\tau}_k^{(l-1)}), k = 0, 1, ..., l-1\}$ of the l triplets $\{(h_k, \theta_k, \tau_k), k = 0, 1, ..., l-1\}$ are available. Then, the processing accomplished within this iteration evolves through the four steps described below.

1. *Coarse estimation of a new DOA* - In this step, a *coarse* estimate $\check{\theta}_l^{(l)}$ of the azimuth θ_l referring to the new (i.e., to the l-th) target is computed on the basis of the *alternating projection* method illustrated in [71]. This requires:
a) computing the $N_R[t] \times N_R[t]$ *covariance matrix*

$$\mathbf{R}_{xx}^{(l)} \triangleq \frac{1}{N} \sum_{n=0}^{N-1} \tilde{\mathbf{H}}_t^{(l)}[n] \left(\tilde{\mathbf{H}}_t^{(l)}[n] \right)^H \tag{3.21}$$

of the *residual* channel response

$$\tilde{\mathbf{H}}_t^{(l)}[n] \triangleq \tilde{\mathbf{H}}_t[n] - \sum_{k=0}^{l-1} \hat{h}_k^{(l-1)} b\left(\hat{\tau}_k^{(l-1)}, f_n \right) \mathbf{a}_t(\hat{\theta}_k^{(l-1)}, f_n); \tag{3.22}$$

b) defining the *orthogonal projection matrix*

$$\mathbf{P}\left(\tilde{\boldsymbol{\theta}}^{(l)} \right) \triangleq \mathbf{A}_t \left(\tilde{\boldsymbol{\theta}}^{(l)} \right) \mathbf{D}_t \left(\tilde{\boldsymbol{\theta}}^{(l)} \right)^{-1} \mathbf{A}_t \left(\tilde{\boldsymbol{\theta}}^{(l)} \right)^H, \tag{3.23}$$

where

$$\mathbf{A}_t \left(\boldsymbol{\theta}^{(l)} \right) \triangleq \left[\mathbf{a}_t\left(\theta_0, f_r\right), \mathbf{a}_t\left(\theta_1, f_r\right), ..., \mathbf{a}_t(\theta_l, f_r) \right], \tag{3.24}$$

is a $N_R[t] \times (l+1)$ matrix, $\mathbf{D}_t(\boldsymbol{\theta}^{(l)}) \triangleq \mathbf{A}_t^H(\tilde{\boldsymbol{\theta}}^{(l)}) \mathbf{A}_t(\tilde{\boldsymbol{\theta}}^{(l)})$, $\boldsymbol{\theta}^{(l)} \triangleq [\theta_0, \theta_1, ..., \theta_l]$, $\tilde{\boldsymbol{\theta}}^{(l)} \triangleq [\hat{\theta}_0^{(l-1)}, \hat{\theta}_1^{(l-1)}, ..., \hat{\theta}_{l-1}^{(l-1)}, \tilde{\theta}]$ and f_r is a proper *reference frequency* (in this work, f_r is always equal to the central frequency f_c of the wideband radiated signal). In fact, given the matrices $\mathbf{R}_{xx}^{(l)}$ (3.21) and $\mathbf{P}(\tilde{\boldsymbol{\theta}}^{(l)})$ (3.23), the estimate $\check{\theta}_l^{(l)}$ is computed as

$$\check{\theta}_l^{(l)} = \arg \max_{\tilde{\theta} \in S_{\tilde{\theta}}} \operatorname{tr}\left(\mathbf{P}\left(\tilde{\boldsymbol{\theta}}^{(l)} \right) \mathbf{R}_{xx}^{(l)} \right), \tag{3.25}$$

where $S_{\tilde{\theta}} = [\theta_0, \theta_{N_\theta - 1}]$ is the search interval considered for the azimuth of the new target, θ_0 and $\theta_{N_\theta - 1}$ represent its lower and upper limits, respectively, and $\operatorname{tr}(\mathbf{X})$ denotes the trace of the square matrix $\mathbf{X}$.

2. *Estimation of target delays* - In this step, an estimate $\hat{\tau}_k^{(l)}$ of the delay τ_k (with $k = 0, 1, ..., l$) is evaluated by solving $(l+1)$ 1D optimization problems. This requires computing first the $(l+1)$-dimensional column vector

$$\hat{\mathbf{u}}_{t,k}^{(l)}[n] \triangleq \mathbf{D}_t \left(\check{\boldsymbol{\theta}}_k^{(l)} \right)^{-1} \mathbf{A}_t \left(\check{\boldsymbol{\theta}}_k^{(l)} \right)^H \tilde{\mathbf{H}}_t^{(l)}[n], \tag{3.26}$$

with $n = 0, 1, ..., N-1$; here, $\check{\boldsymbol{\theta}}_k^{(l)} \triangleq [\hat{\theta}_0^{(l-1)}, \hat{\theta}_1^{(l-1)}, ..., \hat{\theta}_{l-1}^{(l-1)}, \check{\theta}_l^{(l)}]$. Then, the $N \times (l+1)$ matrix $\hat{\mathbf{v}}_{t,k}^{(l)}$ is generated by stacking the N row vectors $\{(\hat{\mathbf{u}}_{t,k}^{(l)}[n])^T, n = 0, 1, ..., N-1\}$ according to their natural order (the order on which they are generated). Finally, $\hat{\tau}_k^{(l)}$ is computed as

$$\hat{\tau}_k^{(l)} = \arg \max_{\tilde{\tau} \in S_{\tilde{\tau}}} \left\| \mathbf{b}^H\left(\tilde{\tau}, f_n\right) \hat{\mathbf{v}}_{t,k}^{(l)} \right\|^2 \tag{3.27}$$

for $k = 0, 1, ..., l$, where $S_{\tilde{\tau}} \triangleq [2R_m/c, 2R_M/c]$, and $\|\mathbf{X}\|$ denotes the Euclidean norm of the complex vector $\mathbf{X}$.

3. *Fine grained estimation of target* DOA - In this step, a fine estimate $\hat{\theta}_k^{(l)}$ of θ_k (with $k = 0, 1, ..., l$) is evaluated by solving $(l+1)$ 1D optimization problems. More specifically, $\hat{\theta}_k^{(l)}$ is computed as

$$\hat{\theta}_k^{(l)} = \arg \max_{\tilde{\theta} \in S_{\tilde{\theta}}} \| \mathbf{A}_t^H(\check{\boldsymbol{\theta}}_k^{(l)}) \, \hat{\mathbf{B}}_{t,k}^{(l)} \|^2, \tag{3.28}$$

where $\check{\boldsymbol{\theta}}_k^{(l)} \triangleq [\hat{\theta}_0^{(l)}, ..., \hat{\theta}_{k-1}^{(l)}, \tilde{\theta}, \hat{\theta}_{k+1}^{(l-1)} ..., \hat{\theta}_l^{(l-1)}]$, $\hat{\mathbf{B}}_{t,k}^{(l)}$ denotes the k-th column of the $N_R[t] \times (l+1)$ matrix

$$\hat{\mathbf{B}}_t^{(l)} = \left[\sum_{n=0}^{N-1} \tilde{\mathbf{H}}_t[n] \, \mathbf{r}^H \left(n, \hat{\boldsymbol{\tau}}^{(l)} \right) \right] \left(\hat{\mathbf{C}}^{(l)} \right)^{-1}, \tag{3.29}$$

$$\mathbf{r} \left(n, \hat{\boldsymbol{\tau}}^{(l)} \right) \triangleq \left[b \left(\hat{\tau}_0^{(l)}, f_n \right), b \left(\hat{\tau}_1^{(l)}, f_n \right), \cdots, b \left(\hat{\tau}_l^{(l)}, f_n \right) \right]^T \tag{3.30}$$

is an $(l+1)$-dimensional column vector, $\hat{\boldsymbol{\tau}}^{(l)} \triangleq [\hat{\tau}_0^{(l)}, \hat{\tau}_1^{(l)}, \cdots, \hat{\tau}_l^{(l)}]$ is an $(l+1)$-dimensional row vector and

$$\hat{\mathbf{C}}^{(l)} \triangleq \sum_{n=0}^{N-1} \mathbf{r} \left(n, \hat{\boldsymbol{\tau}}^{(l)} \right) \mathbf{r}^H \left(n, \hat{\boldsymbol{\tau}}^{(l)} \right) \tag{3.31}$$

is a $(l+1) \times (l+1)$ matrix.

4. *Estimation of target gains* - In this step, the estimate

$$\hat{h}_k^{(l)} = \left\| \mathbf{A}_t \left(\hat{\boldsymbol{\theta}}_k^{(l)} \right) \right\|^{-2} \mathbf{A}_t^H \left(\hat{\boldsymbol{\theta}}_k^{(l)} \right) \hat{\mathbf{B}}_{t,k}^{(l)} \tag{3.32}$$

of the complex gain h_k is computed for $k = 0, 1, ..., l$; here, $\hat{\boldsymbol{\theta}}_k^{(l)} \triangleq [\hat{\theta}_0^{(l)}, \hat{\theta}_1^{(l)}, ..., \hat{\theta}_l^{(l)}]$ and $||\mathbf{A}_t(\hat{\boldsymbol{\theta}}_k^{(l)})||^2 = N \cdot N_R[t]$ ($||\mathbf{A}_t(\hat{\boldsymbol{\theta}}_k^{(l)})||^2 = N_R[t]$ if a single frequency is considered). This concludes the l-th iteration.

The WLA is initialized as follows. A coarse estimate $\check{\theta}_0^{(0)}$ of the first DOA (i.e., of the DOA referring to the dominant target) is computed by means of eq. (3.25), where the covariance matrix $\mathbf{R}_{xx}^{(0)}$ refers to the vector $\tilde{\mathbf{H}}_t^{(0)}[n] = \tilde{\mathbf{H}}_t[n]$ for any n (see eq. (3.9)) and $\mathbf{A}_t(\tilde{\theta}_0) \triangleq \mathbf{a}_t(\tilde{\theta}_0, f_r)$ is employed in the evaluation of the projection matrix $\mathbf{P}(\tilde{\boldsymbol{\theta}}^{(0)})$ (3.23). Then, the initial estimates $\hat{h}_0^{(0)}$ and $\hat{\tau}_0^{(0)}$ of the gain h_0 and the delay τ_0, respectively, are computed on the basis of the procedure illustrated in the second and fourth steps, respectively.

This algorithm deserves various comments. First of all, it is worth mentioning that steps 2. and 3. can be repeated multiple times within the l-th iteration of the WLA (before executing step 4.) in order to progressively refine the estimates of both the delays and the DOA; however, this results in an increase of the overall computational cost. Second, a method for limiting the number of iterations accomplished by the WLA is required. In this case, similarly as the CLEAN algorithm, the MDL or the AIC techniques can be employed to estimate L *before* executing the WLA. However, a simpler alternative to this approach, based on the evaluation of the energy

$$E_{\tilde{H}}^{(l)} = \sum_{n=0}^{N-1} \left\| \tilde{\mathbf{H}}_t^{(l)}[n] \right\|^2 \tag{3.33}$$

of the residual channel response $\tilde{\mathbf{H}}_t^{(l)}[n]$ (3.22) at the beginning of the l-th iteration (with

$l \geq 1$), can be employed for estimating L within the WLA. In fact, the new approach consists in stopping the WLA if

$$E_{\tilde{H}}^{(l-1)} - E_{\tilde{H}}^{(l)} < \varepsilon_{\tilde{H}}, \tag{3.34}$$

where $\varepsilon_{\tilde{H}}$ is a small positive quantity; this means that the new cancellation (see eq. (3.22)), based on the parameters of the target identified in the last (i.e., in the $(l-1)$-th iteration), has not resulted in a significant reduction of the residual energy.

Even if the WLA illustrated above is based on a rigorous derivation, the simulations have evidenced that, in the proposed application, it cannot be employed as it is, since it suffers from severe ill-conditioning in the presence of multiple and highly correlated impinging signals. In fact, when this occurs, some columns of the matrix $\mathbf{A}_t(\check{\boldsymbol{\theta}}^{(l)})$ (3.24) are similar and this makes the computation of the projection matrix $\mathbf{P}(\tilde{\boldsymbol{\theta}}^{(l)})$, of the vectors $\{\hat{\mathbf{u}}_{t,k}^{(l)}\}$ and, consequently, of the estimates $\{\hat{\boldsymbol{\tau}}^{(l)}\}$ (see eqs. (3.23), (3.26) and (3.27), respectively) inaccurate; this results in a poor accuracy in DOA estimation. To circumvent this problem, the WLA has been modified in way that, when estimating the parameters of a new target, the presence of all the previously identified targets is accounted for in the evaluation of the residual $\tilde{\mathbf{H}}_t^{(l)}[n]$ only (see eq. (3.22)). This means that, in the l-th iteration, the availability of the estimates $(\hat{h}_l, \hat{\theta}_l, \hat{\tau}_l)$ computed for l-th target is not exploited to refine the estimates associated with the other detected targets. For this reason, the following changes are introduced in the WLA:

a) In step 1., the residual $\tilde{\mathbf{H}}_t^{(l)}[n]$ is computed on the basis of the recursive formula

$$\tilde{\mathbf{H}}_t^{(l)}[n] \triangleq \tilde{\mathbf{H}}_t^{(l-1)}[n] - \hat{h}_{l-1}\, b\,(\hat{\tau}_{l-1}, f_n)\ \mathbf{a}_t(\hat{\theta}_{l-1}, f_r), \tag{3.35}$$

(which is employed in place of eq. (3.22)) and the matrix $\mathbf{A}_t(\tilde{\boldsymbol{\theta}}^{(l)})$ (3.24) is replaced by the vector $\mathbf{a}_t(\tilde{\theta}, f_c)$ (with $\tilde{\theta} \in S_{\tilde{\theta}}$) when computing the projection matrix $\mathbf{P}(\tilde{\boldsymbol{\theta}})$ (see eq. (3.23)); consequently, the optimization problem (3.25) can be reformulated as (see [71, Sect. IV] for a proof)

$$\check{\theta}_l = \arg\max_{\tilde{\theta} \in S_\theta} \mathbf{a}_{t,n}^H(\tilde{\theta}, f_r)\, \mathbf{R}_{xx}^{(l)}\ \mathbf{a}_{t,n}(\tilde{\theta}, f_r), \tag{3.36}$$

where $\mathbf{a}_{t,n}(\tilde{\theta}, f_r) \triangleq ||\mathbf{a}_t(\tilde{\theta}, f_r)||^{-1}\mathbf{a}_t(\tilde{\theta}, f_r)$ and $\check{\theta}_l$ corresponds to $\check{\theta}_k^{(l)}$.

b) In step 2., the matrix $\mathbf{A}_t(\check{\boldsymbol{\theta}}_k^{(l)})$ is replaced by the vector $\mathbf{a}_t(\check{\theta}_l, f_c)$ in computing the vector $\hat{\mathbf{u}}_{t,n}^{(l)}$ on the basis of eq. (3.26).

c) In step 3., the vector $\mathbf{r}(n, \hat{\boldsymbol{\tau}}^{(l)})$ (3.30) is replaced by the scalar $b(\hat{\tau}_l^{(l)}, f_n)$ in the evaluation of the matrix $\hat{\mathbf{C}}^{(l)}$(3.31) (that turns into a scalar too) and of the matrix $\hat{\mathbf{B}}_{t,k}^{(l)}$ (3.29).

The estimation algorithm resulting from all these modifications is called *modified* WLA (MWLA) in the following. It is worth mentioning that:

1. The frequency domain cancellation formula (3.22) employed in the MWLA is similar to that adopted in the CLEAN algorithm (see eq. (3.20)). For this reason, the accuracy of the MWLA is also affected by the phenomenon of error accumulation described at the end of subsection 3.3.2.

2. In the l-th iteration of the MWLA, the parameters estimated for the l-th target can be progressively refined by repeating steps 2. and 3. multiple times. However, in this case, the modified versions of eqs. (3.27) and (3.28) are employed for $k = l$ only.

3. The MWLA is faster than the CLEAN algorithm; this is mainly due to the fact that the former algorithm, unlike the latter one, requires solving only 1D optimizations

and does not always exploit the information available at the N frequencies (for instance, the projection matrix $\mathbf{P}(\tilde{\boldsymbol{\theta}}^{(l)})$ (3.23) is computed at the reference frequency only). Moreover, computer simulations have evidenced that MWLA is numerically stable and much faster than WLA and that, despite the substantial simplifications adopted in its derivation, achieves a good accuracy. The scheme of the proposed MWLA is described in Algorithm 2.

Algorithm 2: Delays, DOA Estimation: modified WAX-LESHEM algorithm

Input: Measured channel response: $\tilde{\boldsymbol{H}}_t$

Output: Targets position: $\left\{\left(\hat{h}_l, \hat{\theta}_l, \hat{\tau}_l\right)\right\}$, for $l = 0, 1, ..., \bar{L} - 1$.

Initialization:
$l \leftarrow 0$;
a- *Coarse Estimation* of $\check{\theta}_0$ based on eq. (3.36), where $R_{xx}^{(0)}$ matrix is based on (3.21) and $\tilde{\boldsymbol{H}}_{t,r}^{(0)} \leftarrow \tilde{\boldsymbol{H}}_{t,r}$;
Computation of the energy $E_{\tilde{H}}^{(0)}$ (3.33);
for $p \leftarrow 1$ ***to*** 2 **do**
  b- *Target delay estimation* $\hat{\tau}_0$ based on (3.27) and (3.26);
  c- *Fine estimation of target DOA* $\hat{\theta}_0$ (3.28);
end
d- *Estimation of target gain* $\hat{h}_0$ (3.32);
e- *Cancellation of the strongest echo*:
for $r \leftarrow 1$ ***to*** $N_R[t]$ **do**
  $\tilde{\mathbf{H}}_t^{(1)}[n] \triangleq \tilde{\mathbf{H}}_t^{(0)}[n] - \hat{h}_0\, \mathbf{a}_t(\hat{\theta}_0, f_n)\, b\left(\hat{\tau}_0, f_n\right).$
end
f- *Computation of the energy* $E_{\tilde{H}}^{(1)}$ (3.33);
$l \leftarrow 1$;
while $E_{\tilde{H}}^{(l-1)} - E_{\tilde{H}}^{(l)} > \varepsilon_{\tilde{H}}$ **do**
  a- *Coarse Estimation* of $\check{\theta}_l$ based on (3.36), where $R_{xx}^{(l)}$ matrix is baseed on (3.21);
  for $p \leftarrow 1$ ***to*** 2 **do**
    b- *Target delay estimation* of $\hat{\mathbf{u}}_{t,n}^{(l)}$ and $\hat{\mathbf{v}}_t^{(l)}$ based on (3.26) and (3.27);
    c- *Fine estimation of target DOA* $\hat{\theta}_l$ based on (3.28);
  end
  d- *Estimation of target gain* $\hat{h}_l$ based on (3.32);
  e- *Cancellation of the strongest echo*:
  for $r \leftarrow 1$ ***to*** $N_R[t]$ **do**
    $\tilde{\mathbf{H}}_t^{(l+1)}[n] \triangleq \tilde{\mathbf{H}}_t^{(l)}[n] - \hat{h}_l\, \mathbf{a}_t(\hat{\theta}_l, f_n)\, b\left(\hat{\tau}_l, f_n\right).$
  end
  f- *Computation* of the energy $E_{\tilde{H}}^{(l+1)}$ (3.33);
  $l \leftarrow l + 1$;
end
return $\left\{\left(\hat{h}_l, \hat{\theta}_l, \hat{\tau}_l\right)\right\}$, *for* $l = 0, 1, ..., \bar{L} - 1$.

3.3.4 EM-based algorithm for refining the target parameters

As illustrated in the previous two subsections, the estimates $\{(\hat{\theta}_k, \hat{\tau}_k, \hat{h}_k); k = 0, 1, ..., l\}$, referring to $l + 1$ distinct targets, are available at the end of the l-th iteration of both the CLEAN algorithm and MWLA. However, unlike WLA, no attempt is made in these two algoritms to refine the estimates $\{(\hat{\theta}_k, \hat{\tau}_k, \hat{h}_k); k = 0, 1, ..., l - 1\}$ computed in the previous iterations, once the new estimate $(\hat{\theta}_l, \hat{\tau}_l, \hat{h}_l)$ becomes available at the end of the l-th iteration. Moreover, as already mentioned at the end of the previous subsection, MWLA does not fully exploit the information available at all the transmitted frequencies, since part of its processing involves the measurements acquired at a reference frequency only; this may substantially affect its accuracy.

These considerations have motivated the work illustrated in this subsection and concerning the development of a computationally efficient technique that, based on the whole set of available measurements, can refine the estimates made available by the CLEAN algorithm or by the MWLA at the end of a) each iteration or b) their final iteration. The technique developed for this task is based on the EM algorithm [72] and has been inspired by the fact that, for a given l, the n-th measurement $\tilde{H}_{t,r}[n]$ (3.2) acquired on the r-th RX antenna can seen as the superposition of $(l + 1)$ distinct *deterministic* samples $\{h_k\, a_{t,r}(\theta_k, f_n)\, b(\tau_k, f_n); k = 0, 1, ..., l\}$ (characterized by the parameters $\{(h_k, \theta_k, \tau_k), k = 0, 1, ..., l\}$) with a noise sample (including the contribution of both the AWGN and the $(L - 1 - l)$ ignored targets). For this reason, the frequency domain measurement model (3.2) is structurally similar to the time domain model expressed by eq. (50) in ref. [75, Sect. IV] and referring to the case in which a deterministic waveform is received in the presence of channel noise and multipath. This similarity allows us to develop an EM-based iterative algorithm (called EM–*based estimator*, EMBE) potentially able to refine the estimates of the parameters $\{(\theta_k, \tau_k, h_k), k = 0, 1, ..., l\}$ starting from their initial values $\{(\hat{\theta}_k, \hat{\tau}_k, \hat{h}_k); k = 0, 1, ..., l\}$. Each of the EMBE iterations consists of an *estimation* (E) step, followed by a *maximization* (M) step; such steps are described below for the p-th iteration (with $p = 1, 2, ..., N_{EM}$, where N_{EM} denotes the overall number of iterations).

E step - This step aims at computing the $N_R[t] \times N$ matrix

$$\begin{aligned} \mathbf{X}_t^{(p)}[k] = \; & \hat{\mathbf{H}}_t\left(\tilde{h}_k^{(p-1)}, \tilde{\theta}_k^{(p-1)}, \tilde{\tau}_k^{(p-1)}\right) + \\ & + \beta_k^{(l)}\left[\tilde{\mathbf{H}}_t - \sum_{q=0}^{L-1} \hat{\mathbf{H}}_t\left(\tilde{h}_q^{(p-1)}, \tilde{\theta}_q^{(p-1)}, \tilde{\tau}_q^{(p-1)}\right)\right] \end{aligned} \tag{3.37}$$

for $k = 0, 1, ..., l$; here, $(\tilde{\theta}_q^{(p-1)}, \tilde{\tau}_q^{(p-1)}, \tilde{h}_q^{(p-1)})$ denotes the estimate of (θ_q, τ_q, h_q) computed in the previous (i.e., in the $(p-1)$-th) iteration of the EMBE for any q, $\tilde{\mathbf{H}}_t$ is the $N_R[t] \times N$ measurement matrix resulting from the ordered concatenation of the N column vectors $\{\tilde{\mathbf{H}}_t[n], n = 0, 1, ..., N - 1\}$,

$$\hat{\mathbf{H}}_t\left(\tilde{h}, \tilde{\theta}, \tilde{\tau}\right) \triangleq \tilde{h}\; \mathbf{M}_t\left(\tilde{\theta}, \tilde{\tau}\right), \tag{3.38}$$

$\mathbf{M}_t(\tilde{\theta}, \tilde{\tau})$ represents an $N_R[t] \times N$ matrix, whose element lying on its r-th row and its n-th column is defined as

$$m_{t,r,n}(\tilde{\theta}, \tilde{\tau}) \triangleq a_{t,r}\left(\tilde{\theta}, f_n\right)\; b\left(\tilde{\tau}, f_n\right), \tag{3.39}$$

and $\{\beta_k^{(l)}, k = 0, 1, ..., l\}$ are real parameters (also known as *mixing coefficients*) such that: a) $\beta_k \geqslant 0$ for any k; b)

$$\sum_{k=0}^{l} \beta_k^{(l)} = 1. \tag{3.40}$$

M step - In this step, the *three-dimensional* (3D) optimization problem

$$\min_{\tilde{h},\tilde{\theta},\tilde{\tau}} \left\| \mathbf{X}_t^{(p)}[k] - \tilde{h}\ \mathbf{M}_t\left(\tilde{\theta},\tilde{\tau}\right) \right\|^2 \rightarrow \tilde{h}_k^{(p)}, \tilde{\theta}_k^{(p)}, \tilde{\tau}_k^{(p)} \tag{3.41}$$

is solved, so generating the new (and, hopefully, more accurate) estimate $(\tilde{\theta}_k^{(p)}, \tilde{\tau}_k^{(p)}, \tilde{h}_k^{(p)})$ of the triplet (θ_k, τ_k, h_k) (with $k = 0, 1, ..., l$). This problem can be efficiently solved by first evaluating the minimum of the cost function appearing in eq. (3.41) with respect to $\tilde{h}$, given the pair $(\tilde{\theta}, \tilde{\tau})$; this produces the estimate

$$\check{h}_k^{(p)}\left(\tilde{\theta},\tilde{\tau}\right) = \frac{1}{N_R[t]\,N} \sum_{r=0}^{N_R[t]-1} \sum_{n=0}^{N-1} X_{t,r,n}^{(p)}[k]\ \ m_{t,r,n}^{*}\left(\tilde{\theta},\tilde{\tau}\right) \tag{3.42}$$

of h_k, where $X_{t,r,n}^{(p)}[k]$ denotes the element lying on the r-th row and the n-th column of $\mathbf{X}_t^{(p)}[k]$ (3.37). Then, setting $\tilde{h} = \check{h}_k^{(p)}(\tilde{\theta},\tilde{\tau})$ in the RHS of eq. (3.41) results in the 2D optimization problem

$$\min_{\tilde{\theta},\tilde{\tau}} \left\| \mathbf{X}_t^{(p)}[k] - \check{h}_l^{(p)}\left(\tilde{\theta},\tilde{\tau}\right)\ \mathbf{M}_t\left(\tilde{\theta},\tilde{\tau}\right) \right\|^2 \rightarrow \tilde{\theta}_q^{(p)}, \tilde{\tau}_q^{(p)}, \tag{3.43}$$

that can be solved through an exhaustive search over the domain $S_{\tilde{\theta}} \times S_{\tilde{\tau}}$, where $S_{\tilde{\theta}} = [\theta_0,\ \theta_{N_\theta-1}]$ and $S_\tau \triangleq [2R_m/c,\ 2R_M/c]$. Once $\tilde{\theta}_q^{(p)}$ and $\tilde{\tau}_q^{(p)}$ become available, $\tilde{h}_q^{(p)} = \check{h}_k^{(p)}(\tilde{\theta}_q^{(p)}, \tilde{\tau}_q^{(p)})$ is computed on the basis of eq. (3.42). This concludes the M step and, consequently, the p-th iteration of the EMBE. At the end of the last (i.e., of the N_{EM}-th) iteration of the EMBE, the new estimate

$$(\hat{\theta}_k, \hat{\tau}_k, \hat{h}_k) = \left(\tilde{\theta}_k^{(l,N_{EM}+1)}, \tilde{\tau}_k^{(l,N_{EM}+1)}, \tilde{h}_k^{(l,N_{EM}+1)}\right) \tag{3.44}$$

of the triplet (θ_k, τ_k, h_k) becomes available (with $k = 0, 1, ..., l$). Note also that the initialization of the first iteration is simply accomplished by setting $(\tilde{\theta}_k^{(0)}, \tilde{\tau}_k^{(0)}, \tilde{h}_k^{(0)}) = (\hat{\theta}_k, \hat{\tau}_k, \hat{h}_k)$ for $k = 0, 1, ..., l$.

The EMBE deserves the following comments:

1. It does not have to be executed at the end of each iteration of the CLEAN algorithm or of the MWLA; for instance, it can employed once the most important targets (i.e., the targets most contributing to the initial energy $E_{\tilde{H}}^{(0)}$; see eqs. (3.15) and (3.33)) have been identified. Postponing the parameter refinement based on the EMBE can have a significant impact on the computational cost of the overall estimation procedure.

2. Its *accuracy* and *convergence* are influenced by the values of the mixing coefficients $\{\beta_k^{(l)}, k = 0, 1, ..., l\}$. The simplest choice for these coefficients consists in assigning the same value to all of them, so that $\beta_k^{(l)} = 1/(l+1)$ (with $k = 0, 1, ..., l$). However, the computer simulations have evidenced that the estimation accuracy can be improved by selecting

$$\beta_k^{(l)} \propto |\hat{h}_k|^2 \tag{3.45}$$

for $k = 0, 1, ..., l$; in this case, the second term appearing in the RHS of eq. (3.37) plays a more important role in the case of stronger echoes. Note also that no rule is available for a priori selecting a proper value for the parameter N_{EM}; however, it should be expected that the EMBE convergence becomes slower as l increases.

3. Its most computationally intensive task is represented by the solution of the 2D optimization problem (3.43). The computational complexity of this problem can be mitigated by restricting the search domain from $S_{\tilde{\theta}} \times S_{\tilde{\tau}}$ to $[\tilde{\theta}_q^{(p-1)} - \Delta\theta_{EM}/2,\ \tilde{\theta}_q^{(p-1)} + \Delta\theta_{EM}/2] \times [\tilde{\tau}_q^{(p-1)} - \Delta\tau_{EM}/2,\ \tilde{\tau}_q^{(p-1)} + \Delta\tau_{EM}/2]$, where the parameter $\Delta\theta_{EM}$ ($\Delta\tau_{EM}$) represents the size of the new azimuth (delay) domain (the corresponding size for the range domain is $\Delta R_{EM} = 2c/\Delta\tau_{EM}$); the resolutions adopted for the range and the azimuth over the new domain are denoted by R_{res} and θ_{res}, respectively. This strategy is motivated by the fact that the quality of the initial estimates provided by the CLEAN algorithm (or by the MWLA) to the EMBE is usually good; consequently, it is expected that the new estimates $\tilde{\theta}_q^{(p)}$ and $\tilde{\tau}_q^{(p)}$ generated by the EMBE in its p-th iteration will not be too far from $\tilde{\theta}_q^{(p-1)}$ and $\tilde{\tau}_q^{(p-1)}$, respectively. A potential alternative to this approach is represented by the use of *interpolation techniques* in the search of the local minima of the cost function appearing in eq. (3.43) (see [76, Sec. IV]); however, this possibility is not discussed further and is left for future research. The scheme of the EMBE algorithm is shown in Algorithm 3.

Algorithm 3: Expectation-Maximization algorithm: EMBE algorithm

Input: $\{(h_k, \theta_k, \tau_k)\}$, with $k = 0, 1, ..., l$;

Output: $\left\{\left(\tilde{\theta}_k^{(l,N_{EM}+1)}, \tilde{\tau}_k^{(l,N_{EM}+1)}, \tilde{h}_k^{(l,N_{EM}+1)}\right)\right\}$, with $k = 0, 1, .., l$.

1 *Initialization*:

$l \leftarrow 0$;

$\left(\tilde{h}_k^{(0)}, \tilde{\theta}_k^{(0)}, \tilde{\tau}_k^{(0)}\right) = \left(\hat{h}_k, \hat{\theta}_k, \hat{\tau}_k\right)$ for $k = 0, 1, ..., l$;

for $p \leftarrow 1$ ***to*** N_{EM} **do**

  for $k \leftarrow 0$ ***to*** l **do**

    a- *E step*:

    - Computation of $\hat{\boldsymbol{H}}_t\left(\tilde{h}, \tilde{\theta}, \tilde{\tau}\right)$ based on eq. (3.38), eq. (3.39), eq. (3.42);
    - Computation of $\mathbf{X}_t^{(p)}[k]$ based on eqs. (3.37) and (3.40);

    b- *M step*:

    - Computation of $\left(\tilde{h}_k^{(p)}, \tilde{\theta}_k^{(p)}, \tilde{\tau}_k^{(p)}\right)$ based on eqs. (3.41)-(3.43).

  end

end

return $\left\{\left(\tilde{\theta}_k^{(l,N_{EM}+1)}, \tilde{\tau}_k^{(l,N_{EM}+1)}, \tilde{h}_k^{(l,N_{EM}+1)}\right)\right\}$, *with* $k = 0, 1, .., l$.

3.3.5 Computational complexity

The complexity of all the iterative algorithms described in the previous paragraphs has been carefully assessed in terms of number of *floating operations* (flops) to be executed in the detection of L targets. The general criteria adopted in estimating the computational cost of an algorithm are the same as those illustrated in [77, App. A, p. 5420] and a detailed analysis of the costs of the different tasks accomplished by each iterative

algorithm is also provided (see [61, Appendix]). This analysis leads to the conclusion that the computational cost of the CLEAN algorithm, of MWLA, of EMBE, of CLEAN combined with EMBE, of MWLA combined with EMBE, and of MUSIC (see [67]) are approximately of order $\mathcal{O}(N_{CL})$, $\mathcal{O}(N_{MW})$, $\mathcal{O}(N_{EMBE})$, $\mathcal{O}(N_{CLE})$, $\mathcal{O}(N_{MWE})$ and $\mathcal{O}(N_{MU})$, respectively, with

$$N_{CL} = 6\,N\,N_{\tilde{\theta}}\,N_{\tilde{\tau}}\,N_v + 2\,N_{\tilde{\tau}}\,N_{\tilde{\theta}}\,N + 2\,N_{\tilde{\tau}}\,N_{\tilde{\theta}}\,N_v + 15\,N\;N_v \tag{3.46}$$

$$N_{MW} = N_v^2\;(6\,N + 8\,N_\theta) + 30\,N\,N_v + 8\,N_{\tilde{\tau}}N + 30\,N_{\tilde{\theta}}\,N_v \tag{3.47}$$

$$N_{EMBE} = L\;\left[28\,N_v\,N\,N_{\tilde{\tau},EM}\,N_{\tilde{\theta},EM}\right] + 2\,L^2\,N_v\,N \tag{3.48}$$

$$N_{CLE} = 6\,L\,N_v\,N\,N_{\tilde{\tau}}\,N_{\tilde{\theta}} + 28\,L\,N_{EM}\,N_v\,N\,N_{\tilde{\tau},EM}\,N_{\tilde{\theta},EM} \tag{3.49}$$

$$N_{MWE} = L\,N_v^2\;(6\,N + 8\,N_\theta) + 28\,L\,N_{EM}\,N_v\,N\,N_{\tilde{\tau},EM}\,N_{\tilde{\theta},EM} \tag{3.50}$$

and

$$N_{MU} = 17\,(l_1\,l_2)^3 + 16\;(l_1\,l_2)^2\,(p_1\,p_2 + N_{\tilde{\tau}}\,N_{\tilde{\theta}})\,. \tag{3.51}$$

The parameters l_1, l_2, p_1 and p_2 appearing in the last formula are the same as those defined in [67, Sec IV].

3.4 Three-dimensional deterministic methods

In this section, the range and DOA estimation algorithms investigated in Sect. 3.3 are adapted for solving a three-dimensional imaging problem. In the following, these conditions are assumed, for simplicity: a) a single TX antenna (in particular, the t-th TX antenna) and multiple (namely, $N_R[t]$) RX antennas of the URA are exploited for range and DOA estimation; b) the radar received signal is expressed by eq. (3.2).

3.4.1 Standard beamformer

Standard beamformer for 3D detection and estimation requires the evaluation of the *cost function* (e.g., see eq. (3.12) in Sect. 3.3, where a 2D scenario is considered)

$$J_t\left(\tilde{\theta},\tilde{\phi},\tilde{\tau}\right) \stackrel{\Delta}{=} \left|S_t\left(\tilde{\theta},\tilde{\phi},\tilde{\tau}\right)\right|^2, \tag{3.52}$$

with

$$S_t\left(\tilde{\theta},\tilde{\phi},\tilde{\tau}\right) \stackrel{\Delta}{=} \sum_{n=0}^{N-1}\sum_{k=1}^{N_R[t]} \tilde{H}_{t,r_k}[n]\;a^*_{t,r_k}\left(\tilde{\theta},\tilde{\phi},f_n\right)b^*\left(\tilde{\tau},f_n\right), \tag{3.53}$$

over a grid in the 3D space of the trial variables $\tilde{\theta}$, $\tilde{\phi}$ and $\tilde{\tau}$. Note that the selection of a dense and uniform grid results in a computational effort substantially larger than the one required in a 2D scenario (see Sect. 3.3). This problem has been mitigated by adopting the interpolation method investigated in [60], [76], [78] and based on the use of *Chebyshev polynomials*. For this reason, the approximate cost function

$$Q_t\left(\tilde{\theta},\tilde{\phi},\tilde{\tau}\right) = \sum_{j=1}^{P} J_t\left(\breve{\theta}_j,\breve{\phi}_j,\tilde{\tau}\right) l_j^{(P-1)}\left(\tilde{\phi},\breve{\phi}_j\right) l_j^{(P-1)}\left(\tilde{\theta},\breve{\theta}_j\right), \tag{3.54}$$

based on the *Lagrange interpolation* formula, is evaluated in place of $J_t(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$ (3.12) over the set $\{(\tilde{\theta}_m, \tilde{\phi}_n, \tilde{\tau}_p)\}$, collecting $N_{\tilde{\theta}} \cdot N_{\tilde{\phi}} \cdot N_{\tilde{\tau}}$ points, which are uniformly spaced in the 3D domain $[\tilde{\theta}_0, \tilde{\theta}_{N_{\tilde{\theta}}-1}] \times [\tilde{\phi}_0, \tilde{\phi}_{N_{\tilde{\phi}}-1}] \times [\tilde{\tau}_0, \tilde{\tau}_{N_{\tilde{\tau}}-1}]$. Here,

$$l_j^{(P-1)}(\tilde{\theta}, \breve{\theta}_j) \triangleq \prod_{k=0, k\neq j}^{P-1} \left(\tilde{\theta} - \breve{\theta}_k\right) \left(\breve{\theta}_j - \breve{\theta}_k\right)^{-1} \tag{3.55}$$

and

$$l_j^{(P-1)}(\tilde{\phi}, \breve{\phi}_j) \triangleq \prod_{k=0, k\neq j}^{P-1} \left(\tilde{\phi} - \breve{\phi}_k\right) \left(\breve{\phi}_j - \breve{\phi}_k\right)^{-1}, \tag{3.56}$$

with $j = 1, 2, ..., P$, are Lagrange polynomials of degree P depending on the trial azimuth $\tilde{\theta}$ and elevation $\tilde{\phi}$, respectively,

$$\breve{\theta}_k \triangleq \left[(\tilde{\theta}_{N_{\tilde{\theta}}-1} - \tilde{\theta}_0)\, x'_k + (\tilde{\theta}_0 + \tilde{\theta}_{N_{\tilde{\theta}}-1})\right] / 2 \tag{3.57}$$

$$\breve{\phi}_k \triangleq \left[(\tilde{\phi}_{N_{\tilde{\phi}}-1} - \tilde{\phi}_0)\, x'_k + (\tilde{\phi}_0 + \tilde{\phi}_{N_{\tilde{\phi}}-1})\right] / 2 \tag{3.58}$$

and

$$x'_k = \cos\left(\pi(2k+1)/(2P)\right) \tag{3.59}$$

is the k-th node of the P-th order Chebyshev polynomial defined over the interval $[-1, 1]$ (with $k = 0, 1, ..., P-1$). Then, the parameters $\{(\theta_l, \phi_l, \tau_l);\, l = 0, 1, ...,\ \hat{L} - 1\}$ of the $\hat{L}$ targets detected by this method are estimated by identifying the positions of the $\hat{L}$ local maxima of the function $Q_t(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$ (3.54). Note also that, if the l-th local maximum is found at $(\tilde{\theta}, \tilde{\phi}, \tilde{\tau}) = (\hat{\theta}_l, \hat{\phi}_l, \hat{\tau}_l)$, the estimate

$$\hat{h}_l = S_t\left(\hat{\theta}_l, \hat{\phi}_l, \hat{\tau}_l\right) \left(N \cdot N_R[t]\right)^{-1} \tag{3.60}$$

of the complex amplitude associated with the l-th detected target becomes also available (note that this last expression is similar to the one reported in eq. (3.14)). Finally, it is worth mentioning that the accuracy of the adopted approximation depends on the order P of the selected Chebyshev polynomial. In fact, increasing P results in a smaller interpolation error; however, this result is achieved at the price of a higher computational effort. Moreover, increasing P the interpolation solution is more prone to overfit the data.

3.4.2 The CLEAN algorithm

In this paragraph, an extension of the CLEAN algorithm (already developed in Par. 3.3.2 for 2D imaging) to a 3D scenario is discussed. The processing tasks it executes can be divided in the six steps listed below.

1) *Initialization* - Set the iteration index l to 0 and set $\tilde{\mathbf{H}}_t^{(0)}[n] = \tilde{\mathbf{H}}_t[n]$ for $n = 0, 1, ..., N-1$ (see eq. (3.2)).

2) *Computation of the cost function* - Compute the values $\{J_t^{(l)}(\breve{\theta}_j, \breve{\phi}_j, \tilde{\tau}_p)\}$ of the function $J_t^{(l)}(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$ required in the evaluation of the approximate cost function $Q_t^{(l)}(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$ at the points $\{(\tilde{\theta}_m, \tilde{\phi}_n, \tilde{\tau}_p)\}$; here, $J_t^{(l)}(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$ ($Q_t^{(l)}(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$) denotes the function resulting from the replacement of $\tilde{\mathbf{H}}_t[n]$ with $\tilde{\mathbf{H}}_t^{(l)}[n]$ in the *right-hand side* (RHS) of eq. (3.52) (eq. (3.54)).

3) *Estimation of the parameters of a new target* - Search for the global maximum of the cost function $Q_t^{(l)}(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$ over a set of its $N_{\tilde{\theta}} \cdot N_{\tilde{\phi}} \cdot N_{\tilde{\tau}}$ values computed in the 3D space of the variables $(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$; the coordinates of the point associated with the global maximum are denoted by $(\hat{\theta}_l, \hat{\phi}_l, \hat{\tau}_l)$. Then, compute an estimate $\hat{h}_l$ of the complex gain associated with the l-th target by means of eq. (3.60).

4) *Thresholding for identifying false targets* - If $|\hat{h}_l| < T$, where T denotes a proper (positive) threshold, go to step 6), otherwise proceed with the next step.

5) *Cancellation of the last identified target* - Cancel the contribution of the last identified target by computing the *residual frequency response*

$$\tilde{\mathbf{H}}_t^{(l+1)}[n] \triangleq \tilde{\mathbf{H}}_t^{(l)}[n] - \hat{h}_l \, \mathbf{a}_t(\hat{\theta}_l, \hat{\phi}_l, f_n) \, b\left(\hat{\tau}_l, f_n\right), \tag{3.61}$$

for $n = 0, 1, ..., N-1$. Then, increase the iteration index l by one and go to step 2);

6) *End* - The final output provided by the algorithm is expressed by the set of values $\{(\hat{\theta}_l, \hat{\phi}_l, \hat{\tau}_l, \hat{h}_l); l = 0, 1, ..., \hat{L}-1\}$, where $\hat{L}$ represents an estimate of L and is expressed by the last value taken on by the iteration index l. As in the 2D imaging, when CLEAN is used, the following two relevant problems have to be kept into account; a) the selection of a proper value for threshold T (see 3.3.2); b) the error accumulation generated by the iterative cancellation procedure expressed by eq. (3.61). The scheme of the CLEAN algorithm applied for 3D imaging is shown in Algorithm 4.

Algorithm 4: Three dimensional CLEAN algorithm

Input: Measured channel response: $\tilde{\boldsymbol{H}}_t$

Output: Targets position: $\left\{\left(\hat{h}_l, \hat{\theta}_l, \hat{\phi}_l, \hat{\tau}_l\right)\right\}$, for $l = 0, 1, ..., \bar{L}-1$.

Initialisation: Set $l \leftarrow 0$ and $\tilde{\boldsymbol{H}}_t^{(0)} = \tilde{\boldsymbol{H}}_t$;

while $(|\hat{h}_l| > T \vee l = 0)$ **do**

a- *Standard Periodogram*:

- Cost fuctions $J_t^{(l)}\left(\tilde{\theta}, \tilde{\phi}, \tilde{\tau}\right)$ through eq. (3.52)-(3.53) and $Q_t\left(\tilde{\theta}, \tilde{\phi}, \tilde{\tau}\right)$ based on (3.54)-(3.56);

b- *Estimation of the parameters of a new target*:

- Estimation of $(\hat{\theta}_l, \hat{\phi}_l, \hat{\tau}_l) = \arg\max_{\tilde{\theta} \in S_\theta, \tilde{\phi} \in S_\phi, \tilde{\tau} \in S_\tau} Q_t^{(l)}\left(\tilde{\theta}, \tilde{\phi}, \tilde{\tau}\right)$, where $S_\theta \in [\theta_0\,;\, \theta_0 + N_\theta \Delta\theta]$, $S_\phi \in [\phi_0\,;\, \phi_0 + N_\phi \Delta\phi]$, and $S_\tau \in [\tau_0\,;\, \tau_0 + N_\tau \Delta\tau]$;
- Estimation of $\hat{h}_l$ based on eq. (3.60);

c- *Cancellation of the strongest echo*:

for $r \leftarrow 1$ ***to*** $N_R[t]$ **do**

$\tilde{H}_{t,r}^{(l+1)} \leftarrow \tilde{H}_{t,r}^{(l)} - \hat{h}_l \, a_{t,r}\left(\hat{\theta}_l, \hat{\phi}_l, f_n\right) \, b\left(\hat{\tau}_l, f_n\right)$

end

$l = l + 1$

end

return $\left\{\left(\hat{h}_l, \hat{\theta}_l, \hat{\phi}_l, \hat{\tau}_l\right)\right\}$ *for* $l = 0, 1, ..., \bar{L}-1$;

3.4.3 The Wax and Leshem estimation method

In paragraph 3.3.3 a specific instance of the iterative estimation *algorithm* proposed by *M. Wax* and *A. Leshem* in ref. [70] (and called WLA) has been developed for MIMO SFCW radars and a 2D propagation scenario, mitigating the ill-conditioning it suffers from in the presence of multiple and highly correlated echoes; the resulting algorithm has been dubbed *modified* WLA (MWLA). In this subsection, an extension of the MWLA to a 3D scenario is illustrated. The processing tasks executed by this method can be divided in the six steps listed below.

1) *Initialization* - Set the iteration index l to 0, and set $\tilde{\mathbf{H}}_t^{(0)}[n] = \tilde{\mathbf{H}}_t[n]$ for $n = 0, 1, ..., N-1$ (see eq. (3.2)). Then, evaluate the $N_R[t]$-dimensional steering vector $\mathbf{a}_t(\tilde{\theta}, \tilde{\phi}, f_r)$ and its normalised version $\bar{\mathbf{a}}_t(\tilde{\theta}, \tilde{\phi}, f_r) \triangleq ||\mathbf{a}_t(\tilde{\theta}, \tilde{\phi}, f_r)||^{-1}\mathbf{a}_t(\tilde{\theta}, \tilde{\phi}, f_r)$ over the set $S_I \triangleq \{(\check{\theta}_m, \check{\phi}_n)\}$, collecting $P \times P$ points whose coordinates are computed on the basis of eqs. (3.57)-(3.59), at a properly selected reference frequency f_r (typically the central frequency).

2) DOA *coarse estimation* - Evaluate the $N_R[t] \times N_R[t]$ *covariance matrix* $\mathbf{R}_{xx}^{(l)}$ of the *residual* channel response $\tilde{\mathbf{H}}_t^{(l)}[n]$ (see eq. (3.61)). Then, compute the *coarse estimate* $(\check{\theta}_l, \check{\phi}_l)$ of the angular parameters (θ_l, ϕ_l) characterizing the l-th target by solving the 2D optimization problem

$$\left(\check{\theta}_l, \check{\phi}_l\right) = \arg \max_{(\tilde{\theta}_I, \tilde{\phi}_I) \in S_I} \bar{\mathbf{a}}_t^H(\tilde{\theta}_I, \tilde{\phi}_I, f_r)\, \mathbf{R}_{xx}^{(l)}\, \bar{\mathbf{a}}_t(\tilde{\theta}_I, \tilde{\phi}_I, f_r). \tag{3.62}$$

3) *Delay estimation* - Evaluate an estimate $\hat{\tau}_l$ of the delay τ_l associated with the l-th target by solving the 1D optimization problem

$$\hat{\tau}_l = \arg \max_{\tilde{\tau} \in S_{\tilde{\tau}}} \left| \mathbf{b}^H\left(\tilde{\tau}\right)\, \hat{\mathbf{v}}_{t,l} \right|^2, \tag{3.63}$$

where $\hat{\mathbf{v}}_{t,l}$ is a N-dimensional column vector referring to the l-th target and

$$\hat{\mathbf{v}}_{t,l}[n] \triangleq K(\check{\theta}_l, \check{\phi}_l, f_r)\, \mathbf{a}_t(\check{\theta}_l, \check{\phi}_l, f_r)^H\, \tilde{\mathbf{H}}_t^{(l)}[n], \tag{3.64}$$

with $n = 0, 1, ..., N-1$, $K(\check{\theta}_l, \check{\phi}_l, f_r) \triangleq [\mathbf{a}_t^H(\check{\theta}_l, \check{\phi}_l, f_r) \cdot \mathbf{a}_t(\check{\theta}_l, \check{\phi}_l, f_r)]^{-1}$, $S_{\tilde{\tau}}$ denotes the interval $[\tilde{\tau}_0, \tilde{\tau}_{N_{\tilde{\tau}}-1}]$, $\mathbf{b}\left(\tilde{\tau}\right) = [b(\tilde{\tau}, f_0), b(\tilde{\tau}, f_1), ..., b(\tilde{\tau}, f_{N-1})]$ is a N-dimensional row vector.

4) DOA *fine estimation* - Generate a new (and, hopefully, finer) estimate $(\hat{\theta}_l, \hat{\phi}_l)$ of the pair (θ_l, ϕ_l) by solving the 2D optimization problem

$$\left(\hat{\theta}_l, \hat{\phi}_l\right) = \arg \max_{(\tilde{\theta}, \tilde{\phi}) \in S_I} |\mathbf{L}\left(\tilde{\theta}, \tilde{\phi}\right)|^2, \tag{3.65}$$

where

$$\mathbf{L}\left(\tilde{\theta}, \tilde{\phi}\right) \triangleq \mathbf{a}_t(\tilde{\theta}, \tilde{\phi}, f_r)\, \hat{\mathbf{B}}_{t,l} \tag{3.66}$$

is a $P \times P$ matrix,

$$\hat{\mathbf{B}}_{t,l} = \left[\sum_{n=0}^{N-1} \tilde{\mathbf{H}}_t[n]\, b^*\left(\hat{\tau}_l, f_n\right)\right] \hat{C}_l^{-1}, \tag{3.67}$$

is a $N_R[t]$-dimensional column vector and

$$\hat{C}_l \triangleq \sum_{n=0}^{N-1} b\left(\hat{\tau}_l, f_n\right)\, b^*\left(\hat{\tau}_l, f_n\right). \tag{3.68}$$

The computationally efficient method adopted for solving problem (3.65) is based on the Lagrange interpolation procedure illustrated in Sec. 3.4.1. For this reason, the matrix

$\mathbf{L}(\tilde{\theta},\tilde{\phi})$ appearing in the RHS of eq. (3.65) is replaced by its approximation

$$\bar{\mathbf{L}}(\tilde{\theta},\tilde{\phi}) = \sum_{j=1}^{P} \mathbf{L}\left(\check{\theta}_j,\check{\phi}_j\right)\, l_j^{(P-1)}\left(\tilde{\theta},\check{\theta}_j\right)\, l_j^{(P-1)}\left(\tilde{\phi},\check{\phi}_j\right)\,, \tag{3.69}$$

where the polynomials $l_j^{(P-1)}(\tilde{\theta},\check{\theta}_j)$ and $l_j^{(P-1)}(\tilde{\phi},\check{\phi}_j)$ are defined by eqs. (3.55) and (3.56).

5) *Gain estimation* - Compute the estimate

$$\hat{h}_l = \left\|\mathbf{a}_t(\hat{\theta}_l,\hat{\phi}_l,f_r)\right\|^{-2}\mathbf{a}_t(\hat{\theta}_l,\hat{\phi}_l,f_r)^H\,\hat{\mathbf{B}}_{t,l}. \tag{3.70}$$

of the complex amplitude h_l. If $|\hat{h}_l| < T$ (where T denotes a positive threshold), proceed with the next step, otherwise increase the iteration index l by one and go to step 2).

6) *End* - The final output provided by the algorithm is expressed by the set of values $\{(\hat{h}_l,\hat{\theta}_l,\hat{\phi}_l,\hat{\tau}_l); l = 0, 1, ..., \hat{L}-1\}$, where $\hat{L}$ represents an estimate of L and is expressed by the last value taken on by the iteration index l. Finally, it is worth mentioning that steps 3) and 4) can be repeated multiple times within the l-th iteration to progressively refine the estimates of both the delay and the DOA of the new target. The scheme of the MWLA for 3D imaging is shown in Algorithm 5.

Algorithm 5: Three-dimensional modified WAX-LESHEM algorithm

Input: Measured channel response: $\tilde{\boldsymbol{H}}_t$

Output: Targets position: $\left\{\left(\hat{h}_l,\hat{\theta}_l,\hat{\phi}_l,\hat{\tau}_l\right)\right\}$, for $l = 0,1,...,\bar{L}-1$.

Initialization: $l \leftarrow 0$;

a- *Coarse Estimation* of $\left(\check{\theta}_0,\check{\phi}_0\right)$ (3.62); *computation of the energy* $E_{\tilde{H}}^{(0)}$ (3.33);

for $p \leftarrow 1$ ***to*** 2 **do**

  b- *Target delay estimation* $\hat{\tau}_0$ based on (3.63) and (3.64);

  c- *Fine estimation of target DOA* $\left(\hat{\theta}_0,\hat{\phi}_0\right)$ (3.65), (3.69);

end

d- *Estimation of target gain* $\hat{h}_0$ (3.70) and *echo cancellation*:

for $r \leftarrow 1$ ***to*** $N_R[t]$ **do**

  $\tilde{\mathbf{H}}_t^{(1)}[n] \triangleq \tilde{\mathbf{H}}_t^{(0)}[n] - \hat{h}_0\,\mathbf{a}_t(\hat{\theta}_0,\hat{\phi}_0,f_n)\,b\left(\hat{\tau}_0,f_n\right).$

end

f- *Computation of the energy* $E_{\tilde{H}}^{(1)}$ based on (3.33); $l \leftarrow 1$;

while $E_{\tilde{H}}^{(l-1)} - E_{\tilde{H}}^{(l)} > \varepsilon_{\tilde{H}}$ **do**

  a- *Coarse Estimation* of $\left(\check{\theta}_l,\check{\phi}_l\right)$ based on (3.62);

  for $p \leftarrow 1$ ***to*** 2 **do**

    b- *Target delay estimation* $\hat{\tau}_l$ (3.63) and (3.64);

    c- *Fine estimation of target DOA* $\left(\hat{\theta}_l,\hat{\phi}_l\right)$ (3.65);

  end

  d- *Estimation of target gain* $\hat{h}_l$ (3.70) and *echo cancellation*:

  for $r \leftarrow 1$ ***to*** $N_R[t]$ **do**

    $\tilde{\mathbf{H}}_t^{(l+1)}[n] \triangleq \tilde{\mathbf{H}}_t^{(l)}[n] - \hat{h}_l\,\mathbf{a}_t(\hat{\theta}_l,\hat{\phi}_l,f_n)\,b\left(\hat{\tau}_l,f_n\right).$

  end

  f- *Computation* of the energy $E_{\tilde{H}}^{(l+1)}$ based on (3.33); $l \leftarrow l+1$;

end

return $\left\{\left(\hat{h}_l,\hat{\theta}_l,\hat{\phi}_l,\hat{\tau}_l\right)\right\}$, *for* $l = 0,1,...,\bar{L}-1$.

3.4.4 Computational complexity

The computational cost of CLEAN and that of MWLA per iteration are $\mathcal{O}(N_{CL})$ and $\mathcal{O}(N_{MW})$, respectively; here,

$$N_{CL} \approx 6\,N\,N_v\,P^2\,N_{\tilde{\tau}} + 8\,P^2\,N_{\tilde{\theta}} + 15\,N_v\,N \tag{3.71}$$

and

$$N_{MW} \approx N_v^2\,\left(6\,N + 8\,P^2\,\right) + 30\,N\,N_v + 8\,N_{\tilde{\tau}}N + 30\,P^2\,N_v + 8\,P^2\,N_{\tilde{\theta}}. \tag{3.72}$$

It is worth to note that the complexity of the CLEAN algorithm and the MWLA applied for three-dimensional imaging depends on the interpolation order P^2; even when P is low, this quantity is typically higher than the order N_θ and, hence, the complexity of the methods applied for three-dimensional imaging is much heavier respect the bi-dimensional case (see eq. (3.46)-(3.47)).

3.5 Numerical results

In this section, the main results obtained applying the deterministic algorithm described above for 2D and 3D imaging to data collected by a MIMO SFCW Radar are described.

3.5.1 Bi-dimensional imaging

The accuracy of the 2D detection and estimation algorithms illustrated in Section 3.3 has been assessed in an indoor area of small size, because of the limited power radiated by the radar device employed in the considered experiments. The employed radar has been designed and manufactured by *Vayyar Imaging Ltd Company* [62]. It operates in TDM mode and it is equipped with the *uniform rectangular array* (URA) illustrated in Fig. 3.1-a). This array consists of 20 antennas; moreover, four of them (more precisely, those identified by the numbers 1, 10, 11 and 20) can be employed as TX antennas, whereas those identified by the numbers $2-20$ as RX antennas only. The *virtual* array associated with this physical URA is shown in Fig. 3.1-b) and consists of 53 antennas; however, only the portion enclosed within the dashed rectangle and containing 17 virtual elements (forming a ULA) has been exploited by the proposed estimation algorithms. Moreover, the following values have been selected for the parameters characterizing channel sounding in the frequency domain (see eq. (2.24)): $f_0 = 5.05$ GHz, $\Delta f = 9.4$ MHz and $N = 510$ (so that the overall sweep bandwidth is about 5.0 GHz). The numerical results shown in this section refer to two different scenarios. The first scenario is characterized by five targets and, more specifically, by $L = 5$ identical metal discs, all having a diameter equal to 5.5 cm and placed on an horizontal plastic desk. The exact range and azimuth of each of these targets are listed in Table 3.1. In the second scenario, instead, $L = 4$ equal coins, having a diameter of 2.0 cm, are placed at a uniform distance from the radar device (see Fig. 3.2); their range and azimuth are listed in Table 3.2. The raw data acquired by the radar over a single snapshot have been processed by the algorithms illustrated in the previous section. The estimation accuracy of each algorithm has been assessed by evaluating the *root mean square error* (RMSE)

$$\bar{\varepsilon}_X \triangleq \sqrt{L^{-1}\sum_{l=1}^{L}\left[X_l - \hat{X}_l\right]^2} \tag{3.73}$$

and the peak error

$$\hat{\varepsilon}_X \triangleq \max_l \left| X_l - \hat{X}_l \right| \tag{3.74}$$

for the range ($X = R$) and the azimuth ($X = \theta$); here, X_l and $\hat{X}_l$ represent the exact value of the parameter to be estimated for the l-th target (with $l = 0, 1, 2, 3$ and 4) and the corresponding estimate. The analysis of *computational requirements* is based, instead, on assessing both the *computation time* (CT) and the overall computational complexity required for processing the whole set of acquired data and generating the estimates of range and azimuth for all the targets.

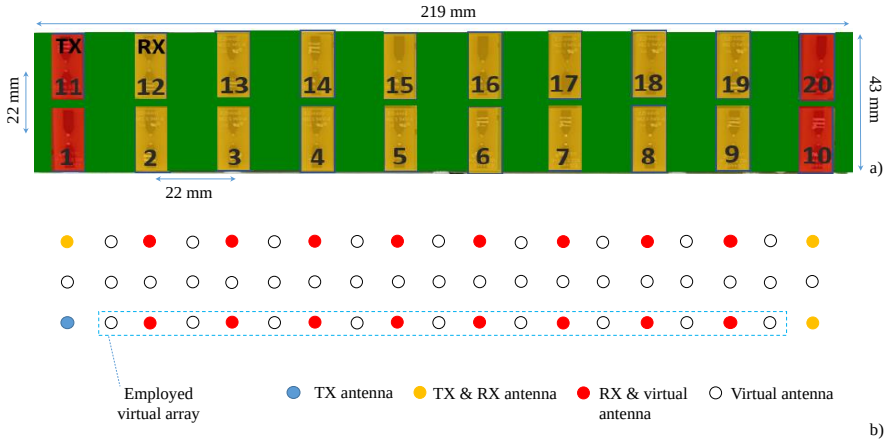


FIGURE 3.1: a) Physical URA of the employed radar device; b) representation of the associated virtual array and of the portion (enclosed in the dashed rectangle) exploited by the estimation algorithms.

Target	Params	1	2	3	4	5
Exact	$\theta(^\circ)$	-18.0	3.0	13.0	24.0	38.0
	R (cm)	82.5	81.0	82.5	73.5	53.0
MU	$\theta(^\circ)$	-24.0	-2.0	18.0	30.0	43.0
	R (cm)	86.0	85.0	87.5	80.0	58.0
CL	$\theta(^\circ)$	-16.0	2.0	12.0	23.0	35.0
	R (cm)	87.5	85.5	86.5	78.0	54.5
MW	$\theta(^\circ)$	-18.0	3.0	13.0	26.0	41.0
	R (cm)	87.0	84.5	86.5	76.0	53.0

TABLE 3.1: Azimuth and range of the targets characterizing scenario # 1, and their estimates computed by three different algorithms (MUSIC (MU), CLEAN (CL) and MWLA (MW)).

The accuracy achieved by the standard beamformer, CLEAN and MWLA in all the considered scenarios has been assessed under the assumption that the number of targets is known a priori; moreover, the following values have been selected for the parameters defining the search domain of these algorithms: a) $\theta_0 = -90^\circ$, $\Delta\theta = 1^\circ$ and $N_\theta = 181$ ($R_m = 20$ cm, $\Delta R = 0.5$ cm , $R_M = 120$ cm and $N_\tau = 201$) for the sequence $\{\theta_k\}$ ($\{R_k\}$) of trial values of the azimuth (range). A contour plot of the cost function $J_t(\tilde{\theta}, \tilde{\tau})$ (3.12) evaluated by the standard beamformer and the CLEAN algorithm in the first considered scenario is illustrated in Fig. 3.3; the estimates of the azimuth and of the range

computed by the CLEAN algorithm, the MWLA and the 2D MUSIC are listed in Table 3.1 (the acronyms CL, MW and MU refer to the CLEAN algorithm, the MWLA and the 2D MUSIC algorithm, respectively), together with the exact values of these quantities. The values of the errors evaluated for each target are shown in Fig. 3.4; the estimates computed by all the algorithms are considered. The corresponding values of $\bar{\varepsilon}_\theta$ and $\bar{\varepsilon}_R$ ($\hat{\varepsilon}_\theta$ and $\hat{\varepsilon}_R$) computed on the basis of eq. (3.73) (eq. (3.74)) are listed in Table 3.3 for all the considered algorithms and the 2D MUSIC algorithm. These results show that all the estimation algorithms achieve similar accuracies, but the best one is provided by the MWLA, which, luckily, requires the shortest CT. Note also that all the iterative algorithms require a substantially shorter CT than the MUSIC algorithm.

FIGURE 3.2: Measurement setup employed in the second scenario. Four metal coins are placed over a rectangular carton box.

Target	Params	1	2	3	4	$\bar{\varepsilon}$	$\hat{\varepsilon}$	CT (s)
Exact	$\theta(^\circ)$	-15.0	-3.0	8.0	25.0			
	R (cm)	65.0	65.0	65.0	65.0			
MU	$\theta(^\circ)$	-14.0	-2.0	6.0	23.0	1.5	2.0	>5 h
	R (cm)	66.0	67.0	68.0	70.0	3.0	5.0	>5 h
CL	$\theta(^\circ)$	-15.0	-2.0	7.0	22.0	1.6	3.0	0.5
	R (cm)	66.0	67.0	67.0	70.0	3.0	5.0	0.5
MW	$\theta(^\circ)$	-16.0	-3.0	7.0	25.0	0.7	1.0	0.01
	R (cm)	67.0	67.0	67.0	68.0	2.0	3.0	0.01
CLE	$\theta(^\circ)$	-15.3	-1.9	7.5	21.8	2.2	3.0	3.9
	R (cm)	66.0	67.0	67.0	70.0	3.0	5.0	3.9
MWE	$\theta(^\circ)$	-15.8	-2.7	7.2	23.4	0.9	1.0	3.4
	R (cm)	66.0	67.0	68.0	69.0	3.0	3.0	3.4

TABLE 3.2: Azimuth and range of the targets characterising scenario # 2, and their estimates computed by five different algorithms: MUSIC, CLEAN, MWLA, CLEAN + EMBE and MWLA + EMBE.

It is also worth mentioning that the MWLA offers a substantial advantage in terms of computational complexity/time even for a different number of targets. This can be

easily inferred from Fig. 3.5, that shows the computational complexity (assessed on the basis of eqs. (3.46)-(3.51)) and the CT characterizing all the considered iterative algorithms for a number of targets ranging from 2 to 8 (no result is shown for the MUSIC algorithm since its complexity and computation time are much higher than those of the other algorithms).

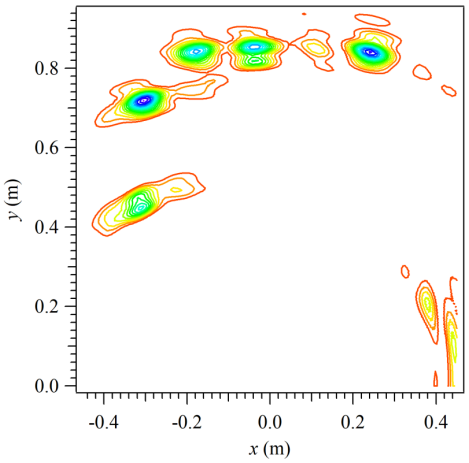
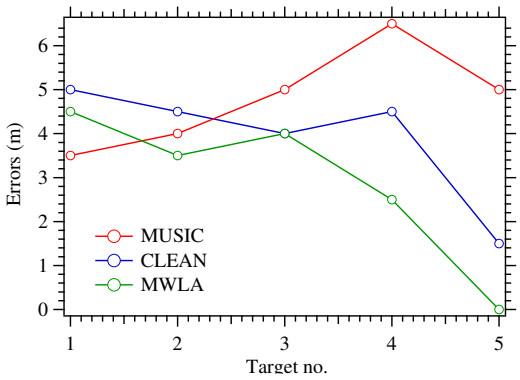
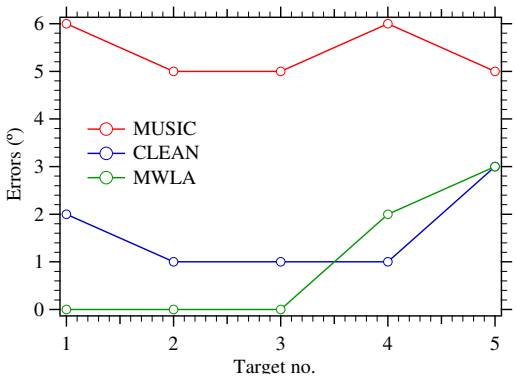


FIGURE 3.3: Contour plot (in Cartesian coordinates) of the cost function $J_t(\tilde{\theta}, \tilde{\tau})$ (3.12) evaluated by the standard beamformer for the considered propagation scenario. The peaks associated with the five targets are clearly visible.



(A) Range errors characterizing scenario # 1 considering the three different algorithms: MUSIC (MU), CLEAN (CL) and MWLA (MW).



(B) Azimuth errors characterizing scenario # 1 considering the three different algorithms: MUSIC (MU), CLEAN (CL) and MWLA (MW).

FIGURE 3.4: Representation of: a) the range errors and b) the azimuth errors characterizing scenario # 1. Three different algorithms, namely, MUSIC (MU), CLEAN (CL) and MWLA (MW) are considered.

The results shown in this figure lead to the following conclusions:

1) MWLA has the lowest complexity. This is due to the fact that this algorithm computes the steering vector at the central frequency only and its estimation of targets parameters does not involve the cost function (3.12).

2) The complexity assessed for the CLEAN algorithm is higher than that of the MWLA combined with the EMBE (MWLA+EMBE), since the most demanding task in the last algorithm involves a limited search domain, as explained at the end of subsection 3.3.4.

	CL	MW	CLE	MWE	MU
$\bar{\varepsilon}_\theta$ (°)	1.8	1.6	1.8	0.6	5.4
$\bar{\varepsilon}_R$ (cm)	4.1	3.3	4.1	3.5	4.9
$\hat{\varepsilon}_\theta$ (°)	3.0	3.0	3.6	1.7	6.0
$\hat{\varepsilon}_R$ (cm)	5.0	4.5	5.0	4.9	6.5
CT	0.7 s	0.02 s	5.3 s	4.6 s	>5 h
$GFlops$	9.46	0.01	11.4	2.0	651.7

TABLE 3.3: Root mean square errors, peak errors and computation times for all the analysed estimation algorithms. Scenario # 1 is considered.

However, the CT observed for the CLEAN algorithm is lower than that of the MWLA combined with the EMBE (MWLA+EMBE). This is due to the fact the MATLAB implementation of the former algorithm is more efficient than that of the latter one. Note, in particular, that in the latter algorithm the parameters of search domain $S_{\tilde{\theta}} \times S_{\tilde{\tau}}$ (and, consequently, the values of the steering vector) need to be re-computed any time that a new target is found. The potential improvement in estimation accuracy provided by the EMBE has been also assessed when it is employed at the end of the last (i.e., of the fourth) iteration of the CLEAN algorithm or of the MWLA, and its mixing coefficients are computed on the basis of eq. (3.45). Moreover, the following values have been selected for the parameters of the EMBE: a) $\Delta R_{EM}/2 = 0.5$ cm and $\Delta\theta_{EM}/2 = 1°$ for the size of the search domain (the adopted resolutions are $R_{res} = 0.1$ cm and $\theta_{res} = 0.1°$ for the range and the azimuth variables, respectively); b) $N_{EM} = 5$ for the overall number of iterations.

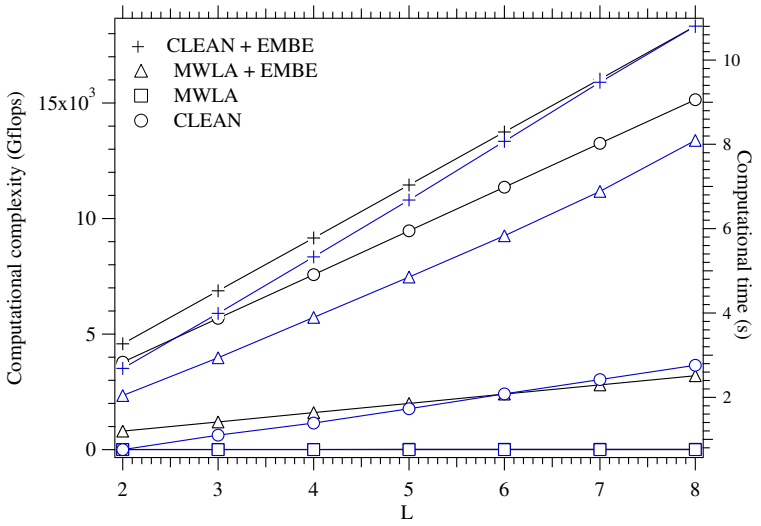


FIGURE 3.5: Computational complexity (black) and computational time (blue) versus the number of targets (L).

The estimation accuracies $\bar{\varepsilon}_\theta$ and $\bar{\varepsilon}_R$ achieved by combining EMBE with CLEAN and with MWLA are listed in Table 3.3 (where the acronyms CLE and MWE refer to the combination of CLEAN with EMBE, and to that of MWLA with EMBE, respectively). From these results it is possible to conclude that:

1) Combining the CLEAN algorithm with the EMBE does not result in a better accuracy in range estimation, and slightly improves the accuracy of azimuth estimates.

2) The best accuracy in azimuth estimation is obtained by combining the MWLA with the EMBE; in fact, this reduces the azimuth RMSE from 1.6° to 0.6°. This improvement can be related to the fact that the MWLA benefits from the *beamforming* expressed by eq. (3.42) and accomplished by the EMBE in estimating h_k, given the trial values $\tilde{\theta}$ and $\tilde{\tau}$ of the target azimuth and delay, respectively.

The computer simulations have also evidenced that, if the EMBE is used at the end of each iteration of the CLEAN algorithm (or of the MWLA), no real improvement is obtained and the computation time becomes substantially larger.

Let us focus now on the second scenario. In this case, the estimates of the azimuth and of the range computed by all the analysed algorithms, together with the corresponding RMSEs, peak errors and CTs, are listed in Table 3.2. These results lead to the conclusion that the MWLA achieves, once again, the best accuracy/CT trade-off. Moreover, the use of the EMBE in combination with the CLEAN algorithm or with the MWLA provides some local improvement in the estimation of the position of the targets; however, no significant variation in terms of RMSE and peak errors is observed. Another interesting result found in this scenario is represented by the fact that the MWLA exhibits a good accuracy (better than that of the other algorithms) even if the azimuth of the targets is not small (e.g., the azimuth of the fourth target is equal to 25.0°).

Further numerical results about multiple variants of the second scenario are given in Table 3.4. In this case, the number of targets, that are all placed at a uniform distance (equal to 65 cm) from the radar device, ranges from two to four and a uniform angular spacing between adjacent targets is selected. Moreover, six distinct configurations for the target geometry are considered (the range and azimuth of each target are given in Table 3.5). The RMSEs and the peak errors resulting from an average over the considered six configurations are listed in Table 3.4. From these results it is easily inferred that the iterative algorithms perform well and the best accuracy is achieved again by the MWLA. The last technical issue analysed in detail is represented by the behaviour of the residual energy $E_{\hat{H}}^{(l)}$ (3.33) available at the end of the l-th iteration in the CLEAN algorithm and in the MWLA. Some results referring to the first scenario are shown in Fig. 3.6.

	CL	MW	CLE	MWE	MU
$\bar{\varepsilon}_\theta$ (°)	1.4	1.2	1.7	1.2	1.3
$\bar{\varepsilon}_R$ (cm)	2.0	2.0	2.0	2.0	1.8
$\hat{\varepsilon}_\theta$ (°)	2.0	1.5	2.0	1.5	1.5
$\hat{\varepsilon}_R$ (cm)	2.8	2.8	2.8	2.8	3.0

TABLE 3.4: Average of root mean square errors and peak errors for all the investigated estimation algorithms. The six configurations described in Table 3.5 are considered for target geometry.

Note that, in this case, two distinct sets of values are given for each algorithm, one computed on the basis of the raw data acquired by the radar (i.e., of the *experimental* data), the other one evaluated on the basis of computer generated data (i.e., of *simulated* data); in the last case, five *point targets*, whose azimuth and range are the same as those in Table 3.1, have been assumed. From these results it is inferred that the iterative cancellation procedure accomplished by the CLEAN algorithm results in a steep decrease of the residual energy in the simulated case. This suggests that a stopping criterion based on the inequality (3.34) could be really used in this case. However, a substantially smaller rate of decrease is observed for the residual energy of the CLEAN algorithm in the case of experimental data. This is mainly due to the fact that the employed metal discs cannot be represented as point targets; for this reason, multiple echoes are received by the radar device from the same target.

Target	Params	1	2	3	4
Exp. 1	$\theta(^\circ)$	-11.0	4.0		
	R (cm)	65.0	65.0		
Exp.2	$\theta(^\circ)$	-25.0	-13.0	4.0	
	R (cm)	65.0	65.0	65.0	
Exp. 3	$\theta(^\circ)$	-25.0	-13.0	4.0	19.0
	R (cm)	65.0	65.0	65.0	65.0
Exp. 4	$\theta(^\circ)$	-3.0	8.0		
	R (cm)	65.0	65.0		
Exp. 5	$\theta(^\circ)$	-3.0	8.0	20.0	
	R (cm)	65.0	65.0	65.0	
Exp. 6	$\theta(^\circ)$	-15.0	-3.0	8.0	25.0
	R (cm)	65.0	65.0	65.0	65.0

TABLE 3.5: Azimuth and range of the targets in the considered six configurations (all related to scenario # 2).

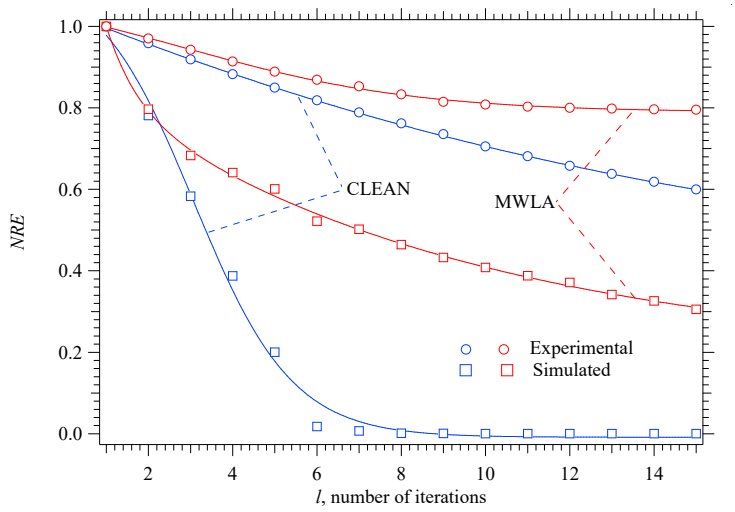


FIGURE 3.6: Normalized residual energy evaluated for the CLEAN algorithm and the MWLA algorithm for the considered propagation scenario. Both experimental and simulated data are considered.

As a matter of fact, when experimental data are processed by the CLEAN algorithm, new echoes, very close to those already identified, appear if its cancellation procedure proceeds beyond the 4-th iteration. A gap between the residual energy computed on the basis of the simulated data and that referring to the experimental data is also observed in the case of the MWLA. However, the rate of decrease is substantially smaller than that evaluated in the case of the CLEAN algorithm in both cases. This depends on the fact that the steering vector $\mathbf{a}_t(\hat{\theta}_{l-1}, f_r)$ is evaluated at the reference frequency f_r only in the computation of the residual $\tilde{\mathbf{H}}_t^{(l)}[n]$ (3.35) for any n; this unavoidably introduces an error. These results suggest that, if the MWLA is employed, the stopping criterion based on the inequality (3.34) can still used, but the threshold $\varepsilon_{\tilde{H}}$ has to be selected very carefully. Despite this problem, the MWLA and the combination of the MWLA with

EMBE represent the best options for range and azimuth estimation in SFCW MIMO radars. Finally, it is worth mentioning that, in the measurement campaigns, other scenarios, characterized by a different numbers of metal discs and target coordinates, have been also considered. However, accuracies similar to the ones measured for the two scenarios described above have been found, provided that the angular coordinates of the employed targets were contained in a limited domain (see eq. (2.64))

3.5.2 Three-dimensional imaging

The accuracy of the algorithms illustrated in Sect. 3.4 has been assessed by both computer simulation and experimental results; in doing this, the measurements acquired through a SFCW MIMO radar manufactured by *Vayyar Imaging Ltd Company* [62] and illustrated in Fig. 3.7-a) have been processed.

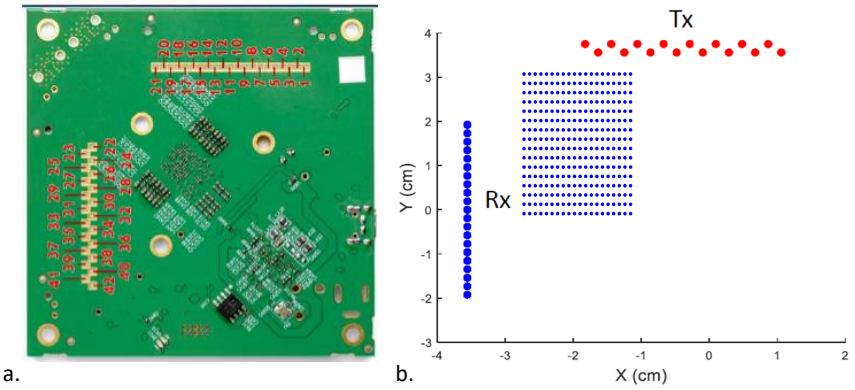


FIGURE 3.7: a. Employed radar device; b. Geometry of the physical antenna arrays and of the corresponding virtual array.

This device operates in the E-band; more specifically, the frequencies of its transmitted signal are expressed by eq. (2.24), with $f_0 = 78$ GHz, $\Delta f = 16.67$ MHz and $N = 121$ (for this reason, the bandwidth of the radiated signal is 2 GHz and its central frequency is $f_c = 79$ GHz). Moreover, it is equipped with $N_T = 16$ TX antennas and $N_R = 21$ RX antennas (see Fig. 3.7-b)). In principle, $16 \cdot 21 = 336$ virtual antennas are available; however, in practice, 16 of them (the ones that are not horizontally aligned) are discarded in order to set up a 20×16 URA. The accuracy of all the algorithms illustrated in the previous section has been assessed by acquiring multiple measurements in a small area, because of the limited power radiated by the radar device employed in this work; in all the experiments proposed in the following, each detectable target is represented by a metal disk, having a diameter of 5.5 cm. The numerical results shown in this section refer to two specific static scenarios.

The first one is an *outdoor* scenario, characterized by a fixed number of detectable targets (namely, $L = 3$ metal disks), which are placed at approximately the same distance from the radar device (see Fig. 3.8); their 3D coordinates are listed in Table 3.6. The second one, instead, is an *indoor* scenario with three distinct configurations, each involving a different number of targets; more specifically, in this case, the chosen configurations involve: 1) a single target placed over a plastic desk; 2) two targets placed over two different plastic desks; 3) three targets placed over three different plastic supports. It is important to remind that in the first scenario the height of all the three targets is equal, while it is varied in the latter in order to test the accuracy of the algorithm also along the elevation direction. In both scenarios, prior knowledge of L is always assumed and the following values are selected for the parameters defining the search domain of the

FIGURE 3.8: Outdoor scenario in which the measurements have been acquired. The three targets, each represented by a metal disk, are clearly visible.

considered algorithms: a) $\tilde{\theta}_{N_{\tilde{\theta}}-1} = \tilde{\phi}_{N_{\tilde{\phi}}-1} = 60°$, $\tilde{\theta}_0 = \tilde{\phi}_0 = -60°$ and $N_\theta = N_\phi = 121$ (so that the step sizes $\Delta\tilde{\theta} = \Delta\tilde{\phi} = 1°$ are obtained); b) $\tau_0 = 2R_m/c$, $\tau_{N_\tau-1} = 2R_M/c$ and $N_\tau = 121$, where $R_m = 0.3$ m ($R_M = 6.0$ m) is the minimum (maximum) measurable range (consequently, a range resolution $\Delta R \approx 5$ cm is achieved). Moreover, the acquired measurements have been always pre-processed by a cancellation technique already available on the employed radar device; this procedure exploits the measurements acquired from the first transmitted frame to cancel out unwanted received echoes. A 2D contour plot of the cost function $Q_t(\tilde{\theta}, \tilde{\phi}, \hat{\tau}_1)$ (3.54) referring to the first scenario is shown in Fig. 3.9. This 2D plot has been generated by evaluating this cost function at the delay $\tilde{\tau} = \hat{\tau}_1$ associated with the first (i.e., with the strongest) echo; this echo originates from target #2. A coarse estimate of the position of each target can be obtained by identifying the peaks of this function. Note that, in the considered scenario, only two peaks (corresponding to the targets #2 and #3) are clearly visible. This can be due to the fact that: a) the tilt angle of target #1 reduces its RCS; b) the range of target #1 is not exactly the same as that of the other two targets.

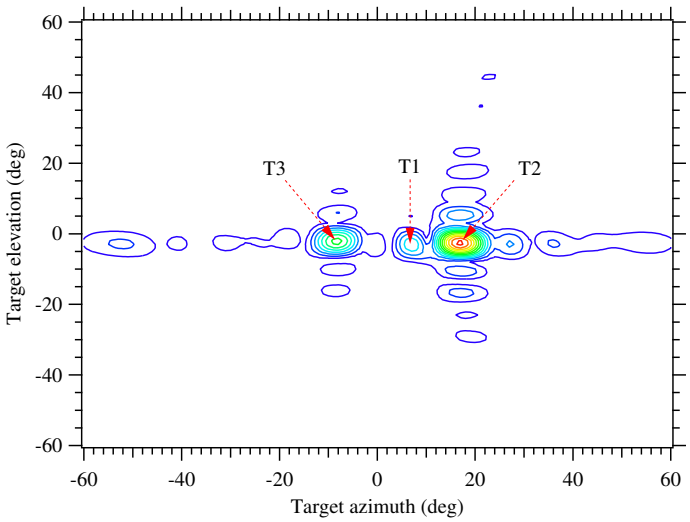
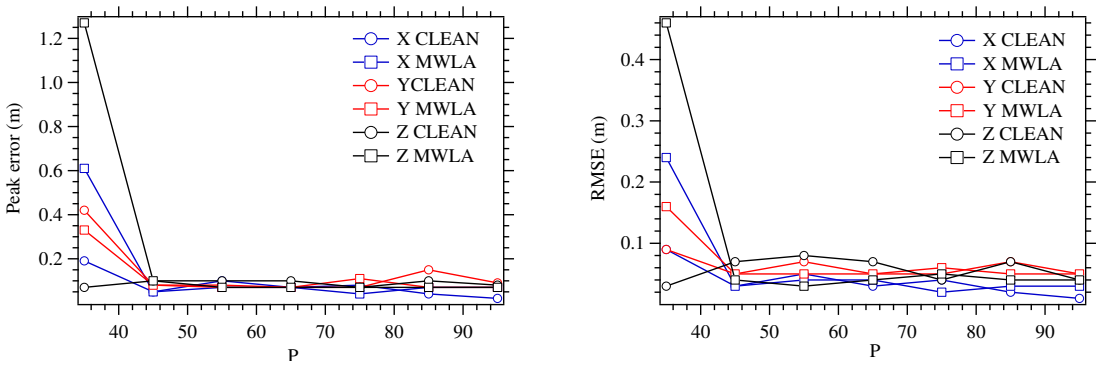


FIGURE 3.9: 2D contour plot (in spherical coordinates) of the function $Q_t(\tilde{\theta}, \tilde{\phi}, \hat{\tau}_1)$ (3.54) evaluated at a fixed delay (i.e., at $\tilde{\tau} = \hat{\tau}_1$) by the standard beamformer in the first scenario.

The range/azimuth/elevation estimates generated by the developed iterative algorithms in this scenario for each of the three targets, together with their *computation time*[1] (CT) and *computational complexity* (in GFlops) are listed in Table 3.6; in this case, the order $P = 45$ is adopted in the interpolation procedure based on Chebyshev polynomials. This choice for the interpolation order represents the best trade off between accuracy and computational complexity, as confirmed by Fig. 3.10. In this figure in fact, the values of the peak error $\hat{\varepsilon}$ and the RMSE $\bar{\varepsilon}$ obtained by CLEAN and MWLA for all the three coordinates X, Y and Z are shown. These errors are related to the first scenario only, described in Table 3.6. The value of the errors in correspondence of $P = 45$ are very low, at the price of a reasonable computational complexity.



(A) Peak error $\hat{\varepsilon}$ obtained by CLEAN and MWLA for all the three coordinates X, Y and Z.

(B) RMSE $\bar{\varepsilon}$ obtained by CLEAN and MWLA for all the three coordinates X, Y and Z.

FIGURE 3.10: Graphical representation of the a) peak error $\hat{\varepsilon}$ and b) the RMSE $\bar{\varepsilon}$ obtained by CLEAN and MWLA for all the three coordinates X, Y and Z.

In the second scenario, the estimation accuracy of each algorithm has been assessed by evaluating the RMSE $\bar{\varepsilon}$ and the *peak error* $\hat{\varepsilon}$ for the estimates of the spatial coordinates of the targets (see eq. (3.73)-(3.74)); $P = 45$ has been selected again. These errors have been computed on the basis of the measurements acquired in the three different configurations mentioned above; the 3D coordinates of the targets and the evaluated errors are listed in Table 3.7 for both the CLEAN algorithm (CL) and the MWLA (ML). Note that, in this case, the values of the RMSE and those of the peak error are quite similar and comparable with the resolution of the employed radar device. These results and those shown for the first scenario lead to the conclusion that the CLEAN algorithm and the MWLA are able to accurately estimate the position of all the targets. Further results, not shown here for space limitations, have evidenced that: a) a similar accuracy is also achieved in the presence of $L = 4$ targets with $P = 45$; b) if $L \geq 5$, achieving a good approximation of the functions $J_t(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$ and $\mathbf{L}(\tilde{\theta}, \tilde{\phi})$ (see eqs. (3.12) and (3.66), respectively) requires increasing the order of the Chebyshev polynomials; c) the errors of the cancellation procedure employed by both the CLEAN algorithm and MWLA accumulate over successive iterations, so that, for a large value of L, a number of iterations greater than L might be required to identify all the targets of interest. The sensitivity of the RMSE and that of the peak error to the order P has been also assessed in the second scenario; for this reason, these errors have been also computed for values of this parameter ranging from 30 to 120. The numerical results have evidenced that: 1) for all the considered values of P, both the RMSE and the peak error are on the order of 1 cm (i.e., they are comparable with the resolution of the employed radar device); b)

[1] All the computation times regarding these experiments refer to those required by a desktop computer equipped with an i7 processor.

when P increases beyond 60, these errors become approximately constant and do not significantly differ from those observed in the absence of interpolation. The impact of P on the CT and overall computational complexity required for processing the whole set of acquired measurements have been also assessed under the assumption of a single detectable target (i.e., for $L = 1$); some results are shown in Fig. 3.11.

Algorithm	Coord.	T1	T2	T3	CT (sec)	GFlops
Exact	$X(m)$	1.9	2.1	2.0		
	$Y(m)$	-0.75	-0.3	0.7		
	$Z(m)$	-0.1	-0.1	-0.1		
CLEAN	$X(m)$	1.89	2.08	2.05	8.7	457
	$Y(m)$	-0.69	-0.29	0.62		
	$Z(m)$	0	0	-0.1		
MWLA	$X(m)$	1.87	2.1	2.05	0.9	15
	$Y(m)$	-0.68	-0.29	0.62		
	$Z(m)$	0	-0.07	-0.1		

TABLE 3.6: Spatial coordinates of the targets characterizing the first scenario and their estimates computed by the CLEAN algorithm and the MWLA. The *computation time* (CT) and computation complexity of these algorithms are also provided.

Exp.	Coord.	Target			$\bar{\varepsilon}$		$\hat{\varepsilon}$	
		1	2	3	CL	ML	CL	ML
1	$X(m)$	1.35			0.03	0.02	0.03	0.02
	$Y(m)$	0.32			0.03	0.03	0.03	0.03
	$Z(m)$	-0.15			0.04	0.04	0.04	0.04
2	$X(m)$	1.38	0.98		0.02	0.02	0.02	0.02
	$Y(m)$	0.32	-0.75		0.05	0.01	0.07	0.02
	$Z(m)$	-0.15	-0.15		0.05	0.04	0.05	0.05
3	$X(m)$	1.38	0.98	2.10	0.02	0.01	0.02	0.02
	$Y(m)$	0.32	-0.75	2.10	0.05	0.04	0.07	0.05
	$Z(m)$	-0.15	-0.18	-0.35	0.04	0.04	0.05	0.06

TABLE 3.7: 3D coordinates of the targets employed in the second scenario. The computed RMSE and the peak errors for both the CLEAN algorithm and the MWLA are also provided.

These results lead to the following conclusions: 1) the computational complexity and the processing time steeply increase with P; 2) the computational time/complexity of the MWLA is substantially lower than that of the CLEAN algorithm.

The last result is due to the fact that: a) the MWLA algorithm requires the computation of the steering vector at a single frequency only (namely, at the reference frequency f_r); b) its estimation of targets parameters does not involve the computation of the cost function $J_t(\tilde{\theta}, \tilde{\phi}, \tilde{\tau})$ (3.12). All the results shown in Fig. 3.11 refer to $L = 1$; increasing L (i.e., the required number of iterations) results in a linear increase of the complexity of both the MWLA and the CLEAN algorithm. As for the 2D imaging algorithms, the last

issue analysed in detail is represented by the behaviour of the *residual energy*

$$E_{\tilde{H}}^{(l)} = \sum_{n=0}^{N-1} \left\| \tilde{\mathbf{H}}_t^{(l)}[n] \right\|^2, \tag{3.75}$$

available at the end of the l-th iteration in CLEAN and in MWLA, as l increases; here, $\tilde{\mathbf{H}}_t^{(l)}[n]$ represents the residual frequency response referring to the n-th frequency and available at the beginning of the l-th iteration (see eq. (3.61)). Moreover, this energy has been normalized with respect to its initial value $E_{\tilde{H}}^{(1)}$: the resulting parameter is called *normalised residual energy*. The NRE computed for the third configuration of the second scenario is shown in Fig. 3.12 (numerical results are identified by marks, whereas continuous lines are drawn for ease of reading).

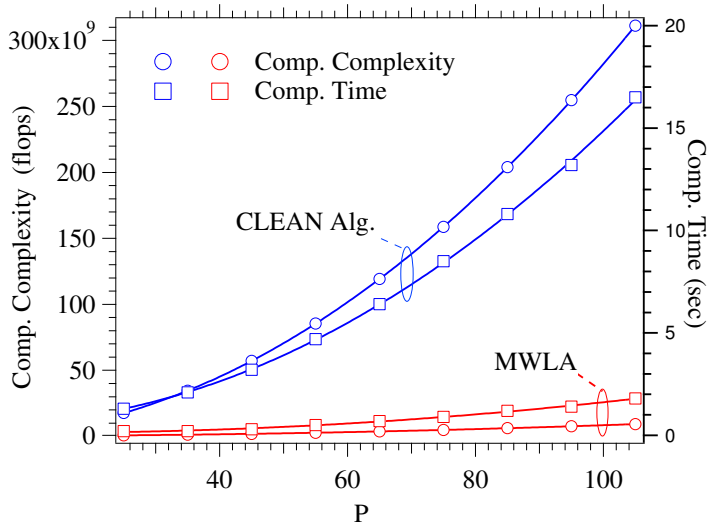


FIGURE 3.11: Computational complexity and computational time versus P; the CLEAN and MWLA algorithms are considered.

Note that, in this case, two sets of results are provided for each algorithm; one is computed on the basis of the raw data acquired by the radar (i.e., of the *experimental* data), the other one on the basis of computer generated data (i.e., of *simulated* data). In the last case, three *point targets*, having the same spatial coordinates as those given in Table 3.7, have been assumed. These results show that the iterative cancellation procedure accomplished by CLEAN and MWLA results in a steep decrease of the NRE in the simulated case. This leads to the same conclusion of the bi-dimensional case, i.e. that, in principle, a stopping criterion based on this quantity could be adopted. However, a slightly smaller rate of decrease is observed in the case of experimental data; this is mainly due to the fact that: a) due to the high frequency of the impinging waves, the metal disks employed in this experiments cannot be represented as point targets, so that multiple echoes are received from each of them; b) the simulated radar signal is characterized by a *signal-to-noise ratio* moderately higher than that one of the real signal received by the radar. This suggests that a stopping criterion based on the evaluation of the NRE can be really adopted in a real radar system, provided that its threshold is adjusted carefully. Note also that the NRE of CLEAN and that of MWLA exhibit a similar rate of decrease in both cases.

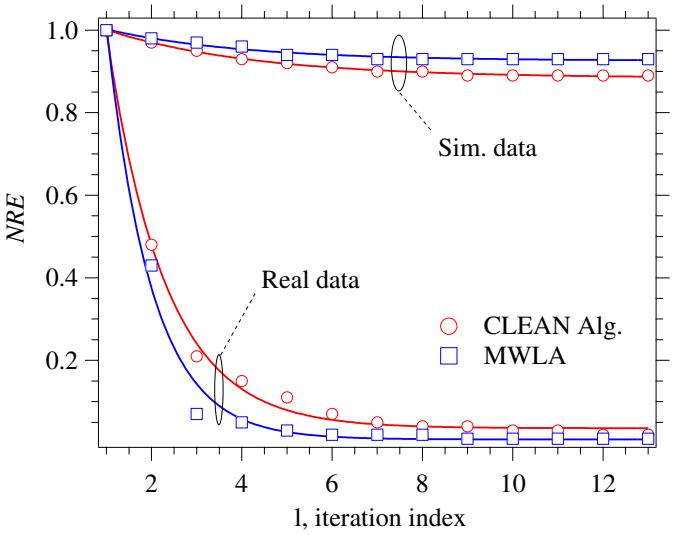


FIGURE 3.12: Normalized residual energy versus the iteration index l; the CLEAN algorithm and the MWLA are considered.

3.6 Conclusions

Five deterministic techniques for jointly estimating ranges and DOA in a MIMO SFCW radar system have been described for 2D radar imaging. All these techniques are deterministic and estimate target parameters through iterative cancellation procedures. The obtained numerical results, based on real measurements, evidence that, in a 2D scenario, the range and azimuth estimates computed by all algorithms are reasonably accurate. However, on the one hand, MWLA achieves a slightly better accuracy and requires a smaller computational effort than CLEAN; on the other hand, limited or no improvement is found if the estimates generated by these techniques are processed by an EM-based algorithm to refine them. Moreover, 3D extensions of CLEAN and MWLA based on an interpolation method to reduce their computational complexity have been described. The obtained numerical results, based on a set of experimental measurements acquired by a commercial radar device, show that these algorithms perform similarly and are able to generate accurate estimates of the 3D position of the detected targets. Similarly as 2D imaging, the MWLA requires a smaller computational effort than CLEAN.

4 A maximum likelihood approach to target detection and estimation

In radar signal processing for autonomous driving applications, *optimal* (i.e., *maximum likelihood*, MLI) techniques for the estimation of the overall number of targets and of their spatial coordinates cannot be employed in practice, since they require solving complicated multidimensional optimization problems and, consequently, entail a huge computational effort, even in the presence of a small number of targets [54]. This has motivated the development of various *sub-optimal* estimation techniques able to achieve good estimation accuracy at a manageable computational cost. A well known sub-optimal technique employed in real world radar systems is the one described in section 2.5 for FMCW radar systems; this requires: a) the computation of a multidimensional *Fast Fourier Transform* (FFT) of the matrix collecting the time-domain samples of the signals acquired through the receive array of the employed radar device; b) the search for the peaks of the resulting amplitude spectrum over a range-azimuth (range-azimuth-elevation) domain for 2D (3D) imaging. As already devised in Par. 2.5, despite the practical importance of this technique, it suffers from the following relevant drawback: it can fail to locate targets whose electromagnetic echoes are weaker than those generated by other spatially close targets; this is due to the fact the spectral contribution due to weak echoes is usually hidden by the leakage originating from stronger echoes. This drawback may substantially affect the overall quality of radar imaging in the presence of extended targets, since such targets can be usually modelled as a cluster of point targets characterized by different radar cross sections.

Alternative sub-optimal techniques available in the literature are based on the idea of turning a complicated multidimensional optimization problem into a series of simpler and interconnected optimization sub-problems, each of which involves a search for the local maxima of a specific cost function over a limited 1D or 2D parameter space. Examples of this approach are offered in Sect. 3.3 (3.4), where range-DOAs estimators for SFCW MIMO radars, respectively, are proposed. In particular, various *iterative deterministic methods* applicable to a 2D and a 3D propagation scenario are derived. These methods have the following relevant features: 1) they process a single snapshot of the received signal (acquired over the whole antenna array); 2) they estimate a new target in each iteration; 3) they do not require prior knowledge of the overall number of targets; 4) they involve 1D or 2D maximizations only; 5) they achieve a good accuracy at a reasonable computational cost; 6) the computational effort they require can be easily controlled by setting a threshold on the maximum number of targets to be detected.

In this chapter, other two novel detection and estimation algorithms named *Range and angle serial cancellation algorithm* for 2D and 3D radar imaging are described. These techniques, that have been developed and tested by me, similarly as CLEAN and MWLA

(see Sect. 3.3 and 3.4) they are deterministic, process a single snapshot, operate in an iterative fashion and are computationally efficient; this last feature can be related to the fact they require the evaluation of 1D Inverse Fourier Transforms only and the search for the global maximum of proper cost functions over 1D (delay, azimuth or elevation) domains. Moreover, they first extract the range of each detected target from the spectra of the received signals and, then, fuse the information originating from such spectra to extract DOA information.

In addition, they exploit a novel iterative estimation technique, dubbed *complex single delay estimation and cancellation*, based on a serial cancellation approach for evaluating the parameters of multiple overlapped complex exponentials in the presence of additive noise.
This method, in turn, exploits the *complex single delay estimator* that is based on:

a) expressing the dependence of the MLI metric on the delay residual in an approximate polynomial form through standard approximations of trigonometric functions;

b) exploiting the *alternating minimization* technique for the maximization of this metric (e.g., see [71, Par. IV-A]).

Moreover, its most relevant feature is represented by the fact that it requires the evaluation of spectral coefficients that are not exploited by all the other related estimation methods available in the literature. The accuracy of the developed 2D and 3D radar imaging techniques is assessed on the basis of the measurements acquired through an SFCW commercial MIMO system in different scenarios. The obtained results lead to the conclusion that the proposed methods achieve a good probability of convergence and accuracy in the presence of multiple targets in both bidimensional and three-dimensional spaces.

The remaining part of this chapter is organized as follows. The target detection and delay estimation methods, on which the algorithms are based, are illustrated in section 4.1, whereas the algorithms for bi-dimensional and three-dimensional imaging are presented in section 4.2. The performance of the algorithms is analysed in section 4.5, where various numerical results, based on experimental measurements, are illustrated. Finally, some conclusions and final comments are offered in section 4.6.

4.1 Complex single delay estimators

In this section, a novel method for estimating the parameters of a complex exponential is described. Then, it can be exploited how this method can be exploited to detect multiple echoes and estimate their parameters through a deterministic procedure based on successive cancellation. Finally, the computational complexity of the developed estimation methods is analysed in detail.

4.1.1 Complex single delay estimator

Let us focus first on the problem of estimating the parameters (namely, the delay and complex amplitude) of a single exponential contained in the *complex* sequence $\{x_{c,n}; n = 0, 1, ..., N-1\}$, whose n-th sample is expressed by eq. (2.25) with $L = 1$[1], i.e. as

$$x_{c,n} = A \exp(-j2\pi nF) + w_{c,n} \tag{4.1}$$

[1] the target index l and the channel index v are omitted for simplicity.

where A (2.26) and F are the complex amplitude and the normalised delay, respectively, of the exponential itself. It is well known that the MLI estimates F_{MLI} and A_{MLI} of the parameters F and A, respectively, represent the solution of the least square problem (e.g., see [79, eq. (22)])

$$(F_{\mathrm{MLI}}, A_{\mathrm{MLI}}) \triangleq \arg\min_{\tilde{F},\tilde{A}} \varepsilon(\tilde{F}, \tilde{A}), \tag{4.2}$$

where $\tilde{F}$ and $\tilde{A}$ represent trial values of F and A, respectively,

$$\varepsilon(\tilde{F}, \tilde{A}) \triangleq \frac{1}{N} \sum_{n=0}^{N-1} \varepsilon_n(\tilde{F}, \tilde{A}) \tag{4.3}$$

is the *mean square error* (MSE) evaluated over the whole observation interval,

$$\varepsilon_n(\tilde{F}, \tilde{A}) \triangleq \left[x_{c,n} - s_n(\tilde{F}, \tilde{A})\right]^2 \tag{4.4}$$

is the *square error* between the noisy sample $x_{c,n}$ (4.1) and its useful component

$$s_n(\tilde{F}, \tilde{A}) \triangleq \tilde{A} \exp\left(-j2\pi n\tilde{F}\right), \tag{4.5}$$

evaluated under the assumption that $F = \tilde{F}$ and $A = \tilde{A}$.

The cost function $\varepsilon(\tilde{F}, \tilde{A})$ (4.3) can be easily reformulated in a different way as follows. To begin, substituting the RHS of eq. (4.5) in that of eq. (4.4) produces, after some manipulation,

$$\varepsilon_n(\tilde{F}, \tilde{A}) = |x_{c,n}|^2 + 2\left[\tilde{A}_R^2 + \tilde{A}_I^2\right] - 2\Re\{x_{c,n}\tilde{A}^*\}\cos(\tilde{\phi}_n) + 2\Im\{x_{c,n}\tilde{A}^*\}\sin(\tilde{\phi}_n), \tag{4.6}$$

where

$$\tilde{\phi}_n \triangleq 2\pi n\tilde{F}, \tag{4.7}$$

and $\Re\{x\}$ ($\Im\{x\}$) denotes the real (imaginary) part of the complex variable x. Then, substituting the RHS of eq. (4.6) into eq. (4.3) yields

$$\varepsilon(\tilde{F}, \tilde{A}) = \varepsilon_x + \tilde{A}_R^2 + \tilde{A}_I^2 - 2\tilde{A}_R\bar{X}_R(\tilde{F}) - 2\tilde{A}_I\,\bar{X}_I(\tilde{F}), \tag{4.8}$$

where

$$\varepsilon_x \triangleq \frac{1}{N} \sum_{n=0}^{N-1} |x_{c,n}|^2, \tag{4.9}$$

$\bar{X}_R(\tilde{F}) \triangleq \Re\{\bar{X}(\tilde{F})\}$, $\bar{X}_I(\tilde{F}) \triangleq \Im\{\bar{X}(\tilde{F})\}$, $\tilde{A} = \tilde{A}_R + j\tilde{A}_I$ and

$$\bar{X}(\tilde{F}) \triangleq \frac{1}{N} \sum_{n=0}^{N-1} x_{c,n} \exp\left(j\tilde{\phi}_n\right) = \frac{1}{N} \sum_{n=0}^{N-1} x_{c,n} \exp\left(j2\pi n\tilde{F}\right), \tag{4.10}$$

is, up to a scale factor, the *Inverse Fourier transform* (IFFT) of the sequence $\{x_{c,n}\}$.

Based on eq. (4.8), it is not difficult to show that the optimization problem expressed by eq. (4.2) does not admit a closed form solution because of the non-linear dependence of the function $\varepsilon(\tilde{F}, \tilde{A})$ on its variable $\tilde{F}$. However, an approximate solution to this problem can be derived by

a) Exploiting an iterative method, known as *alternating minimization* (AM) [71]. This allows us to transform the *two-dimensional* optimization problem expressed by eq. (4.2) into two interconnected *one-dimensional* problems, one involving the variable $\tilde{F}$ only (conditioned on the knowledge of $\tilde{A}$), the other one involving the variable $\tilde{A}$ only (conditioned on the knowledge of $\tilde{F}$).

b) Expressing the dependence of the function $\varepsilon(\tilde{F}, \tilde{A})$ on the variable $\tilde{F}$ through the pair $(F_c\,, \tilde{\delta})$ such that

$$\tilde{F} = F_c + \tilde{\delta}\, F_{\text{IDFT}}, \tag{4.11}$$

where F_c is a given *coarse estimate* of F, $\tilde{\delta}$ is a real variable called *residual*,

$$F_{\text{IDFT}} = \frac{1}{N_0} \tag{4.12}$$

is the normalized *fundamental delay* associated with the N_0-th order *inverse discrete Fourier transform* (IDFT)

$$\mathbf{X}_0 = \left[X_{0,0}, X_{0,1}, ..., X_{0,N_0-1}\right]^T \tag{4.13}$$

of the zero padded version

$$\mathbf{x}_{0,\text{ZP}} = \left[\mathbf{x}_0^T\, \mathbf{0}_{(M-1)N}^T\right]^T, \tag{4.14}$$

of the vector

$$\mathbf{x}_0 \triangleq \left[x_{c,0}, x_{c,1}, ..., x_{c,N-1}\right]^T \tag{4.15}$$

collecting all the elements of the sequence $\{x_{c,n}\}$, M is a positive integer (dubbed *over-sampling factor*), $\mathbf{0}_D$ is a $D-$dimensional (column) null vector and $N_0 \triangleq M \cdot N$. It is important to note that eq. (4.11)-(4.15) does not concern with the alternating maximization method described in [71], but they are derived on the basis of the physics of the problem.

c) Expressing the dependence of the function $\varepsilon(\tilde{F}, \tilde{A})$ (4.8) on the variable $\tilde{\delta}$ through its powers $\{\tilde{\delta}^l;\ 0 \leq l \leq 3\}$; this result is achieved by approximating various trigonometric functions appearing in the expression of $\varepsilon(\tilde{F}, \tilde{A})$ with their Taylor expansions truncated to a proper order (three) in order to have a sufficiently accurate result.

Let us show how these principles can be put into practice. First of all, the exploitation of the above mentioned AM approach requires solving the following two sub-problems: **P1**) minimizing the cost function $\varepsilon(\tilde{F}, \tilde{A})$ (4.8) with respect to $\tilde{A}$, given $\tilde{F} = \hat{F}$; **P2**) minimizing the same function with respect to $\tilde{F}$, given $\tilde{A} = \hat{A}$. Sub-problem **P1** can be easily solved thanks to the polynomial dependence of the cost function $\varepsilon(\hat{F}, \tilde{A})$ on the variables $\tilde{A}_R$ and $\tilde{A}_I$. In fact, the minimum of the function $\varepsilon(\tilde{F}, \tilde{A})$ with respect to the last variables results in

$$\tilde{A} = \hat{A} = \bar{X}(\hat{F}) \tag{4.16}$$

Therefore, given $\tilde{F} = \hat{F}$, the optimal value $\hat{A}$ of the variable $\tilde{A}$ can be computed exactly through the last equation; this requires the evaluation of $\bar{X}(\hat{F})$ that can be computed exactly through (4.10) or, in an approximate fashion, through a computationally efficient procedure based on the fact that the vector

$$\bar{\mathbf{X}}_s \triangleq M\, \mathbf{X}_0 \tag{4.17}$$

collects N_0 uniformly spaced samples of the function $\bar{X}(F)$ (4.10). In fact, the k-th element

$$X_{0,k} \triangleq \frac{1}{N_0} \sum_{n=0}^{N-1} x_{c,n} \exp\left(j \frac{2\pi n k}{N_0}\right), \tag{4.18}$$

of vector $\mathbf{X}_0$ (4.13) can be expressed as

$$X_{0,k} = \frac{N}{N_0} \bar{X}\left(F_k\right) = \frac{1}{M} \bar{X}\left(F_k\right), \tag{4.19}$$

where

$$F_k \triangleq k\, F_{\text{IDFT}} \tag{4.20}$$

with $k = 0, 1, ..., N_0 - 1$. For this reason, an approximate evaluation of the quantity $\bar{X}(\hat{F})$ at a normalised delay $\hat{F}$ different from any multiple of F_{IDFT} (4.12) can be accomplished by interpolating the elements of the vector $\bar{\mathbf{X}}_s$ (4.17); the last vector, in turn, can be easily computed after evaluating the N_0-th IFFT of $\mathbf{x}_{0,\text{ZP}}$ (4.14), i.e. the vector $\mathbf{X}_0$ (4.13).

Let us now take into consideration now sub-problem **P2**. Unluckily, this sub-problem does not admit a closed form solution. For this reason, an approximate solution is developed in the following. Such a solution is based on representing the normalized delay F in the same form as $\tilde{F}$ (see eq. (4.11)), i.e. as

$$F = F_c + \delta\, F_{\text{IDFT}} \tag{4.21}$$

and on a novel method for estimating the real residual δ, i.e., for accomplishing the *fine estimate* of F. This method is derived as follows. Representing the trial normalized delay $\tilde{F}$ according to eq. (4.11) allows us to express the variable $\tilde{\phi}_n$ (4.7) as

$$\tilde{\phi}_n = \hat{\theta}_n + n\tilde{\Delta}, \tag{4.22}$$

where

$$\tilde{\Delta} \triangleq 2\pi\tilde{\delta}\, F_{\text{IDFT}} \tag{4.23}$$

is a new variable and

$$\hat{\theta}_n \triangleq 2\pi n\, F_c. \tag{4.24}$$

Then, substituting the RHS of eq. (4.22) into eq. (4.6) (with $\tilde{A} = \hat{A}$) yields

$$\begin{aligned}\varepsilon_n(\tilde{F}, \hat{A}) = |x_{c,n}|^2 + \hat{A}_R^2 + \hat{A}_I^2 - 2\Re\{x_{c,n}\hat{A}^*\}\cos\left(n\tilde{\Delta} + \hat{\theta}_n\right) \\ + 2\Im\{x_{c,n}\hat{A}^*\}\sin\left(n\tilde{\Delta} + \hat{\theta}_n\right)\end{aligned} \tag{4.25}$$

Based on the last equation, the following expression can be derived

$$\begin{aligned}\varepsilon_n(\tilde{F}, \hat{A}) = |x_{c,n}|^2 + \hat{A}_R^2 + \hat{A}_I^2 - 2(x_{c,n}^{(R)}\hat{A}_R + x_{c,n}^{(I)}\hat{A}_I)\cdot\left[\cos(\hat{\theta}_n)\cos(n\tilde{\Delta}) - \sin(\hat{\theta}_n)\sin(n\tilde{\Delta})\right] \\ + 2(x_{c,n}^{(I)}\hat{A}_R - x_{c,n}^{(R)}\hat{A}_I)\cdot\left[\sin(\hat{\theta}_n)\cos(n\tilde{\Delta}) + \cos(\hat{\theta}_n)\sin(n\tilde{\Delta})\right].\end{aligned} \tag{4.26}$$

where $\hat{A}_R \triangleq \Re\{\hat{A}\}$, $\hat{A}_I \triangleq \Im\{\hat{A}\}$, $x_{c,n}^{(R)} \triangleq \Re\{x_{c,n}\}$, $x_{c,n}^{(I)} \triangleq \Im\{x_{c,n}\}$ and $\tilde{\phi}_n$ is defined by eq. (4.22) (see also eqs. (4.23)-(4.24)). If the normalised delay F_{IDFT} is sufficiently small, i.e. if the IFFT order is large enough, and assuming that:

$$\cos\left(n\tilde{\Delta}\right) \approx 1 - 2n^2\tilde{\Delta}^2 \tag{4.27}$$

and

$$\sin\left(n\tilde{\Delta}\right) \approx n\tilde{\Delta} - n^3\frac{\tilde{\Delta}^3}{6} \tag{4.28}$$

are valid approximations, the following approximate expression can be obtained after few algebraic manipulations

$$\begin{aligned}\varepsilon_{\text{CSDE}}\left(\tilde{\Delta},\hat{A}\right) \triangleq\ & \varepsilon_x + \hat{A}_R^2 + \hat{A}_I^2 \\ & - \frac{1}{3}\tilde{\Delta}^3\left(\hat{A}_R\,\bar{X}_{3,\rho}^{(I)} - \hat{A}_I\,\bar{X}_{3,\rho}^{(R)}\right) \\ & + \tilde{\Delta}^2\left(\hat{A}_R\,\bar{X}_{2,\rho}^{(R)} + \hat{A}_I\,\bar{X}_{2,\rho}^{(I)}\right) \\ & + 2\tilde{\Delta}\left(\hat{A}_I\,\bar{X}_{1,\rho}^{(R)} - \hat{A}_R\,\bar{X}_{1,\rho}^{(I)}\right) \\ & - 2\left(\hat{A}_R\,\bar{X}_{0,\rho}^{(R)} + \hat{A}_I\,\bar{X}_{0,\rho}^{(I)}\right)\end{aligned} \tag{4.29}$$

can be derived for the function $\varepsilon(\tilde{F},\hat{A})$ (4.8); here,

$$\rho \triangleq F_c/F_{\text{IDFT}}, \tag{4.30}$$

$\bar{X}_{k,\rho}^{(R)} \triangleq \Re\{\bar{X}_{k,\rho}\}$, $\bar{X}_{k,\rho}^{(I)} \triangleq \Im\{\bar{X}_{k,\rho}\}$, for any k, p and $x \geq 0$,

$$\bar{X}_{k,\rho} \triangleq \frac{1}{N}\sum_{n=0}^{N-1} x_{k,n}\exp\left(j\frac{2\pi n\rho}{N_0}\right), \tag{4.31}$$

for any ρ and $k = 1, 2$, and

$$x_{k,n} \triangleq n^k \cdot x_{c,n} \tag{4.32}$$

with $n = 0, 1, ..., N-1$. It is important to point out that: a) if ρ is an integer, the quantity $\bar{X}_{k,\rho}$ (4.31) (with $k = 1$ and 2) represents the ρ-th element of the vector

$$\mathbf{X}_k = M\left[X_{k,0}, X_{k,1}, ..., X_{k,N_0-1}\right]^T \tag{4.33}$$

generated by the N_0-th order IDFT of the zero padded version

$$\mathbf{x}_{k,\text{ZP}} = \left[\mathbf{x}_k^T\ \mathbf{0}_{(M-1)N}^T\right]^T, \tag{4.34}$$

of the vector

$$\mathbf{x}_k \triangleq \left[x_{k,0}, x_{k,1}, ..., x_{k,N-1}\right]^T; \tag{4.35}$$

b) if ρ is not an integer, the quantity $\bar{X}_{k,\rho}$ can be evaluated exactly using eq. (4.31) or, in an approximate fashion, by interpolating a subset of adjacent elements of the vector $\mathbf{X}_k$ (4.33); c) the evaluation of the vectors $\{\mathbf{X}_k; k = 1, 2\}$ requires two *additional* IFFTs.

Since the function $\varepsilon_{\text{SDE}}(\tilde{\Delta},\hat{A})$ (4.29) is a polynomial of degree 3 in the variable $\tilde{\Delta}$, an estimate $\hat{\Delta}$ of Δ and, consequently, an estimate (see eq. (4.23))

$$\hat{\delta} = \frac{\hat{\Delta}}{2\pi F_{\text{IDFT}}} \tag{4.36}$$

of δ, can be obtained by computing the derivative of this function with respect to $\tilde{\Delta}$, setting it to zero and solving the resulting quadratic equation

$$a(\rho)\ \tilde{\Delta}^2 + b(\rho)\ \tilde{\Delta} + c(\rho) = 0, \tag{4.37}$$

in the variable $\tilde{\Delta}$; here,

$$a(\rho) \triangleq \frac{1}{2}\Im\left\{\hat{A}^*\bar{X}_{3,\rho}\right\}, \tag{4.38}$$

$$b(\rho) \triangleq -\Re\{\hat{A}^* \bar{X}_{2,\rho}\} \tag{4.39}$$

and

$$c(\rho) \triangleq -\Im\{\hat{A}^* \bar{X}_{1,\rho}\}. \tag{4.40}$$

Note that only one of the two solutions of eq. (4.37), namely

$$\hat{\Delta} = -\frac{b(\rho) + \sqrt{(b(\rho))^2 - 4a(\rho)\ c(\rho)}}{2a(\rho)} \tag{4.41}$$

has to be employed. A simpler estimate of Δ is obtained neglecting the contribution of the first term in the left-hand side of eq. (4.37), i.e. setting $a(\rho) = 0$. This leads to a first-degree equation, whose solution is

$$\hat{\Delta} = -c(\rho)/b(\rho). \tag{4.42}$$

Given an estimate $\hat{\Delta}$ of Δ (and, consequently, and estimate $\hat{\delta}$ of δ; see eq. (4.36)), the *fine-grained estimate*

$$\hat{F} = F_c + \hat{\delta}\, F_{\text{IDFT}} = F_c + \frac{\hat{\Delta}}{2\pi} \tag{4.43}$$

of F can be evaluated on the basis of eq. (4.21).

The mathematical results derived above allow us to devise a novel estimation algorithm, called *complex single delay estimator* (also abbreviated as CSDE in the following), for iteratively estimating the normalised delay F and the complex amplitude A. This algorithm is initialised by

1) Evaluating: a) the vector $\mathbf{X}_0$ (4.13); b) the initial coarse estimate $\hat{F}_c^{(0)}$ of F as

$$\hat{F}_c^{(0)} = \hat{\alpha}\, F_{\text{IDFT}}, \tag{4.44}$$

where the integer $\hat{\alpha}$ is computed by means of the well known *periodogram method* (e.g., see [79, Sec. IV] or [80, Sec. I]), i.e. as

$$\hat{\alpha} = \arg \max_{\tilde{\alpha} \in \{0,1,\ldots,N_0-1\}} |\bar{X}_{0,\tilde{\alpha}}|; \tag{4.45}$$

c) the quantity (see eq. (4.30))

$$\hat{\rho}^{(0)} \triangleq \frac{\hat{F}_c^{(0)}}{F_{\text{IDFT}}} = \hat{\alpha}; \tag{4.46}$$

d) the initial estimate $\hat{A}^{(0)}$ of A on the basis of eq. (4.16) with $\hat{F} = \hat{F}_c^{(0)}$; e) the spectral coefficients $\bar{X}_{1,\hat{\alpha}}$ and $\bar{X}_{2,\hat{\alpha}}$ on the basis of eq. (4.31); f) the coefficients $\{a(\hat{\alpha}), b(\hat{\alpha}), c(\hat{\alpha})\}$ ($\{b(\hat{\alpha}), c(\hat{\alpha})\}$ according to eqs. (4.38)–(4.40) and the first estimate $\hat{\Delta}^{(0)}$ of Δ on the basis of eq. (4.41); g) the first fine estimate $\hat{F}^{(0)}$ of F on the basis of eqs. (4.21) and (4.44), i.e. as

$$\hat{F}^{(0)} = \hat{F}_c^{(0)} + \frac{\hat{\Delta}^{(0)}}{2\pi} \tag{4.47}$$

2) Setting its iteration index i to 1.

Then, an iterative procedure is started. The i-th iteration is fed by the estimates $\hat{F}^{(i-1)}$ and $\hat{A}^{(i-1)}$ of F and A, respectively, and produces the new estimates $\hat{F}^{(i)}$ and $\hat{A}^{(i)}$ of the

same quantities (with $i = 1, 2, ..., N_{\text{CSDE}}$, where N_{CSDE} is the overall number of iterations); the procedure employed for the evaluation of $\hat{F}^{(i)}$ and $\hat{A}^{(i)}$ consists of the two steps described below (the p-th step is denoted by **CSDE-Sp**).

CSDE-S1 - The new estimate $\hat{\Delta}^{(i)}$ of Δ is computed through eq. (4.41) (eq. (4.42)); in the evaluation of the coefficients $\{a(\rho), b(\rho), c(\rho)\}$ ($\{b(\rho), c(\rho)\}$) appearing in the RHS of these equations, $\hat{A} = \hat{A}^{(i-1)}$ and

$$\rho = \hat{\rho}^{(i-1)} \triangleq \hat{F}^{(i-1)} / F_{\text{IDFT}} \tag{4.48}$$

are assumed. Then,

$$\hat{F}^{(i)} = \hat{F}^{(i-1)} + \frac{\hat{\Delta}^{(i)}}{2\pi} \tag{4.49}$$

is evaluated.

CSDE-S2 - The new estimate $\hat{A}^{(i)}$ of $\hat{A}$ is evaluated through eq. (4.16); $\hat{F} = \hat{F}^{(i)}$ is assumed in this case. Moreover, the index i is incremented by one before starting the next iteration.

At the end of the last (i.e., of the N_{CSDE}-th) iteration, the fine estimates $\hat{F} = \hat{F}^{(N_{\text{CSDE}})}$ and $\hat{A} = \hat{A}^{(N_{\text{CSDE}})}$ of F and A, respectively, become available. A proper value of the number of iteration N_{CSDE} can be obtained when the absolute distance $|\hat{F}^{(i)} - \hat{F}^{(i-1)}| << 0$ and $|\hat{A}^{(i)} - \hat{A}^{(i-1)}| << 0$ are lower than zero. The CSDE is summarized in Algorithm 6.

Algorithm 6: Complex single delay estimator

1 **Initialisation:**
a- Evaluate $\mathbf{X}_0$ (4.13) and $\hat{\alpha}$ based on eq. (4.45); then, compute the initial estimate $\hat{A}^{(0)}$ of A (4.16) and set $\hat{\rho}^{(0)} = \hat{\alpha}$ (see eq. (4.46)).
b- Evaluate the quantities $\bar{X}_{k,\hat{\alpha}}$ according to eq. (4.31) for $k = 1, 2$; then, compute $\{a(\hat{\alpha}), b(\hat{\alpha}), c(\hat{\alpha})\}$ according to eqs. (4.38)–(4.40). Finally compute $\hat{\Delta}^{(0)}$ and $\hat{F}^{(0)}$ according to eqs. (4.41) and eq. (4.43), respectively.

2 **Refinement: for** $i = 1$ *to* N_{CSDE} **do**
c- *Estimation of A*:
Set $\hat{F} = \hat{F}^{(i-1)}$; then, evaluate $\bar{X}(\hat{F})$ according to eq. (4.10) or by interpolating a few adjacent elements of $\bar{\mathbf{X}}_s$ (4.17). Finally, compute $\hat{\rho}^{(i-1)}$ and $\hat{A}^{(i)}$ according to eqs. (4.48) and (4.16), respectively.
d- *Estimation of F*:
Set $\hat{A} = \hat{A}^{(i)}$; then, compute $\bar{X}_{k,\hat{\rho}^{(i-1)}}$ according to eq. (4.31) or or by interpolating a few adjacent elements of $\bar{\mathbf{X}}_k$ (4.33); then, compute $\{a(\hat{\rho}^{(i-1)}), b(\hat{\rho}^{(i-1)}), c(\hat{\rho}^{(i-1)})\}$ (4.38)–(4.40). Finally, compute $\hat{\Delta}^{(i)}$ and $\hat{F}^{(i)}$ according to eqs. (4.41) and (4.49), respectively.
end

It is worth observing that:

a) The estimate $\hat{\delta}^{(i)}$ of the residual δ computed by the CSDE in its i-th iteration is expected to become smaller as i increases, since $\hat{F}^{(i)}$ should progressively approach F if the algorithm converges.

b) The quantities $\{\bar{X}_{k,\hat{\rho}^{(i-1)}}; k = 1, 2\}$ required in the first step of the i-th iteration can be computed exactly on the basis of eq. (4.31). However, they can be also evaluated,

in an approximate fashion, by interpolating I adjacent elements of the N_0-dimensional vectors $\mathbf{X}_k$ (4.33), where I denotes the selected interpolation order.

c) The estimate $\hat{\Delta}^{(i)}$ evaluated according to eq. (4.42) is expected to be less accurate than that computed on using eq. (4.41). However, the obtained numerical results have evidenced that both solutions achieve similar accuracy. Despite, the use of eq. (4.41) is assumed in Algorithm 6 for generality.

d) The CSDE can be employed even if the single exponential appearing in the RHS of eq. (4.1) is replaced by the superposition of L distinct exponentials. In this case, the strongest (i.e., the dominant) exponential is detected through the periodogram method (see eq. (4.45)) and the parameters of this exponential are estimated in the presence of both Gaussian noise and the interference due to the remaining exponentials. Therefore, the estimation accuracy of the CSDE is affected by both the amplitudes and the delays of the other $(L-1)$ exponentials.

e) A stopping criterion, based on the trend of the sequence $\{\hat{\Delta}^{(i)};\ i = 1, 2, ...\}$, can be easily formulated for the CSDE. For instance, the execution of its two steps can be stopped if, at the end of the i-th iteration, the condition

$$\left|\hat{\Delta}^{(i)}\right| < \varepsilon_{\Delta} \tag{4.50}$$

is satisfied; here, ε_{Δ} represents a proper threshold.

4.1.2 Complex single delay estimation and cancellation

Let us analyse now in detail how the technique derived in the previous paragraph can be exploited to estimate the multiple complex exponentials that form the useful component of the complex sequence $\{x_{c,n}\}$, when its n-th sample is expressed by eq. (2.25), with $L > 1$. The recursive method developed to achieve this based on the following basic principles:

1. Exponentials are *sequentially* detected and estimated.

2. The detection of a new exponential and the estimation of its parameters are based on the procedure developed for the CSDE in the previous paragraph; in addition, a cancellation algorithm is incorporated in both these methods to remove the contribution of previously detected exponentials from all the spectral information (namely, the spectrum $\bar{X}(F)$ (4.10), the vector $\mathbf{X}_0$ (4.13) and the coefficients $\{\bar{X}_{k,\rho}\}$ (4.31)), that are processed to detect and estimate the new exponential.

3. After detecting a new echo and estimating its parameters, a *re-estimation technique* is executed to improve the accuracy of both this exponential and the previously estimated ones; the proposed technique is inspired by the related methods described in refs. [81], [82] and [83].

4. A proper criterion is adopted to stop recursions. This allows to estimate the (unknown) number of targets, that is the value of the parameter L.

The recursive method relying on these principles is called *complex single delay estimation and cancellation* (briefly CSDEC in the following). The CSDEC algorithm is initialised by:

1) Executing the CSDE, fed by the complex sequence $\{x_{c,n}\}$, to generate, through N_{CSDE} iterations, the initial estimates $\hat{F}_0[0]$ and $\hat{A}_0[0]$ of the normalized delay and the complex amplitude, respectively, of the first detected echo.

2) Setting the recursion index r to 1.

Then, a recursive procedure is started. The r-th recursion is fed by the vectors

$$\hat{\mathbf{F}}\left[r-1\right]=\left[\hat{F}_0\left[r-1\right],\hat{F}_1\left[r-1\right],...,\hat{F}_{r-1}\left[r-1\right]\right]^T, \tag{4.51}$$

and

$$\hat{\mathbf{A}}\left[r-1\right]=\left[\hat{A}_0\left[r-1\right],\hat{A}_1\left[r-1\right],...,\hat{A}_{r-1}\left[r-1\right]\right]^T, \tag{4.52}$$

collecting the delays and the associated complex amplitudes characterizing the r exponentials detected and estimated in the previous $(r-1)$ recursions, and generates the new vectors

$$\hat{\mathbf{F}}\left[r\right]=\left[\hat{F}_0\left[r\right],\hat{F}_1\left[r\right],...,\hat{F}_r\left[r\right]\right]^T \tag{4.53}$$

and

$$\hat{\mathbf{A}}\left[r\right]=\left[\hat{A}_0\left[r\right],\hat{A}_1\left[r\right],...,\hat{A}_r\left[r\right]\right]^T \tag{4.54}$$

after: a) *estimating* the delay $\hat{F}_r\left[r\right]$ and the associated complex amplitude $\hat{A}_r\left[r\right]$ of a new (i.e., of the r-th) echo (if any); b) *refining* the estimates of the r echoes available at the beginning of the considered recursion. The procedure employed for accomplishing all this consists of the three steps described below (the p-th step is denoted by **CSDEC-Sp**).

CSDEC-S1 (CSDE *initialisation with cancellation*) - In this step, the following quantities are evaluated (see the *initialisation* part of Algorithm 6):

a) The *residual spectrum*

$$\begin{aligned}\mathbf{X}_0\left[r\right] &= \left[X_{0,0}\left[r\right],X_{0,1}\left[r\right],...,X_{0,N_0-1}\left[r\right]\right]^T \\ &\triangleq \mathbf{X}_0-\mathbf{C}_0\left(\hat{\mathbf{A}}\left[r-1\right],\hat{\mathbf{F}}\left[r-1\right],r\right),\end{aligned} \tag{4.55}$$

where $\mathbf{X}_0$ is the N_0-th order IDFT of the zero padded version $\mathbf{x}_{0,\mathrm{ZP}}$ of the vector $\mathbf{x}_0$ collecting all the elements of the sequence $\{x_{c,n}\}$ (see eqs. (4.14)–(4.15)) and the N_0-dimensional vector

$$\mathbf{C}_0\left(\hat{\mathbf{A}}\left[r-1\right],\hat{\mathbf{F}}\left[r-1\right],r\right)\triangleq\sum_{l=0}^{r-1}\bar{\mathbf{C}}_0\left(\hat{A}_l\left[r-1\right],\hat{F}_l\left[r-1\right]\right), \tag{4.56}$$

represents the contribution given by all the estimated echoes to $\mathbf{X}_0$. In particular,

$$\bar{\mathbf{C}}_0(\hat{A}_l[r-1],\hat{F}_l[r-1])=\bar{A}_l\ \bar{\mathbf{W}}_0^{(l)}, \tag{4.57}$$

where $\bar{\mathbf{W}}_0^{(l)}$ denotes, up to a scale factor, the N_0-th order IDFT of the vector

$$\bar{\mathbf{w}}_0^{(l)}\triangleq\left[1,\bar{w}_l,\bar{w}_l^2,...,\bar{w}_l^{N-1},0,...,0\right]^T. \tag{4.58}$$

and

$$\bar{w}_l\triangleq\exp(-j2\pi\hat{F}_l[r-1]). \tag{4.59}$$

Then, the m-th element of the vector $\bar{\mathbf{W}}_0^{(l)}$ is given by

$$\bar{W}_0^{(l)}[m]=\frac{1}{N}\sum_{n=0}^{N-1}\bar{w}_l^n\exp\left(j\frac{2\pi m}{N_0}n\right)=\frac{1}{N}\sum_{n=0}^{N-1}\left(q\left[m\right]\right)^n, \tag{4.60}$$

where

$$q\left[m\right]\triangleq\exp\left(-j2\pi\left(\hat{F}_l[r-1]-\frac{m}{N_0}\right)\right). \tag{4.61}$$

Therefore, the identity

$$\sum_{n=0}^{N-1} q^n = \frac{q^N - 1}{q - 1}, \tag{4.62}$$

holding for any $q \in \mathbb{C}$, can be exploited for an efficient computation of all the elements of the vector $\bar{\mathbf{W}}_0^{(l)}$.
If the overall energy

$$\mathcal{E}_0[r] \triangleq |\mathbf{X}_0[r]|^2 \tag{4.63}$$

satisfies the inequality

$$\mathcal{E}_0[r] < \mathcal{T}_{\text{CSDEC}}, \tag{4.64}$$

where $\mathcal{T}_{CSDEC}$ is a proper threshold, the algorithm stops and the estimate $\hat{L} = r$ of L is generated.

b) The integer (see eq. (4.45))

$$\hat{\alpha}[r] = \arg \max_{\tilde{\alpha} \in \{0,1,\ldots,N_0-1\}} |X_{0,\tilde{\alpha}}[r]|, \tag{4.65}$$

that represents the index of the element of $\mathbf{X}_0[r]$ (4.55) having the largest absolute value.

c) The preliminary estimate

$$\bar{A}_r[r] = \bar{X}(\hat{F}_{c,r}[r]) - \bar{X}_{\text{lk},0}\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], \hat{F}_{c,r}[r]\right) \tag{4.66}$$

of the complex amplitude of the new echo; here,

$$\hat{F}_{c,r}[r] = \hat{\alpha}[r]\, F_{\text{IDFT}} \tag{4.67}$$

and

$$\bar{X}_{\text{lk},0}\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], \hat{F}_{c,r}[r]\right) \triangleq \sum_{l=0}^{r-1} \bar{X}_0\left(\hat{A}_l[r-1], \hat{F}_l[r-1], \hat{F}_{c,r}[r]\right) \tag{4.68}$$

represent the *coarse estimate* of the delay of the new echo (see eq. (4.44)) and the contribution given to $\bar{X}(F)$ by all the estimated echoes (i.e., the *leakage*) at the delay $F = \bar{F}_{c,r}[r]$, where $\bar{X}_0(\hat{A}_l[r-1], \hat{F}_l[r-1], \hat{F}_{c,r}[r])$ is the leakage due to the l-th echo.

d) The spectral coefficient

$$\bar{X}_{k,\rho[r]}[r] = \bar{X}_{k,\rho[r]} - \hat{X}_{\text{lk},k}\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], \hat{F}_{c,r}[r]\right) \tag{4.69}$$

for $k = 1, 2$ and 3; here, we have that (see eq. (4.46))

$$\rho[r] = \hat{F}_{c,r}[r]/F_{\text{IDFT}} = \hat{\alpha}[r] \tag{4.70}$$

and

$$\hat{X}_{\text{lk},k}\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], \hat{F}_{c,r}[r]\right) \triangleq \sum_{l=0}^{r-1} \bar{X}_{\text{lk},k}\left(\hat{A}_l[r-1], \hat{F}_l[r-1], \hat{F}_{c,r}[r]\right) \tag{4.71}$$

is the contribution given to $\bar{X}_{k,\rho[r]}[r]$ by all the estimated exponentials (i.e., the *leakage*) at the normalised delay $\hat{F}_{c,r}[r]$ (in particular, $\bar{X}_{\text{lk},k}(\hat{A}_l[r-1], \hat{F}_l[r-1], \hat{F}_{c,r}[r])$ represents the leakage due to the l-th estimated echo). This last function, for a generic index k can be expressed as

$$\bar{X}_{\text{lk},k}\left(\hat{A}_l[r-1], \hat{F}_l[r-1], \hat{F}_{c,r}[r]\right) = \bar{C}_l\, \bar{W}_k^{(l)}(\hat{F}_{c,r}[r]), \tag{4.72}$$

where the quantity

$$\bar{W}_k^{(l)}(\hat{F}_{c,r}[r]) = \frac{1}{N}\sum_{n=0}^{N-1} n^k \bar{w}_l^n \exp\left(j2\pi n\hat{F}_{c,r}[r]\right) = \frac{1}{N}\sum_{n=0}^{N-1} n^k \left(p^{(m)}\right)^n \tag{4.73}$$

with

$$p^{(m)} \triangleq \exp\left(-j2\pi\left(\hat{F}_l[r-1] - \hat{F}_{c,r}[r]\right)\right). \tag{4.74}$$

For a more efficient computation of eq. (4.73) using (4.74) considering

$$q(x) \triangleq \exp\left(-j\frac{2\pi x}{N_0}\right), \tag{4.75}$$

the following identities can be employed:

$$(q-1)^2\sum_{n=0}^{N-1} n\,q^n = (N-1)q^{N+1} - Nq^N + q, \tag{4.76}$$

$$(q-1)^3\sum_{n=0}^{N-1} n^2q^n = (N-1)^2q^{N+2} + \left(-2N^2+2N+1\right)q^{N+1} + N^2q^N - q^2 - q \tag{4.77}$$

and

$$\begin{aligned}(q-1)^4\sum_{n=0}^{N-1} n^3q^n &= q + 4q^2 + q^3 - N^3q^N + \left(3N^3 - 3N^2 - 3N - 1\right)q^{N+1} \\ &\quad + \left(-3N^3 + 6N^2 - 4\right)q^{N+2} + (N-1)^3q^{N+3}\end{aligned} \tag{4.78}$$

e) The coefficients $a(\hat{\alpha}[r])$, $b(\hat{\alpha}[r])$ and $c(\hat{\alpha}[r])$, the residual $\hat{\Delta}^{(0)}[r]$ and the normalized delay

$$\hat{F}_r^{(0)} = \hat{F}_{c,r}[r] + \frac{\hat{\Delta}^{(0)}[r]}{2\pi} \tag{4.79}$$

on the basis of eqs. (4.38)–(4.40), eq. (4.41) (or eq. (4.42)), and eqs. (4.44) and (4.47). Note that $\hat{F}_r^{(0)}$ represents the *initial fine estimate* of the normalized delay of the new exponential. The evaluation of the delay $\hat{F}_r^{(0)}$ (4.79) concludes the initialization of the modified CSDE executed for the detection and the estimation of the new echo.

CSDEC-S2 (CSDE *refinement with cancellation*) - After carrying out the first step, N_{CSDE} iterations[2] are executed to refine the estimate of the parameters of the new exponential. The processing accomplished in this step follows closely that described in the *refinement* part of Algorithm 6. For this reason, in each iteration, a new estimate of the complex amplitude and of the delay residual of the r-th exponential are computed. This requires re-using eqs. (4.66)–(4.68) and (4.69), respectively, in order to remove the leakage in the spectrum $\bar{X}(F)$ and in the coefficients $\bar{X}_{k,\rho}$ (see steps **c** and **d**, respectively). At the end of the last iteration, the delay $\hat{F}_{\text{CSDE},r}[r]$ and the associated complex amplitude $\hat{A}_{\text{CSDE},r}[r]$ of the new exponential are available; these represent $\hat{F}_r[r]$ and $\hat{A}_r[r]$, respectively, if the following re-estimation step is not accomplished.

[2]The potential dependence of the parameter N_{CSDE} on the recursion index r is ignored here for simplicity.

CSDEC-S3 (*re-estimation*) - This step is fed by the $(r+1)$ normalized delays $\{\hat{F}_0[r-1], \hat{F}_1[r-1], ..., \hat{F}_{r-1}[r-1], \hat{F}_{\mathrm{CSDE},r}[r]\}$ and the associated complex amplitudes $\{\hat{A}_0[r-1], \hat{A}_1[r-1], ..., \hat{A}_{r-1}[r-1], \hat{A}_{\mathrm{CSDE},r}[r]\}$. It consists in repeating the previous step for each of the detected echoes, starting from the first one and ending with the $(r+1)$-th one. This means that, when re-estimating the l-th echo, the leakage due to the echoes whose index belong to set $\{0, 1, .., l-1, l+1, ..., r\}$ has to be removed by exploiting equations similar to (4.66) and (4.69)–(4.71), with $l = 0, 1, ..., r$. This allows us to progressively refine the amplitude and the delay of each echo, so generating the final delays $\{\hat{F}_0[r], \hat{F}_1[r], ..., \hat{F}_r[r]\}$ and their complex amplitudes $\{\hat{A}_0[r], \hat{A}_1[r], ..., \hat{A}_r[r]\}$. Note that, in principle, the re-estimation can be accomplished multiple (say, N_{RES}) times.

Algorithm 7: Complex single delay estimation and cancellation (CSDEC)

a- Initialisation:

- Set the index $r \leftarrow 0$
- evaluate $\hat{F}_0[0]$ and $\hat{A}_0[0]$;
- Computation of the initial energy $\mathcal{E}_0[r]$ through eq. (4.63).

while $\mathcal{E}_0[r] > \mathcal{T}_{CSDEC}$ **do**

- $r \leftarrow r+1$
- *Estimation of* $\mathbf{X}_0[r]$ (4.55)-(4.56);
- *Coarse estimation* of $\bar{A}_r[r]$ and $\bar{F}_{c,r}[r]$ through eqs. (4.65)-(4.68).
- *Evaluation* of $\bar{X}_{k,\rho[r]}[r]$ through eqs. (4.69)-(4.71).
- *Evaluation* of $a(\hat{\alpha}[r])$, $b(\hat{\alpha}[r])$, $c(\hat{\alpha}[r])$ and $\hat{F}_r^{(0)}$ based on eqs. (4.38)-(4.40) and eq. (4.79), respectively;
- $\hat{A}_{\mathrm{CSDE},r} \leftarrow \bar{A}_r[r]$;
- $\hat{F}_{\mathrm{CSDE},r} \leftarrow \bar{F}_r[r]$;

b- Refinement:

Considering $\{\hat{A}_0[r-1], \hat{A}_1[r-1], ..., \hat{A}_{r-1}[r-1], \hat{A}_{\mathrm{CSDE},r}[r]\}$ and $\{\hat{F}_0[r-1], \hat{F}_1[r-1], ..., \hat{F}_{r-1}[r-1], \hat{F}_{\mathrm{CSDE},r}[r]\}$:

for $i = 1$ *to* N_{RES} **do**

for $i = 1$ *to* N_{CSDE} **do**

- *Coarse estimation* of $\bar{A}_r[r]$ and $\bar{F}_{c,r}[r]$ through eqs. (4.65)-(4.68).
- *Evaluation* of $\bar{X}_{k,\rho[r]}[r]$ based on eqs. (4.69)-(4.71).
- *Evaluation* of $a(\hat{\alpha}[r])$, $b(\hat{\alpha}[r])$, $c(\hat{\alpha}[r])$ and $\hat{F}_r^{(i)}$ based on eqs. (4.38)-(4.40) and eq. (4.79), respectively.

end

end

c- Computation of the energy $\mathcal{E}_0[r]$ through eq. (4.63)

end

The estimates generated by the CSDEC algorithms are biased. However, this bias can be removed by running an additional (i.e., a fourth) step after that these algorithms have been executed. In this final step, the estimation algorithm developed by *Ye* and *Aboutanios* in ref. [84], [85] is carried out after initializing it with the estimates $\{\hat{F}_0[\hat{L}],\ \hat{F}_1[\hat{L}],\ ...,\ \hat{F}_{\hat{L}-1}[\hat{L}]\}$ and their complex amplitudes $\{\hat{A}_0[\hat{L}],\ \hat{A}_1[\hat{L}],\ ...,\ \hat{A}_{\hat{L}-1}[\hat{L}]\}$ ($\{\hat{C}_0[\hat{L}],\ \hat{C}_1[\hat{L}],\ ...,\ \hat{C}_{\hat{L}-1}[\hat{L}]\}$) generated by the CSDE. The hybrid techniques resulting from interconnecting the CSDEC algorithms with the above mentioned algorithm are dubbed *hybrid* CSDEC (HCSDEC). The scheme for the CSDEC is proposed in Algorithm 7. The CSDE and CSDEC algorithms are employed in the bi-dimensional and three-dimensional imaging described below in Par. 4.2.

It is worth pointing out that:

a) The oversampling factor M adopted in the computation of the vectors $\{\mathbf{X}_k^{(l)}\}$ and the stopping criterion employed by the CSDE (i.e. the maximum value of N_{CSDE}) need to be carefully adjusted to achieve a good accuracy in the estimation of the parameters of each new exponential.

b) Poor estimation of the normalised delay F_l and/or the complex amplitude A_l may lead to significant *error accumulation* if **CSDEC-S3** is removed; it is important to keep in mind that a fundamental role in accurate cancellation is played by the accuracy of the estimated delay residual.

c) The threshold $\mathcal{T}_{\text{CSDEC}}$ needs to be properly adjusted in order to ensure that the probability that $\hat{L}$ equals to L is close to unity. On the one hand, a large value of $\mathcal{T}_{\text{CSDEC}}$ may lead to miss weaker echoes; on the other hand, a small value of this parameter may lead to the identification of non-existent echoes and, consequently, of false targets in a radar system.

4.1.3 Computational complexity

The complexity of the estimation algorithms developed in paragraphs (4.1.1) and (4.1.2) has been carefully assessed in terms of number of *floating operations* (flops) to be executed in the detection of L targets. This analysis leads to the conclusion that these are approximately of order $\mathcal{O}(M_{\text{CSDE}})$, with

$$M_{\text{CSDE}} = N_0 \log_2 N_0 + K_{\text{CSDE}}\, N_{\text{CSDE}}\, I^2; \tag{4.80}$$

here, N_{CSDE} represents the overall number iterations accomplished by the CSDE and $K_{\text{CSDE}} = 1/2$. Computer simulations have evidenced that, in the considered scenarios, a small value of I is required if the so called *barycentric interpolation* is employed (see [76]). For this reason, the contribution of the second term appearing in the RHS of both eq. (4.80) can be neglected; therefore, the order of the overall computational cost of both estimators is approximated well by the first term, which originates from IFFT processing. Moreover, based on the last results, it is not difficult to show that the computational costs of the CSDEC are approximately of order $\mathcal{O}(M_{\text{CSDEC}})$, with

$$M_{\text{CSDEC}} = N_0 \log_2(N_0) + K_{\text{CSDE}}\, L\, N_{\text{CSDE}}\, I^2, \tag{4.81}$$

if no re-estimation is accomplished (see **CSDEC-S3** in the description of the CSDEC algorithms) and the algorithms stop after detecting the last echo. Note that the first term appearing in the RHS of eq. (4.81) accounts for the initialization (and, in particular, for the computation of the vectors $\mathbf{X}_0$ (4.13) and $\{\mathbf{X}_k\}$ (4.33)), whereas the second term for

the fact that, in the CSDEC, the CSDE is executed L times. It is worth noting that the costs due to the evaluation of the estimated echoes detected after the first one and to their time domain cancellation do not play an important role in this case. However, if *re-estimation is accomplished*, the parameter L appearing in the RHS of 4.81) is replaced by L^2, since this task involves all the estimated exponentials. Despite this, the increase in the overall computational cost of the CSDEC with respect to the CSDE is limited since the use of re-estimation allows these algorithms to achieve convergence with a smaller value of the parameter N_{CSDE}.

4.2 Bi-dimensional and three-dimensional imaging

All the algorithms developed in the following section can be considered as specific instances of the general detection and estimation method described by the block diagram shown in Fig. 4.1. The processing accomplished by the blocks this diagram consists of can be summarized as follows. Each vector of the set $\{\mathbf{x}_c^{(v)}\}$, collecting N_V vectors (see eq. (2.25)), undergoes IFFT processing, so that the analysis of the acquired measurements is moved from the frequency-domain to time-domain. The output of the IFFT block is processed by the *range profile estimator* (RPE), that generates the so called *target range profile* (TRP), i.e. a collection of the ranges at which the relevant echoes are detected and of the associated energies. Note that the last quantities allow us to rank each range on the basis of its preceptual importance. The output of the IFFT processing and the target range profile are processed by the *spatial estimator* (SPE). This block detects all the targets associated with each range appearing in the TRP and estimates their angular parameters; moreover, it may generate a finer estimate of their range. The SPE output is represented by the set $\{(\hat{R}_l, \hat{\theta}_l, \hat{\phi}_l); l = 0, 1, ..., \hat{L} - 1\}$, where $\hat{L}$ represents an estimate of the parameter L (i.e., of the overall number of point targets), whereas $\hat{R}_l$, $\hat{\theta}_l$ and $\hat{\phi}_l$ represent an estimate of the range R_l, azimuth θ_l and elevation ϕ_l, respectively, of the l-th target (with $l = 0, 1, ..., \hat{L} - 1$).

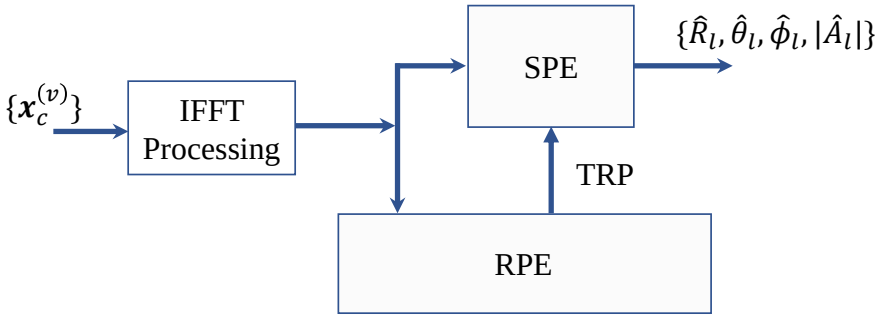


FIGURE 4.1: Block diagram describing the general method proposed in this work.

It is important to point out that:

1. If this method is adopted, range estimation is decoupled from angular estimation, so that a 3D (2D) detection and estimation problem is turned into a) a 1D detection/estimation problem involving the detection of multiple targets and the estimation of their ranges only plus b) a 2D (1D) estimation problem concerning the targets detected at the same range and the estimation of their azimuth and elevation (azimuth) only. Consequently, the overall problem of detecting multiple targets and estimating their range and angles is turned into a pair of simpler detection and estimation problems.

2. The SPE exploits the range information generated by the RPE in order to *concentrate its computational effort on a set of well defined ranges*; this allows reducing the size of the search space involved in spatial estimation. This explains also why the processing accomplished by the SPE cannot start before that at a least a portion of the range/energy information (i.e., a portion of the TRP) generated by the RPE becomes available.

3. Various techniques can be exploited in the RPE and in the SPE to develop computationally efficient implementations of this method.

As far as the last point is concerned, the following techniques can be adopted by the RPE to mitigate its complexity:

a) *Antenna selection* – This consists in feeding the RPE with a subset of the outputs of the IFFT block; such outputs are generated on the basis of N_A of the N_V VAs. Note that, on the one hand, a larger N_A allows to compute more accurate TRP; on the other hand, selecting a smaller N_A results in a reduction of the overall effort required for the computation of the TRP.

b) *Antenna-by-antenna processing* – The measurements acquired through the selected N_A VAs can be efficiently processed by adopting a two-step procedure. In the first step, target range estimation is accomplished on each VA *independently of all the other* VAs, i.e. the acquired measurements are processed on an antenna-by-antenna basis; this is beneficial when parallel computing hardware is employed in the execution of the first step. In the second step, instead, the target range information extracted from each of the selected N_A VAs are fused to generate the TRP.

c) *Serial target cancellation in the range domain* – Target detection and range estimation on each VA represents a *multidimensional problem* since they aim at detecting multiple targets and estimating their ranges. In the considered method, this multidimensional problem is turned into a sequence of 2D estimation problems by adopting a *serial interference cancellation* (SIC) approach (like the one proposed in Par. 3.3 and Par. 3.4). This means that the noisy signal observed on each VA is processed in an iterative fashion. At each iteration, a single (and, in particular, the strongest) target is detected, and its range and complex amplitude are estimated. Then, the contribution of this target to the received signal is estimated and subtracted from the signal itself (i.e., the detected target is treated as a form of *interference* to be cancelled), so generating a *residual signal*. The last signal represents the input of the next iteration. This procedure is repeated until the overall energy of the residual drops below a given threshold. Note also that the use of this SIC-based approach allows us to mitigate the impact of the spectral leakage due to strong targets, that can potentially hide weak targets having similar ranges. The range and complex estimation in each selected VA is implemented by the CSDEC algorithm thoroughly described in paragraph 4.1.2.

In the SPE, instead, the following techniques can be employed to reduce its overall computational complexity:

a) *Alternating maximization* – As for MWLA (see paragraph 3.3 and 3.4), the *alternating maximization* technique is exploited to develop iterative algorithms that alternate the estimation of the elevation of a given target with that of its azimuth; for this reason, a 2D optimization problem is turned into a pair of interacting 1D optimization problems. Actually, this technique is also exploited by the RPE in its

serial target cancellation in the range domain; in fact, this allows to decouple the estimation of the normalised delay characterizing a given target from that of its complex amplitude.

b) *Serial target cancellation in the angular domain* – Each of the ranges collected in the TRP is associated with an unknown number of targets; for this reason, the processing accomplished by SPE aims at resolving all the targets associated with a given range and estimating their angular coordinates. In the considered method, a SIC approach is exploited to turn this multidimensional problem into a sequence of 2D (1D) estimation problems if a 3D (2D) propagation scenario is considered. This means that the noisy data referring to a specific range and acquired on all the VAs are iteratively processed to detect a single (and, in particular, the strongest) target, and to estimate its angular coordinates and complex amplitude. Then, the contribution of this target to the outputs of the IFFT block is estimated and subtracted from them, so generating a set of *residual data*. This *detection/estimation/cancellation* procedure is iteratively applied to the residual data until its overall energy drops below a given threshold. Moreover, in a 3D propagation scenario, this procedure is combined with the AM approach described in the previous point; this allows to detect and estimate the angular parameters of a single target (i.e., to solve a 2D optimization problem) by means of an iterative procedure alternating the estimation of its elevation with that of its azimuth (i.e., by means of an algorithm solving two intertwined 1D optimization problems). Note also that the use of serial cancellation allows us to mitigate the impact of the spectral leakage due to strong targets, that can potentially hide weak targets having similar spatial coordinates.

c) *Parallel processing of the data associated with different ranges* – The detection and the estimation of the targets associated with distinct ranges of the TRP can be accomplished in a *parallel fashion* or in a *sequential fashion*. The first approach is more efficient than the second one if spatial estimation is executed on parallel computing hardware. In fact, in this case, multiple spatial estimation algorithms can be run *in parallel*, one for each of the ranges appearing in the TRP. Note, however, that the price to be paid for this is represented by the fact that the target information generated by any parallel procedure need to be fused at its end. In fact, the analysis of the measurements referring to close ranges appearing in the TRP may lead to detecting the same target more than once. This problem can be circumvented by adopting the second approach, since, in this case, the measurements associated with each of the ranges appearing in the TRP are processed *sequentially*. In doing so, the followed order is based on the importance of these ranges, i.e. on the associated energies.

Based on the general method outlined above and on the techniques listed for the RPE and the SPE, two specific algorithms are developed in the following. These algorithms are denoted by *range & angle serial cancellation algoritm #X* (briefly RASCA#*X* in the following), with $X = 1$ and 2. In their description it is assumed, without loss of generality, that the available measurements are acquired through the $N_V = N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ virtual *uniform rectangular array* (URA) represented in Fig. 4.2 in the case of 3D imaging and through an horizontal ULA, consisting of N_{VH} virtual antennas, in the case of 2D imaging. In the first case, the horizontal (vertical) spacing between adjacent antennas is denoted by d_{VH} (d_{VV}), whereas, in the second one, is denoted by d_{VH}. Therefore, in this considerations, it has been assumed that a *reference virtual antenna*, identified by

$(p,q) = (p_R, q_R)$ in the 3D (2D) case[3], is selected in the virtual array, as exemplified by Fig. 4.2.

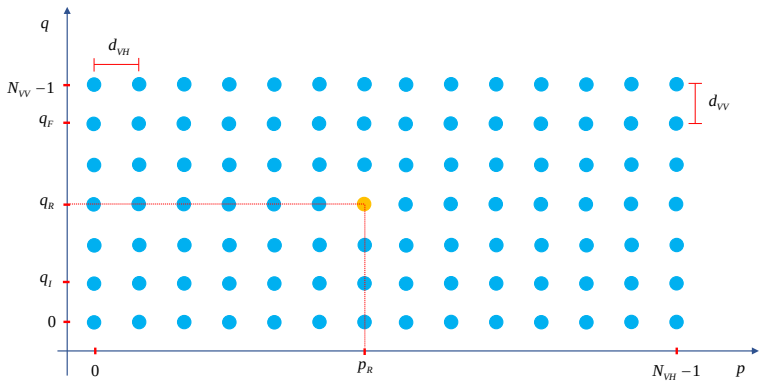


FIGURE 4.2: Virtual antenna array considered in the description of the detection and estimation algorithms. The selected reference virtual antenna is also shown.

4.2.1 Three-dimensional range & angle serial cancellation algorithm #1

The inner structure of the RASCA#1 is described by the block diagram shown in Fig. 4.3.

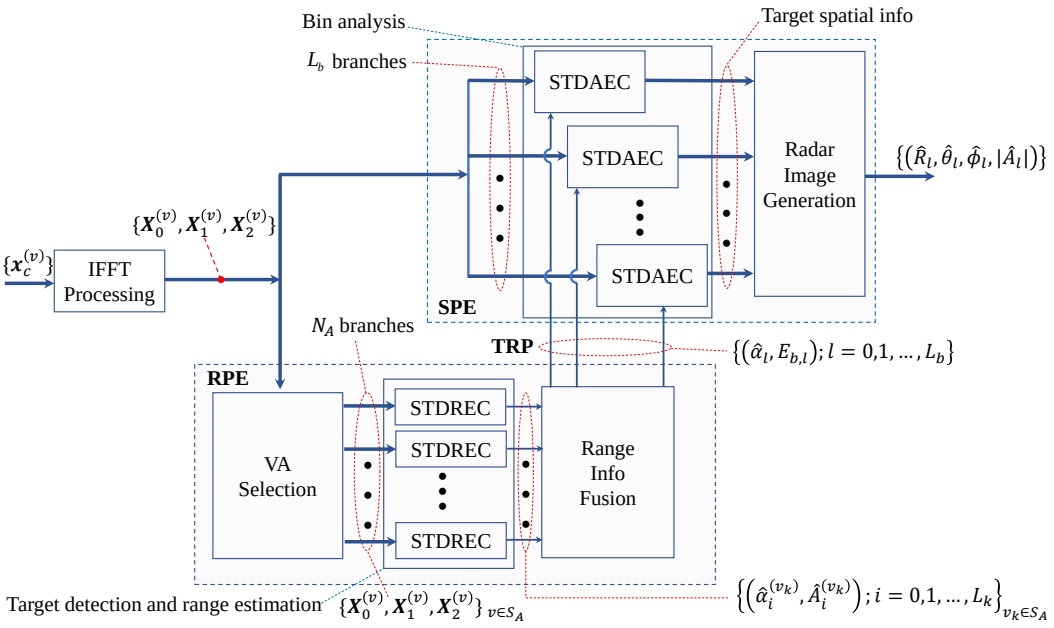


FIGURE 4.3: Block diagram describing the overall structure of the RASCA#1.

The processing accomplished inside the blocks appearing in that figure can be summarized as follows. The IFFT *processing* block allows to turn the *frequency domain* information provided by the set of N_V vectors $\{\mathbf{x}_c^{(v)}\}$ into the *time domain* information feeding both the RPE and the SPE blocks. This transformation requires the evaluation of $3N_V$ IFFTs, all of order N_0. In fact, it consists in the evaluation of the triad $(\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)})$,

[3]in this chapter the indexes pair $(p\,q)$ is employed instead of $(t\,r)$ for indicating a specific virtual channel; in particular, the p-th (q-th) index refers to an element of the horizontal (vertical) ULA.

collecting three N_0-dimensional vectors, on the basis of $\mathbf{x}_c^{(v)}$, for $v = 0, 1, ..., N_V - 1$. For this reason, the output of the considered block is represented by the set

$$\mathcal{S}_{\text{IFFT}} \triangleq \left\{ \left(\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)} \right) ; v = 0, 1, ..., N_V - 1 \right\}, \tag{4.82}$$

consisting of $3 \cdot N_V$ N_0-dimensional vectors. Note that, however, a portion of this set is discarded by the RPE, since this block processes the information originating from N_A distinct VAs only. The triads selected by the RPE form the subset

$$\bar{\mathcal{S}}_{\text{IFFT}} \triangleq \left\{ \left(\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)} \right) ; v \in \mathcal{S}_A \right\}, \tag{4.83}$$

of $\mathcal{S}_{\text{IFFT}}$ (4.82); here,

$$\mathcal{S}_A \triangleq \{ v_0, v_1, ..., v_{N_A - 1} \}, \tag{4.84}$$

represents the set of the values of the VA index v identifying the elements of $\mathcal{S}_{\text{IFFT}}$ that belong to $\bar{\mathcal{S}}_{\text{IFFT}}$. Each of the triads of $\bar{\mathcal{S}}_{\text{IFFT}}$ is processed, independently of all the other ones, by a novel iterative estimation algorithm called *single target detection, range estimation and cancellation* (STDREC). This algorithm detects the most relevant targets on the selected antenna and estimates their ranges, i.e. the delays associated with these ranges; (see eqs. (2.10)-(2.11)) and their complex amplitudes (see eq. (2.26)). Its name originates from the fact that this algorithm, in each of its iterations, *detects* a single target (namely, the strongest target), *estimates* its parameters (and, in particular, the delay characterizing it, i.e., its range) and *cancels* the contribution given by it to the received signal; the residual signal resulting from target cancellation represents the input of the next iteration. The output of the STDREC algorithm that processes the raw data originating from the v_k-th VA is represented by the set

$$\mathcal{S}_{v_k} \triangleq \left\{ \left(\hat{\alpha}_i^{(v_k)}, \hat{A}_i^{(v_k)} \right), i = 0, 1, \ldots, L_k - 1 \right\}, \tag{4.85}$$

with $k = 0, 1, ..., N_A - 1$; here, L_k is the overall number of targets detected on the considered VA, whereas $\hat{A}_i^{(v_k)}$ and $\hat{\alpha}_i^{(v_k)}$ represent the estimate of the complex amplitude of the i-th target and the index of the delay bin[4] in which this target has been detected (with $i = 0, 1, \ldots, L_k - 1$). Finally, the information provided by the N_A sets $\{\mathcal{S}_{v_k}\}$ are merged to generate the single set

$$\mathcal{S}_{\text{RPE}} \triangleq \{ (\hat{\alpha}_l, E_{b,l}), l = 0, 1, \ldots, L_b - 1 \}, \tag{4.86}$$

where L_b is the overall number of targets detected on all the selected VAs, $\hat{\alpha}_l$ is the index of the *delay bin* in which the l-th target has been detected and $E_{b,l}$ is the average energy estimated for it (with $l = 0, 1, \ldots, L_k - 1$). Note that:

- The cardinality L_b of the set $\mathcal{S}_{\text{RPE}}$ represents a preliminary estimate of the overall number of targets; in fact, multiple targets having the same range or ranges whose mutual differences are below the resolution of the employed radar system are detected as a single target and no effort is made at this stage to separate their contributions.
- The energies $\{E_{b,l}\}$ represent the perceptual importance of the identified delay bins, in the sense that a larger energy is associated with a more important delay bin.

[4]The index of the delay bin is computed on the basis of the N_0-dimensional vector $\mathbf{X}_0^{(v)}$ (see eq. (4.82)) collecting the elements of the N_0 order IFFT of the input time domain vector $\{\mathbf{x}_c^{(v)}\}$

Both sets $\mathcal{S}_{\text{IFFT}}$ (4.82) and $\mathcal{S}_{\text{RPE}}$ (4.86) feed the SPE. The aim of this block is to analyse the spectral information associated with the ranges (i.e., with the delay bins) identified by the RPE in order to: a) estimate the angular coordinates (i.e., azimuth and elevation) of the targets contributing to each delay bin; b) detect additional targets associated with adjacent delay bins and potentially hidden by the spectral leakage due to stronger targets; c) estimate the angular coordinates (i.e., azimuth and elevation) of such additional targets and compute a finer estimate of their range.

The first stage of the processing accomplished by the SPE involves the whole set $\mathcal{S}_{\text{IFFT}}$ (4.82) and is executed on a bin-by-bin basis, since it aims at: a) detecting all the targets that contribute to the energy of each bin contained in the TRP and b) estimating their angular coordinates. For this reason, this stage consists of L_b estimators running *in parallel*; each estimator focuses on one of the L_b delay bins (i.e., ranges) appearing in the TRP (see Fig. 4.3). Moreover, each estimator executes a novel iterative estimation algorithm, called *single target detection, angular estimation and cancellation* (STDAEC). The l-th STDAEC algorithm processes the spectral information available on the whole virtual receive array and referring to the $\hat{\alpha}_l$-th delay bin only (with $l = 0, 1, ..., L_b - 1$), detects $L[l]$ targets and, for each detected target, computes: a) an estimate of its complex amplitude; b) an estimate of its angular coordinates (i.e., its azimuth and its elevation); c) a refined estimate of its range (do not forget that the preliminary estimate of this range is provided by the index $\hat{\alpha}_l$ of the considered delay bin). If $D[l]$ iterations are accomplished by the l-th STDAEC algorithm, $D[l]$ distinct targets are detected in the $\hat{\alpha}_l$-th delay bin if none of them is classified as a *false* (i.e., *ghost*) *target*. In addition, all the estimates generated by this algorithm are collected in the set

$$\mathcal{T}_l \triangleq \left\{ \left(\hat{A}_i\left[l\right], \hat{F}_i\left[l\right], \hat{\alpha}_i\left[l\right], \hat{F}_{\text{V},i}[l], \hat{F}_{\text{H},i}[l] \right);\ i = 0, 1, \ldots, D[l] - 1 \right\}, \tag{4.87}$$

with $l = 0, 1, \ldots, L_b - 1$; here, $\hat{A}_i\left[l\right]$, $\hat{F}_i\left[l\right]$ and $\hat{\alpha}_i\left[l\right]$ denote the estimates of the complex amplitude $A_i\left[l\right]$, of the normalized delay $F_i\left[l\right]$ and of the delay bin $\alpha_i\left[l\right]$, respectively, characterizing the i-th target detected *on the reference antenna* and in the l-th delay bin, whereas $\hat{F}_{\text{H},i}[l]$ and $\hat{F}_{\text{V},i}[l]$ represent the estimates of the normalized horizontal spatial delay and of the normalized vertical spatial delay referring to the above mentioned target.

Finally, in the second (and last) stage of the SPE, the spatial coordinates of all the detected targets are computed on the basis of the spatial information collected in the L_b sets $\{\mathcal{T}_l\}$ and an overall image of the propagation scenario is generated in the form of a *point cloud*. The derivation of the STDAEC algorithm relies on the fact that: a) each target provides an *additive contribution* to the temporal spectra evaluated on all the VAs; b) the phase of this contribution exhibits *periodic variations* as we move horizontally or vertically along the considered virtual array (see Fig. 4.2). In fact, if we assume that the intensity of the echo received by each VA from the i-th target detected in the l-th delay bin does not change from antenna to antenna, the complex amplitude $A_i[p, q, l]$ observed on the (p, q)-th VA can be expressed as[5] (see eqs. (2.26) and (2.10))

$$A_i[p, q, l] = A_i[l] \exp\left\{ -j\frac{4\pi}{\lambda} \left[d_{\text{VH}} \left(p - p_R\right) \cos(\phi_i[l]) \sin(\theta_i[l]) + d_{\text{VV}} \left(q - q_R\right) \sin(\phi_i[l]) \right] \right\}; \tag{4.88}$$

[5]Note that the phase of this amplitude is similar to the one of the steering vector proposed in paragraph 3.2.

here, $\lambda = c/f_0$ is the wavelength associated with the start frequency (see eq. (2.24)), (p_R, q_R) is the pair of integers identifying the selected *reference antenna*, $\theta_i[l]$, $\phi_i[l]$ and $R_i[l]$ are the azimuth, the elevation and the range, respectively, characterizing the considered target, and $A_i[l]$ is its complex amplitude observed on the reference antenna. If eq. (4.88) holds, the rate of the phase variations observed in the complex amplitudes $\{A_i[p,q,l]\}$ for a given l as we move horizontally or vertically along the considered virtual receive array is proportional to

$$F_{\mathrm{H},i}[l] \triangleq \frac{2d_{\mathrm{VH}}}{\lambda}\cos(\phi_i[l])\sin(\theta_i[l]) \tag{4.89}$$

and

$$F_{\mathrm{V},i}[l] \triangleq \frac{2d_{\mathrm{VV}}}{\lambda}\sin(\phi_i[l]), \tag{4.90}$$

respectively. In fact, the quantity $F_{\mathrm{H},i}[l]$ ($F_{\mathrm{V},i}[l]$) represents the normalized horizontal (vertical) spatial delay characterizing the i-th target detected in the l-th delay bin; if both these delays are known, the elevation and the azimuth of this target can be evaluated as

$$\phi_i[l] = \arcsin\left(\frac{\lambda}{2d_{\mathrm{VV}}}F_{\mathrm{V},i}[l]\right) \tag{4.91}$$

and

$$\theta_i[l] = \arcsin\left(\frac{\lambda}{2d_{\mathrm{VH}}\cos(\phi_i[l])}F_{\mathrm{H},i}[l]\right), \tag{4.92}$$

respectively.

Moreover, in the development of the STDAEC algorithm, the following two techniques are exploited:

Serial cancellation of targets – This technique is conceptually similar to the cancellation strategy exploited by the STDREC algorithm and allows us to detect multiple targets in the same delay bin and, in particular, to identify targets having similar spatial coordinates. It is important to keep in mind that the delays associated with distinct targets detected in the same delay bin do not necessarily belongs to that bin; in fact, they can belong to adjacent bins, so that the tails (not the peak) of their spectra are really observed in the considered delay bin. This problem originates from the fact that, generally speaking, *the contribution of a point target to the spectrum computed on each* VA *is not a line*, unless the associated delay is exactly a multiple of the IDFT delay term f_{IDFT} (4.103); consequently, such a contribution is spread over multiple adjacent delay bins (i.e., spectral leakage is observed).

Spatial folding – As already stated above, the delay associated with a target detected in the l-th delay bin does not necessarily fall exactly in that bin. The technique dubbed *spatial folding* has been devised to; a) evaluate a more accurate estimate of the delay associated with a target detected in a given bin; b) discriminate real targets from ghost targets. Spatial folding is based on the following idea. Once the horizontal and the vertical spatial delays associated with a target detected in a given delay bin have been estimated (see eqs. (4.89) and (4.90)), the spectra computed for all the VAs can be combined *in a constructive fashion* by

- taking a *reference* VA (identified by $(p,q) = (p_R, q_R)$; see Fig. 4.2), and compensating for the phase differences, estimated for that target, between the reference VA

and all the other VAs, or

- taking a *reference* ULA and compensating for the phase differences, estimated for that target, between the reference ULA and other ULAs *parallel* to the reference ULA.

In case 1), folding generates a single spectrum, dubbed *folded spectrum*, and has the beneficial effects of a) averaging out the effects of the noise affecting different VAs and b) combining, in an unconstructive fashion, the contributions of all the targets different from the one which the employed spatial delays refer to. For this reason, in analysing the amplitude of the folded spectrum, a well defined peak in its amplitude is expected in the l-th delay bin or in a bin close to it. When this occurs, the position of this peak allows to compute a refined estimate of the delay (and, consequently, of the range) characterizing the target for which folding has been accomplished. On the contrary, if this does not occur, the detected target is actually a *ghost target*. In case 2), folding generates as many *folded spectra* as the number of antennas of the reference ULA and offers the same advantages as case 1).

In the remaining part of this thesis, when folding is used, the following terminology is adopted:

Vertical folding – This refers to the case in which folding involves a reference *horizontal* ULA (HULA) on which other HULAs are folded.

Overall folding – This refers to the case in which folding involves all the spectra, i.e. the overall URA; a single folded spectrum is computed in this case.

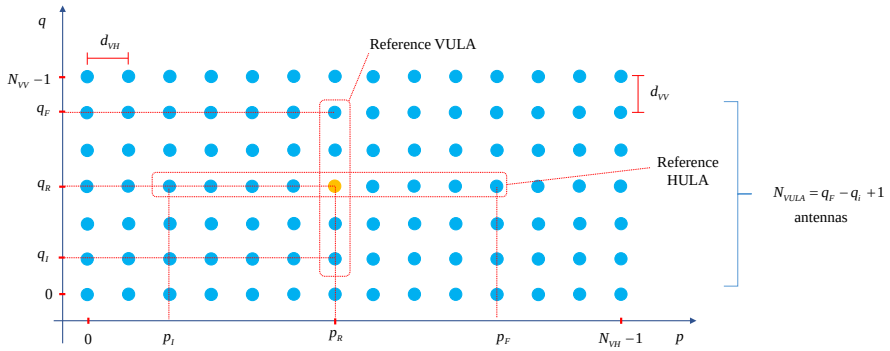


FIGURE 4.4: Example of reference VULA and reference HULA including the reference antenna.

Note that, in any case, folding may involve the *whole* virtual receive array or *a portion of it*. The exploitation of a subset of the available VAs is motivated by the fact that, in practice, in computing a folded spectrum referring to the l-th delay bin, the *estimates* $\hat{F}_{\mathrm{H},i}[l]$ and $\hat{F}_{\mathrm{V},i}[l]$ of the delays $F_{\mathrm{H},i}[l]$ and $F_{\mathrm{V},i}[l]$, respectively, are employed, so that the quality of the phase compensation factors computed for the antennas that are *farther* from the reference antenna or the reference HULA may be affected by significant *estimation errors*. The exploitation of spectral folding in the STDAEC algorithm requires:

1. Selecting a *reference* VULA, that consists of N_{VULA} adjacent and vertically aligned VAs (with $N_{\mathrm{VULA}} \leq N_{\mathrm{VV}}$), within the virtual array; in the following, let us assume, that the reference VULA includes the reference antenna and, consequently, is identified by $q = q_I, q_I+1, ..., q_F$, with $q_I \leq q_R \leq q_F$ and $p = p_R$, so that $N_{\mathrm{VULA}} = q_F - q_I + 1$ (see Fig. 4.4).

2. Selecting a *reference* HULA, that consists of N_{HULA} adjacent and horizontally aligned VAs; in the following, we assume that the reference HULA is the horizontal ULA containing the reference antenna and, consequently, is identified by $q = q_R$ and $p = p_I, p_I + 1, ..., p_F$, with $p_I \leq p_R \leq p_F$, so that $N_{\text{HULA}} = p_F - p_I + 1$ (see Fig. 4.4).

3. Selecting a set of HULAs, different from the reference HULA and having the same size of it (i.e., the same number of VAs); in the following, we assume that these HULAs, called *vertically folded* HULAs, correspond to $q = q_I^{(\text{VF})}, q_I^{(\text{VF})} + 1, ..., q_R - 1, q_R + 1, ..., q_F^{(\text{VF})}$, with $q_I^{(\text{VF})} \leq q_R \leq q_F^{(\text{VF})}$, as illustrated in Fig. 4.5; note that the overall number of involved HULAs is $N_{\text{HULA}}^{(\text{VF})} = q_F^{(\text{VF})} - q_I^{(\text{VF})} + 1$.

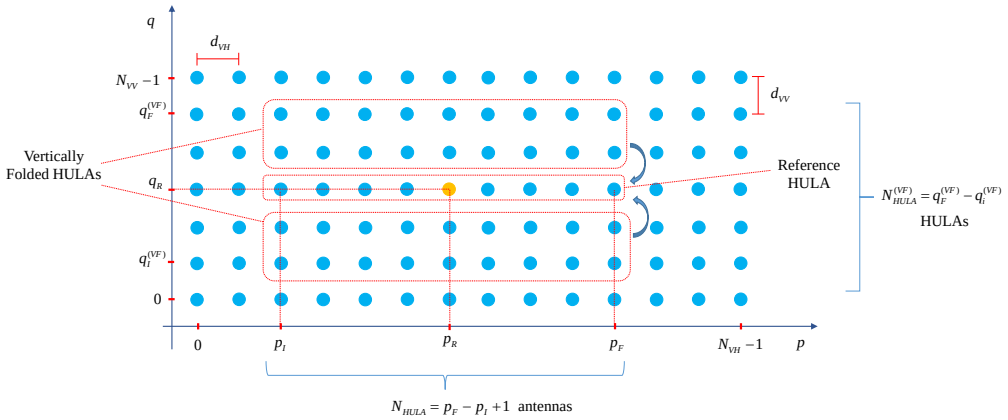


FIGURE 4.5: Representation of a set of vertically folded HULAs.

In the following, the RASCA#1 processing is divided in three tasks, each associated with one of the blocks appearing in Fig. 4.3 (the i-th task is denoted by **Ti** in the following). A description of each task is provided below.

T1 – IFFT processing The processing accomplished within this task can be summarized as follows. Given the vector $\mathbf{x}_c^{(v)}$ (see eq. (2.25)), the N-dimensional vectors

$$\mathbf{x}_1^{(v)} \triangleq \left[x_{1,0}^{(v)}, x_{1,1}^{(v)}, \ldots, x_{1,N-1}^{(v)}\right]^T \tag{4.93}$$

and

$$\mathbf{x}_2^{(v)} \triangleq \left[x_{2,0}^{(v)}, x_{2,1}^{(v)}, \ldots, x_{2,N-1}^{(v)}\right]^T \tag{4.94}$$

are evaluated for $v = 0, 1, ..., N_{\text{V}} - 1$; here,

$$x_{m,n}^{(v)} \triangleq n^m \, x_{c,n}^{(v)} \tag{4.95}$$

with $n = 0, 1, ..., N - 1$ and $m = 1, 2$. Then, the vectors $\mathbf{x}_c^{(v)}$, $\mathbf{x}_1^{(v)}$ and $\mathbf{x}_2^{(v)}$ undergo *zero padding* (ZP) for any v; this produces the N_0-dimensional vectors

$$\mathbf{x}_{0,\text{ZP}}^{(v)} = \left[(\mathbf{x}_c^{(v)})^T \, \mathbf{0}_{(M-1)N}^T\right]^T, \tag{4.96}$$

$$\mathbf{x}_{1,\text{ZP}}^{(v)} = \left[(\mathbf{x}_1^{(v)})^T \, \mathbf{0}_{(M-1)N}^T\right]^T \tag{4.97}$$

and

$$\mathbf{x}_{2,\mathrm{ZP}}^{(v)} = \left[(\mathbf{x}_2^{(v)})^T \, \mathbf{0}_{(M-1)N}^T \right]^T, \tag{4.98}$$

respectively; here, M is a positive integer (dubbed *oversampling factor*), $\mathbf{0}_D$ is a D-dimensional (column) null vector and

$$N_0 \triangleq M\,N. \tag{4.99}$$

Finally, the N_0-dimensional vectors

$$\begin{aligned} \mathbf{X}_m^{(v)} &= \left[X_{m,0}^{(v)}, X_{m,1}^{(v)}, ..., X_{m,N_0-1}^{(v)} \right]^T = \mathbf{X}_m\,[p,q] \\ &= \left[X_{m,0}\,[p,q]\,, X_{m,1}\,[p,q]\,, ..., X_{m,N_0-1}\,[p,q] \right]^T \\ &\triangleq \mathrm{IDFT}_{N_0} \left[\mathbf{x}_{m,\mathrm{ZP}}^{(v)} \right], \end{aligned} \tag{4.100}$$

with $m = 0,\ 1,\ 2$, are computed for any v (i.e., for any p and q); here, $\mathrm{IDFT}_{N_0}\,[\mathbf{x}]$ denotes, up to a scale factor, the N_0 order *inverse discrete Fourier transform* of the N_0-dimensional vector $\mathbf{x}$. More specifically, we assume that

$$X_{m,k}^{(v)} \triangleq \frac{1}{N} \sum_{n=0}^{N-1} n^m \, x_{c,n}^{(v)} \exp\left(j \frac{2\pi n k}{N_0} \right), \tag{4.101}$$

with $k = 0, 1, ..., N_0 - 1$ and $m = 0,\ 1,\ 2$. [6]

T2 – RPE The processing accomplished within this task consists of three consecutive steps listed below (the i-th step is denoted by T2-Si in the following); each step is associated with one of the blocks included in the RPE block, as shown in Fig. 4.3.

T2-S1) VA *selection* – In this step, the set $\bar{\mathcal{S}}_{\mathrm{IFFT}}$ (4.83) is built. This requires generating the set $\mathcal{S}_A$ (4.84), i.e. a set of N_A integers identifying the selected VAs (the elements of $\mathcal{S}_A$ are generated by *randomly* extracting N_A distinct integers from the set $\{0, 1, ..., N_V - 1\}$).

T2-S2) *Target detection and range estimation* – The processing carried out within this step is executed by the STDREC algorithm; this operates on an antenna-by-antenna basis. The STDREC processing for the v_k-th VA (with $k = 0, 1, ..., N_A - 1$) can be summarized as follows. A simple initialization is accomplished first by setting

$$\mathbf{X}_m^{(v_k)}[0] \triangleq \mathbf{X}_m^{(v_k)}, \tag{4.102}$$

with $m = 0,\ 1,\ 2$, and the iteration index i to 0. Then, the execution of the STDREC iterations is started; in the i-th iteration, the three steps described below are accomplished to detect a new target and cancel its contribution to the triad $(\mathbf{X}_0^{(v_k)}[i], \mathbf{X}_1^{(v_k)}[i], \mathbf{X}_2^{(v_k)}[i])$ (the p-th step of the considered i-th iteration is denoted by **STDREC-Sp** in the following).

STDREC-S1) *Detection of a new target and estimation of its parameters* – The triad $(\mathbf{X}_0^{(v_k)}[i], \mathbf{X}_1^{(v_k)}[i], \mathbf{X}_2^{(v_k)}[i])$ is processed to detect a new (i.e., the i-th) target, and to estimate the normalized delay $F_i^{(v_k)}$ and the complex amplitude $A_i^{(v_k)}$ associated with it. Note that, generally speaking, the normalized delay $F_i^{(v_k)}$ is not a multiple of the *fundamental delay*

$$F_{\mathrm{IDFT}} = t_s / N_0 \tag{4.103}$$

[6] Note that this is the same expression derived in eq. (4.31) for $\bar{X}_{k,\rho}$.

associated with the IFFT processing executed in **T1** ($t_s = 1/\Delta f$); for this reason, it can be expressed as

$$F_i^{(v_k)} = F_{c,i}^{(v_k)} + \delta_i^{(v_k)}\, F_{\text{IDFT}}, \tag{4.104}$$

where $F_{c,i}^{(v_k)}$ represents a *coarse estimate* of $F_i^{(v_k)}$ and $\delta_i^{(v_k)}$ is a real parameter called *residual*. This step consists in executing the *complex single delay estimator* (CSDE), that represent the most important part of the STDREC algorithm, and which detailed description is provided in paragraphs 4.1.1. In short, the CSDE computes the estimates $\hat{A}_i^{(v_k)}$, $\hat{F}_{c,i}^{(v_k)}$, $\hat{\delta}_i^{(v_k)}$, (see eqs. (4.48) and (4.49))

$$\hat{\alpha}_i^{(v_k)} = \left\lfloor \hat{F}_{c,i}^{(v_k)} / F_{\text{IDFT}} \right\rfloor \tag{4.105}$$

and

$$\hat{F}_i^{(v_k)} = \hat{F}_{c,i}^{(v_k)} + \hat{\delta}_i^{(v_k)}\, F_{\text{IDFT}} \tag{4.106}$$

of the parameters $A_i^{(v_k)}$, $F_{c,i}^{(v_k)}$, $\delta_i^{(v_k)}$, $\alpha_i^{(v_k)}$ and $F_i^{(v_k)}$, respectively, on the basis of the triad $(\mathbf{X}_0^{(v_k)}[i], \mathbf{X}_1^{(v_k)}[i], \mathbf{X}_2^{(v_k)}[i])$; here, $\hat{\alpha}_i^{(v_k)}$ represents the index of the delay bin in which the i-th target is detected on the v_k-th antenna. Note that the parameter $\hat{F}_i^{(v_k)}$, even if useless in the construction of the set $\mathcal{S}_{v_k}$ (4.85), is exploited in the next step.

STDREC-S2) *Cancellation of the new target* – The contribution $(\mathbf{C}_{X_0}^{(v_k)}[i], \mathbf{C}_{X_1}^{(v_k)}[i], \mathbf{C}_{X_2}^{(v_k)}[i])$, given by the i-th (i.e., by the last) target detected on the v_k-th VA to the triad $(\mathbf{X}_0^{(v_k)}[i], \mathbf{X}_1^{(v_k)}[i], \mathbf{X}_2^{(v_k)}[i])$ is

$$\mathbf{C}_{X_0}^{(v_k)}[i] = \hat{A}_i^{(v_k)}\, \bar{\mathbf{W}}_0^{(v_k)}[i], \tag{4.107}$$

$$\mathbf{C}_{X_1}^{(v_k)}[i] = \hat{A}_i^{(v_k)}\, \bar{\mathbf{W}}_1^{(v_k)}[i] \tag{4.108}$$

and

$$\mathbf{C}_{X_2}^{(v_k)}[i] = \hat{A}_i^{(v_k)}\, \bar{\mathbf{W}}_2^{(v_k)}[i], \tag{4.109}$$

respectively; here, $\bar{\mathbf{W}}_m^{(v_k)}[i]$ denotes, up to a scale factor, the N_0-th order IDFT of the vector

$$\bar{\mathbf{w}}_m^{(v_k)}[i] \triangleq \left[0, 1^m \cdot \bar{w}_i^{(v_k)}, 2^m \cdot \left(\bar{w}_i^{(v_k)}\right)^2, \ldots,\ (N-1)^m \cdot \left(\bar{w}_i^{(v_k)}\right)^{N-1}, 0, \ldots, 0\right]^T, \tag{4.110}$$

with $m = 0$, 1 and 2,

$$\bar{w}_i^{(v_k)} \triangleq \exp(-j2\pi\, \hat{F}_i^{(v_k)}), \tag{4.111}$$

$$\hat{F}_i^{(v_k)} \triangleq \hat{\tau}_i^{(v_k)} \Delta f \tag{4.112}$$

and $\hat{A}_i^{(v_k)}$ and $\hat{\tau}_i^{(v_k)}$ represent the estimates of the complex amplitude $A_i^{(v_k)}$ and of the delay $\tau_i^{(v_k)}$, respectively, characterizing the i-th target and the v_k-th selected channel. An efficient method can be used for the computation of the vector $\bar{\mathbf{W}}_m^{(v_k)}[i]$ appearing in the RHS of eqs. (4.107)–(4.109) (with $m = 0$, 1 and 2), since, for any m, this vector is, up to a scale factor, the N_0-th order IDFTs of the sequence $\{n^m\ (\bar{w}_i^{(v_k)})^n;\ n = 0, 1, ..., N-1\}$. In

fact, the l-th element of the vectors $\bar{\mathbf{W}}_m^{(v_k)}[i]$ is given by

$$\bar{W}_m^{(v_k)}[i,l] = \frac{1}{N}\sum_{n=0}^{N-1} n^m \left(\bar{w}_i^{(v_k)}\right)^n \exp\left(j\frac{2\pi l}{N_0}n\right) = \frac{1}{N}\sum_{n=0}^{N-1} n^m (\bar{q}[l])^n, \tag{4.113}$$

where

$$\bar{q}[l] \triangleq \exp\left(-j2\pi\left(\hat{F}_i^{(v_k)} - \frac{l}{N_0}\right)\right). \tag{4.114}$$

Even in this case, the identities listed in eqs. (4.76)-(4.78) can be exploited for an efficient computation of the RHS of eq. (4.113). Cancellation consists in the computation of the new *residual* triad

$$\mathbf{X}_m^{(v_k)}[i+1] = \left[X_{m,0}^{(v_k)}[i+1], ..., X_{m,N_0-1}^{(v_k)}[i+1]\right]^T \triangleq \mathbf{X}_m^{(v_k)}[i] - \mathbf{C}_{X_m}^{(v_k)}[i], \tag{4.115}$$

with $m = 0,\ 1,\ 2$.

STDREC-S3) *Computation of the residual energy in the time domain* – The energy

$$E_{i+1}^{(v_k)} \triangleq \left\|\mathbf{X}_0^{(v_k)}[i+1]\right\|^2 = \sum_{p=0}^{N_0-1}\left|X_{0,p}^{(v_k)}[i+1]\right|^2 \tag{4.116}$$

characterizing the residual spectrum vector $\mathbf{X}_0^{(v_k)}[i+1]$ (4.115) is computed and compared with the positive threshold T_{STDREC} (which may exhibit a dependence on range, i.e. on the detected delay). If this energy is below the threshold, the STDREC algorithm stops and $L_k = i$ relevant targets are detected on the v_k-th VA; otherwise, the recursion index i is increased by one and a new recursion is started by going back to **STDREC-S1**.

T2-S3) *Fusion of range information* – This step aims at merging the information provided by the N_A data sets $\{\mathcal{S}_{v_k}\}$ evaluated in the previous step. Its output is represented by the set $\mathcal{S}_{\text{RPE}}$ (4.86), whose elements (i.e., the L_b pairs $\{(\hat{\alpha}_l, E_{b,l})\}$) are evaluated as follows. If we define the set

$$\mathcal{A}_b^{(v_k)} \triangleq \{\hat{\alpha}_i^{(v_k)}; i = 0, 1, \ldots, L_k - 1\}, \tag{4.117}$$

identifying all the bins in which at least one target has been detected on the v_k-th VA (with $k = 0,\ 1,\ ...,\ N_A - 1$), the set

$$\mathcal{A}_b \triangleq \{\hat{\alpha}_l; l = 0, 1, \ldots, L_b - 1\} \tag{4.118}$$

is generated by putting together all the distinct integers that appear at least once in the N_A sets $\{\mathcal{A}_b^{(v_k)}; k = 0, 1, ..., N_A - 1\}$. Then, the average energy $E_{b,l}$ associated with the $\hat{\alpha}_l$-th bin (with $l = 0,\ 1,\ \ldots,\ L_b - 1$) is computed as

$$E_{b,l} = \frac{1}{N_{b,l}}\sum_{k=0}^{N_A-1}\sum_{i=0}^{L_k-1}\left|\hat{A}_i^{(v_k)}\right|^2 \delta\left[\hat{\alpha}_i^{(v_k)} - \hat{\alpha}_l\right], \tag{4.119}$$

where

$$N_{b,l} = \sum_{k=0}^{N_A-1}\sum_{i=0}^{L_k-1}\delta\left[\hat{\alpha}_i^{(v_k)} - \hat{\alpha}_l\right] \tag{4.120}$$

represents the overall number of antennas that contribute to this energy (here, $\delta[z] = 1$ if $z = 0$ and $\delta[z] = 0$ if $z \neq 0$).

T3 – SPE The processing accomplished within this task consists of the two consecutive steps listed below (the i-th step is denoted by T3-Si in the following); each step is associated with one of the blocks contained in the SPE representation shown in Fig. 4.3.

T3-S1) *Bin analysis* – Within this step, L_b STDAEC algorithms are run *in parallel*, one for each of the L_b ranges (i.e., delay bins) appearing in the TRP. A schematic description of l-th STDAEC algorithm is provided below (with $l = 0, 1, ..., L_b - 1$). This algorithm consists of three steps (its r-th step is denoted by **STDAEC-Sr** in the following) and is initialised by

1. Setting its iteration index i to 0.
2. Setting
$$\mathbf{X}^{(0)}[l] \triangleq \mathbf{X}[l], \tag{4.121}$$
where
$$\mathbf{X}[l] \triangleq [X_{0,\hat{\alpha}_l}[p,q]], \tag{4.122}$$
is a $N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ matrix collecting the spectral information available on the whole virtual receive array and referring to the $\hat{\alpha}_l$-th delay bin only.

Then, the STDAEC algorithm starts executing its iterations. Within its i-th iteration, it accomplishes the three steps described below.

STDAEC-S1) *Detection of a new target and estimation of its angular parameters* – In this step, the $N_V = N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ matrix
$$\mathbf{X}^{(i)}[l] \triangleq \left[X_l^{(i)}[p,q]\right], \tag{4.123}$$
is processed to detect the strongest target contributing to it, and to compute the estimates $\hat{\theta}_i[l]$, $\hat{\phi}_i[l]$, and $\hat{A}_i[l]$, of $\theta_i[l]$, $\phi_i[l]$) and $A_i[l]$, respectively (note that this target represents the i-th one detected in the considered delay bin, since $(i-1)$ targets have been detected in the previous recursions). The processing accomplished within this step is accomplished by a novel iterative detection and estimation algorithm called *single target detection and angular estimation* (STDAE); this algorithm consists in the five steps described below (its r-th step is denoted by **STDAE-Sr** in the following).

STDAE-S1) IFFT *processing on the reference* VULA – The portion of the initial spectral information referring to the reference VULA is extracted from the matrix $\mathbf{X}^{(i)}[l]$ and stored in the N_{VULA}-dimensional vector
$$\begin{aligned}\mathbf{S}_{\mathrm{VULA},0}^{(i)}[l] &= \left[S_{0,0}^{(i)}[l], S_{0,1}^{(i)}[l], ..., S_{0,N_{\mathrm{VULA}}-1}^{(i)}[l]\right]^T \\ &\triangleq \left[X_l^{(i)}[p_R,q_I], ..., X_l^{(i)}[p_R,q_F]\right]^T.\end{aligned} \tag{4.124}$$
Based on the last vector, the N_{VULA}-dimensional vector
$$\mathbf{S}_{\mathrm{VULA},k}^{(i)}[l] \triangleq \left[S_{k,0}^{(i)}[l], S_{k,1}^{(i)}[l], ..., S_{k,N_{\mathrm{VULA}}-1}^{(i)}[l]\right]^T, \tag{4.125}$$

is generated for $k = 1, 2$ and 3; here,

$$S_{k,p}^{(i)}[l] \triangleq p^k \, X_{\mathrm{VULA},p}^{(i)}[l] = p^k \, X_l^{(i)}[p_R, q_I + p] \tag{4.126}$$

with $p = 0, 1, ..., N_{\mathrm{VULA}} - 1$. The vector $\mathbf{S}_{\mathrm{VULA},k}^{(i)}[l]$ (with $k = 0, 1, 2$ and 3), after ZP, feeds a IDFT of order

$$\bar{N}_0 \triangleq \bar{M} \cdot N_{\mathrm{VULA}} \tag{4.127}$$

where $\bar{M}$ represents the adopted oversampling factor. This produces the vector

$$\begin{aligned} \mathbf{s}_{\mathrm{VULA},k}^{(i)}[l] &= \left[s_{k,0}^{(i)}[l], s_{k,1}^{(i)}[l], ..., s_{k,\bar{N}_0-1}^{(i)}[l] \right]^T \\ &\triangleq \mathrm{IDFT}_{\bar{N}_0} \left[\bar{\mathbf{S}}_{\mathrm{VULA},k}^{(i)}[l] \right], \end{aligned} \tag{4.128}$$

where $\bar{\mathbf{S}}_{\mathrm{VULA},k}^{(i)}[l]$ denotes the zero padded version of $\mathbf{S}_{\mathrm{VULA},k}^{(i)}[l]$. Then, we have that

$$s_{k,m}^{(i)}[l] \triangleq \frac{1}{N_{\mathrm{VULA}}} \sum_{p=0}^{N_{\mathrm{VULA}}-1} S_{k,p}^{(i)}[l] \exp\left(j \frac{2\pi p m}{\bar{N}_0} \right), \tag{4.129}$$

with $m = 0, 1, ..., \bar{N}_0 - 1$.

STDAE-S2) *Single angle estimation* – The $\bar{N}_0$-dimensional vectors $\{\mathbf{s}_{\mathrm{VULA},k}^{(i)}[l]; k = 0, 1, 2, 3\}$ are processed by the *complex single delay estimator* (CSDE) to detect the i-th (strongest target) appearing in l-th delay bin and estimate its angular coordinates (this target represents the i-th one detected in the considered delay bin, since $(i-1)$ targets have been detected in the previous iterations). The CSDE computes the estimates $\hat{A}_{\mathrm{V},i}[l]$ and $\hat{F}_{\mathrm{V},i}[l]$ of the parameters $A_i[l]$ and $F_{\mathrm{V},i}[l]$ (see eq. (4.90)), respectively. Note that the quantity $\hat{A}_{\mathrm{V},i}[l]$ is not exploited in the following since, it represents a *preliminary estimate* of $A_i[l]$.

STDAE-S3) *Vertical folding* – The estimate $\hat{F}_{\mathrm{V},i}[l]$ of the normalized vertical delay $F_{\mathrm{V},i}[l]$ (4.90) is employed to compensate for the phase difference between each of the HULAs to be vertically folded and the reference HULA (i.e., for the phase differences along the *vertical* direction), so that the spectral data associated with all these HULAs can be combined (i.e., summed) in a constructive fashion. In practice, first, the *phase rotation factor*

$$R_i^{(\mathrm{VF})}[l, q] \triangleq \left[\exp\left(j \Delta\psi_i^{(\mathrm{VF})}[l] \right) \right]^{q - q_R}, \tag{4.130}$$

with

$$\Delta\psi_i^{(\mathrm{VF})}[l] \triangleq 2\pi \hat{F}_{\mathrm{V},i}[l], \tag{4.131}$$

is computed for the q-th HULA (with $q = q_I^{(\mathrm{VF})}, q_I^{(\mathrm{VF})} + 1, ..., q_R - 1, q_R + 1, ..., q_F^{(\mathrm{VF})}$). Then, vertical folding is accomplished by computing the N_{HULA}-dimensional vector

$$\mathbf{X}_i^{(\mathrm{VF})}[l] = \mathbf{X}^{(i)}[l, q_R] + \sum_{\substack{q = q_I^{(\mathrm{VF})} \\ q \neq q_R}}^{q_F^{(\mathrm{VF})}} \mathbf{X}^{(i)}[l, q] \; R_i^{(\mathrm{VF})}[l, q], \tag{4.132}$$

that represents a set of N_{HULA} vertically folded spectra; here,

$$\mathbf{X}^{(i)}[l,q] \triangleq \left[X_l^{(i)}[p_I,q], X_l^{(i)}[p_I+1,q], ..., X_l^{(i)}[p_F,q]\right]^T, \tag{4.133}$$

is a N_{HULA}-dimensional row vector extracted from the q-th row of the matrix $\mathbf{X}^{(i)}[l]$ (4.123).

STDAE-S4) IFFT *processing and horizontal delay estimation* – The processing accomplished in this step is very similar to that carried out in **STDAE-S1** and **STDAE-S2**. In fact, the only difference is represented by the fact that the N_{VULA}-dimensional vector $\mathbf{S}_{\mathrm{VULA},0}^{(i)}[l]$ (4.124) is replaced by the N_{HULA}-dimensional vector $\mathbf{X}_i^{(\mathrm{VF})}[l]$ (4.132) generated in the previous step. Therefore, in this case, the CSDE is exploited to estimate the horizontal delay $F_{\mathrm{H},i}[l]$ and, again, the complex amplitude $A_i[l]$ associated with the i-th target. The CSDE computes the estimates $\hat{A}_{\mathrm{H},i}[l]$ and $\hat{F}_{\mathrm{H},i}[l]$ of the parameters $A_i[l]$ and $F_{\mathrm{H},i}[l]$, respectively. Note that: a) the quantity $\hat{A}_{\mathrm{H},i}[l]$ is not exploited in the following since, it represents a preliminary estimate of $A_i[l]$; b) the estimates $\hat{A}_{\mathrm{V},i}[l]$ and $\hat{A}_{\mathrm{H},i}[l]$ can be significantly different if multiple targets having similar horizontal delays or similar vertical delays contribute to the considered delay bin.

STDAE-S5) *Overall folding and delay/amplitude estimation*– In this step, the angular information (i.e., the delays $\hat{F}_{\mathrm{V},i}[l]$ (4.90) and $\hat{F}_{\mathrm{H},i}[l]$ (4.89)) computed in the previous step are exploited to accomplish overall folding[7]; this step involves the whole spectrum computed on the selected VAs. If the whole receive antenna array is exploited, overall folding consists in computing the N_0-dimensional vector

$$\mathbf{X}_{0,\mathrm{OF}}[l] \triangleq \sum_{p=0}^{N_{\mathrm{VH}}-1}\sum_{q=0}^{N_{\mathrm{VV}}-1} \mathbf{X}_0[p,q]\ R_i^{(\mathrm{HV})}[l,p,q], \tag{4.134}$$

where

$$R_i^{(\mathrm{HV})}[l,p,q] \triangleq R_i^{(\mathrm{VF})}[l,q]\ R_i^{(\mathrm{HF})}[l,p] \tag{4.135}$$

is a phase rotation factor, $R_i^{(\mathrm{VF})}[l,q]$ is expressed by eq. (4.130),

$$R_i^{(\mathrm{HF})}[l,p] \triangleq \left[\exp\left(j\Delta\psi_i^{(\mathrm{HF})}[l]\right)\right]^{p-p_R} \tag{4.136}$$

and

$$\Delta\psi_i^{(\mathrm{HF})}[l] \triangleq 2\pi\hat{F}_{\mathrm{H},i}[l]; \tag{4.137}$$

note that $R_i^{(\mathrm{HV})}[l,p_R,q_R]=1$. Given $\mathbf{X}_{0,\mathrm{OF}}[l]$ (4.134), the sequence of the absolute values of its elements is analysed to verify the presence of an amplitude peak in the $\hat{\alpha}_l$-th delay bin or in a bin close to it; for this reason, the difference $|\hat{\alpha}_{\mathrm{OF}}-\hat{\alpha}_l|$ is compared with the positive threshold T_{OF}, where

$$\hat{\alpha}_{\mathrm{OF}} = \arg\max_{\tilde{a}\in\{0,1,...,N_0\}} |\mathbf{X}_{0,\mathrm{OF}}[\tilde{a}]|. \tag{4.138}$$

[7]As already mentioned above, a portion of the whole virtual array can used to mitigate the impact of the estimation errors affecting the two spatial delays.

If this difference exceeds T_{OF}, the presence of a ghost target is detected; otherwise, the N_0-dimensional vector

$$\mathbf{X}_{m,\mathrm{OF}}[l] \triangleq \sum_{p=0}^{N_{\mathrm{VH}}-1} \sum_{q=0}^{N_{\mathrm{VV}}-1} \mathbf{X}_m[p,q]\ R_i^{(\mathrm{HV})}[l,p,q], \tag{4.139}$$

is computed for $m = 1$ and 2, and the CSDE algorithm is run to estimate, on the basis of the vectors $\mathbf{X}_{0,\mathrm{OF}}[l]$, $\mathbf{X}_{1,\mathrm{OF}}[l]$ and $\mathbf{X}_{2,\mathrm{OF}}[l]$ (see eq. (4.134)), the *final estimates* $\hat{F}_i[l]$ and $\hat{A}_i[l]$ of the parameters $F_i[l]$ and $A_i[l]$ characterizing the i-th target detected in the l-th delay bin. Note that the integer part $\hat{\alpha}_i[l]$ of the delay $\hat{F}_i[l]$ does not necessarily coincide with $\hat{\alpha}_l$ (see eq. (4.105)) but, if it differs, it is certainly close to $\hat{\alpha}_l$. When this happens, if the quantity $\hat{\alpha}_i[l]$ appears in one of the pairs of set $\mathcal{S}_{\mathrm{RPE}}$ (4.86), it is discarded, because the corresponding bin is already being analysed. Otherwise, the new pair

$$\left(\hat{\alpha}_i[l], E_{b,L_b}\right), \tag{4.140}$$

where $E_{b,L_b} = |\hat{A}_i[l]|^2$, is added to the set $\mathcal{S}_b$ and the number of its elements (denoted by L_b) is increased by one. This means that the STDAEC algorithm has to be run on the (new) $\hat{\alpha}_i[l]$-th bin.

STDAEC-S2) *Target cancellation* – The contribution $\mathbf{C}_{\mathrm{X}_0}^{(i)}[l]$, given by the i-th target detected in the l-th delay bin, to the vector $\mathbf{X}^{(i)}[l]$ (4.123) is computed considering

$$\mathbf{C}_{\mathrm{X}_0}^{(i)}[l] = \left[C_{\mathrm{X}_0}^{(i)}[p,q,l]\right] \tag{4.141}$$

given by the i-th (i.e., by the last) target detected in the l-th delay bin on the whole array (see eq. (4.143)) and the expression

$$C_{\mathrm{X}_0}^{(i)}[p,q,l] = \hat{A}_i[l]\ \exp\left\{-j2\pi\left[(p-p_R)\hat{F}_{\mathrm{H},i}[l]\ \ +(q-q_R)\hat{F}_{\mathrm{V},i}[l]\right]\right\}, \tag{4.142}$$

is employed for any VA (i.e., for any p and q). Finally, the cancellation consists in this case in the computation of the new *residual* vector

$$\mathbf{X}^{(i+1)}[l] \triangleq \mathbf{X}^{(i)}[l] - \mathbf{C}_{\mathrm{X}_0}^{(i)}[l]. \tag{4.143}$$

STDAEC-S3) *Residual energy test* – The energy

$$E^{(i+1)}[l] \triangleq \left\|\mathbf{X}^{(i+1)}[l]\right\|^2 = \sum_{p=0}^{N_{\mathrm{VH}}-1} \sum_{q=0}^{N_{\mathrm{VV}}-1} \left|X_l^{(i+1)}[p,q]\right|^2 \tag{4.144}$$

of the residual spectrum vector $\mathbf{X}^{(i+1)}[l]$ (4.143) is compared with the positive threshold T_{STDAEC} (which may exhibit an angular dependence). If this energy is below the threshold, the STDAEC algorithm stops; otherwise, the recursion index i is increased by one and a new iteration is started by going back to **STDAEC-S1**. If $D[l]$ iterations are accomplished by the STDAEC algorithm operating on the $\hat{\alpha}_l$-th delay bin, no more than $D[l]$ distinct targets are identified in that bin ($D[l]$ targets are found if none of them is deemed to be a ghost target). All the target information acquired from the $\hat{\alpha}_l$-th delay bin are collected in the set $\mathcal{T}_l$ (4.87).

T3-S2) *Evaluation of the target spatial coordinates and generation of the overall image* – In this step, the estimates of the range, of the elevation and of the azimuth of the i-th target

detected in the $\hat{\alpha}_l$-th delay bin are computed as

$$\hat{R}_i[l] = \frac{c}{2}\hat{\tau}_i\,[l]\,, \tag{4.145}$$

$$\hat{\phi}_i[l] = \arcsin\left(\frac{\lambda}{2d_{\mathrm{VV}}}\hat{F}_{\mathrm{V},i}[l]\right) \tag{4.146}$$

and

$$\hat{\theta}_i[l] = \arcsin\left(\frac{\lambda}{2d_{\mathrm{VH}}\cos(\hat{\phi}_i[l])}\hat{F}_{\mathrm{H},i}[l]\right), \tag{4.147}$$

respectively; here, $\hat{\tau}_i\,[l] = \hat{F}_i[l]/\Delta f$ (see eq. (2.27)). Finally, these information are fused to generate the overall set

$$\mathcal{I}_t \triangleq \left\{\left(\hat{R}_l, \hat{\theta}_l, \hat{\phi}_l, \left|\hat{A}_l\right|\right); l = 0, 1, \ldots, \hat{L}-1\right\}, \tag{4.148}$$

describing the generated radar image, which, generally speaking, is a cloud of $\hat{L}$ points. The set $\mathcal{I}_t$ results from the union of all the sets $\{\mathcal{I}_t^{(l)}\}$, where

$$\mathcal{I}_t^{(l)} \triangleq \left\{\left(\hat{R}_i[l], \hat{\theta}_i[l], \hat{\phi}_i[l], \left|\hat{A}_i[l]\right|\right); i = 0, 1, \ldots, D[l]-1\right\}, \tag{4.149}$$

with $l = 0, 1, \ldots, L_b - 1$.

The structure of the RASCA#1 deserves a number of comments, that are listed below for the different tasks and the steps they consist of.

T1 – In this task, each of the vector $\{\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)}\}$, defined by eq. (4.100), is computed by executing a N_0 order IFFT. The vector $\mathbf{X}_0^{(v)}$ (4.100) consists of N_0 equally spaced samples of the spectrum of the sequence $\{x_{c,n}^{(v)}\}$ acquired on the v-th VA. The vectors $\mathbf{X}_1^{(v)}$ and $\mathbf{X}_2^{(v)}$, instead, consist of, up to a scale factor, N_0 equally spaced samples of the *first* and the *second derivatives*, respectively, of the same spectrum.

T2-S1 – The exploitation of a subset of the available antennas is motivated by the need of reducing the computational effort required by T2 as much as possible. The adoption of a *deterministic* method for the selection of N_A antennas (with $N_A < N_V$) is not recommended. In fact, when multiple consecutive snapshots are processed to generate independent images, randomly changing the subset of N_A antennas from snapshot to snapshot allows the considered radar system to benefit from *antenna diversity*.

T2-S2 – The STDREC algorithm deserves the following comments:

a) The availability of *accurate* estimates of the normalized delay $F_i^{(v_k)}$ and of the complex amplitude $A_i^{(v_k)}$ plays an important role in this step, since these parameters are exploited in the serial cancellation procedure based on eq. (4.115). In particular, ignoring the delay residual $\delta_i^{(v_k)}$ of the normalized delay $F_i^{(v_k)}$ (4.104) in this procedure (i.e., assuming that $\hat{F}_i^{(v_k)} = \hat{\alpha}_i^{(v_k)}$; see eq. (4.105)) may result in a significant *error accumulation* in that procedure.

b) A threshold on the maximum computational effort required by the STDREC algorithm can be set by requiring that the recursion index i never exceeds a fixed threshold; this is equivalent to limit the overall number of targets that can be detected on each VA.

c) The STDREC algorithm generates N_A different data sets; the k-th data set consists of the triplets $\{(\hat{\alpha}_i^{(v_k)}, \hat{F}_i^{(v_k)}, \hat{A}_i^{(v_k)}); i = 0, 1, \ldots, L_k - 1\}$ ($\{(\hat{\alpha}_i^{(v_k)}, \hat{F}_i^{(v_k)}, \hat{A}_i^{(v_k)}); i = 0, 1, \ldots, L_k - 1\}$), characterizing the L_k targets detected on the v_k-th antenna (with $k = 0, 1, ..., N_A - 1$). Note that the overall number of targets may change from antenna to antenna, especially in the presence of extended targets; this is due to the fact that the signals acquired on different VAs can exhibit significant differences in their spectral content.

d) The following important interpretation of the processing accomplished by the STDREC algorithm on the v_k-th VA can be given. The vector $\mathbf{X}_0^{(v_k)}$ can be seen as a collection of noisy spectral information referring to N_0 distinct *delay bins* (i.e., to N_0 distinct *range bins*) and is usually *dense* in the presence of multiple extended targets. The STDREC allows to extract a *discrete delay* (i.e., *range*) *profile* from the vector $\mathbf{X}_0^{(v_k)}$. In various real world scenarios, this profile turns out to be *sparse*, even in the presence of a dense vector $\mathbf{X}_0^{(v_k)}$; this is beneficial, since allows to concentrate the RPE computational effort on a set of *specific ranges* (i.e., delay bins). The *range profile* referring to the v_k-th VA is described by the set of L_k pairs $\mathcal{S}_{v_k} = \{(\hat{\alpha}_i^{(v_k)}, \hat{A}_i^{(v_k)}); i = 0, 1, \ldots, L_k - 1\}$, with $k = 0, 1, ..., N_A - 1$; the parameter $\hat{\alpha}_i^{(v_k)}$ identifies the delay bin associated with the i-th target (or targets) detected on this VA, whereas the absolute value of $\hat{A}_i^{(v_k)}$ represents an estimate of the *strength* of the echo associated with it.

e) The STDREC algorithm can be used for detecting multiple targets and accurately estimating their range in a monostatic radar, i.e., equipped with a single TX antenna and a single RX antenna.

f) The STDREC algorithm can be easily extended in a way that *multiple targets* are detected and estimated in parallel in each of its iterations. If we focus on its i-iteration and the v_k-th VA, this result is achieved by running multiple (say, $m_i^{(v_k)}$) instances of the CSDEC algorithm in parallel. Each of these instances is initialised with the delay associated with the absolute maximum or a relative maximum detected in the sequence of the absolute values of the elements of the vector $\mathbf{X}_0^{(v_k)}[i]$ (see eq. (4.100)). A constraint is set on the minimum spacing between the $m_i^{(v_k)}$ detected delays in order to minimize the interference between the instances running in parallel. Moreover, after identifying the absolute maximum in the above mentioned sequence, a threshold, proportional to such a maximum, is set on the minimum value of the acceptable relative maximum/maxima, so that unrelevant delays are discarded. It is also worth stressing that, if a *cluster* of $m_i^{(v_k)}$ distinct delays is estimated, each of the components of the triplet $(\mathbf{C}_{X_0}^{(v_k)}[i], \mathbf{C}_{X_1}^{(v_k)}[i], \mathbf{C}_{X_2}^{(v_k)}[i])$ appearing in the RHS of eq. (4.115) consists of the sum of $m_i^{(v_k)}$ terms, each associated with one of these delays.

g) As shown in Par. 4.1.1, the estimates generated by the STDREC algorithm are biased if the values selected for the parameters N_{CSDE} and N_{RES} are not large enough. In principle, this bias can be arbitrarily reduced by increasing the values of these parameters. However, it is found out that a computationally efficient alternative to this approach is represented by running an additional step (i.e., STDREC-S4) once that the STDREC algorithm has been executed. In this final step, the Alg-YA is carried out after initializing it with the estimates of the normalized frequencies and the associated complex amplitudes generated by the STDREC. The hybrid

technique resulting from interconnecting the STDREC algorithm with the above mentioned algorithm is dubbed *hybrid* STDREC (HSTDREC) in the following; note that this algorithm represents an instance of the hybrid CSDE proposed in ref. 4.1.1.

T3-S1 – This step is the most complicated of the whole algorithm and deserves the following comments:

a) In principle, the horizontal and vertical spatial delays (see eqs. (4.89) and (4.90)) of multiple targets contributing to the l-th delay bin can be detected by first computing a 2D IDFT of the matrix $\mathbf{X}[l]$ (4.122) and, then, by looking for the local maxima over the absolute values of the elements of the resulting 2D matrix; note that the matrix $\mathbf{X}[l]$ can be also zero-padded before computing its 2D IFFT to improve the resulting spectral resolution. This procedure may require a significant computational effort and its accuracy is affected by the spectral leakage due to any potential strong target. In the STDAEC algorithm, instead, 2D processing is avoided by alternating vertical and horizontal 1D IDFTs; consequently, relevant spatial delays are estimated by searching for the peaks of 1D amplitude spectra (i.e., $\mathbf{S}^{(i)}_{\mathrm{VULA},0}[l]$ and $\mathbf{X}^{(\mathrm{VF})}_{i}[l]$) (in other words, an AM approach is adopted). This allows to mitigate computational complexity and to detect weak targets hidden by close strong targets through successive cancellations.

b) Each of the four vectors $\{\mathbf{s}^{(i)}_{\mathrm{VULA},k}[l];\ k = 0, 1, 2, 3\}$ defined in **STDAE-S1** can be computed by executing a $\bar{N}_0$-th order IFFT (see eq. (4.128)). Note that, on the one hand, the vector $\mathbf{s}^{(i)}_{\mathrm{VULA},0}[l]$ collects $\bar{N}_0$ equally spaced samples of the spectrum of the sequence $\{X^{(i)}_{\mathrm{VULA},p};\ p = 0, 1, ..., N_{\mathrm{VULA}} - 1\}$ (see eq. (4.126)). On the other hand, the k-th vector $\mathbf{s}^{(i)}_{\mathrm{VULA},k}[l]$ (with $k = 1, 2$ and 3) collects, up to a scale factor, N_0 equally spaced samples of the k-th order derivative of the same spectrum.

c) The processing accomplished in **STDAE-S4** is very similar to that carried out in **STDAE-S1** and **STDAE-S2**. In fact, the only difference is represented by the fact that the N_{VULA}-dimensional vector $\mathbf{S}^{(i)}_{\mathrm{VULA},0}[l]$ (4.124) is replaced by the N_{HULA}-dimensional vector $\mathbf{X}^{(\mathrm{VF})}_{i}[l]$ (4.132) generated in **STDAE-S3**. Therefore, in this case, the CSDE is exploited to estimate the horizontal delay $F_{\mathrm{H},i}[l]$ and, again, the complex amplitude $A_i[l]$ associated with the i-th target.

d) As already suggested for the STDREC algorithm, the STDAEC algorithm can be employed to detect and estimate multiple angles in parallel; this requires running multiple instances of the CSDE algorithm in parallel.

The scheme of RASCA #1 for three-dimensional imaging is shown in Algorithm 8.

Algorithm 8: Range & Angle Serial Cancellation Algorithm (RASCA) #1

T1 – IFFT: Compute $\{\mathbf{x}_{0,\mathrm{ZP}}^{(v)}, \mathbf{x}_{1,\mathrm{ZP}}^{(v)}, \mathbf{x}_{2,\mathrm{ZP}}^{(v)}\}$ and $\{\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)}\}$ (4.96)-(4.100).
T2 – RPE: S1) Extract N_A VAs from the available VAs; build the set $\bar{\mathcal{S}}_{\mathrm{FFT}}$ (4.83).
for $k = 0$ *to* $N_A - 1$ **do**
  S2) Set $\mathbf{X}_m^{(v_k)}[0] \triangleq \mathbf{X}_m^{(v_k)}$ for $m = 0, 1, 2$ (4.102), $i = 0$; compute $E_0^{(v_k)}$ (4.116).
  while $E_i^{(v_k)} > T_{\mathrm{STDREC}}$ **do**
    STDREC-S1) Compute $(\hat{A}_i^{(v_k)}, \hat{F}_i^{(v_k)})$ from $(\mathbf{X}_0^{(v_k)}[i], \mathbf{X}_1^{(v_k)}[i], \mathbf{X}_2^{(v_k)}[i])$.
    STDREC-S2) Compute $(\mathbf{C}_{X_0}^{(v_k)}[i], \mathbf{C}_{X_1}^{(v_k)}[i], \mathbf{C}_{X_2}^{(v_k)}[i])$ (4.107)–(4.109) and $(\mathbf{X}_0^{(v_k)}[i+1], \mathbf{X}_1^{(v_k)}[i+1], \mathbf{X}_2^{(v_k)}[i+1])$ (4.115).
    STDREC-S3) Compute the residual energy $E_{i+1}^{(v_k)}$ according to eq. (4.116).
  end
end
S3) Build the set $\mathcal{S}_{\mathrm{RPE}}$ (4.86) (see eqs. (4.117) and (4.120).
T3 – SPE: S1) Set $i = 0$ and $\mathbf{X}^{(0)}[l]$ (4.121); then, compute $E^{(0)}[l]$ (4.144).
Parallel For $l = 0$ *to* $l = L_b - 1$ **do**
  while $E^{(i)}[l] > T_{\mathrm{STDAEC}}$ **do**
    STDAEC-S1) Compute $(\hat{A}_{\mathrm{V},i}[l], \hat{F}_{\mathrm{V},i}[l])$ (4.128), $R_i^{(\mathrm{VF})}[l, q]$ (4.130) and $\mathbf{X}_i^{(\mathrm{VF})}[l]$ (4.132). Then, compute $(\hat{A}_{\mathrm{H},i}[l], \hat{F}_{\mathrm{H},i}[l])$, $R_i^{(\mathrm{HF})}[l, p]$ (4.136) and finally $\{\mathbf{X}_{m,\mathrm{OF}}[l]; m = 0, 1, 2\}$ (4.134), evaluating $(\hat{A}_i[l], \hat{F}_i[l])$.
    STDAEC-S2) Compute $\mathbf{C}_{X_0}^{(i)}[l]$ (4.141)–(4.142) and $\mathbf{X}^{(i+1)}[l]$ (4.143).
    STDAEC-S3) Compute $E^{(i+1)}[l]$ (4.144).
  end
  S2) Compute $\hat{R}_i[l]$, $\hat{\phi}_i[l]$, $\hat{\theta}_i[l]$ according to eq. (4.145)–(4.147).
end

4.2.2 Three-dimensional range & angle serial cancellation algorithm #2

The inner structure of the RASCA#2 is described by the block diagram shown in the below Fig. 4.6.

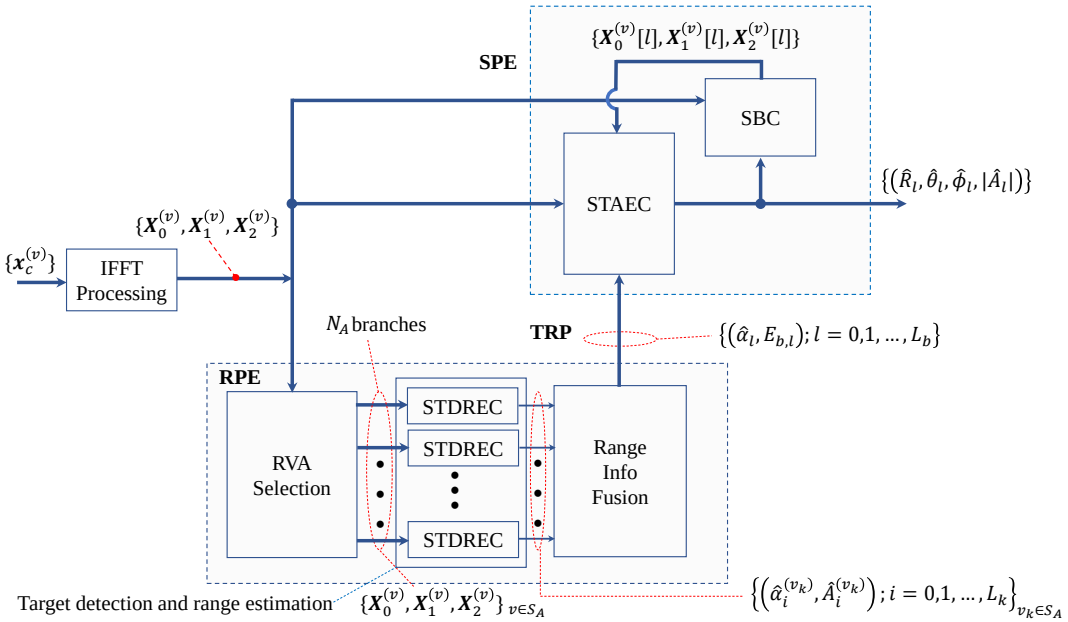


FIGURE 4.6: Block diagram describing the overall structure of the RASCA#2.

The processing accomplished by this algorithm can be divided into three tasks, each associated with one of the blocks appearing in that figure (the i-th task is denoted by **Ti** in the following). In all these tasks, a single snapshot is processed. The first two tasks (namely, those concerning the IFFT processing and the RPE) coincide with those of RASCA#1, since the only differences about the last algorithm and RASCA#2 concern their SPEs. For this reason, in the remaining part of this paragraph, a schematic description of the third task only is provided. In this last task of this technique, spatial estimation is accomplished *serially* i.e., *on a bin-by-bin basis*; moreover, the delay bins selected by the RPE are analysed in an ordered way and, more precisely, according to their decreasing energies; in the following it is assumed, for simplicity, that the bins appearing in the set $\mathcal{S}_{\text{RPE}}$ (4.86) have been ordered in a way that

$$E_{b,l} \geq E_{b,l-1} \tag{4.150}$$

for $l = 0, 1, \ldots, L_b - 1$. The SPE is initialised by setting

$$\mathbf{X}_m^{(v)}[0] \triangleq \mathbf{X}_m^{(v)}, \tag{4.151}$$

with $m = 0,\ 1,\ 2$ and the bin index l to 0. Then, L_b consecutive iterations are run. In the l-th iteration, the STDAEC algorithm is run to a) estimate, on the basis of the $N_{\text{VH}} \times N_{\text{VV}}$ matrix

$$\mathbf{X}^{(i)}[l] \triangleq \left[X_l^{(i)}[p,q]\right], \tag{4.152}$$

the angular parameters of the $\bar{D}[l]$ distinct targets contributing to that bin. A refined estimate of the range for all the targets detected in the considered delay bin is computed on the basis of set of triplets $\{(\mathbf{X}_0^{(v)}[l], \mathbf{X}_1^{(v)}[l], \mathbf{X}_3^{(v)}[l])\}$. Then, the estimates of the range and of the angular parameters of the $\bar{D}[l]$ detected targets are: a) made available at the

output of the SPE; b) processed by another algorithm, called *single bin cancellation* (SBC). The SBC algorithm estimates the contributions $\{(\mathbf{C}_{X_0}^{(v)}[l], \mathbf{C}_{X_1}^{(v)}[l], \mathbf{C}_{X_2}^{(v)}[l])\}$ of these targets (i.e., of the l-th delay bin) to the set of triads $\{(\mathbf{X}_0^{(v)}[l], \mathbf{X}_1^{(v)}[l], \mathbf{X}_3^{(v)}[l])\}$ (see eq. (4.100)) and cancels it. In particular, the terms $(\mathbf{C}_{X_0}^{(v)}[l], \mathbf{C}_{X_1}^{(v)}[l], \mathbf{C}_{X_2}^{(v)}[l])$ employed in the single bin cancellation procedure of RASCA#2 (see eq. (4.156)) are evaluated as

$$\mathbf{C}_{X_0}^{(v)}[l] \triangleq \sum_{i=0}^{D[l]-1} \hat{A}_i[l] \ \bar{W}_0^{(v)}[i,l] \tag{4.153}$$

$$\mathbf{C}_{X_1}^{(v)}[l] \triangleq \sum_{i=0}^{D[l]-1} \hat{A}_i[l] \ \bar{W}_1^{(v)}[i,l] \tag{4.154}$$

and

$$\mathbf{C}_{X_2}^{(v)}[l] \triangleq \sum_{i=0}^{D[l]-1} \hat{A}_i[l] \ \bar{W}_2^{(v)}[i,l] \tag{4.155}$$

respectively, with $v = 0, 1, ..., N_V - 1$. Here, $D[l]$ represents the overall number of distinct targets detected in the $\hat{\alpha}_l$-th time bin and $\hat{A}_i[l]$ denotes the complex amplitude associated with the i-th target detected in the $\hat{\alpha}_l$-th time bin. Moreover, the complex elements $\bar{W}_0^{(v)}[i,l]$, $\bar{W}_1^{(v)}[i,l]$ and $\bar{W}_2^{(v)}[i,l]$ are based on (4.113) (with $k = 0, 1$ and 2), where, however, the normalized delay $\hat{F}_i[l]$ is used in place of $\hat{F}_i^{(v_k)}$ (see eq. (4.112)) in the computation of the quantity $\bar{W}_i^{(v)}$ (see eq. (4.111)). Cancellation consists in the computation of the new residual triad

$$\mathbf{X}_m^{(v)}[l+1] \triangleq \mathbf{X}_m^{(v)}[l] - \mathbf{C}_{X_m}^{(v)}[l], \tag{4.156}$$

with $m = 0, 1, 2$ and $v = 0, 1, ..., N_V - 1$. Finally, if $l < L_b - 1$, l is increased by one and a new iteration is started; otherwise, if $l = L_b - 1$, processing is over.

4.2.3 Bi-dimensional range & angle serial cancellation algorithms

The two algorithms described in the previous two paragraphs can be easily adapted to the case in which the considered colocated MIMO radar system is equipped with an HULA and, consequently, can be exploited for 2D imaging only. In this case, RASCA#1 and RASCA#2 can be employed after making minor changes in their structures. Such changes concern only the SPE and can be summarized as follows:

1. The first three steps of the STDAE in **T3-S1** are not performed; therefore, the fourth step of that algorithm is the first one to be executed. Moreover, the matrix $\mathbf{X}^{(i)}[l]$ (4.123) is replaced by the N_{VH}-dimensional vector

$$\mathbf{X}^{(i)}[l] \triangleq \left[X_l^{(i)}[p]\right], \tag{4.157}$$

   collecting the spectral information available on the whole virtual receive array and referring to the l-th echo only.

2. The delay $\hat{F}_{V,i}[l]$ (4.90) is not available and, therefore, it is not included in the set $\mathcal{T}_l$ (4.87) (the elevation angle $\hat{\phi}_i[l]$ (4.146) is not estimated). As consequence, the

amplitude $A_i[p,l]$ observed on the p-th VA can be expressed in this case as:

$$A_i[p,l] = A_i[l] \exp\left\{-j\frac{4\pi}{\lambda}\left[d_{\mathrm{VH}}\left(p-p_R\right)\sin(\theta_i[l])\right]\right\}; \tag{4.158}$$

where the dependence on the vertical element q is ignored respect eq. (4.88). In the bi-dimensional version of RASCA #1 and RASCA #2 the whole overall folding is the horizontal folding:

$$\mathbf{X}_{0,\mathrm{OF}}[l] \triangleq \sum_{p=0}^{N_{\mathrm{VH}}-1} \mathbf{X}_0[p]\ R_i^{(\mathrm{HV})}[l,p], \tag{4.159}$$

where

$$R_i^{(\mathrm{HV})}[l,p] \triangleq R_i^{(\mathrm{HF})}[l,p] \tag{4.160}$$

is a phase rotation factor and

$$R_i^{(\mathrm{HF})}[l,p] \triangleq \left[\exp\left(j\Delta\psi_i^{(\mathrm{HF})}[l]\right)\right]^{p-p_R} \tag{4.161}$$

with

$$\Delta\psi_i^{(\mathrm{HF})}[l] \triangleq 2\pi\hat{F}_{\mathrm{H},i}[l]; \tag{4.162}$$

and

$$F_{\mathrm{H},i}[l] \triangleq \frac{2d_{\mathrm{VH}}}{\lambda}\sin(\theta_i[l]) \tag{4.163}$$

Similar changes are made in the RASCA#2, since this is makes use of the same techniques adopted in the RASCA#1.

4.2.4 Unequal response of virtual antennas

The derivation of the RASCA#1 and RASCA#2 for SFCW radar systems relies on the assumption that the sample sequence made available by the v-th VA is expressed by eq. (2.25). The adopted signal models holds if the amplitudes of the L overlapped oscillations contributing to the useful component of the received signal do not change from antenna to antenna. The experiments accomplished on commercial colocated SFCW MIMO radars have evidenced that: a) these amplitudes are not constant across the whole virtual array; b) their differences are influenced by the azimuth and the elevation of each target. I believe that all this is due to the different behavior of the multiple receive chains employed in each MIMO device and to the mismatches in the receive antenna patterns. This problem, that affects the quality of collected data can be mitigated by enriching the physical array with a set of surrounding *passive antennas*; in this case, the array is artificially extended with new antennas along all its sides, so that the behavior of all its active antennas becomes more uniform.

In principle, this phenomenon can be accounted for, by including its effects in the received signal model. For instance, eq. (2.25) can be generalised as

$$x_{c,n}^{(v)} = \sum_{l=0}^{L-1} \alpha_v\left(\theta_l,\phi_l\right)\ a_l\ \exp\left(-j\left(2\pi n F_l^{(v)} + \psi_l^{(v)}\right)\right) + w_{c,n}^{(v)}, \tag{4.164}$$

where $\alpha_v(\theta_l,\phi_l)$ represents an attenuation factor depending on the angular coordinates of the l-th target and v is the VA index. Neglecting the presence of the factor $\alpha_v\left(\theta_l,\phi_l\right)$ in the development of the algorithms has the following implications:

a) An error is introduced by the STDAEC algorithm in its cancellation procedure (see **STDAEC-S2** in paragraph 4.2.1; eq. (4.143)) of the RASCA#1. Note, in particular, that the estimate $\hat{A}_i[l]$ of the complex amplitude characterizing to the i-th target detected in the $\hat{\alpha}_l$-th bin is computed *after* the overall spatial folding (i.e., after **STDAE-S5**); consequently, its absolute value represents a sort of spatial average computed over all the involved VAs. Moreover, only the phase variations of this complex gain are accounted for in the computation of the contribution $\mathbf{C}_{X_0}^{(i)}[l]$ of this target to the matrix $\mathbf{X}^{(i)}[l]$ (see eqs. (4.141)–(4.142)).

b) An error in the cancellation procedure is introduced by the SBC algorithm in **T3** of the RASCA#2 (see paragraph 4.2.2). In fact, in the computation of the vector $\mathbf{C}_{X_m}^{(v)}[l]$ (with $m = 0$, 1 and 2) appearing in eq. (4.156), the variations in the absolute value of the complex gain $A_i[l]$ are not accounted for (see eqs. (4.153)–(4.155)).

In principle, if the functions $\{\alpha_v(\theta_l, \phi_l)\}$ were known for all the VAs, their effect could be compensated for *after* evaluating the estimates $(\hat{\theta}_i, \hat{\phi}_i)$ of the angular coordinates of the i-th target; in fact, this result could be achieved by replacing the estimate $\hat{A}_i[l]$ of the complex gain $A_i[l]$ with

$$\hat{A}_i[v, l] \triangleq \hat{A}_i[l]\ \alpha_v(\hat{\theta}_i, \hat{\phi}_i). \tag{4.165}$$

in the evaluation of the term $\mathbf{C}_{X_0}^{(i)}[l]$ appearing in eqs. (4.141)–(4.142) or eqs. (4.153). Estimating the function $\alpha_v(\theta, \phi)$, however, is a time consuming task, since it requires a proper measurement setup and a calibration procedure in an anechoic chamber, knowing the scattering properties of the radiated target. This problem can be circumvented by: a) exploiting *deep learning techniques* [86] in the SPE; b) adopting a *data-driven approach* [87].

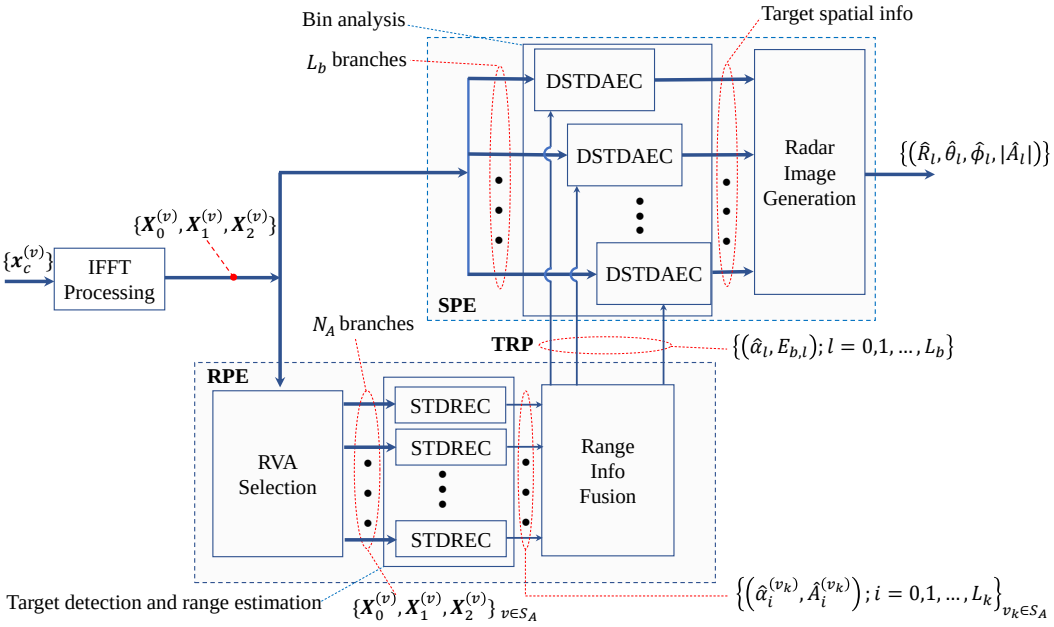


FIGURE 4.7: Block diagram representing the RASCA #1 method; a compensation technique based on deep-learning methods (dubbed as DSTDAEC) is employed in the SPE.

The proposed approach is motivated by the fact that:

a) Deep learning techniques can be employed to approximate complicated functions, that do not lend themselves to a simple parametric representation and without requiring particular expertise in data pre-processing.

b) A data-driven approach allows to train different models on the basis of data collected in a real scenario or synthetically generated, without prior knowledge about the parametric representation of the considered problem.

The training procedure is important, because, through it, the involved network acquires the ability to generate a correct prediction on the basis of never seen data available at its input. In practice, the adoption of this approach requires modifying the STDAEC technique employed in the RASCA#1 (see Fig. 4.3) and, in particular, embedding a deep neural network in it. This network is employed to estimate the distorted amplitudes of all the targets detected in the l-th delay bin (with $l = 0, 1, \ldots, L_b - 1$), so that accurate cancellation becomes possible. If the proposed method is adopted in RASCA #1, the block diagram illustrated in Fig. 4.3 is replaced by that shown in Fig. 4.7. As it can be easily inferred from comparing the first figure with the second one, the STDAEC block is replaced by a new block, that implements a new algorithm, called *deep* STDAEC (DSTDAEC). The initialization procedure is the same of the STDAEC and the DSTDAEC algorithm consists of three steps (its r-th step is denoted by **DSTDAEC-Sr** in the following) and the processing accomplished within each of them is sketched below for the l-th delay bin (with $l = 0, 1, \ldots, L_b - 1$).

DSTDAEC-S1) *Detection of a new target and estimation of its angular parameters* - This step is identical to the first step of the STDAEC technique. Therefore, the $N_{VH} \times N_{VV}$ matrix $\mathbf{X}^{(i)}[l]$ is processed to detect the *strongest* target (i.e., the i-th target) contributing to the l-th delay bin and to estimate its parameters $(\theta_i[l], \phi_i[l], R_i[l], A_i[l])$.

DSTDAEC-S2) *Estimation of the distorted amplitudes* - In this step, a *deep neural network* is employed to *predict* the absolute value[8] of $\tilde{A}_i[v, l]$ (i.e., to predict a new estimate $\tilde{A}_i[v, l]$ of $|\hat{A}_i[v, l]|$; on the basis of the absolute value of the amplitude $\hat{A}_i[l]$ (estimated in **STDAE-S5**). This network is designed to solve a *regression* problem through a *supervised learning* approach. Network training is based on the dataset

$$\mathcal{D} = \left\{ \left(\hat{\mathbf{\Lambda}}_q, \tilde{\mathbf{\Lambda}}_q \right) ; \quad q = 1, 2, ..., N_t \right\}, \tag{4.166}$$

collecting N_t distinct points; here,

$$\hat{\mathbf{\Lambda}}_q = \left[\left| \hat{A}_i[1, l] \right|, \left| \hat{A}_i[2, l] \right|, ..., \left| \hat{A}_i[N_V, l] \right| \right]^T \tag{4.167}$$

and

$$\tilde{\mathbf{\Lambda}}_q = \left[\left| \tilde{A}_i[1, l] \right|, \left| \tilde{A}_i[2, l] \right|, ..., \left| \tilde{A}_i[N_V, l] \right| \right]^T \tag{4.168}$$

are N_V-dimensional vectors representing the *input* of the network and its *response*, respectively, in the q-th trial. In network training, the following assumptions are made: a) the vectors $\tilde{\mathbf{\Lambda}}_q$ and $\hat{\mathbf{\Lambda}}_q$ are known for any q; b) in the q-trial, the radar system observes a *single point target* and the target is detected in the frequency bin $\hat{b}_l$. It is important to note that:

[8] The network is used for *amplitude prediction*, since we assume that the antenna array is introduces an amplitude distortion only.

a) The evaluation of the elements $\{\hat{A}_i[v,l] = \hat{A}_i[p,q,l]\}$ in vector $\hat{\mathbf{\Lambda}}_q$ (4.167) is based on the estimate $\hat{A}_i[l]$ computed in **STDAE-S5**; for this reason we have that

$$\left|\hat{A}_i[1,l]\right| = \left|\hat{A}_i[2,l]\right| = ... = \left|\hat{A}_i[N_V,l]\right| = \left|\hat{A}_i[l]\right|. \tag{4.169}$$

b) If the vector $\tilde{\mathbf{\Lambda}}_q$ (4.168) is known, an estimate $|\check{A}_i[l]|$ of the absolute value of $A_i[l]$ can be also computed as

$$|\check{A}_i[l]| = N_V^{-1} \sum_{v=1}^{N_V} |\tilde{A}_i[v,l]|. \tag{4.170}$$

The architecture proposed for the deep neural network has been inspired by that of a reasonably simple *feed-forward* neural networks [86, Chapter 5]. This architecture is characterized by input and output layers of the same size (N_V) and *K hidden layers*. Each layer is composed by neurons and each neuron provides a non-linear combination of its inputs to the next layer. The amount of layers and neurons in the network are hyperparameters that they can be freely chosen, so that the *weights* and *bias* in each neuron can be optimized. Note, however, that the size of each layer decreases from the input layer to the output layer. During training, the network has the ability to tune its parameters in order to learn a compressed representation of the input data, so generating an estimate of the amplitude distortion $\alpha_v(\theta,\phi)$.

DSTDAEC-S3) *Target cancellation* - In this step, the contribution $\mathbf{C}_X^{(i)}[l]$, given by the i-th target detected in the l-th delay bin, to the matrix $\mathbf{X}^{(i)}[l]$ is computed. Eqs. (4.141)-(4.142) in **STDAEC-S2** can be still exploited; however, eq. (4.142) is modified as

$$\mathbf{C}_X^{(i)}[l] = |\tilde{A}_i[p,q,l]| \exp^{\angle \hat{A}_i[l]} \exp\left\{-j2\pi\left[(p-p_R)\hat{F}_{H,i}[l] + (q-q_R)\hat{F}_{V,i}[l]\right]\right\} \tag{4.171}$$

where $|\tilde{A}_i[p,q,l]| = |\tilde{A}_i[v,l]|$ and $|\tilde{A}_i[v,l]|$ is predicted by the network. Then, cancellation is accomplished by computing the new *residual* vector $\mathbf{X}^{(i+1)}[l]$ on the basis of eq. (4.143).

DSTDAEC-S4) *Residual energy test* - The energy $E^{(i+1)}[l]$ of the residual spectrum vector $\mathbf{X}^{(i+1)}[l]$ computed on the basis of eq. (4.144) is compared with the positive threshold T_{STDAEC}. If this energy is below the threshold, the DSTDAEC algorithm stops; otherwise, the recursion index i is increased by one and a new iteration is started by going back to **DSTDAEC-S1**.

All the target information referring to the $\hat{b}_l$-th delay bin are collected in the set $\mathcal{T}_l$; note that the quantity $|\hat{A}_i[l]|$ appearing in the RHS of the definition (4.149) is replaced by $|\check{A}_i[l]|$ estimated by the network.

The method for amplitude compensation illustrated for the RASCA #1 can be easily adapted to RASCA #2 (see Fig. 4.6). In this case, the STDAEC technique is replaced by the DSTDAEC technique. Moreover, the SBC procedure is still based on eqs. (4.156); however, in the computation of the vectors $\mathbf{C}_{X_m}^{(v)}[l]$, on the basis of eqs. (4.153)-(4.155), the dependence of the absolute value of the quantity $\hat{A}_i^{(v)} = \check{A}_i[v,l]$ on the antenna index v has to be taken into account.

4.3 Other target detection and estimation techniques

The detection and estimation algorithms described above have been compared, in terms of accuracy and complexity, with two different types of algorithms that, similarly as the RASCAs, are able to generate radar images in the form of point clouds. The algorithms of the first type are called FFT-*based algorithms* (FFT-BAs), since they rely on multidimensional IFFT processing for the evaluation of all the spatial coordinates of targets (i.e., their range and DOA); such algorithms have been inspired by the FFT-based algorithm proposed by Texas Instrument in [88]. The algorithms of the second type, instead, are called MUSIC-*based algorithms* (MUSIC-BAs); these make use of the same method as the first type for range estimation, but the MUSIC algorithm for DOA estimation [32].

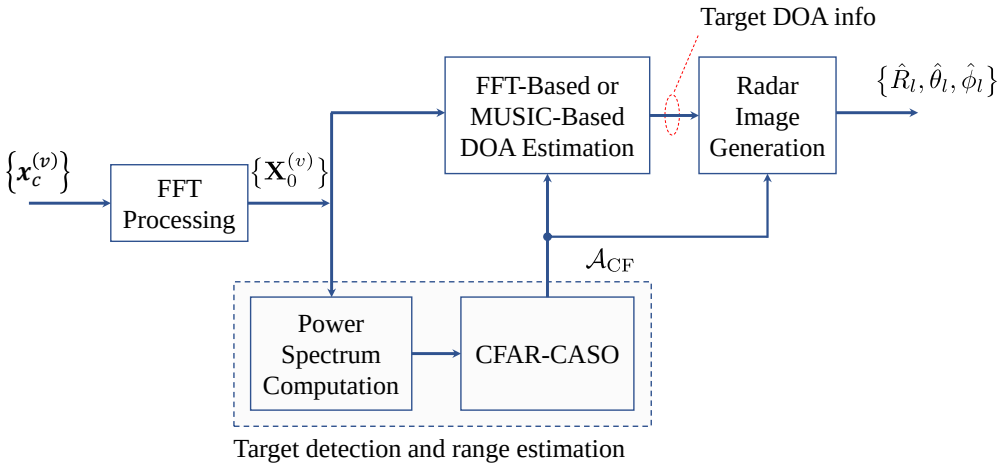


FIGURE 4.8: Block diagram describing the overall structure of the FFT-BAs and the MUSIC-BAs.

In the remaining part of this section, a brief description is provided for both types referring to a SFCW radar system. The inner structure of both types of algorithms is described by the block diagram shown in Fig. 4.8. The processing accomplished by the blocks this diagram consists of, can be summarized as follows. Each vector of the set $\{\mathbf{x}_c^{(v)}\}$, collecting N_{V} vectors (see (2.25)), undergoes, after ZP, a N_0 order inverse FFT; this produces a set of N_0-dimensional vectors $\{\mathbf{X}_0^{(v)}\}$ (see (4.96), (4.100) and (4.101)). Based on this set of vectors, the N_0-dimensional power spectrum

$$\mathbf{P}_0 = [P_{0,0}, P_{0,1}, ..., P_{0,N_0-1}]^T \tag{4.172}$$

is computed; here,

$$P_{0,i} \triangleq \frac{1}{N_{\mathrm{VR}}} \sum_{v=0}^{N_{\mathrm{VR}}-1} \left(X_{0,i}^{(v)}\right)^2 \tag{4.173}$$

with $i = 0, 1, ..., N_0 - 1$. The vector $\mathbf{P}_0$ (4.172) feeds the *cell-averaging smallest of - constant false alarm rate* (CFAR-CASO) algorithm developed in [89]. Based on this algorithm, a target is detected in the i-th delay bin if

$$P_{0,i} > T_{\mathrm{CFAR}}, \tag{4.174}$$

where $i \in \{i_m, i_m + 1, ..., i_M\}$. Here,

$$T_{\mathrm{CFAR}} = K_0 \min\left(\bar{P}_l\,,\, \bar{P}_u\right) \tag{4.175}$$

represents a decision threshold, K_0 is a real parameter whose value is selected on the basis of the required *false alarm rate*, and

$$\bar{P}_l = \frac{1}{C_s} \sum_{k=i-(G_s+C_s)}^{i-(G_s+1)} P_{0,k} \tag{4.176}$$

and

$$\bar{P}_u = \frac{1}{C_s} \sum_{k=i+G_s+1}^{i+G_s+C_s} P_{0,k} \tag{4.177}$$

represent the average of the power spectrum computed over C_s adjacent bins positioned on the left and on the right, respectively, with respect to the i-th delay bin. Moreover, G_s and C_s are two integer parameters defining the size and the position (with respect to the i-th bin), respectively, of the set of frequency bins involved in the computation of $\bar{P}_l$ (4.176) and $\bar{P}_u$ (4.177), whereas i_m and i_M are two non negative integers such that $i_m \geq (G_s + C_s)$ and $i_M \leq N_0 - 1 - (G_s + C_s)$. In our work, the inequality

$$P_{0,i} > P_{l,u} \tag{4.178}$$

is also required to be satisfied together with the condition (4.174), where $P_{l,u}$ represents the largest element of the set $\{P_{0,i+l};\, l = -(G_s + C_s), -(G_s + C_s) + 1, -G_s - 1, G_s + 1, G_s + C_s\}$. This allows us to reduce the overall number of detected targets, so reducing the density of the generated point cloud. The CFAR-CASO algorithm generates the vector

$$\mathcal{A}_{\mathrm{CF}} = \left[\hat{\alpha}_0, \hat{\alpha}_1, ..., \hat{\alpha}_{L_b-1}\right]^T \tag{4.179}$$

where $\hat{\alpha}_l$ represents the index of the delay bin in which the l-th target has been detected (with $l = 0, 1, \ldots, L_b - 1$) and L_b is the overall number of detected targets. This vector is processed for DOA estimation. The two options (associated with the above mentioned types of algorithms) are considered for this task and are described in the remaining part of this paragraph.

IFFT-*based* DOA *estimation* – Let us focus first on the case in which a virtual HULA, consisting of N_{VH} virtual elements, is employed for resolving the targets associated with a given delay bin and estimating their azimuth. In this case, azimuth estimation consists of the following two steps:

1) The N_{VH}-dimensional column vector (see (4.157))

$$\mathbf{X}\left[l\right] \triangleq \left[X_{0,\hat{\alpha}_l}^{(0)}, X_{0,\hat{\alpha}_l}^{(1)}, ..., X_{0,\hat{\alpha}_l}^{(N_{\mathrm{VH}}-1)}\right]^T, \tag{4.180}$$

collecting the spectral information available on the whole array and referring to the $\hat{\alpha}_l$-th delay bin (with $l = 0, 1, ..., L_b - 1$) is applied to an $\bar{N}_0$ order IFFT algorithm; let $\mathbf{s}[l] = [s_0[l], s_1[l], ..., s_{\bar{N}_0-1}[l]]^T$ denote the $\bar{N}_0$-dimensional IFFT output.

2) The dominant peaks[9] in the sequence $\{|s_k[l]|;\, k = 0, 1, ..., \bar{N}_0 - 1\}$ are identified; each peak corresponds to a distinct target. If $k_i[l]$ denotes the index of i-th peak (with $i = 0, 1, ..., L_h[l] - 1$, where $L_h[l]$ is the overall number of targets detected in the considered frequency bin), the estimate of the azimuth of the i-th target is evaluated as

$$\hat{\theta}_i\left[l\right] = \arcsin\left(h_{\bar{N}_0}\left[k_i\left[l\right]\right]\right) \tag{4.181}$$

[9]It is important to distinguish peaks associated with different targets from side-lobes; in our simulations, a candidate peak is classified as a side-lobe (and, consequently, ignored) if its amplitude differs by more than 1 dB from that of a close dominant peak, as suggested in [88].

where

$$h_{\bar{N}_0}[x] \triangleq 2\left(x - \bar{N}_0/2\right)/\bar{N}_0 \,. \tag{4.182}$$

Let us now consider the case in which the URA represented in Fig. 4.2 is employed for resolving the targets associated with each delay bin, and estimating their azimuth and elevation. The algorithm employed in this case involves the $N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ matrix $\mathbf{X}[l] \triangleq [X_{0,\hat{\alpha}_l}[p,q]]$ (4.122), collecting the spectral information available on the whole array for the $\hat{\alpha}_l$-th delay bin. This algorithm consists of the following four steps:

1) The p_R-th row of the matrix $\mathbf{X}[l]$ is processed to generate the $\bar{N}_0$-dimensional column vector $\mathbf{s}_{\mathrm{VULA},0}[l] = [s_{0,0}[l], s_{0,1}[l], ..., s_{0,\bar{N}_0-1}[l]]^T$ on the basis of (4.124); here, p_R represents the column index of the reference antenna in the considered URA (see Fig. 4.2).

2) The dominant peaks of the sequence $\{|s_{0,r}[l]|; r = 0, 1, ..., \bar{N}_0 - 1\}$ are identified. If $r_i[l]$ denotes the index of i-th peak (with $i = 0, 1, ..., L_v[l] - 1$, where $L_v[l]$ is the overall number of targets detected in the considered delay bin), the estimate of the elevation $\hat{\phi}_i[l]$ of the associated target is evaluated as

$$\hat{\phi}_i[l] = \arcsin\left(h_{\bar{N}_0}[r_i[l]]\right). \tag{4.183}$$

3) The 2D inverse FFT of the matrix $\mathbf{X}[l]$ is computed; this produces the $\bar{N}_0 \times \bar{N}_0$ matrix $\bar{\mathbf{S}}[l] = [\bar{S}_{k,r}[l]]$, such that

$$\bar{S}_{k,r}[l] \triangleq \frac{1}{N_{\mathrm{V}}} \sum_{q=0}^{N_{\mathrm{VV}}-1} \sum_{p=0}^{N_{\mathrm{VH}}-1} X_{0,\hat{\alpha}_l}[p,q] \cdot \exp\left(j\frac{2\pi}{\lambda}\psi_{r,k}\right), \tag{4.184}$$

where

$$\psi_{r,k} \triangleq q\, h_{\bar{N}_0}[r]\, d_{\mathrm{VV}} + p\, h_{\bar{N}_0}[k]\, d_{\mathrm{VH}}. \tag{4.185}$$

4) The dominant peaks of the sequence $\{[|\bar{S}_{k,r_i[l]}[l]|; k = 0, 1, ..., \bar{N}_0 - 1\}$ are identified (with $i = 0, 1, ..., L_v[l] - 1$); let $L_h[i,l]$ denote their overall number. If the m-th peak is found for $k = k_{m,i}[l]$ (with $m = 0, 1, ..., L_h[i,l] - 1$), the azimuth $\hat{\theta}_{i,r_i[l]}[l]$ of the associated target is evaluated as

$$\hat{\theta}_{i,r_i[l]}[l] = \arcsin\left(\frac{h_{\bar{N}_0}[k_{m,i}[l]]}{\cos\left(\hat{\phi}_i[l]\right)}\right), \tag{4.186}$$

where $\hat{\phi}_i[l]$ is expressed by (4.183); consequently, the angular coordinates of the i-th target detected in the $\hat{\alpha}_l$-th delay bin are $(\hat{\theta}_{i,r_i[l]}[l], \hat{\phi}_i[l])$, whereas its range is computed on the basis of $\hat{\alpha}_l$. The last step concludes our description of the FFT-BAs. Note that the overall number of detected targets is given by

$$\hat{L} = \sum_{l=0}^{L_b-1} \sum_{i=0}^{L_v[l]-1} L_h[i,l]\,. \tag{4.187}$$

MUSIC-*based* DOA *estimation* – Similarly as our description of the FFT-BAs, we first focus on the case in which a virtual HULA, consisting of N_{VH} virtual elements, is employed for resolving the targets associated with a given delay bin and estimating their azimuth. In this case, the algorithm considered for DOA estimation consists of the following three steps:

1) The $N_{\mathrm{VH}} \times N_{\mathrm{VH}}$ autocorrelation matrix

$$\mathbf{R}_X[l] = \mathbf{X}[l]\ \mathbf{X}[l]^H \tag{4.188}$$

is computed; here, $\mathbf{X}[l]$ is defined by (4.180).

2) The $\bar{N}_0$-dimensional *pseudo-spectrum* $\mathbf{P}_{\mathrm{MU}}^{(l)}$ is evaluated; its k-th element is given by

$$\mathcal{P}_{\mathrm{MU}}^{(l)}[k] = \frac{1}{\mathbf{a}^H[k]\ \mathbf{Q}_{N_{\mathrm{VR}}}\mathbf{Q}_{N_{\mathrm{VR}}}^H\mathbf{a}[k]} \tag{4.189}$$

with $k = 0, 1, ..., \bar{N}_0 - 1$; here, $(\cdot)^H$ denotes the conjugate and transpose operator, $\mathbf{Q}_{N_{\mathrm{VH}}}$ is a matrix having size $N_{\mathrm{VH}} \times (N_{\mathrm{VH}}-1)$ and whose columns are the $(N_{\mathrm{VH}}-1)$ noise eigenvectors (associated with the $(N_{\mathrm{VH}}-1)$ smallest eigenvalues) of $\mathbf{R}_{\mathrm{X}}[l]$ (4.188) and $\mathbf{a}[k]$ is a N_{VH}-dimensional steering vector, whose n-th element $a_n[k]$ is given by

$$a_n[k] = \exp\left(-j\pi n\, h_{\bar{N}_0}[k]\right), \tag{4.190}$$

with $n = 0, 1, ..., N_{\mathrm{VH}} - 1$.

3) The dominant peaks appearing in the sequence $\{\mathcal{P}_{\mathrm{MU}}^{(l)}[k];\ k = 0, 1, ..., \bar{N}_0 - 1\}$, consisting of the ordered elements of $\mathbf{P}_{\mathrm{MU}}^{(l)}$, are identified; let $L_h[l]$ denote their overall number. If the i-th peak is found for $k = k_i[l]$ (with $i = 0, 1, ..., L_h[l] - 1$), the azimuth $\hat{\theta}_i[l]$ of the associated target is evaluated on the basis of (4.181)–(4.182).

Let us consider now the case in which the uniform rectangular array shown in Fig. 4.2 is employed for resolving the targets associated with each delay bin, and estimating their azimuth and elevation. In this case, the adopted procedure involves the $N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ matrix $\mathbf{X}[l] \triangleq [X_{0,\hat{\alpha}_l}[p,q]]$ (4.122) for any $\hat{\alpha}_l$ and consists of the following four steps:

1) The pseudo-spectrum referring to the reference VULA (that consists of N_{VULA} virtual elements) is evaluated. In this step, we assume that the p_R-th row of $\mathbf{X}[l]$ is employed for the evaluation of the autocorrelation matrix $\mathbf{R}_X[l]$ (4.188) and that the $\bar{N}_0$-dimensional vector $\mathbf{P}_{\mathrm{MU}}^{(\mathrm{VULA})}[l]$ is computed on the basis of (4.189)–(4.190) (note that N_{V} and $\delta[k]$ are replaced by N_{VULA} and $\delta[r]$, respectively).

2) The dominant peaks appearing in the sequence of the ordered elements of $\mathbf{P}_{\mathrm{MU}}^{(\mathrm{VULA})}[l]$ are identified; let $L_v[l]$ denote their overall number. If the i-th peak is found for $r = r_i[l]$ (with $i = 0, 1, ..., L_v[l] - 1$), the elevation $\hat{\phi}_i[l]$ of the associated target is evaluated on the basis of (4.183).

3) The pseudo-spectrum $\mathbf{P}_{\mathrm{MU}}^{(\mathrm{HULA})}[l,i]$ associated with the i-th estimated elevation is evaluated for the whole virtual array. In this step, if we assume that the autocorrelation matrix $\mathbf{R}_X$ is computed according to (4.188) (where, however, $\mathbf{X}[l]$ is the $N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ matrix defined above), the $\bar{N}_0$-dimensional vector $P_{\mathrm{MU}}^{(\mathrm{HULA})}[l,i]$ is generated on the basis of (4.189). Note that, in this case, N_{V} is replaced by N_{HULA} and that the n-th element $a_n[k]$ of the N_{HULA}-dimensional steering vector $\mathbf{a}[k]$ is

$$a_n[k] = \exp\left(-j\pi n\, h_{\bar{N}_0}[k]\cos\left(\hat{\phi}_i[l]\right)\right) \tag{4.191}$$

with $n = 0, 1, ..., N_{\mathrm{HULA}}$.

4) The dominant peaks appearing in the sequence of the ordered elements of $\mathbf{P}_{\mathrm{MU}}^{(\mathrm{HULA})}[l]$ are identified; let $L_h[i,l]$ denote their overall number. If the m-th peak is found for $k = k_{m,i}[l]$ (with $m = 0, 1, ..., L_h[i,l] - 1$), the azimuth $\hat{\theta}_{i,r_i[l]}[l]$ of the associated target is evaluated as

$$\hat{\theta}_{i,r_i[l]}[l] = \arcsin\left(h_{\bar{N}_0}[k_{m,i}[l]]\right). \tag{4.192}$$

Consequently, the angular coordinates of this target are $(\hat{\theta}_{i,r_i[l]}[l], \hat{\phi}_i[l])$, whereas its range is computed on the basis of its bin index $\hat{\alpha}_l$. The last step concludes our description of the MUSIC-BAs. Finally it is important to point out that:

a) The overall number of targets detected by these algorithms is still expressed by (4.187).

b) The order adopted in the computation of the pseudo-spectra (first the vertical pseudo spectrum $\mathbf{P}_{\mathrm{MU}}^{(\mathrm{VULA})}[l]$, then the horizontal pseudo-spectra $\{\mathbf{P}_{\mathrm{MU}}^{(\mathrm{HULA})}[l,i]\}$) is dictated by the fact $\mathbf{P}_{\mathrm{MU}}^{(\mathrm{HULA})}[l,i]$ depends on the elevation estimate $\hat{\phi}_i[l]$ for any i.

The performance of the FFT-BAs and the MUSIC-BAs has been assessed for both 2D and 3D propagation scenarios. It is important to worth that this MUSIC implementation is different from that one proposed in chapter 3, since this algorithm is based also on the concept of Alternating Maximization, in order to reduce the computational complexity of the algorithm.

4.4 Computational complexity

The computational cost of the algorithms described in section 4.2 and 4.3 has been carefully assessed in terms of *floating point operations* (flops) to be executed in the detection of L targets[10]. Our analysis leads to the conclusion that the overall cost of RASCA applied in a bi-dimensional scenario (range and azimuth estimation only) is approximately of order $\mathcal{O}(M_{\mathrm{R-S2}})$, respectively, where

$$\begin{aligned} M_{\mathrm{R-S2}} =& 24N_{\mathrm{VH}}\,N_0\log_2(N_0) + 26N_A\,K_{\mathrm{T}_2}N_0 \\ &+ L_b\,K_{\mathrm{T}_3}\left(18N_{\mathrm{VH}}\,N_0 + 8\bar{N}_0\log_2(\bar{N}_0)\right); \end{aligned} \tag{4.193}$$

while the overall cost of RASCA applied in a 3D scenario is of order $\mathcal{O}(M_{\mathrm{R-S3}})$ where

$$\begin{aligned} M_{\mathrm{R-S3}} =& \; 24N_{\mathrm{V}}\,N_0\log_2\left(N_0\right) + 26N_A\,K_{\mathrm{T}_2}\,N_0 \\ &+ L_b\,K_{\mathrm{T}_3}\left(18N_{\mathrm{VH}}\,N_{\mathrm{VV}}\,N_0 + 16\bar{N}_0\log_2\left(\bar{N}_0\right)\right) \end{aligned} \tag{4.194}$$

here, K_{T_2} (K_{T_3}) represents the overall number of iterations carried out by the STDREC (STDAEC) algorithm.

The cost of the FFT-based and MUSIC-based algorithms described in Section 4.3 are approximately of order $\mathcal{O}(M_{\mathrm{F-S3}})$, $\mathcal{O}(M_{\mathrm{F-S2}})$, $\mathcal{O}(M_{\mathrm{M-S3}})$ and $\mathcal{O}(M_{\mathrm{M-S2}})$, for FFT-BA and MUSIC-BA applied to 3D and 2D scenarios, respectively, where

$$\begin{aligned} M_{\mathrm{F-S3}} =& 8N_{\mathrm{V}}N_0\log_2(N_0) \\ &+ 8L_b\left(\bar{N}_0^2\log_2(\bar{N}_0^2) + \bar{N}_0\log_2(\bar{N}_0)\right), \end{aligned} \tag{4.195}$$

$$M_{\mathrm{F-S2}} = 8\big(N_{\mathrm{VH}}N_0\log_2(N_0) + L_b\bar{N}_0\log_2(\bar{N}_0)\big), \tag{4.196}$$

$$\begin{aligned} M_{\mathrm{M-S3}} =& 8N_{\mathrm{V}}N_0\log_2(N_0) + L_b\bar{N}_0(N_{\mathrm{VV}}^3 + N_{\mathrm{VH}}^3) \\ &+ 16L_b\bar{N}_0(N_{\mathrm{VV}}^2 + N_{\mathrm{VH}}^2) \end{aligned} \tag{4.197}$$

and

$$M_{\mathrm{M-S2}} = 8N_{\mathrm{VH}}N_0\log_2(N_0) + L_b\bar{N}_0(N_{\mathrm{VH}}^3 + 16N_{\mathrm{VH}}^2). \tag{4.198}$$

It is important to keep in mind that a comparison among the computational costs listed above does not fully account for the gap that can be observed in the execution speed

[10] In the remaining part of this section, the overall number of estimated targets ($\hat{L}$) is assumed to be equal to L, for simplicity.

of the corresponding algorithms. In fact, in practice, a portion of the computation time is absorbed by the procedure employed to find the dominant peaks of real sequences in both the FFT-BAs and the MUSIC-BAs. Moreover, the vector $\mathcal{A}_{\mathrm{CF}}$ (4.179), collecting the indices of the delay bins in which *at least* one target has been detected, may include *ghost* targets; as evidenced by the numerical results, the impact of this phenomenon on the overall computation time may not be negligible.

4.5 Numerical results

In this section, the accuracy of the proposed algorithms is assessed on the basis of experimental data collected by an SFCW commercial and colocated MIMO radar. In particular, after the description of the employed radar device and the adopted experimental setup, the accuracy achieved by RPE in range and phase estimation on multiple antennas of the same array and the accuracy of the 2D/3D images generated by the RASCA#1 algorithm are analysed in detail.

4.5.1 Employed radar devices and adopted experimental setup

A measurement campaign has been accomplished in the building of the Department of Engineering "Enzo Ferrari" (University of Modena and Reggio Emilia, Modena, Italy) to acquire data through an SFCW MIMO radar operating in the E-band. The device, dubbed VIC SFCW radar in the following, is the *Vito-In-Car* radar manufactured by *Vayyar Imaging Ltd Company* [62][11]; it is classified as a *short range radar* (SRR). Its main parameters are: a) initial transmit frequency $f_0 = 78$ GHz; b) frequency spacing $\Delta f = 16.67$ MHz; c) overall number of frequencies $N = 121$. Therefore, the bandwidth and the central frequency of the radiated signal are $B_{\mathrm{VIC}} = 2.0$ GHz and $f_c = 79$ GHz, respectively. This device is equipped with the *uniform rectangular array* (URA) shown in Fig. 4.9-a) this is composed of $N_T = 16$ TX antennas and $N_R = 21$ RX antennas, so that an URA of $16 \cdot 21 = 336$ VAs is available, as shown in Fig. 4.9-b). However, in this work, only $N_{VV} = 20$ HULAs, each consisting of $N_{HULA} = 16$ virtual channels characterized by the interantenna spacing[12] $d_{VH} = \lambda/4$, have been exploited; note that the vertical distance between two adjacent HULAs is $d_{VV} = \lambda/4$. The available array, made of $N_V = 320$ virtual elements, allows us to achieve a reasonable range resolution ($\Delta R_3 = c/(2B_{\mathrm{VIC}}) = 7.5$ cm), and good azimuthal and elevation resolutions (given by $\Delta\theta_3 = \lambda/(2d_H(N_{HULA}-1)) \cdot (180/\pi) = 7.6^\circ$ and $\Delta\phi_3 = \lambda/(2d_V(N_V-1) \cdot (180/\pi) = 6.0^\circ$, respectively). In this work, the HULA contained in the red rectangle appearing in Fig. 4.9-b) (the whole array) has been exploited for 2D (3D) imaging.

The measurement campaigns have been conducted in a large empty room (whose width, depth and height are 10 m, 8 m and 2.5 m, respectively). The employed radar device has been mounted on an horizontal wooden bar together with a pico-flexx camera manufactured by *PMD Technologies Inc.* [90] and has been lifted by a tripod at an height of roughly 1.60 m from ground, as shown in Fig. 4.10. The employed camera is based on a near-infrared *vertical cavity surface emitting laser* (VCSEL), and is able to provide a depth map or, equivalently, a 3D point-cloud of a small region of the observed environment (its maximum depth is equal to 4 m, whereas its *field of view* is $62^\circ \times 45^\circ$). The measurements acquired through the radar have been always pre-processed by the cancellation

[11] This radar is the same used for evaluating the accuracy of three dimensional CLEAN algorithm and MWLA described in Par. 3.5

[12] the spacing between two adjacent horizontal virtual channels (e.g. see definition in eqs. (2.1)–(2.2)) in Par. 2.3.1 and Fig. 4.9-b)

algorithm already available on this device; this algorithm exploits the measurements acquired from the first transmitted frame to remove unwanted received echoes.

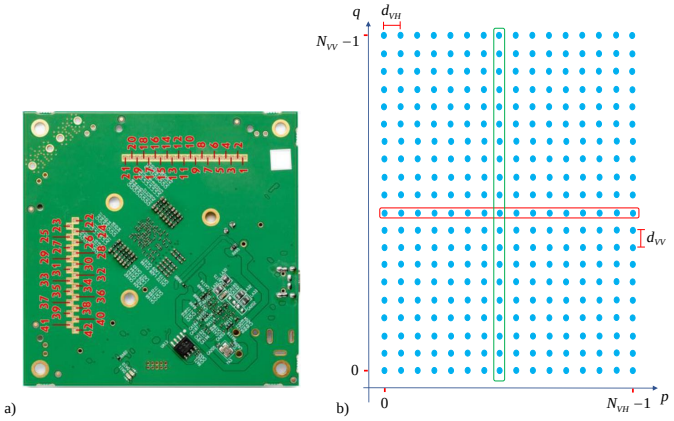


FIGURE 4.9: Representation of: a) the physical array of the VIC SFCW radar; b) the corresponding virtual array (the red rounded rectangle contains HULA employed for 2D imaging, whereas the green rounded rectangle the vertical array chosen as a reference for 3D imaging).

As far as the acquired measurements are concerned, it is important to point out that: a) target ranges have been estimated with respect to the central virtual channel of the employed ULA; b) the exact target positions have been acquired with respect to the centre of the pico-flexx camera; c) data processing has been accomplished in the MATLAB environment; d) all the detection and estimation algorithms have been run on a desktop computer equipped with a single i7 processor. Three different sets of experiments have been carried out to assess the accuracy of: a) the range and the amplitude estimates evaluated by the RPE (and, in particular, by the STDREC algorithm) in the presence of a single target and of multiple targets; b) the 2D and 3D images generated by the RASCA #1 method in the presence of multiple targets.

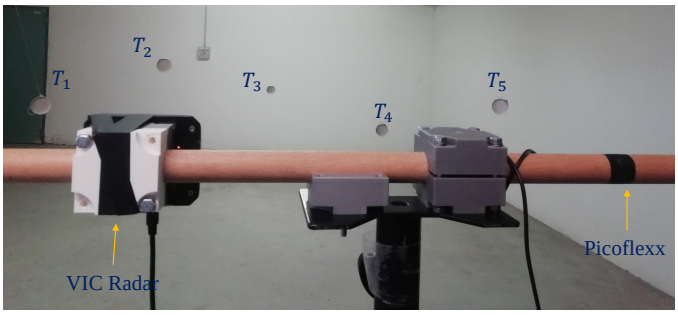


FIGURE 4.10: Experimental set-up developed for the considered measurement campaigns. The radar device and the reference sensor (pico-flexx) are mounted on a wooden bar. A group of metal targets, placed at the different height respect to the sensors, is also visible.

4.5.2 Range and amplitude estimation

In this paragraph, the accuracy of the STDREC algorithm employed by the RPE is analysed. The numerical results illustrated in this subsection refer to two specific static scenarios. The first scenario is characterized by a single detectable target, represented by a small metal disk[13] of size 5.5 cm. The target range R has been chosen in the interval $(1.0\text{ m}, 3.0\text{ m})$, with a step of 0.5 m; in all cases, its azimuth θ has remained within

[13]Each target is hung from the ceiling: a nylon thread has been used for suspending them.

the interval $(-40°, 40°)$. The exact range and azimuth characterizing the ten distinct positions considered for the targets are listed in Table 4.1 for all the considered radar devices (the data referring to the i-th position are collected in the column identified by T_i, with $i = 1, 2, ..., 10$). The second scenario, instead, is characterized by the presence of an overall number of targets ranging from 1 to 9 (so that $1 \leq L \leq 9$). The targets are represented by small coins with a diameter of 2 cm; the range and azimuth characterizing their exact positions are listed in Table 4.2 (the data referring to the i-th target are collected in the column identified by T_i, with $i = 1, 2, ..., 9$); they have been sequentially added in the scenario, starting from a single target and then increasing the number of targets up to 9. This has allowed to assess how the performance of the STDREC algorithm is influenced by the value of the parameter L in the presence of *closely spaced targets*. In processing all the acquired measurements, prior knowledge of L has been always assumed [14]; moreover, the following values have been selected for the parameters of the STDREC algorithm: $N = 121$, $N_0 = N \cdot M = 1936$, $M = 16$, $N_{\text{CSDE}} = 5$ and $I = 7$. It is worth nothing that: a) the value of the parameters has been selected in way to guarantee approximately a sufficient bin resolution for the STDREC algorithm (given by $\Delta R_{bin} = \frac{cM}{2B} = \frac{\Delta R_x}{M}$, with $x = 1, 2, 3$). In analysing the data acquired in both scenarios, the accuracy of range estimates has been assessed by evaluating the RMSE $\bar{\varepsilon}_R$ and the peak error $\hat{\varepsilon}_R$ (see eq. (3.73) and (3.74), respectively, considering $X_l = R_l$ ($\hat{X}_l = \hat{R}_l$)). Moreover, since the RCS of the considered targets was unknown, the analysis of the complex gains available over the 16 channels of the considered virtual ULA and associated with the same target has concerned only their (unwrapped) phase.

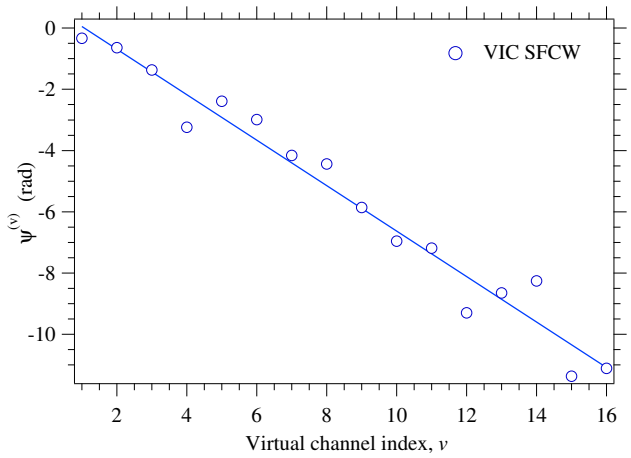


FIGURE 4.11: Unwrapped phase of the complex gain versus index of the virtual channel of the reference HULA; a single target is assumed.

An example of the phases trajectory estimated by the STDREC algorithm over the considered reference HULA, consisting of 16 VAs (the reference HULA is composed by the VAs contained inside the red rounded rectangles appearing in Fig. 4.9-b)) and associated with a target placed at relatively small azimuth angle with respect to the centre of the radars is shown[15] in Fig. 4.11. Since the horizontal distance d_{VH} between adjacent virtual channels is constant, the estimated phases exhibit a linear dependence on the index of the virtual channel (see, in particular, eqs. (2.10) and (2.12) in Sect. 2.4). Moreover, if a linear fitting is drawn for these data, it should be expected that the slope of the resulting straight line is proportional to $\sin(\theta)$ (see eq. (2.42)); this is confirmed by the results shown in Fig. 4.11. To assess the quality of the estimated phases, their

[14] Some methods for estimating the number of target L are available in [74].

[15] The exact coordinates can be found in the column labelled by T_7 in Table 4.1.

RMSE[16] $\bar{\varepsilon}_\psi$ has been evaluated in all the scenarios; in doing so, the linear fitting of the phases estimated over the whole ULA has been taken as a reference with respect to which the error of each phase estimate has been computed. The estimate of the target range generated by the STDREC algorithms in each of the $N_m = 10$ distinct positions considered in the first scenario are listed in Table 4.1; in the same table, the value of $\bar{\varepsilon}_\psi$ computed for each position is also given. The target ranges and their estimates listed in Table 4.1 are also represented in Fig. 4.12. The errors $\bar{\varepsilon}_R$ (3.73) and $\hat{\varepsilon}_R$ (3.74), the mean of $\bar{\varepsilon}_\psi$ (denoted by $\bar{\varepsilon}_{m,\psi}$ and generated by taking the average of the N_m values available for $\bar{\varepsilon}_\psi$) and the average *computation time* (CT) evaluated on the basis of these results are listed in Table 4.3.

Methods		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Exact	R (m)	1.2	1.08	1.5	1.42	2.05	2.1	2.75	2.45	2.88	3.09
	θ (°)	-35	30	-20	20	-12	15	-14	15	- 18	25
STDREC	$\hat{R}$ (m)	1.242	1.06	1.517	1.413	2.025	2.06	2.78	2.409	2.905	3.041
	$\bar{\varepsilon}_\psi$ (rad)	0.25	0.35	0.67	0.345	0.422	0.45	0.686	0.919	0.379	0.54

TABLE 4.1: Exact positions (range and azimuth) of the considered target (first scenario), estimated ranges and RMSE evaluated for the phase fitting over the considered sixteen virtual channels of the VIC SFCW device.

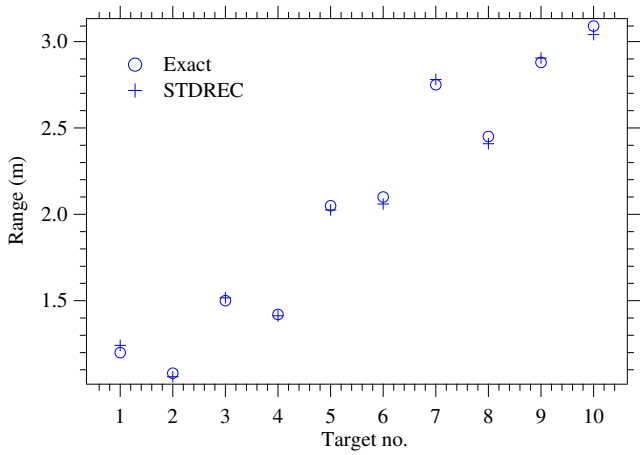


FIGURE 4.12: Representation of the ranges estimated by the STDREC algorithm (first experimental scenario).

These results and those listed in Table 4.1 have led us to the following conclusions:

1. In all the considered cases, the STDREC is able to accurately estimate the range and the amplitudes of a single target.
2. All the values of $\bar{\varepsilon}_R$ and $\hat{\varepsilon}_R$ are comparable, reasonably low and in the order of the resolution of the devices.
3. The value of $\bar{\varepsilon}_\psi$ characterizing the VIC SFWC radar is lower when the target is closer to this device. This result is mainly due to the fact: a) the far-field condition

[16]The evaluation of this RMSE is based on a formula similar to eq. (3.73).

of the VIC SFWC device is satisfied at short ranges, since the antenna array of the former device is more compact than that of the latter one; b) the VIC SFCW radar is a SRR system. In fact, for these reasons, the measurements provided by the VIC SFCW radar allow us to obtain accurate results at a shorter distance.

4. The CT is in the order of few milliseconds.

Method	T_1 (m)	T_2 (m)	T_3 (m)	T_4 (m)	T_5 (m)	T_6 (m)	T_7 (m)	T_8 (m)	T_9 (m)
Exact	1.860	1.900	1.980	2.110	2.190	2.220	2.370	2.410	2.460
STDREC	1.837	1.920	1.98	2.081	2.192	2.266	2.355	2.427	2.518
HSTDREC	1.838	1.920	1.976	2.081	2.199	2.265	2.351	2.423	2.530

TABLE 4.2: Exact positions of the nine targets characterizing the second scenario. The range estimates computed by the STDREC, HSTDREC algorithms are also provided.

Methods	$\bar{\varepsilon}_R$ (m)	$\hat{\varepsilon}_R$ (m)	$\bar{\varepsilon}_{m,\psi}$ (rad)	CT (msec)
STDREC	0.032	0.049	0.501	0.3

TABLE 4.3: *Root mean square error* $\bar{\varepsilon}_R$, *peak error* $\hat{\varepsilon}_R$, *mean error* $\bar{\varepsilon}_{m,\psi}$ and *computation time* (CT) evaluated for the STDREC algorithm in the first scenario.

Methods	$\bar{\varepsilon}_R$ (m)	$\hat{\varepsilon}_R$ (m)	CT (msec)
STDREC	0.03	0.06	15
HSTDREC	0.03	0.07	16

TABLE 4.4: *Root mean square error* $\bar{\varepsilon}_R$, *peak error* $\hat{\varepsilon}_R$, and *computation time* (CT) evaluated in the second scenario.

Let us focus on the second scenario now. In this case, the range estimates have been generated by the STDREC algorithm and its hybrid version, i.e. the HSTDREC algorithm. The obtained results are listed in Table 4.2. The errors $\bar{\varepsilon}_R$ and $\hat{\varepsilon}_R$, and the CT assessed in this case are listed in Table 4.4. From these results it is easily inferred that:

1. The STDREC and the HSTDREC algorithms works properly for $L = 1$ up to $L = 9$.
2. In the considered scenario, the HSTDREC algorithm does not provide a better accuracy than the STDREC algorithms; this means that estimation bias of the last algorithms does not play an important role in this case.
3. The estimated RMSEs and peak errors are comparable to the order of the resolution of the device, even if the estimation of the first target is not as good as expected. This is probably due to the fact that the energy received from this target has been found to be lower than that coming from the other four targets.

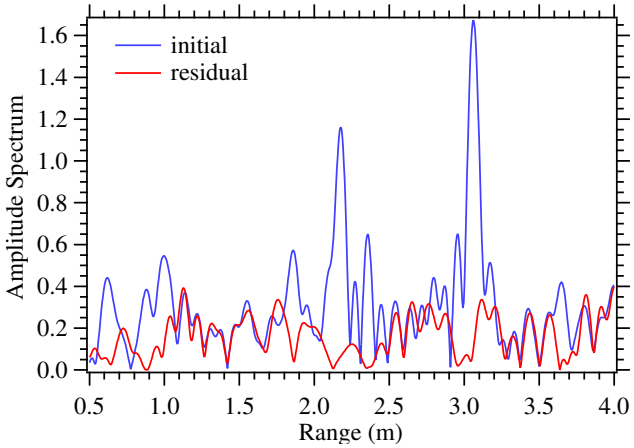


FIGURE 4.13: Representation of the initial amplitude spectrum of the signal observed on the central virtual channel (blue line) and of the final *residual* amplitude spectrum generated by the STDREC algorithm (red line).

Finally, it is important to stress on the fact that the robustness of STDREC and HST-DREC algorithms is related to the accuracy of the estimation and cancellation procedure they accomplish. This is exemplified by Fig. 4.13, where the *initial amplitude spectrum* of the signal received on the central virtual channel of the SFCW radar in the second scenario and *its residual*, resulting from the cancellation of the spectral contributions due to the five detected targets, are shown.

4.5.3 Two-dimensional and three-dimensional imaging

The numerical results illustrated in this subsection have been generated to assess the accuracy of the 2D and 3D images generated by the RASCA #1 algorithm. Two different groups of experiments have been done. The first (second) group has allowed to assess the performance achieved by the above mentioned algorithm in 2D (3D) imaging. In both cases, the measurements have been acquired in the presence of an increasing number of targets through the radar device; in practice, these targets, each represented by a small metal disk of size 5.5 cm, have been sequentially introduced in the propagation environment, so that their overall number (L) has ranged from one to five (i.e., $1 \leq L \leq 5$). In the first group of experiments, the following choices were made:

1. The targets have been placed in a very small region and at the same height. The range R and azimuth θ of each of them are listed in Table 4.5; note that these parameters belong to the intervals [2.2 m, 2.7 m] and $[-15\,^{\circ},\ 30^{\circ}]$, respectively.

2. The measurements have been acquired through a virtual ULA, consisting of 16 VAs.

The range R, azimuth θ and elevation ϕ of the targets considered in the second group of experiments are listed in Table 4.7; the values of these parameters belong to the intervals [1.9 m, 2.8 m], $[-30\,^{\circ},\ 35\,^{\circ}]$ and their elevation $[-10\,^{\circ}, 10\,^{\circ}]$. In this case, the measurements have been acquired through the whole virtual array of the employed radar device.

In analysing the acquired measurements, the following choices have been made for the parameters of RASCA #1: a) $N_A = 16$ ($N_A = 10$) in the RPE employed in 2D (3D) imaging; b) $N_{CSDE} = 10$ in both the STDREC and the STDAEC algorithms; c) the values

of the parameters N_0 and M are equal to those employed for the STDREC in the previous paragraph; d) the oversampling factor $\bar{M} = 16$, obtaining a value of $\bar{N}_0 = 320$ when the entire array is employed and e) $T_{OF} = 0$. Moreover, the following values have been selected for the parameters of the FFT-BAs and the MUSIC-BAs: $C_s = 6$, $G_s = 6$ and $K_0 = 2$. Furthermore, prior knowledge of L has been always assumed (otherwise a minimum threshold $T_{\text{STDREC}} = 0.2 \cdot E_0^{(v_k)}$ (4.116) can be employed for both 2D and 3D) and the threshold T_{STDAEC} has been selected in the range $[0.01, 0.9] \cdot E^{(0)}[l]$ (4.144) (its value has been adjusted on the basis of the SNR of the received signal and the number of detectable targets).

Exp.	Methods	Params.	T_1	T_2	T_3	T_4	T_5
	Exact	R (m)	2.26	2.51	2.44	2.68	2.21
		θ (°)	-12.7	-4.5	10.6	18.0	28.3
1)	RASCA	R (m)	2.18				
		θ (°)	-11.8				
2)	RASCA	R (m)	2.25	2.55			
		θ (°)	-12.8	-6.3			
3)	RASCA	R (m)	2.31	2.54			2.16
		θ (°)	-13.2	-7.2			25.5
4)	RASCA	R (m)	2.31	2.54		2.75	2.16
		θ (°)	-13.2	-6.7		17.5	25.5
5)	RASCA	R (m)	2.30	2.52	2.53	2.7	2.13
		θ (°)	-12.5	-7.0	4.0	17.1	27.0

TABLE 4.5: Exact range and azimuth of the five targets considered in the first group of experiments and corresponding estimates generated by the RASCA#1.

Let us focus first on the case of 2D imaging. The estimates of range and azimuth generated by the RASCA #1 are listed in Table 4.5, whereas the values of RMSE, peak error and CT computed by averaging the RMSEs in each single experiment are listed in Table 4.6. From these results it can be easily inferred that all the range and azimuth errors are reasonably low and comparable to the resolution of the device.

Method	$\bar{\varepsilon}$		$\hat{\varepsilon}$		CT
	R (m)	θ (°)	R (m)	θ (°)	(sec)
RASCA	0.07	2.5	0.09	3.8	0.3
FFT-BA	0.07	3.0	0.09	4.0	0.45
MUSIC-BA	0.06	2.5	0.08	3.5	0.45

TABLE 4.6: Average *root mean square error* $\bar{\varepsilon}$, *peak error* $\hat{\varepsilon}$, and *computation time* (CT) evaluated on the basis of first measurement for RASCA, FFT-BA and MUSIC-BA.

The good accuracy achieved by the RASCA#1 is also evidenced by Fig. 4.14, where the *range-azimuth map*[17] generated on the basis of the measurement acquired through

[17]This is generated though standard 2D IFFT processing [58].

the SFCW radar is represented as a contour plot[18]; in the same figure, the exact position of the five targets employed in the first group of experiments and their estimates are shown. Let us consider now the case of 3D imaging.

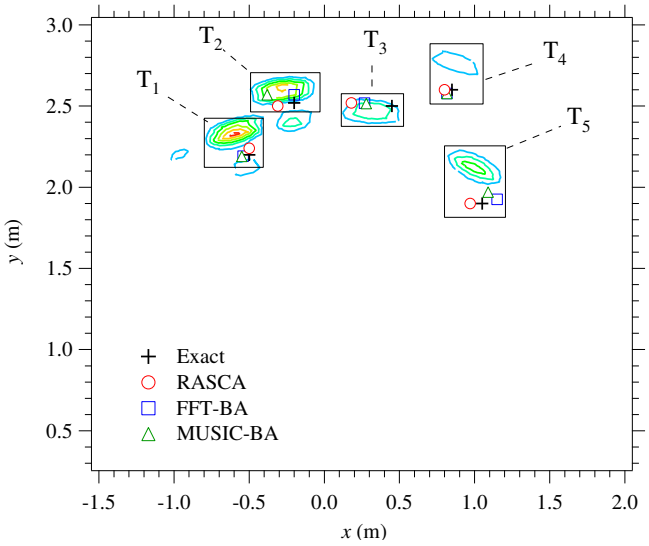


FIGURE 4.14: Representation of the range-azimuth map (in $x-y$ coordinates) in the presence of five targets. The exact position of each target and its estimate (shown in Table 4.5) is also shown.

Exp.	Methods	Params.	T_1	T_2	T_3	T_4	T_5
	Exact	R (m)	1.94	2.34	2.75	2.49	2.08
		θ (°)	-27.8	-9.9	0	14.0	35.2
		ϕ (°)	-6.0	2.0	-2.1	-7.0	-2.0
1)	RASCA	R (m)	1.85				
		θ (°)	-20.9				
		ϕ (°)	-3.5				
2)	RASCA	R (m)	1.91				2.05
		θ (°)	-21.0				28.5
		ϕ (°)	-3.5				-1.5
3)	RASCA	R (m)	1.89	2.24			2.1
		θ (°)	-21.0	-9.5			29.2
		ϕ (°)	-4.0	5.0			-3.0
4)	RASCA	R (m)	1.85	2.23		2.60	2.06
		θ (°)	-21	-9		15	29
		ϕ (°)	-3.5	0		-7	-2.5
5)	RASCA	R (m)	1.82	2.26	2.82	2.61	2.1
		θ (°)	-21.0	-9.0	2.1	14.6	29.0
		ϕ (°)	-3.5	0	-2.5	-8.2	-1.0

TABLE 4.7: Exact range, azimuth and elevation of the five targets considered in the second group of experiments and corresponding estimates generated by the RASCA#1.

[18]Note that $x-y$ coordinates are employed in this case, in place of range and azimuth; the position of the radar system corresponds to the origin of the reference system.

Methods	$\bar{\varepsilon}$			$\hat{\varepsilon}$			CT
	R (m)	θ (°)	ϕ (°)	R (m)	θ (°)	ϕ (°)	(sec)
RASCA	0.07	5.2	1.7	0.09	5.3	1.8	2.2
FFT-BA	0.08	5.1	2.1	0.11	6.5	2.8	1.0
MUSIC-BA	0.11	5.3	2.1	0.18	6.7	3.1	1.1

TABLE 4.8: Average *root mean square error* $\bar{\varepsilon}$, *peak error* $\hat{\varepsilon}$, and *computation time* (CT) evaluated in the second experiment. Target range, azimuth and elevation are taken into consideration.

The estimates of range, azimuth and elevation generated by the RASCA#1 are listed in Table 4.7, whereas the values of RMSE, peak error and CT computed by averaging the RMSEs, peak errors and CTs evaluated in each single experiment are listed in Table 4.8. In this last table, the values of RMSE, peak error and CT for the employed FFT-BAs and MUSIC-BAs are also provided. These results deserve the following conclusions:

a) The RMSE and the peak errors evaluated for target range, azimuth and elevation are reasonably low and comparable with those obtained in the case of 2D imaging for range and azimuth only (see Table 4.6);

b) The accuracy achieved by the VIC SFCW radar in the estimation of target elevation is better than that in azimuth estimation. This is due to the fact that the URA of the device has more channels, i.e. a finer angular resolution along the vertical direction respect the horizontal one.

c) The average CT is in the order of few seconds for the proposed algorithm.

The good resolution provided by the RASCA#1 is also highlighted by Fig. 4.15, where the exact positions of the five targets employed in the second group of experiments and their estimates are shown; note, in particular, that a good accuracy is achieved even in the presence of closely spaced targets, like T_4 and T_5.

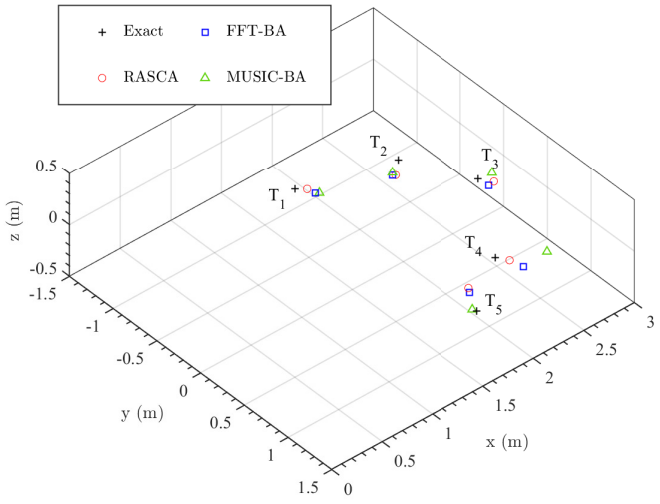


FIGURE 4.15: Representation of a 3D scenario characterized by five targets. The exact position of each target and its estimates are shown.

4.6 Conclusions

In this chapter, a novel algorithm for detecting and estimating the parameters of a single exponential using an SFCW radar system has been proposed; moreover, its has been shown how it can be exploited to estimate the parameters of multiple exponentials through a serial cancellation procedure inspired by the CLEAN algorithm and the MWLA. These two methods, namely the CSDE and CSDEC, have been exploited for deriving two imaging algorithms for the detection and estimation of multiple targets in a colocated MIMO SFCW radar system, dubbed RASCA#1 and RASCA#2. These algorithms allow to obtain reasonably accurate estimates of range, DOAs and amplitude in the presence of single or multiple closely spaced targets in both 2D and 3D scenario, representing a valid alternative to the methods proposed in Sect 3.3 and 3.4. The accuracy and robustness of STDREC (that includes the CSDEC) and RASCA #1 have been also assessed on the basis of various measurements acquired through a state of the art MIMO SFCW commercial radars. The experimental results have led to the conclusion that a) the STDREC method is able to generate very accurate estimates regarding the range and the complex amplitude of a single or multiple targets; b) the RASCA#1 is able to generate accurate 2D and 3D radar images.

It is important to underline that the STDREC and RASCA methods can be potentially used in other applications, not necessarily related to the field of radar systems, in which other estimators solving the same problem are adopted. In particular, the STDREC algorithm may be employed for estimating the *channel impulse response* (CIR) of a channel. In fact, such a CIR may be represented by a sequence of lines (i.e., of delayed delta functions), each positioned in a specific *time bin* and characterized by a specific complex gain. Moreover, the received signal model expressed by eq. (2.25) has the same mathematical structure as the signal available at the output of the IDFT evaluated by the demodulator in a digital communication system employing the *orthogonal frequency division multiplexing* (OFDM) modulation (e.g., see [91, Par.4.4.4, eq. (4.129)]) when *known* (i.e., *pilot*) *channel symbols* are transmitted. For this reason, the STDREC algorithm may be exploited for accomplishing *pilot-based channel estimation* of the impulse response of the communication channel in this system. This result has significant technical implications, since the OFDM has been adopted in the air interface of *fourth generation* (4G) and *fifth generation* (5G) standards for cellular communications [92], [93]. It is worth mentioning that, to the best of my knowledge, the use of a SIC approach to channel estimation has been proposed in ref. [94, Par. II-A] for a *long term evolution*-advanced (LTE-advanced) cellular system; however, the estimation technique proposed in that work is completely different from the STDREC technique devised for a SFCW radar system. The novelty of the solution is mainly represented by the STDREC technique and by the simplicity of the adopted cancellation procedure. The STDREC technique, in fact, is able to generate an accurate estimate of the parameters of each exponential through the computation of both the integer part and the fractional part of its delay; the accuracy of delay estimation plays a fundamental role in the efficacy of the *serial cancellation procedure*, mitigating the *error accumulation* problem.

5 | Learning techniques for colocated MIMO radars

5.1 Introduction

Based on what has been shown in chapters 3 and 4, it is possible to state that the full exploitation of the potentialities offered by modern colocated MIMO radar devices cannot be separated by the use of proper detection and estimation methods [22], [95].
In particular, the proposed methods are *model-based*, since they require the full knowledge of the employed radar device and rely on a parametric description of the propagation environment; note that, in such a description, targets are usually represented as *points* reflecting electromagnetic energy [58], [59], [96], [97]; broadly speaking, in many cases, these methods allow to achieve good estimation accuracy at the price of an acceptable computational effort.

Unluckily, in a number of recent applications, MIMO radars operate in extremely complex, highly dynamic and time varying scenarios, affected by multipath propagation, clutter and interference, and in the presence of *extended* targets. In such conditions, deterministic algorithms may fail, since they are unable to achieve acceptable estimation accuracy and are prone to generate ghost targets [98]. When this occurs, *machine learning* (ML) and *deep learning* (DL) techniques represent an appealing alternative or the only viable technical solution. A relevant example of this class of techniques is represented by *neural networks* (NNs) [87], [99]. These networks can automatically learn specific data patterns and extract useful information directly from raw data, even in the presence of strong interference. In fact, they can be trained to recognise interference and remove it, so making the recovery of useful signal components possible.

Unfortunately, the application of NNs and related methods to MIMO radars is challenging, because, on the one hand, the problems tackled in this field are often substantially different from those to which such methods have been applied for a number of years (e.g., processing of RGB images in computer vision); on the other hand, the large radar dataset required for the proper training of a NN may be unavailable. Another critical issue emerging from the exploitation of ML and DL methods in real world applications is represented by the fact that a trained machine is, by and large, a black box mapping inputs to outputs; for this reason, generally speaking, it cannot be inferred why a given output has been based on the provided input data [100]. This explains why, in various radar applications, a model-based approach could be preferred.
Despite this relevant limitation, it is widely accepted that the use of ML and DL methods in colocated MIMO radars will allow to solve a number of real world problems. For instance, recent work has evidenced that they can be successfully exploited in the classification of human activities and gestures, in the detection of human falls [101] and in the classification of dynamic targets in dense and dynamic urban scenarios [102].

This chapters aims at providing an overview of the ML and DL methods employed in most of the possible radar applications, analysing their pros and cons, discussing the main lessons that have been learnt from their use and illustrating some trends in this research area. Furthermore, the description of learning methods is enriched with various numerical examples on synthetically generated dataset and an entire section is devoted to the analysis of numerical results generated the measurements acquired through a commercial colocated MIMO radar. In particular, differently from what done in the previous chapter, I preferred to focus on ML methods for an FMCW MIMO Radar, since these devices are the most popular in the automotive market and, as far as I know, a lot of works regarding ML algorithms and radar systems available in the recent literature are concerning with such devices.

The remaining part of this chapter is organized as follows. In sections 5.2 and 5.3 the most relevant ML and DL methods currently being investigated for their use in colocated MIMO radars are described, respectively. An overview of the specific applications of these techniques to colocated MIMO radars is illustrated in section 5.4; these applications regards human-motion and human-gesture classification, healthcare monitoring, and target detection and localization. Some trends emerging in the current research activities about the application of DL techniques to colocated MIMO radars are illustrated in section 5.5. Various ML and DL methods are compared, in terms of accuracy and computational complexity, in section 5.6, where their use in human activity classification, and in the detection and position estimation of a moving target is illustrated. Finally, some conclusions are offered in section 5.7.

5.2 Machine learning based methods

In paragraph 2.5 one of the simplest deterministic approaches for target detection and estimation of its parameters is proposed for an FMCW radar. An alternative to such algorithm is offered by ML methods [87]. If ML techniques are employed, the inner structure of the considered radar system and the physical laws on which its operation is based can be ignored, since the required information are automatically extracted by an algorithm able to *learn the regularities characterizing the set of observed data*. Let us reconsider now the detection and estimation problem described in paragraph 2.5 from this new perspective and show how a solution based on ML methods can be devised. To this aim, I take into consideration again a FMCW radar system equipped with the antenna array shown in Fig. 2.8-a) and assume that it is employed to perform a measurement campaign. In this campaign, N_t independent trials are accomplished in the presence of a single point target or in the absence of it; in each trial, the pair $[\mathbf{x}^{(0)}, \mathbf{x}^{(1)}]$ of noisy vectors (see eq. (2.48)) is acquired and stored in a database together with the target range and azimuth (when the target is present). In the following, $[\mathbf{r}_{q,0}, \mathbf{r}_{q,1}]$ and

$$\mathbf{t}_q \triangleq \left[d_q, R_q, \phi_q\right]^T \tag{5.1}$$

denote the value of the pair $[\mathbf{x}^{(0)}, \mathbf{x}^{(1)}]$ and the associated *label* acquired in the q-th trial (with $q = 0, 1, \ldots, N_t - 1$) and used as ground truth; here, $d_q = -1$ (1) if the target is absent (present), and R_q and ϕ_q represent the target range and azimuth[1], respectively, in the same trial if $d_q = 1$ (if $d_q = -1$, the values of both R_q and ϕ_q are irrelevant). In this

[1]In this chapter the azimuth angle is always indicated by the Greek letter ϕ, in order to avoid confusion with the vector of parameters of the network.

case, the ML approach consists in processing the dataset

$$\mathcal{D}_i \triangleq \left\{ \left[\mathbf{r}_{q,0}, \mathbf{r}_{q,1} \right], \mathbf{t}_q; q = 0, 1, ..., N_t - 1 \right\} \tag{5.2}$$

to learn how to detect the presence of a target on the basis of a new pair $[\mathbf{x}^{(0)}, \mathbf{x}^{(1)}]$ and, if a target is detected, how to estimate its position. The accuracy of the algorithm resulting from the learning phase (i.e., from *training*) depends not only on the adopted ML method, but also on the size of $\mathcal{D}_i$ (i.e., on N_t). Generally speaking, the use of ML methods requires the availability of a large set of measurements, i.e. a large N_t (say, at least, a few thousands). Unluckily, any ML method extracting the required knowledge directly from $\mathcal{D}_i$ (5.2) has to process high dimensional vectors if the size N of the vectors $\mathbf{r}_{q,0}$ and $\mathbf{r}_{q,1}$ is large. Actually, the dimensionality of the given problem can be reduced by exploiting prior knowledge about the problem itself. In fact, in developing the proposed deterministic algorithm, we have learnt that essential information for target detection and estimation is provided by the complex pair $[X_{\hat{l}}^{(0)}, X_{\hat{l}}^{(1)}]$ (see eqs. (2.50)-(2.51)), where $\hat{l}$ is expressed by eq. (2.55). These considerations suggest to:

a) *Pre-process* the pair $(\mathbf{r}_{q,0}, \mathbf{r}_{q,1})$ in order to generate the vector

$$\mathbf{X}_q \triangleq \left[X_q^{(0)}, X_q^{(1)} \right]^T, \tag{5.3}$$

where $X_q^{(0)}$ and $X_q^{(1)}$ are the values taken on by the quantities $X_{\hat{l}}^{(0)}$ and $X_{\hat{l}}^{(1)}$, respectively, in the q-th trial (with $q = 0, 1, ..., N_t - 1$); $X_q^{(0)}$ and $X_q^{(1)}$ can be considered as highly informative data extracted from the received signal, i.e., briefly, as the *features* available for the considered problem.

b) Replace the set $\mathcal{D}_i$ (5.2) with the new set

$$\mathcal{D} \triangleq \{\mathbf{X}_q, \mathbf{t}_q; q = 0, 1, ..., N_t - 1\}, \tag{5.4}$$

that consists of low dimensional vectors only, and use it to train the considered ML method; when this occurs, the last set is called *training set*.

Once training is over, the ML algorithm resulting from it is able to infer the unknown value of $\mathbf{t}_q$ (5.1) for any new vector $\mathbf{X}_q$ (5.3) of noisy data (with $q > N_t - 1$); in other words, it is able to predict: a) d_q; b) R_q and ϕ_q if a target is detected. It is important to remark that any ML algorithm predicting d_q solves a *binary classification problem*, since it assigns a new observation to one of two *categories* of noisy data, one associated with the presence of a target, the other one with its absence; in other words, *the algorithm is exploited to recognise a specific pattern in the noisy observations*. If the considered ML algorithm is also able to *predict* the value of the pair (R_q, ϕ_q) (i.e., of two continuous variables), it solves a *regression problem* too. In the considered radar system, different ML algorithms can be employed to learn classification and regression rules from the training set $\mathcal{D}$ (5.4). Moreover, all such algorithms can be considered as specific instances of the so called *supervised learning* methodology, as shown in the following paragraph. Generally speaking, supervised learning techniques can be employed when:

1. A training set

$$\mathcal{D} \triangleq \left\{ \left(\mathbf{r}_q, \mathbf{t}_q \right); q = 0, 1, ..., N_t - 1 \right\}, \tag{5.5}$$

collecting N_t, D_r-dimensional real *observations* (also called *covariates*, or *domain points*, or *explanatory variables*) $\{\mathbf{r}_q; q = 0, 1, ..., N_t - 1\}$, with

$$\mathbf{r}_q \triangleq \left[r_{q,0}, r_{q,1}, ..., r_{q,D_r-1}\right]^T, \tag{5.6}$$

and the associated D_t-dimensional real *labels* (also called *dependent variables* or *responses*) $\{\mathbf{t}_q; q = 0, 1, ..., N_t - 1\}$, with

$$\mathbf{t}_q \triangleq \left[t_{q,0}, t_{q,1}, ..., t_{q,D_t-1}\right]^T, \tag{5.7}$$

is available.

2. There exists some mechanism relating the variable $\mathbf{r}_q$ to the variable $\mathbf{t}_q$ for any q.

The last point is a fundamental one, since it does not make any sense to develop rules applicable to unseen examples in the absence of some assumptions about the mechanism underlying data generation; the set of these assumptions is known as the *inductive bias*.

5.2.1 The supervised learning problem

Generally speaking, supervised learning concerns the identification of the conditional *probability density function* (pdf) $f(\mathbf{t}|\mathbf{r})$ (also called *predictive distribution*) minimizing the average *generalization loss*

$$L_p(\tilde{\mathbf{t}}) \triangleq \mathrm{E}_{f(\mathbf{t},\mathbf{r})}\left\{\ell\left(\mathbf{t}, \tilde{\mathbf{t}}\left(\mathbf{r}\right)\right)\right\}; \tag{5.8}$$

here, $\mathrm{E}_{f(\mathbf{x})}\{\cdot\}$ denotes the expectation evaluated with respect to the joint $f(\mathbf{x})$, $\tilde{\mathbf{t}}(\mathbf{r})$ is a *prediction* of the label $\mathbf{t}$ computed from the observation $\mathbf{r}$ and $\ell(\cdot,\cdot)$ is a given *cost function*. If the label of each observation is *one-dimensional* (1D) and is real, the cost functions

$$\ell_2\left(t, \hat{t}\right) = (t - \hat{t})^2 \tag{5.9}$$

and

$$\ell_0\left(t, \hat{t}\right) = \begin{cases} 1 \text{ if } t = \hat{t} \\ 0 \text{ elsewhere} \end{cases} \tag{5.10}$$

are often employed for regression and binary classification, respectively. It is well known that, if the posterior pdf $f(\mathbf{t}|\mathbf{r})$ is *known*, the minimum value of the loss $L_p(\tilde{\mathbf{t}})$ (5.8) is achieved by selecting the *optimal prediction* (e.g., see [87, Par. III.C, eq. (4)])

$$\hat{t}\left(\mathbf{r}\right) = \arg\min_{\tilde{\mathbf{t}}} \mathrm{E}_{f(\mathbf{t}|\mathbf{r})}\left[\ell\left(\mathbf{t}, \tilde{\mathbf{t}}\right)|\mathbf{r}\right], \tag{5.11}$$

whatever cost function is selected.

Supervised learning methods are employed when the conditioned pdf $f(\mathbf{t}|\mathbf{r})$ (or the joint pdf $f(\mathbf{t}, \mathbf{r})$) is unknown, but a training set $\mathcal{D}$, collecting N_t distinct data generated on the basis of it and structured according to eq. (5.4), is available. The objective of these methods is to generate a *predictor*, denoted by $\hat{t}_{\mathcal{D}}(\mathbf{r})$, exclusively based on $\mathcal{D}$ and whose performance, in terms of generalization loss, is as close as possible to that of the optimal predictor $\hat{t}(\mathbf{r})$ (5.11); this means that the loss evaluated for the prediction of the label associated with a new observation should be as small as possible. The derivation of the predictor $\hat{t}_{\mathcal{D}}(\mathbf{r})$ can be formulated as an optimization problem, whose form depends on the specific assumptions about the model that is being learnt. The *frequentist approach*

or a *Bayesian approach* can be adopted for solving this problem, as illustrated in the following paragraph. In particular, the general principles of these two approaches can be employed to solve a specific regression problem concerning the FMCW radar system described in chapter 2 and equipped with the array shown in Fig. 2.8-a). In this case, we assume: a) the presence of a single point target placed at a fixed and known range R; b) the availability of the synthetically generated dataset (see eq. (5.5))

$$\mathcal{D} \triangleq \{r_q, t_q; q = 0, 1, ..., N_t - 1\}, \tag{5.12}$$

where[2]

$$t_q \triangleq \phi_q, \tag{5.13}$$

$$r_q = \Delta\hat{\psi}_q \tag{5.14}$$

is an estimate of the phase difference

$$\Delta\psi_q \triangleq \psi_q^{(1)} - \psi_q^{(0)} \tag{5.15}$$

and $\psi_q^{(0)}$ ($\psi_q^{(1)}$) is the phase of the sinusoidal oscillation associated with the considered target and observed on the first (second) RX antenna for any q (see eqs. (2.33)-(2.38) and (2.42)).

5.2.2 The frequentist and Bayesian approach to supervised learning

The frequentist approach relies on the assumption that all the points of the set $\mathcal{D}$ (5.12) are generated *independently* on the basis of the same *unknown* joint pdf $f(\mathbf{r}, \mathbf{t})$, that is

$$\left(\mathbf{r}_q, \mathbf{t}_q\right) \sim f\left(\mathbf{r}, \mathbf{t}\right) = f\left(\mathbf{t}|\mathbf{r}\right) f\left(\mathbf{r}\right), \tag{5.16}$$

with $q = 0, 1, ..., N_t - 1$. Under this assumption, two possible approaches can be adopted to derive the above mentioned predictor $\hat{t}_{\mathcal{D}}(\mathbf{r})$, namely: a) *separate learning* and *inference*; b) direct inference via *empirical risk minimization* (ERM). The first approach consists in learning an approximation, denoted by $f_{\mathcal{D}}(\mathbf{t}|\mathbf{r})$, of the conditional pdf $f(\mathbf{t}|\mathbf{r})$, and in using the former pdf in place of the latter one to derive the expression of the predictor $\hat{t}_{\mathcal{D}}(\mathbf{r})$ on the basis of eq. (5.11). The second approach, instead, aims at directly learning $\hat{t}_{\mathcal{D}}(\mathbf{r})$ by solving the problem

$$\hat{t}_{\mathcal{D}}\left(\mathbf{r}\right) = \arg\min_{\tilde{\mathbf{t}}} L_{\mathcal{D}}\left(\tilde{\mathbf{t}}\left(\mathbf{r}\right)\right), \tag{5.17}$$

where

$$L_{\mathcal{D}}\left(\tilde{\mathbf{t}}\left(\mathbf{r}\right)\right) \triangleq \frac{1}{N_t} \sum_{q=0}^{N_t-1} \ell\left(\mathbf{t}_q, \tilde{\mathbf{t}}\left(\mathbf{r}_q\right)\right) \tag{5.18}$$

is the so called *empirical loss*. In both cases, the optimization of a *set of parameters* characterizing the model selected for the conditional pdf $f_{\mathcal{D}}(\mathbf{t}|\mathbf{r})$ or that chosen for the predictor $\hat{t}_{\mathcal{D}}(\mathbf{r})$ is required. However, the first approach is more flexible than the second one since, in principle, the approximate pdf $f_{\mathcal{D}}(\mathbf{t}|\mathbf{r})$ it generates can be exploited to derive the predictor $\hat{t}_{\mathcal{D}}(\mathbf{r})$ for any cost function; on the contrary, the solution of the problem (5.17) holds for a specific cost function only. Moreover, it should be kept in mind that, if the first approach is adopted, two options are available. The first consists in learning a *discriminative probabilistic model*, i.e. in learning directly an approximation of the

[2]Note that, in this case, $d_q = 1$ and $R_q = R$ in eq. (5.1), so that the label $\mathbf{t}_q$ turns into a scalar.

posterior $f(\mathbf{t}|\mathbf{r})$. On the contrary, the second option consists in learning a *generative probabilistic model*, i.e. in learning the joint pdf $f(\mathbf{t}, \mathbf{r})$ and, then, in deriving an estimate of the posterior $f(\mathbf{t}|\mathbf{r})$ from it.

Considering a frequentist approach, the predictor of the azimuth ϕ_q (5.13) associated with the new observation $\Delta\hat{\psi}_q$ for any $q > N_t - 1$ can be derived considering the discriminative probabilistic model

$$f\left(t|r_q, \mathbf{w}\right) = \mathcal{N}\left(t; \mu\left(r_q, \mathbf{w}\right), \beta^{-1}\right), \tag{5.19}$$

where

$$\mu\left(r_q, \mathbf{w}\right) \triangleq \sum_{j=0}^{\bar{M}} w_j \, \varphi_j\left(r_q\right) = \mathbf{w}^T \boldsymbol{\varphi}\left(r_q\right), \tag{5.20}$$

$\bar{M}$ is the *order* of the model,

$$\mathbf{w} \triangleq \left[w_0, w_1, ..., w_{\bar{M}}\right]^T \tag{5.21}$$

is a vector collecting $\bar{M} + 1$ distinct real parameters (called *weights*),

$$\boldsymbol{\varphi}\left(r_q\right) \triangleq \left[\varphi_0(r_q), \varphi_1(r_q), ..., \varphi_{\bar{M}}(r_q)\right]^T \tag{5.22}$$

is the so called *feature* vector, $\{\varphi_j(x); j = 0, 1, ..., \bar{M}\}$ are $\bar{M} + 1$ non linear functions and β^{-1} is the variance of the noise affecting the labels. In the following, let us assume that

$$\varphi_j(x) = x^j \tag{5.23}$$

for $j = 0, 1, ..., \bar{M}$; consequently, eq. (5.20) becomes

$$\mu(r_q, \mathbf{w}) \triangleq w_0 + \sum_{j=1}^{\bar{M}} w_j \, r_q^j. \tag{5.24}$$

It is worth noting that:

a) Adopting the probabilistic model (5.19) with the mean $\mu(r_q, \mathbf{w})$ (5.24) is amounts to postulating a polynomial dependence of the label ϕ_q on the corresponding observation $\Delta\hat{\psi}_q$.

b) The selected model depends on its order $\bar{M}$ and on the $(\bar{M} + 2)$-dimensional parameter vector $\boldsymbol{\theta} \triangleq \left[\mathbf{w}^T, \beta\right]^T$.

c) The parameter $\bar{M}$ defines the number of degrees of freedom available in the model and, consequently, determines its *bias*.

As far as the last point is concerned, it is worth mentioning that, if $\bar{M}$ is too small, the resulting predictor may *underfit* the observations, since it is unable to accurately represent this dependence on their labels. On the contrary, if $\bar{M}$ is too large, the model is able to account for the observations of the training set, but it may generate inaccurate predictions; in other words, it memorizes the training set, but it is unable to generalise what has learnt to new examples. The last problem is known as *overfitting*. For instance, in the considered problem, good results are obtained if $\bar{M} = 3$ is selected.

If the ERM approach is adopted to adjust the parameters of the probabilistic model (5.19) (and, in particular, the weight vector $\mathbf{w}$ (5.21)) in an optimal fashion, the obtained result depends on the selected cost function and cannot be always put in a closed form. However, if the cost function $\ell_2(t, \hat{t})$ (5.9) is chosen and noise is neglected (i.e., β^{-1} is

assumed to be very small), a closed form expression can be derived for $\hat{\mathbf{w}}$ for any $\bar{M}$. In fact, under these assumptions, it can be proved that:

1. The optimal predictor $\hat{t}_{\mathcal{D}}(\mathbf{r})$ (5.17) becomes (e.g., see [86, Sect. 3.1.1, eq. (3.20)])

$$\hat{t}_{\mathcal{D}}(r_q) = \mu\left(r_q, \hat{\mathbf{w}}\right), \tag{5.25}$$

where

$$\hat{\mathbf{w}} = \arg\min_{\tilde{\mathbf{w}}} L_{\mathcal{D}}\left(\tilde{\mathbf{w}}\right), \tag{5.26}$$

$\tilde{\mathbf{w}}$ denotes a trial value of $\mathbf{w}$ and

$$L_{\mathcal{D}}\left(\tilde{\mathbf{w}}\right) \triangleq \frac{1}{N_t}\sum_{q=0}^{N_t-1}\left(t_q - \mu\left(r_q, \hat{\mathbf{w}}\right)\right)^2 \tag{5.27}$$

is the empirical loss (see eq. (5.18)).

2. The solution of the minimization problem appearing in the RHS of eq. (5.26) is

$$\hat{\mathbf{w}} = \left(\boldsymbol{\Phi}_{\mathcal{D}}^T\boldsymbol{\Phi}_{\mathcal{D}}\right)^{-1}\boldsymbol{\Phi}_{\mathcal{D}}^T\,\mathbf{t}_{\mathcal{D}}, \tag{5.28}$$

where

$$\boldsymbol{\Phi}_{\mathcal{D}} \triangleq [\boldsymbol{\varphi}(r_0), \boldsymbol{\varphi}(r_1), ..., \boldsymbol{\varphi}(r_{N_t-1})] \tag{5.29}$$

is a $N_t \times (\bar{M}+1)$ matrix and

$$\mathbf{t}_{\mathcal{D}} \triangleq [t_0, t_1, ..., t_{N_t-1}]^T. \tag{5.30}$$

Given the weight vector $\hat{\mathbf{w}}$ (5.28), the estimate

$$\hat{\beta}^{-1} \triangleq \frac{1}{N_t}\sum_{q=0}^{N_t-1}\left(t_q - \hat{\mathbf{w}}^T\boldsymbol{\varphi}\left(r_q\right)\right)^2, \tag{5.31}$$

of the noise variance β^{-1} can be easily evaluated.

The Bayesian approach consists in formulating the uncertainty about the parameters of the adopted probabilistic model in statistical terms, i.e. in treating its parameter vector $\boldsymbol{\theta}$ as a *random vector*. Let us assume that each observation and its label are 1D (i.e., $D_t = D_r = 1$), so that all the labels of the training set $\mathcal{D}$ (5.12) and the associated observations can be collected in the N_t-dimensional vectors $\mathbf{t}_{\mathcal{D}}$ (5.30) and

$$\mathbf{r}_{\mathcal{D}} \triangleq [r_0, r_1, ..., r_{N_t-1}]^T, \tag{5.32}$$

respectively. If the discriminative probabilistic model (5.19) introduced in the previous paragraph is exploited, a Bayesian method based on it can be developed as follows. To begin, the joint pdf

$$f\left(t, \mathbf{t}_{\mathcal{D}}, \mathbf{w} | r_q, \mathbf{r}_{\mathcal{D}},\right) = f\left(\mathbf{t}_{\mathcal{D}}, \mathbf{w} | \mathbf{r}_{\mathcal{D}}, \alpha, \beta\right)\, f\left(t | r_q, \mathbf{w}\right) \tag{5.33}$$

is considered in place of the pdf $f(t|r_q, \mathbf{w})$ (5.19); here, β^{-1} is the variance of the noise affecting the labels,

$$f\left(\mathbf{t}_{\mathcal{D}}, \mathbf{w} | \mathbf{r}_{\mathcal{D}}, \alpha, \beta\right) = f\left(\mathbf{t}_{\mathcal{D}} | \mathbf{r}_{\mathcal{D}}, \mathbf{w}, \beta\right)\, f\left(\mathbf{w} | \alpha\right) \tag{5.34}$$

is the joint probability of the $(\bar{M}+1)$-dimensional weight vector $\mathbf{w}$ (5.21) and the label vector $\mathbf{t}_{\mathcal{D}}$ (5.30) conditioned on $\mathbf{r}_{\mathcal{D}}$ (5.32), on the *hyperparameter* α and on the parameter β, and $f(\mathbf{w}|\alpha)$ is the prior pdf of $\mathbf{w}$. The Gaussian model

$$f(\mathbf{w}|\alpha) = \mathcal{N}\left(\mathbf{w}; \mathbf{0}, \alpha^{-1}\mathbf{I}_{\bar{M}+1}\right) = \left(\frac{\alpha}{2\pi}\right)^{(\bar{M}+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right\} \tag{5.35}$$

is employed for the second pdf appearing in the RHS of eq. (5.34) (e.g., see [86, Sect. 1.2.4, p. 30, eq. (1.65)]); here, $\mathbf{I}_N$ is the $N \times N$ unit matrix and α represents the precision of the last pdf. The first pdf appearing in the RHS of eq. (5.34), instead, represents a *likelihood function* expressing how likely the response $\mathbf{t}_{\mathcal{D}}$ are, given $\mathbf{r}_{\mathcal{D}}$, $\mathbf{w}$ and β; this function can be factored as

$$f\left(\mathbf{t}_{\mathcal{D}}|\mathbf{r}_{\mathcal{D}}, \mathbf{w}, \beta\right) = \prod_{k=0}^{N_t-1} f\left(t_k|r_k, \mathbf{w}, \beta\right), \tag{5.36}$$

and, consequently, can be expressed in terms of the probabilistic model (5.19).

Given the joint pdf $f(t, \mathbf{t}_{\mathcal{D}}, \mathbf{w}, |r_q, \mathbf{r}_{\mathcal{D}})$ (5.33), the *predictive distribution* $f(t|r_q, \mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}})$ can be evaluated as

$$f\left(t|r_q, \mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}\right) = \frac{1}{f\left(\mathbf{t}_{\mathcal{D}}|\mathbf{r}_{\mathcal{D}}, \alpha, \beta\right)} \int f\left(t, \mathbf{t}_{\mathcal{D}}, \mathbf{w}|r_q, \mathbf{r}_{\mathcal{D}}\right) d\mathbf{w}, \tag{5.37}$$

where

$$f(\mathbf{t}_{\mathcal{D}}|\mathbf{r}_{\mathcal{D}}, \alpha, \beta) = \int f\left(\mathbf{t}_{\mathcal{D}}, \mathbf{w}|\mathbf{r}_{\mathcal{D}}, \alpha, \beta\right) d\mathbf{w} = \int f\left(\mathbf{t}_{\mathcal{D}}|\mathbf{r}_{\mathcal{D}}, \mathbf{w}, \beta\right) f\left(\mathbf{w}|\alpha\right) d\mathbf{w} \tag{5.38}$$

is a *marginal likelihood*. The expression (5.37) can be also reformulated as follows. Substituting the RHS of eq. (5.34) in that of eq. (5.33) and the resulting factorization in the RHS of eq. (5.37) yields

$$f\left(t|r_q, \mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}\right) = \int \frac{f\left(\mathbf{t}_{\mathcal{D}}|\mathbf{r}_{\mathcal{D}}, \mathbf{w}, \beta\right) f\left(\mathbf{w}|\alpha\right)}{f\left(\mathbf{t}_{\mathcal{D}}|\mathbf{r}_{\mathcal{D}}, \alpha, \beta\right)} \cdot f\left(t|r_q, \mathbf{w}\right) d\mathbf{w}. \tag{5.39}$$

Then, since

$$\frac{f\left(\mathbf{t}_{\mathcal{D}}|\mathbf{r}_{\mathcal{D}}, \mathbf{w}, \beta\right) f\left(\mathbf{w}|\alpha\right)}{f\left(\mathbf{t}_{\mathcal{D}}|\mathbf{r}_{\mathcal{D}}, \alpha, \beta\right)} = f\left(\mathbf{w}|\mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}, \alpha, \beta\right), \tag{5.40}$$

eq. (5.39) can be rewritten as

$$f\left(t|r_q, \mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}\right) = \int f\left(\mathbf{w}|\mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}, \alpha, \beta\right) f\left(t|r_q, \mathbf{w}\right) d\mathbf{w}. \tag{5.41}$$

The last equation shows how the predictive distribution is influenced by the uncertainty about the weight vector; such an uncertainty is expressed by the pdf $f(\mathbf{w}|\mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}, \alpha, \beta)$.

If the pdf $f(\mathbf{w}|\mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}, \alpha, \beta)$ is assumed to be Gaussian and, in particular,

$$f\left(\mathbf{w}|\mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}, \alpha, \beta\right) = \mathcal{N}(\mathbf{w}|\mu_{\mathcal{D}}, \sigma^2_{\mathcal{D}}), \tag{5.42}$$

where (e.g., see [86, Sec. 3.3, p. 153, eqs. (3.53)-(3.54)])

$$\mu_{\mathcal{D}} = \beta\, \sigma^2_{\mathcal{D}}\, \mathbf{\Phi}^T_{\mathcal{D}} \mathbf{t}_{\mathcal{D}}, \tag{5.43}$$

$$\sigma^2_{\mathcal{D}} = \left(\alpha \mathbf{I}_{\bar{M}+1} + \beta \mathbf{\Phi}^T_{\mathcal{D}} \mathbf{\Phi}_{\mathcal{D}}\right)^{-1} \tag{5.44}$$

and the $N_t \times (\bar{M}+1)$ matrix $\mathbf{\Phi}_{\mathcal{D}}$ is given by eq. (5.29), the expression

$$f\left(t|r_q, \mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}\right) = \mathcal{N}(t; \mu(r_q), \sigma^2(r_q)) \tag{5.45}$$

can be derived from eq. (5.41) (e.g., see [86, Sec. 1.2.4, p. 31, eq. (1.69)] for a proof of this result); here,

$$\mu\left(r_q\right) = \beta \boldsymbol{\varphi}\left(r_q\right)^T \mathbf{S} \sum_{k=0}^{N_t-1} \boldsymbol{\varphi}\left(r_k\right)\, t_k, \tag{5.46}$$

$$\sigma^2\left(r_q\right) = \beta^{-1} + \boldsymbol{\varphi}\left(r_q\right)^T \mathbf{S}\, \boldsymbol{\varphi}\left(r_q\right), \tag{5.47}$$

$\boldsymbol{\varphi}(r_q)$ is the $(\bar{M}+1)$-dimensional vector (5.22) and

$$\mathbf{S}^{-1} \triangleq \alpha\, \mathbf{I}_{\bar{M}+1} + \beta \sum_{k=0}^{N_t-1} \boldsymbol{\varphi}\left(r_k\right)\ \boldsymbol{\varphi}\left(r_k\right)^T \tag{5.48}$$

is an $(\bar{M}+1) \times (\bar{M}+1)$ matrix. It is important to underline that the variance $\sigma^2(r_q)$ (5.47) of the predictive distribution $f\left(t|r_q, \mathbf{r}_{\mathcal{D}}, \mathbf{t}_{\mathcal{D}}\right)$ (5.45) (and, consequently, the accuracy of the prediction), unlike that of the Gaussian model $f(t|r_q, \mathbf{w})$ (5.19), is given by the sum of two terms; the first term originates from the noise affecting the labels, whereas the second one from the uncertainty about the parameter vector $\mathbf{w}$. Moreover, the second term is influenced by the considered observation (i.e., it depends on r_q); in practice, smaller values of the standard deviation $\sigma(r_q)$ are usually obtained when r_q is close to the observations of the training set.

Both the two topologies of algorithms illustrated above have been trained on the basis of the available training set $\mathcal{D}$ (5.12). Once the training has been carried out, the generalization capability of the resulting algorithm can be assessed by evaluating the *empirical loss* (5.18) on the basis of a different dataset, called *test set* $\mathcal{D}_{ts}$ and collecting $\bar{N}_t$ observations generated in the same way as the ones of $\mathcal{D}$, but in an independent fashion. In the considered computer simulations, the training set $\mathcal{D}$ (5.12) and the test set $\mathcal{D}_{ts}$ consist of $N_t = 200$ and $\bar{N}_t = 25$ observations, respectively. In generating the considered dataset, the following choices have been made:

a) the distance d between adjacent virtual channels is equal to $\lambda/4$;

b) the target range R is equal to 3.0 m, whereas the target azimuth ϕ_q is uniformly distributed over the interval $[\phi_m,\ \phi_M] = [-60^\circ\ 60^\circ]$, respectively, for any q (this interval is comparable to the horizontal FOV of a real radar system);

c) the amplitude $a_q^{(v)}$ characterizing the sinusoid observed on the v-th virtual antenna is randomly selected in the interval $[0.4, 1.2]$ V for any q (see eq. (2.7));

d) the random variable $a_q^{(v)}$ is independent of $a_p^{(u)}$ for any $u \neq v$ and/or $p \neq q$;

e) the observation r_q (5.14) is generated on the basis of eqs. (2.61) and (5.3), i.e. as $\Delta\hat{\psi}_q = \angle X_q^{(0)}\, (X_q^{(1)})^*$ for any q.

Moreover, the following choices have been made for the parameters of the radar system:

a) the generated frequency modulated waveform is characterized by $\mu = 7.8125 \cdot 10^{12}$ Hz s^{-1}, $T = 256\,\mu$s and $T_R = 64\,\mu$s;

b) the sampling period employed at the receive side is $T_s = 0.25\ \mu$s and $N = 512$ time-domain samples are acquired from each of the two RX antennas;

c) the standard deviation of the noise affecting these samples is $\sigma_w = \sqrt{2}$ V (see eq. (2.7));

d) the oversampling factor $M = 4$ (see eq. (2.58)) and the threshold $P_{th} = 0.5\ \mathrm{V^2Hz^{-1}}$ are employed by the detection algorithm based on eqs. (2.55)-(2.56). The points of these data sets are represented in Figs. 5.1 and 5.2, respectively.

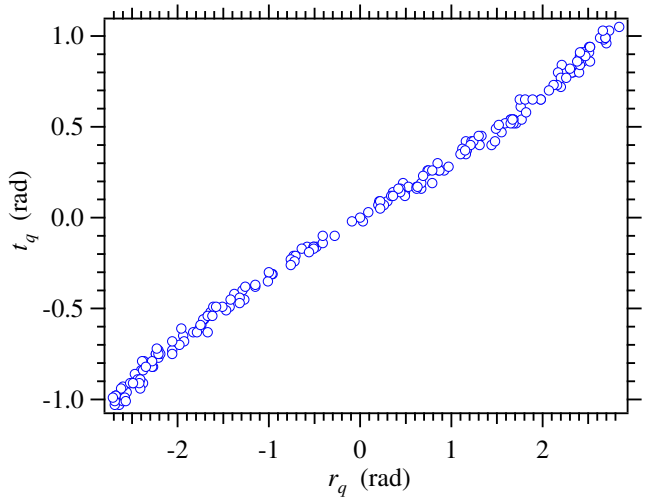


FIGURE 5.1: Representation of the points of the synthetically generated training set $\mathcal{D}$ (5.12); $N_t = 200$ is assumed.

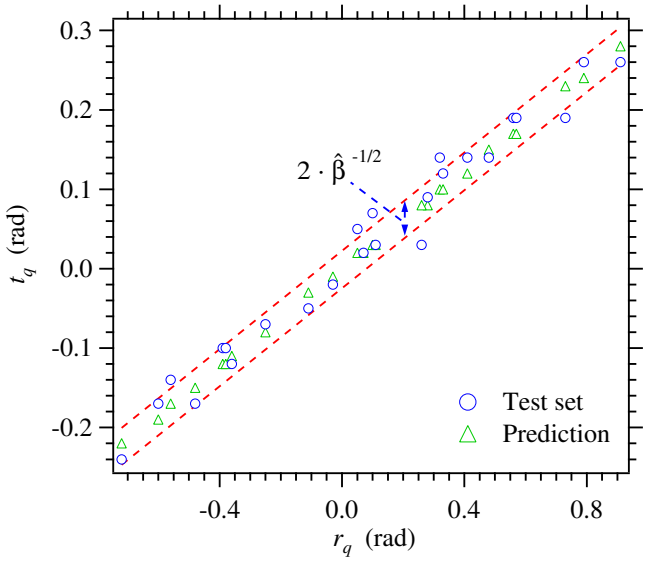


FIGURE 5.2: Representation of the points of the synthetically generated test set $\mathcal{D}_{ts}$ (blue circles) and of the corresponding predictions (green triangles) evaluated on the basis of eq. (5.25); $\bar{N}_t = 25$ is assumed. Two straight lines, expressed by eq. (5.49), are also shown.

First, the weight vector $\hat{\mathbf{w}}$ (5.28) and the estimate $\hat{\beta}^{-1}$ (5.31) of the noise variance have been computed on the basis of $\mathcal{D}$. Then, the accuracy of the resulting regression algorithm has been assessed on $\mathcal{D}_{ts}$. The predictions obtained through the frequentist approach associated with the points of $\mathcal{D}_{ts}$ are represented in Fig. 5.2; in this figure, two (red) straight lines, generated on the basis of the linear equations

$$t = \mu(r, \hat{\mathbf{w}}) \pm \hat{\beta}^{-1/2}, \tag{5.49}$$

are also shown to highlight the meaning of the noise standard deviation $2\hat{\beta}^{-1/2}$. These results lead to the conclusion that, in the considered scenario, the developed regression

method is able to predict the azimuth of a target with good accuracy. This is confirmed by the fact that the empirical loss computed over the set $\mathcal{D}_{ts}$ (i.e., the *generalization loss*) is close to the empirical loss $L_{\mathcal{D}}(\hat{\mathbf{w}})$ evaluated over the set $\mathcal{D}$ (see eq. (5.18)); in fact, the *root mean square error*[3] (RMSE) evaluated over $\mathcal{D}$ is equal to[4] 1.7°, whereas that computed over $\mathcal{D}_{ts}$ is equal to 1.3°.

The accuracy of the Bayesian regression algorithm described above has been assessed on the test set shown in Fig. 2.6 after training it on the set illustrated in Fig. 2.5; moreover, $\alpha = 0.05$ has been selected in this case. The prediction $\mu(r_q)$ evaluated on the basis of eq. (5.46) for each observation of the test set and the corresponding standard deviation $\sigma(r_q)$ (i.e., the square root of the RHS of eq. (5.47)) are represented in Fig. 5.3. The RMSE evaluated over the test set is equal to 1.4° and is approximately equal to that one computed over the training set. Note that this value is comparable to those ones computed for the predictor described in the previous paragraph (and based on a frequentist approach). For this reason, in this case, the Bayesian approach does not offer any advantage with respect to the frequentist one.

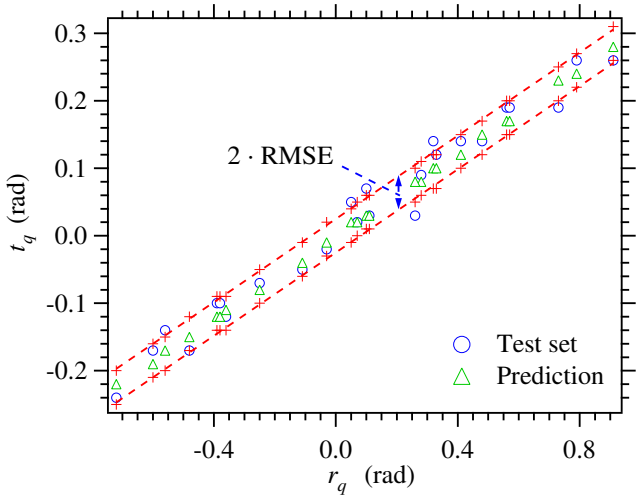


FIGURE 5.3: Representation of the regression technique based on the probabilistic model (5.45). The blue circles represent the true domain points, whereas the green triangles the corresponding predictions; the red curves are generated by interpolating the points generated on the basis of the two equations $t_q = \mu(r_q) \pm \sigma(r_q)$, with $r_q \in \mathcal{D}_{ts}$.

In general, a specific discriminative regression problem is represented by the parametric pdf $f(\mathbf{t}|\mathbf{r}, \boldsymbol{\theta})$, a closed form expression for the optimal value

$$\hat{\boldsymbol{\theta}} = \arg\min_{\tilde{\boldsymbol{\theta}}} L_{\mathcal{D}}\left(\tilde{\boldsymbol{\theta}}\right), \tag{5.50}$$

of the D_θ-dimensional parameter vector $\boldsymbol{\theta}$ is unavailable in most cases. When this occurs, iterative optimization techniques, like the *stochastic gradient descent* (SGD) method, can be employed to estimate $\hat{\boldsymbol{\theta}}$. The application of the SGD to the considered problem leads easily to the recursive equation

$$\hat{\boldsymbol{\theta}}^{(i+1)} = \hat{\boldsymbol{\theta}}^{(i)} + \gamma^{(i+1)} N_S^{-1} \sum_{q \in \mathcal{S}} \nabla_{\tilde{\boldsymbol{\theta}}} \ell\left(\mathbf{t}_q, \hat{\mathbf{t}}\left(\mathbf{r}_q, \tilde{\boldsymbol{\theta}}\right)\right)\Big|_{\tilde{\boldsymbol{\theta}}=\hat{\boldsymbol{\theta}}^{(i)}}, \tag{5.51}$$

[3] This parameter represents the square root of the empirical loss.
[4] The RMSE computed over $\mathcal{D}$ is given by $\hat{\beta}^{-1/2}$.

with $i = 0, 1, ..., N_E - 1$; here, $\hat{\boldsymbol{\theta}}^{(i)}$ denotes estimate of $\hat{\boldsymbol{\theta}}$ computed in the i-th recursion, $\mathcal{S}$ is a set of N_S integers randomly selected in the set $\{0, 1, ..., N_t - 1\}$ (with $N_S < N_t$), $\gamma^{(i)}$ is the *learning rate* adopted in the i-th iteration and $\nabla_{\mathbf{x}} f(\mathbf{x})$ denotes the gradient of the function $f(\mathbf{x})$.

It can be proved that, if the learning rate schedule (i.e., the sequence $\{\gamma^{(i)}\}$) satisfies the so called *Robbins-Monro conditions*, the SGD converges to the optimal solution, provided that the function $L_{\mathcal{D}}(\tilde{\boldsymbol{\theta}})$ is strictly convex. The initial value $\hat{\boldsymbol{\theta}}^{(0)}$ can be either randomly selected or it can be inherited from the training procedure accomplished another model; the last solution represents a specific application of the so called *transfer learning* principle. Iterations are stopped when negligible variations are observed in the estimates generated by consecutive recursions or an upper limit set on the overall number of recursions is reached. Once the final estimate of $\hat{\boldsymbol{\theta}}$ has been computed from the available training set, the generalization capability of the resulting algorithm can be assessed by evaluating the *empirical loss* (5.18) on a given *test set* $\mathcal{D}_{ts}$.

Finally, it is worth mentioning that the selection of the parameter M (i.e., of the *model complexity*) plays a fundamental role. In fact, if its value is too small (too large), the resulting regression method can suffer from underfitting (overfitting). Overfitting is usually prevented by including a *regularization* term in the training of the adopted model. For instance, if the optimization problem (5.50) is considered, this result can be achieved by adopting the cost function

$$L_D\left(\tilde{\boldsymbol{\theta}}\right) + \frac{\lambda}{N_t}\left\|\tilde{\boldsymbol{\theta}}\right\|^2, \tag{5.52}$$

where λ is a real positive weight influencing the predictive capability of the resulting solution and $||\mathbf{x}||$ is the *Euclidean norm* of the vector $\mathbf{x}$.

5.2.3 Specific methods for binary classification

In the remaining part of this section the focus is moved on a specific supervised problem, namely *binary classification*; two classification methods, based on *discriminative deterministic models*, are proposed to solve it. Moreover, it is shown how different classification methods can be combined to improve the overall accuracy. Note that, in general, classification methods based on *discriminative deterministic models* are able to represent the deterministic mapping between domain points and labels through specific functions, called *discriminant functions*. In the field of radar systems, these methods can be exploited for target detection.

The first method taken into consideration in this paragraph is the *support vector machine* (SVM) technique; in the following, the analysis is limited to its *linear form*, for simplicity, and the label of each observation can take on only the values ± 1 (consequently, $D_t = 1$). The SVM technique processes the training set $\mathcal{D}$ (5.5) to find the *maximum-margin hyperplane*; this divides the subset of observations for which $t_q = 1$ from that for which $t_q = -1$ in a way that the distance between itself and the nearest point from either group is maximized. In the considered case, the above mentioned hyperplane can be defined as the set of points satisfying the equation

$$y(\mathbf{r}_q, \mathbf{w}) = 0 \tag{5.53}$$

for any q, where

$$y(\mathbf{r}_q, \mathbf{w}) \triangleq \mathbf{w}^T \mathbf{r}_q + b\,, \tag{5.54}$$

$\mathbf{r}_q$ is the q-th observation of $\mathcal{D}$ (see the definition (5.6)), $\mathbf{w}$ represents a D_r-dimensional weight vector (expressed by eq. (5.21), with $\bar{M} = D_r - 1$) and b is a real parameter called *bias*. The adoption of a classification strategy based on the approach illustrated above relies on the implicit assumption that, if the parameters $\mathbf{w}$ and b appearing in eq. (5.54) are properly selected, the dataset $\mathcal{D}$ (5.5) is *linearly separable in the feature space*. In fact, when this occurs, two parallel hyperplanes separating the above mentioned two subsets of observations and having their mutual distances as large as possible can be found. If the observations of the set $\mathcal{D}$ are normalised, the hyperplanes delimiting the subsets of observations associated with $t_q = 1$ and $t_q = -1$ can be represented by the equations

$$y(\mathbf{r}_q, \mathbf{w}) = 1 \tag{5.55}$$

and

$$y(\mathbf{r}_q, \mathbf{w}) = -1, \tag{5.56}$$

respectively, i.e. briefly as

$$t_q \, y(\mathbf{r}_q, \mathbf{w}) = 1. \tag{5.57}$$

The last formula expresses the *canonical representation* of the decision hyperplanes. Based on the last result, the constraint according to which each point of the set $\mathcal{D}$ (5.5) must lie on the correct side of each of the two hyperplanes (i.e., that it must fall in the correct decision region) can be formulated as

$$t_q \, y(\mathbf{r}_q, \mathbf{w}) \geq 1 \tag{5.58}$$

for any q.

A method for the optimization of the parameters b and $\mathbf{w}$ appearing in eq. (5.54) can be developed as follows. The perpendicular distance of the point $\mathbf{r}_q$ from the decision hyperplane can expressed as

$$\frac{t_q \, y(\mathbf{r}_q)}{||\mathbf{w}||} = \frac{t_q \left(\mathbf{w}^T \mathbf{r}_q + b\right)}{||\mathbf{w}||} \tag{5.59}$$

for any q; its minimum value over the set $\mathcal{D}$ is known as *margin*. The optimal choice $(\hat{\mathbf{w}}, \hat{b})$ of the parameters $(\mathbf{w}, b)$ is the one *maximizing the margin* and, consequently, can be evaluated as

$$\left(\hat{\mathbf{w}}, \hat{b}\right) = \arg\max_{\tilde{\mathbf{w}}, \tilde{b}} \left\{ \frac{1}{||\tilde{\mathbf{w}}||} \min_q \left[t_q (\tilde{\mathbf{w}}^T \mathbf{r}_q + \tilde{b}) \right] \right\}, \tag{5.60}$$

where $(\tilde{\mathbf{w}}, \tilde{b})$ denotes a trial value of the pair $(\mathbf{w}, b)$; the data points closest to the max-margin hyperplane are called *support vectors*. Unluckily, the optimization problem appearing in the RHS of eq. (5.60) does not admit a simple solution. However, since there is always at least one support vector satisfying eq. (5.57), this problem can be reformulated in a simpler form, i.e. as the maximization of $||\tilde{\mathbf{w}}||^{-1}$ or, equivalently, as

$$\hat{\mathbf{w}} = \arg\min_{\tilde{\mathbf{w}}} \frac{||\tilde{\mathbf{w}}||^2}{2} \tag{5.61}$$

under the constraint expressed by eq. (5.58); note that the parameter $\tilde{b}$ is no more visible in the last formulation, but its value is implicitly determined by the above mentioned constraint. To solve the constrained optimization problem (5.61), the *Lagrangian* function

$$\mathcal{L}\left(\tilde{\mathbf{w}}, \tilde{b}, \tilde{\mathbf{a}}\right) \triangleq \frac{||\tilde{\mathbf{w}}||^2}{2} - \sum_{q=0}^{N_t-1} \tilde{a}_q \{t_q(\tilde{\mathbf{w}}^T \mathbf{r}_q + \tilde{b}) - 1\}. \tag{5.62}$$

is defined; this function depends not only on the parameters $\tilde{\mathbf{w}}$ and $\tilde{b}$, but also on the non negative parameters $\{\tilde{a}_q\}$, called *Lagrange multipliers* and collected in the vector $\tilde{\mathbf{a}} \triangleq [\tilde{a}_0, \tilde{a}_1, ..., \tilde{a}_{N_t-1}]^T$ (the q-th element of this vector is associated with the q-th constraint expressed by eq. (5.58)). Taking the partial derivatives of the function $\mathcal{L}(\tilde{\mathbf{w}}, \tilde{b}, \tilde{\mathbf{a}})$ (5.62) with respect to $\tilde{\mathbf{w}}$ and $\tilde{b}$ and setting them to zero results in

$$\sum_{q=0}^{N_t-1} \tilde{a}_q \, t_q = 0 \tag{5.63}$$

and

$$\tilde{\mathbf{w}} = \sum_{q=0}^{N_t-1} \tilde{a}_q \, t_q \, \mathbf{r}_q, \tag{5.64}$$

respectively. Then, substituting eqs. (5.63)-(5.64) in the RHS of eq. (5.62) produces the so called *dual representation* of the margin maximization problem. Solving the last problem requires maximizing the function

$$\mathcal{L}\left(\tilde{\mathbf{a}}\right) \triangleq \sum_{q=0}^{N_t-1} \tilde{a}_q - \frac{1}{2} \sum_{q=0}^{N_t-1} \sum_{k=0}^{N_t-1} \tilde{a}_q \, \tilde{a}_k \, t_q \, t_k (\mathbf{r}_q^T \, \mathbf{r}_k) \tag{5.65}$$

with respect to the vector $\tilde{\mathbf{a}}$, under the set of constraints $\{\tilde{a}_q \geq 0 \text{ for any } q\}$ and the constraint expressed by eq. (5.63) and produces the optimal value $\hat{\mathbf{a}}$ of the vector $\tilde{\mathbf{a}}$. Given $\hat{\mathbf{a}}$, the optimal values $\hat{\mathbf{w}}$ and $\hat{b}$ of $\tilde{\mathbf{w}}$ and $\tilde{b}$, respectively, are computed as (see eq. (5.64))

$$\hat{\mathbf{w}} = \sum_{q=0}^{N_t-1} \hat{a}_q \, t_q \, \mathbf{r}_q, \tag{5.66}$$

and

$$\hat{b} = N_{\mathcal{S}_M}^{-1} \sum_{q \in \mathcal{S}_M} \left(t_q - \sum_{k \in \mathcal{S}_M} \hat{a}_k \, t_k \, \mathbf{r}_q^T \, \mathbf{r}_k \right), \tag{5.67}$$

respectively, where $\mathcal{S}_M$ and $N_{\mathcal{S}_M}$ denote the set of support vectors and its cardinality, respectively. Given $(\hat{\mathbf{w}}, \hat{\mathbf{a}}, \hat{b})$, the classification of a new data point $(\mathbf{r}_q, t_q)$ (with $q > N_t - 1$) is accomplished on the basis of the sign of the quantity (see eq. (5.54))

$$y(\mathbf{r}_q, \hat{\mathbf{w}}) \triangleq \hat{\mathbf{w}}^T \, \mathbf{r}_q + \hat{b}, \tag{5.68}$$

that can be also expressed as (see eq. (5.66))

$$y(\mathbf{r}_q) = \sum_{k=0}^{N_t-1} \hat{a}_k \, t_k \, \mathbf{r}_q^T \, \mathbf{r}_k + \hat{b}. \tag{5.69}$$

As mentioned above, this classification method is derived under the assumption that the set of feature vectors $\{\mathbf{r}_q\}$ is linearly separable. When this does not occur, a specific *kernel function*, denoted by $\boldsymbol{\phi}(\cdot)$, can be used to map the vector $\mathbf{r}_q$ (5.6) into the new feature vector $\boldsymbol{\phi}(\mathbf{r}_q)$ for any q (e.g., see [86, Chap. 6]). The objective is transforming the available classification space into a one characterized by linear boundaries; in principle,

the dimensionality of $\boldsymbol{\phi}(\cdot)$ may differ from D_r. Well known examples of the kernels employed with the SVM method are the polynomial, the Gaussian and the Laplace kernels. It is important to note that kernel selection is very critical, since its choice can significantly influence classification accuracy.

The second method taken into consideration is the so called *K nearest-neighbour* (K-NN) technique [57], that represents an example of non-parametric approach to the classification problem. In the case of binary classification, it can be summarised as follows. The points of the training set $\mathcal{D}$ (5.5) are partitioned into two classes, denoted by $\mathcal{C}_0$ and $\mathcal{C}_1$, where

$$\mathcal{C}_k \triangleq \{(\mathbf{r}_{q_k}, t_{q_k}); q_k = 0, 1, ..., N_k - 1\}, \tag{5.70}$$

with $k = 0$ and 1, and N_k denotes the number of points belonging to the k-th class, so that

$$\sum_{k=0}^{1} N_k = N_t. \tag{5.71}$$

Let us assume now that a new D_r-dimensional observation, denoted by $\mathbf{r}_q$ (with $q > N_t - 1$) and called *query instance*, becomes available. The K-NN strategy classifies $\mathbf{r}_q$, i.e. assigns it to one of the two classes defined above, on the basis of the votes of its K *nearest neighbours* (i.e., of the K points of $\mathcal{D}$ closest to $\mathbf{r}_q$); here, K is an integer parameter, whose value is usually small and odd. The identification of the nearest neighbours unavoidably requires the computation of the distance of $\mathbf{r}_q$ from all the points of the set $\mathcal{D}$; if the Euclidean distance is employed, the distance of $\mathbf{r}_q$ from $\mathbf{r}_t \in \mathcal{D}$ is given by

$$d_q \triangleq ||\mathbf{r}_t - \mathbf{r}_q||, \tag{5.72}$$

with $t = 0, 1, ..., N_t - 1$. Given the set $\{d_q\}$, consisting of N_t distances, the nearest neighbours $\{\mathbf{r}_{\mathrm{nn},j}; j = 0, 1, .., K - 1\}$ are identified by searching for the K points of $\mathcal{D}$ that satisfy the inequality

$$d_q < V_q \tag{5.73}$$

where V_q is a fixed threshold, such that all the required K points are found. Then, if K_k denotes the number of nearest neighbours belonging to $\mathcal{C}_k$ (i.e., the number of *representatives* of $\mathcal{C}_k$), $\mathbf{r}_q$ is assigned to the class having the largest number of representatives, i.e. to $\mathcal{C}_0$ ($\mathcal{C}_1$) if $K_0 > K_1$ ($K_1 > K_0$).

It is worth pointing out that the parameter K controls the degree of smoothing, i.e. the size of the regions assigned to each class. In fact, a small value of K usually results in many small regions assigned to each class, whereas a large one leads to fewer larger regions [86, Par. 2.5.2]. Moreover, if $K = 1$ is selected, a *nearest-neighbour* classifier is obtained; in this case, if the dataset is quite large, it can be shown that the error rate of a K-NN classifier is never larger than twice the minimum achievable error rate of an optimal classifier[5], i.e. of a classifier having full knowledge of the pdf of the observations [103].

Multiple classification methods can be combined to improve the overall accuracy; this idea leads to the development of the so called *ensemble classifiers* [86, Ch. 14.2]. Specific examples of these classifiers are represented by the so called *bootstrap aggregating* (also known as *bagging* [104]) and *boosting* methods [86]. The first method can be employed when M_B predictions, denoted by $\{y^{(m)}(\mathbf{r}_q); m = 0, 1, ..., M_B - 1\}$ and generated

[5]The optimal classification strategy can be easily formulated on the basis of eq. (5.11) (see paragraph 5.2.1).

by M_B different classifiers (called *base classifiers*), are available; the output is computed as

$$Y_M \triangleq \frac{1}{M_B} \sum_{m=0}^{M_B-1} y^{(m)}\left(\mathbf{r}_q\right), \tag{5.74}$$

i.e. as an average of all the above mentioned predictions and the predicted class is identified by the sign of this quantity; this reduces the impact of the error due to each single classifier M_B times. This method is really effective when the errors originating from distinct classifiers are uncorrelated; unluckily, in some cases, such errors may be significantly correlated. When this occurs, classification accuracy can be improved through boosting and, in particular, through the *adaptive boosting* method, also known as *AdaBoost* [105]. In fact, the AdaBoost technique can achieve good accuracy even if its M_B base classifiers do not perform well (say, their behaviour is only slightly better than random), i.e. they are *weak learners*. If a *binary classification problem* is considered, the training phase of this method evolves through M_B classification stages, each involving a distinct base classifier; moreover, this method is initialised assigning the same weight to all the observations, i.e. setting $\tilde{w}_q^{(0)} = 1/N_t$ for any q, where $\tilde{w}_q^{(0)}$ denotes the initial weight assigned to the q-th observation. The m-th stage (with $m = 0, 1, ..., M_B - 1$) evolves through the following steps:

1) The m-th base classifier is trained to minimise the weighted error function

$$J^{(m)} \triangleq \sum_{q=0}^{N_t-1} \tilde{w}_q^{(m)} I(y^{(m)}(\mathbf{r}_q)), \tag{5.75}$$

where

$$I\left(y^{(m)}\left(\mathbf{r}_q\right)\right) \triangleq \begin{cases} 1 \text{ if } \; y^{(m)}\left(\mathbf{r}_q\right) \neq t_q \\ 0 \quad \text{otherwise} \end{cases} \tag{5.76}$$

and $\{\tilde{w}_q^{(m)}\}$ is a set of non negative weights such that

$$\sum_{q=0}^{N_t-1} \tilde{w}_q^{(m)} = 1. \tag{5.77}$$

2) The weighted measure of the *error rate*

$$\varepsilon^{(m)} \triangleq \frac{\sum_{q=0}^{N_t-1} \tilde{w}_q^{(m)} I\left(y^{(m)}\left(\mathbf{r}_q\right)\right)}{\sum_{q=0}^{N_t-1} \tilde{w}_q^{(m)}} \tag{5.78}$$

and the *weighting coefficient* (e.g., see [86, Par. 14.3, eq. (14.16)])

$$\alpha^{(m)} \triangleq \ln\left(\frac{1-\varepsilon^{(m)}}{\varepsilon^{(m)}}\right) \tag{5.79}$$

are computed.

3) The weight assigned to the q-th data point is updated using the recursive formula (e.g., see [86, Par. 14.3, eq. (14.18)])

$$\tilde{w}_q^{(m+1)} = \tilde{w}_q^{(m)} \exp\left(\alpha^{(m)} I\left(y^{(m)}\left(\mathbf{r}_q\right)\right)\right) \tag{5.80}$$

for any q.

These steps force the classifier employed in each stage to put more emphasis on

those points that have been misclassified by previous classifiers. In fact, a higher error rate entails a larger increase in the weight assigned to the q-th observation (see eqs. (5.79) and (5.80)), provided that it has not been correctly classified (i.e., that $I(y^{(m)}(\mathbf{r}_q)) = 1$). The final prediction generated by the AdaBoost technique is

$$Y_{B_M}(\mathbf{r}_q) = \text{sign}\left(\sum_{m=0}^{M_B-1} \alpha^{(m)}\, y^{(m)}(\mathbf{r}_q)\right). \tag{5.81}$$

In assessing the accuracy of any classification method, $\bar{N}$-fold cross validation can be used when the size of the available dataset is not so large. This consists in:

a) randomly partitioning the available dataset into $\bar{N}$ blocks;

b) assessing the classification accuracy on the n-th block (taken as test set) after that the considered method has been trained on the basis of the remaining $(\bar{N}-1)$ blocks (with $n = 0, 1, ..., \bar{N}-1$).

At the end of this procedure, $\bar{N}$ distinct accuracies are available; the final score is expressed by their average.

Let us focus now on a specific application of the SVM and K-NN techniques to an FCMW radar system equipped with the antenna array shown in Fig. 2.8-b) (and characterized by $d = \lambda/4$) and operating in the presence of *at most a single point target*. In the q-th trial, the set $\{\mathbf{x}_q^{(0)}, \mathbf{x}_q^{(1)}, \mathbf{x}_q^{(2)}, \mathbf{x}_q^{(3)}\}$, consisting of four N-dimensional noisy vectors, each associated with one of the $N_V = 4$ virtual receive channels, is available for any q (see eq. (2.48)). The 4-*dimensional* (4D) feature vector

$$\mathbf{R}_q = \left[R_q^{(0)}, R_q^{(1)}, R_q^{(2)}, R_q^{(3)}\right]^T \triangleq \left[\left|X_q^{(0)}\right|, \left|X_q^{(1)}\right|, \left|X_q^{(2)}\right|, \left|X_q^{(3)}\right|\right]^T; \tag{5.82}$$

can be considered in place of the noisy channels; here,

$$X_q^{(v)} = M\; X_{q,\hat{l}}^{(v)} \tag{5.83}$$

and $X_{q,\hat{l}}^{(v)}$ is computed from eqs. (2.51), (2.55) and (2.58), i.e. by sampling the spectrum $\bar{X}_q^{(v)}(f)$ (2.46) of the zero-padded sample sequence acquired on the v-th virtual antenna at the target frequency $\hat{f}_m$ (2.57) (with $v = 0, 1, 2$ and 3). The target detection strategy adopted in this case differs from that presented in one illustrated in paragraph 2.5 and based on the computation of the average power

$$P_q \triangleq N_V^{-1} \sum_{v=0}^{N_V-1} |X_q^{(v)}|^2, \tag{5.84}$$

and on its comparison with a threshold (see eqs. (2.54) and (2.56)). This choice is motivated by the fact that, the amplitude $a_q^{(v)}$ of the sinusoid observed on the v-th virtual channel and associated with the detected point target is assumed to depend on the antenna index[6] v; the last assumption allows to account for: a) the dependence of the target reflectivity on the direction of observation; b) the differences in the amplifications introduced by distinct receive chains of the employed MIMO radar. In fact, in the considered

[6]For this reason, the assumption we made in writing eq. (2.7) does not hold any more.

radar system, a target is detected if the inequality

$$\max_{v\in\{0,1,2,3\}} \left| X_q^{(v)} \right|^2 > P_{da}, \tag{5.85}$$

holds, i.e. if $|X_q^{(v)}|^2 > P_{da}$ for at least a single value of v.

In the considered experiment, the training set

$$\mathcal{D} \triangleq \{(\mathbf{R}_q, t_q); q = 0, 1, ..., N_t - 1\}, \tag{5.86}$$

referring $N_t = 100$ independent trials, has been synthetically generated. Half of its data are associated with the detection of a *real* target, the remaining half with the detection of a *false* target; for this reason, the vector $\mathbf{R}_q$ (5.82) is labelled by $t_q = 1$ (-1) in the presence of a *real* (*false*) target. Moreover, the following assumptions have been made in generating the q-th observation of the training set $\mathcal{D}$ and the test set $\mathcal{D}_{ts}$:

a) The amplitude $a_q^{(v)}$ of the sinusoid observed on the v-th antenna in the presence of a real target is uniformly distributed over the interval $[0, 1]$ V;

b) The random variable $a_q^{(v)}$ is independent of $a_p^{(u)}$ for any $u \neq v$ and/or $p \neq q$.

c) The range R_q and the azimuth ϕ_q of the target (if present) are uniformly distributed over the intervals $[R_m\,,\, R_M] = [1.0 \text{ m}, 5.0 \text{ m}]$ and $[\phi_m\,,\, \phi_M] = [-60^\circ, 60^\circ]$, respectively, for any q.

The values selected for most of the parameters of the considered radar system are equal to those listed in the examples of paragraph 5.2.2, the only differences being represented by the fact that:

a) the standard deviation of the noise affecting the received signal samples is $\sigma_w = 1.0$ V (see eq. (2.7));

b) the threshold $P_{da} = 0.3$ V^{2}Hz^{-1} is employed by the detection algorithm based on eq. (5.85).

The dataset $\mathcal{D}$ (5.86) has been employed to train the linear SVM, K-NN and Adaboost techniques; $K = 4$ (this number is the best found for K, but also other odd number are possible) and $M_B = 100$ has been selected for the second classifier and the third one, respectively. Moreover, the weak learner employed in the m-th step of the Adaboost technique consists in comparing one of the components of the vector $\mathbf{R}_q$ (5.82) with a threshold[7]. More specifically, the classification criterion adopted by each weak learner can be expressed as

$$R_q^{(v)} \underset{t_q=-1}{\overset{t_q=1}{\gtrless}} P^{(v)} \tag{5.87}$$

where the index v is randomly selected in the set $\{0, 1, 2, 3\}$ and $P^{(v)} \sim \mathcal{U}(\min_q(R_q^{(v)}), \max_q(R_q^{(v)}))$ is the decision threshold associated with the v-th feature $R_q^{(v)}$ acquired in the q-th trial. Note that the classification criterion (5.87) leads to partitioning the observation space into two regions, separated by a hyperplane (perpendicular to one of the reference axes).

In this case, the aim of the three classifiers is discriminating between the presence of a real target and that of a false target any time a target is detected; for this reason, they are exploited to reduce the false alarm probability. Some numerical results are shown in

[7]This simple classifier can be interpreted as a form of a decision tree known as *decision stump* and characterized by a single node (e.g., see [86, Ch. 14.3-14.4]

Figs. 5.4, 5.5 and 5.6, that refer to SVM, K-NN and Adaboost, respectively; in all these figures, the set of points[8] $\{(R_q^{(0)}, R_q^{(1)})\}$ extracted from the dataset $\mathcal{D}$ (5.86) are represented on a Cartesian plane and are identified by a green (blue) circle if associated with a false (real) target.

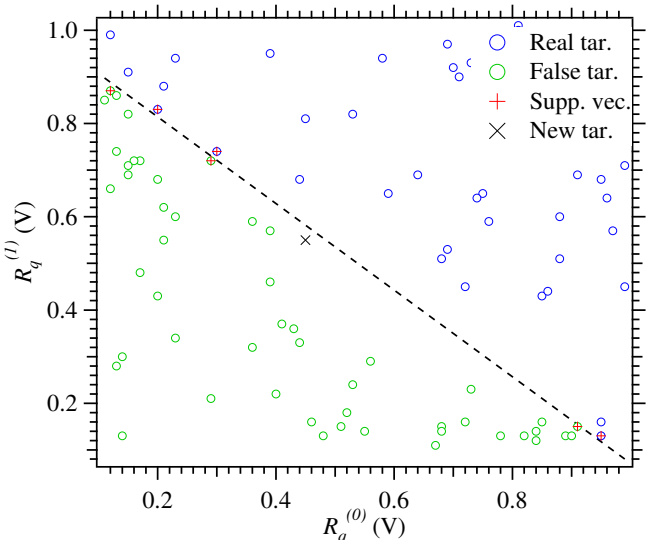


FIGURE 5.4: Representation of the decision mechanism employed by a linear SVM classifier. The points of the training set corresponding to false (real) targets are identified by the green (blue) circles. The decision boundary of the SVM is represented by a dashed line, whereas the red crosses identify support vectors. A new observation, identified by a black cross, is classified as a false target, since it falls in the lower half plane delimited by the decision boundary.

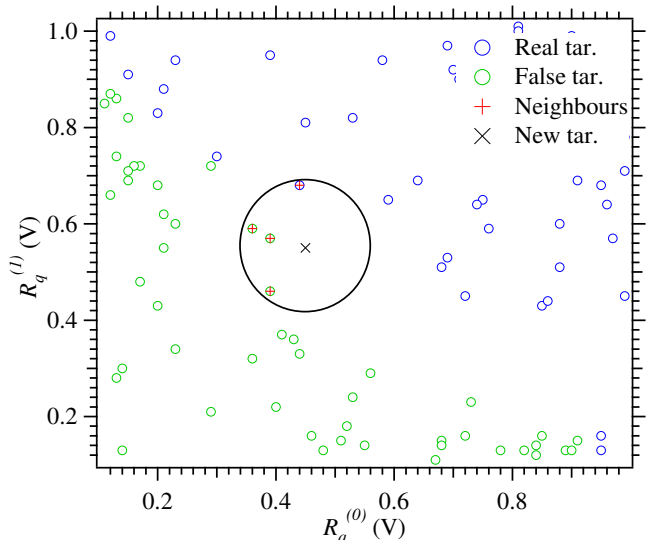


FIGURE 5.5: Representation of the decision mechanism employed by a K-NN classifier (with $K = 4$). The points of the training set corresponding to false (real) targets are identified by the green (blue) circles. A new observation, identified by a black cross, is classified as a false target, since class $\mathcal{C}_1$ is the one having the largest number of representatives contained in the black circle.

[8]Note that the observations of the dataset belong to a 4D space in this case; for this reason, all their components cannot be represented in the same figure.

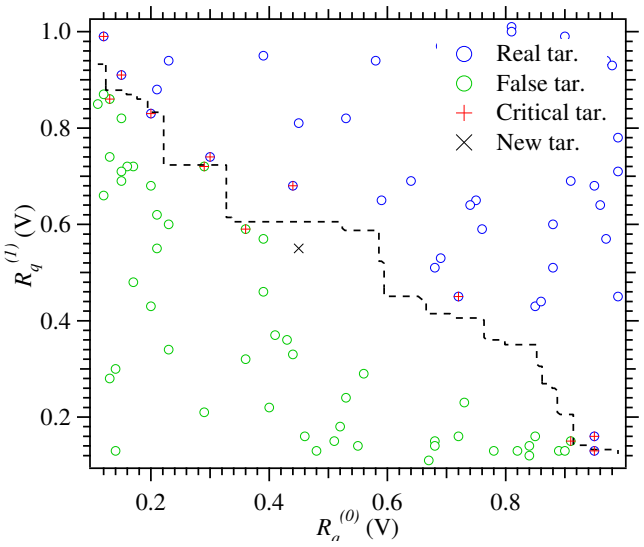


FIGURE 5.6: Representation of the decision mechanism employed by the Adaboost classifier. The points of the training set corresponding to false (real) targets are identified by the green (blue) circles. The decision boundary of the Adaboost is represented by a dashed line, whereas the red crosses identify critical targets. A new observation, identified by a black cross, is classified as a false target, since it falls in the lower region delimited by the decision boundary.

These results deserve the following comments:

1) SVM training leads to generating the linear decision boundary shown in Fig. 5.4; in this figure, a new observation is classified as a false target, since it falls in the lower decision region.

2) The K-NN method classifies the new observation shown in Fig. 5.5 as a false target, since class $\mathcal{C}_1$ is the one having the largest number of representatives contained in the black circumference (having radius equal to $V_q = 0.07$ V and centered at the new observation).

3) Adaboost training leads to generating the decision boundary shown in Fig. 5.6. In the same figure, the *critical points* of the base classifiers (i.e., their misclassified points) are also shown; as it can be easily inferred from eq. (5.80), their weights of these points tend to increase with iterations.

In the same figure, a new observation is classified as a false target, since it falls in the lower decision region. In the considered scenario, our computer simulations have evidenced that the accuracy achieved by the considered classification techniques is around 90%, assuming N-fold cross validation with $N = 5$; in particular, the accuracies of SVM, K-NN and Adaboost are 91%, 89% and 93%, respectively. The binary classification methods illustrated above can be also exploited to develop solutions to multi-class problems; in fact, in general, any problem of this type can be represented as a sequence of binary classification problems [106]. This approach is exemplified in paragraph 5.4.1 and, in more detail, in paragraph 5.6.1, where its application to the classification of three human activities is illustrated. In particular, in paragraph 5.6.1, it is shown how a specific solution to this problem can be devised by exploiting *pairwise classification* (also known as *round-robin* class binarization). If K denotes the overall number of classes, this classification method is based on a) combining $L = K(K-1)/2$ binary classifiers (called *base* learners) and b) using the so called *one-versus-one* coding scheme. In this case, each binary classifier is trained assuming one class as positive, another class as negative (the labels associated with the q-th observation are $t_q = 1$ and $t_q = -1$ for the first class

and the second one, respectively), and ignoring all the other classes (the label associated with this case is $t_q = 0$). When a new observation is available, it is processed by each binary classifier, so that all its possible assignments to every class pair are taken into consideration. This procedure leads to generating a codeword of size L for each class; the l-th element of this codeword can take on the values ± 1 or 0 on the basis of the class membership established by the l-th binary learner (with $l = 0, 1, ..., L-1$). The K codewords produced by the L learners in response to the q-th observation represent the rows of the $K \times L$ *design matrix* $\mathbf{T}_q = [t_{k,l}^{(q)}]$; note that the presence of a '0' on the k-th row of the l-th column for any q means that all the observations associated with the k-th class are ignored by the l-th classifier. The class predicted for the q-th observation is the one minimizing the average of the binary losses over the L different binary learners [107]; in practice, the value of the class index for the q-th observation is computed as

$$\hat{k}_q = \arg \min_{k \in \{0,1,...,K-1\}} \frac{\sum_{l=0}^{L-1} |t_{k,l}^{(q)}| \, g\left(t_{k,l}^{(q)}, y_{q,l}\right)}{\sum_{l=0}^{L-1} |t_{k,l}^{(q)}|}, \tag{5.88}$$

where $y_{q,l}$ is the *score*[9] assigned by the l-th binary learner to the considered observation and

$$g\left(t_{k,l}^{(q)}, y_{q,l}\right) \triangleq \frac{1}{2} \exp\left(-t_{k,l}^{(q)} y_{q,l}\right) \tag{5.89}$$

is the *binary loss* function.

5.2.4 Unsupervised learning

Unsupervised learning is less well defined than its supervised counterpart, since it deals with learning some specific properties of the mechanism on which the generation of the considered set of observations is based. Unlike supervised methods, unsupervised learning works with *unlabelled* datasets. In the following, it is assumed that:

1) Learning is based on the dataset

$$\mathcal{D} \triangleq \{\mathbf{r}_q; q = 0, 1, ..., N_t - 1\}, \tag{5.90}$$

that consists of N_t i.i.d. *unlabelled* D_r-dimensional observations;

2) All the available observations are realizations of the same random variable $\mathbf{r}$, characterized by its *unknown* pdf $f(\mathbf{r})$.

The goal of unsupervised methods is to learn some useful properties of the pdf $f(\mathbf{r})$. It is important to bear in mind that the D_r elements which the random vector $\mathbf{r}$ is made of can be highly correlated. These mutual dependencies are often modelled by introducing a new vector, denoted by $\mathbf{z}$ and collecting the so called *latent* or *hidden variables*. This approach allows to model the dependencies between the elements of the observations indirectly, i.e., through the direct dependencies between such elements and the hidden vector. The relationship between the vectors $\mathbf{z}$ and $\mathbf{r}$ can be modelled in different ways. This results in different models that can be adopted in unsupervised learning; further details can be found in ref. [108].

[9] This quantity can be computed on the basis of eq. (5.69) (eq. (5.81)) if the SVM (Adaboost) method is used.

In the remaining part of this paragraph, a list of the typical unsupervised problems tackled in the field of MIMO radar systems is provided. Then, two specific unsupervised methods and their application to specific problems in that field are described.

Unsupervised learning methods can be exploited to solve the following four relevant technical problems:

a) *Clustering* - Data clustering consists in partitioning the dataset $\mathcal{D}$ (5.90) in a number of groups such that data points in the same group are dissimilar from the data points belonging to all the other groups. In clustering problems, an hidden random variable, called *class variable*, is usually added to all the elements of the dataset; this variable describes the cluster membership for every observation in the dataset. In the last years, significant attention has been paid to the use of clustering methods in automotive radar systems, since distinct clusters can be related to different types of targets, like pedestrians, cars or obstacles. A description of two clustering methods employed in the above mentioned field is provided in paragraph 5.4.4.

b) *Dimensionality reduction* - This aims at generating a reduced dimensionality representation of the observations. Such a representation eases the visualization and interpretation of the dataset, and the identification of specific patterns in it. A well known technique for dimensionality reduction is the *principal component analysis* (PCA); its description is provided in paragraph 5.4.3, whereas its application to a dataset referring to a specific MIMO radar system is illustrated in paragraph 5.2.5.

c) *Feature extraction* - This consists in deriving a vector-valued function, denoted by $\mathbf{g}(\cdot)$ and such that $\mathbf{g}(\mathbf{r})$ represents a useful and lower-dimensional representation of the feature vector $\mathbf{r}$; the vector $\mathbf{g}(\mathbf{r})$ can be used as an input to a supervised learning method. A well known method for synthetizing the function $\mathbf{g}(\cdot)$ is represented by the *autoencoder*, as illustrated in paragraph 5.3.5. A simple method for feature extraction in a MIMO radar system has been described in paragraph 2.5; other techniques are illustrated in paragraphs 5.4.1 and 5.4.4, where their use of radar in human motion characterization and in autonomous driving, respectively, is considered.

d) *Generation of new samples* - This aims at producing new samples of a random vector $\mathbf{r}$ in a way that these are approximately distributed according to its true pdf $f(\mathbf{r})$. Methods for generating new samples can be exploited to de-noise data and for interference mitigation in autonomous driving applications, as illustrated in paragraph 5.4.4.

5.2.5 Selected unsupervised methods

In this paragraph two specific unsupervised methods, namely the PCA technique for dimensionality reduction [109] and the *K-means* algorithm for data clustering [110] are presented. The PCA method is employed to project the dataset $\mathcal{D}$ (5.90) onto a new space, called *principal subspace* and having a dimension $D_r' < D_r$; in doing so, the variance of the projected data is maximised, in order to retain the most relevant variations characterizing the original dataset. This method can be understood by illustrating its application to the case in which $D_r = 4$, $D_r' = 1$ and $\mathbf{r}_q = \mathbf{R}_q$, where the 4D vector $\mathbf{R}_q$ is expressed by eq. (5.82). In this case, the 4D observation $\mathbf{R}_q$ is projected onto the scalar

$$R_q^{'} \triangleq \mathbf{u}_0^T \, \mathbf{R}_q, \tag{5.91}$$

where $\mathbf{u}_0$ is a 4D unit vector [86]. If we define the data covariance matrix

$$\mathbf{\Delta} \triangleq \frac{1}{N_t} \sum_{q=0}^{N_t-1} \left(\mathbf{R}_q - \bar{\mathbf{R}}\right) \left(\mathbf{R}_q - \bar{\mathbf{R}}\right)^T, \tag{5.92}$$

where

$$\bar{\mathbf{R}} \triangleq \frac{1}{N_t} \sum_{q=0}^{N_t-1} \mathbf{R}_q, \tag{5.93}$$

is the data mean, the variance

$$\sigma_R^2 = \mathbf{u}_0^T \, \mathbf{\Delta} \, \mathbf{u}_0 \tag{5.94}$$

of the projected dataset is maximized if

$$\mathbf{u}_0^T \mathbf{\Delta} \, \mathbf{u}_0 = \lambda_0, \tag{5.95}$$

where λ_0 is the largest eigenvalue of the matrix $\mathbf{\Delta}$ (5.92) and $\mathbf{u}_0$ (that represents the *first principal component*) is the associated eigenvector.

In general, if a D'_r-dimensional projection space is considered, the principal components are represented by D'_r eigenvectors $\{\mathbf{u}_l; l = 0, 1, ..., D'_r - 1\}$ of the data covariance matrix $\mathbf{\Delta}$; these eigenvectors are associated with its D'_r largest eigenvalues $\{\lambda_l; l = 0, 1, ..., D'_r - 1\}$ and are chosen to be *orthonormal*. The quality of the resulting transformation can be assessed by evaluating the *distortion measure* (e.g., see [86, Par. 12.1.2, eq. (12.18)])

$$J \triangleq \sum_{l=D'_r}^{D_r-1} \lambda_l, \tag{5.96}$$

i.e., the sum of the eigenvalues associated with the eigenvectors that are orthogonal to the principal subspace; the smaller is the value taken on by this parameter, the better is the original dataset approximation. Actually, if $D'_r = D_r$, the PCA does not cause any distortions, but it simply restates the input dataset in a new domain, where the features of the datasets are uncorrelated. In this particular case, the value of J is zero (see (5.96)).

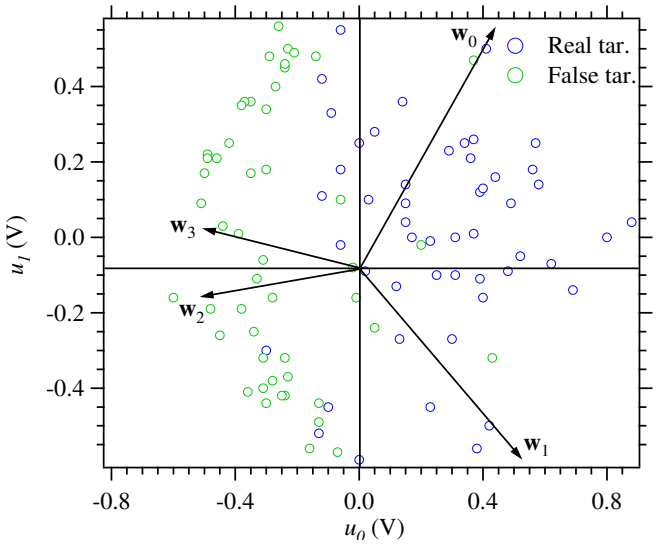


FIGURE 5.7: Biplot of the dataset $\mathcal{D}'$ generated by the PCA technique. The points of the reduced dataset corresponding to false (real) targets are identified by the green (blue) circles.

In the considered experiment, the PCA method has been applied to extract a 2D dataset from the 4D dataset which Figs. 5.4-5.6 refer to (see paragraph 5.2.3). The 2D points of the new dataset, denoted by $\mathcal{D}'$, are represented in the *principal component bi-plot*[10] shown in Fig. 5.7. In this figure, the axes of the Cartesian plane are associated with the principal components, whereas the vector $\mathbf{w}_i$, represented by an oriented segment, allows to quantify, through its amplitude and orientation, the weight of the contribution provided by the i-th component of the original feature vectors (i.e., of the set $\{\mathbf{R}_q\}$; see eq. (5.82)) to the principal components (with $i = 0$, 1, 2 and 3). From Fig. 5.7 it is easily inferred that:

1. The weights of the contributions due to $R_q^{(2)}$ and $R_q^{(3)}$ are similar and are about half of those provided by $R_q^{(0)}$ and $R_q^{(1)}$.
2. The new 2D observations referring to real (false) targets are spread over the right (left) half plane of the Cartesian plane.

The *K-means* method allows partitioning the available dataset $\mathcal{D}$ into K clusters, each collecting the samples whose mutual distances are small with respect to the distances from the points outside the cluster itself. In practice, if the *center* of the k-th cluster is denoted by $\boldsymbol{\mu}_k$ (with $k = 0, 1, ..., K-1$), the K-means method assigns the q-th data point $\mathbf{r}_q$ to the cluster whose center is closest to $\mathbf{r}_q$. This strategy can be formalised as the one minimizing the so-called *distortion measure*

$$V \triangleq \sum_{q=0}^{N_t-1} \sum_{k=0}^{K-1} p_{q,k} ||\mathbf{r}_q - \boldsymbol{\mu}_k||^2, \tag{5.97}$$

with respect to the variables $\{p_{q,k}\}$ and the vectors $\{\boldsymbol{\mu}_k\}$; here, $p_{q,k}$ is a binary indicator variable implementing the 1-of-K coding scheme, i.e. such that $p_{q,k} = 1$ ($p_{q,k} = 0$) if $\mathbf{r}_q$ is (is not) assigned to the k-th cluster. The problem of minimizing the function V (5.97) is solved by means of an iterative procedure, whose iterations consist of two steps. In the first step, known as *expectation*, the metric V is minimized with respect to each of the variables $\{p_{q,k}\}$, keeping the centers $\{\boldsymbol{\mu}_k\}$ fixed; on the contrary, in the second step, called *maximization*, the same metric is minimized with respect to the vectors $\{\boldsymbol{\mu}_k\}$, keeping the variables $\{p_{q,k}\}$ fixed. More specifically, in the first step, the values of the variables $\{p_{q,k}\}$ employed are computed as

$$p_{q,k} \triangleq \begin{cases} 1 \text{ if } \quad k = \arg\min_j ||\mathbf{r}_q - \boldsymbol{\mu}_j||^2 \\ 0 \text{ otherwise} \end{cases} \tag{5.98}$$

for any q (in other words, the q-th data point is assigned to the cluster whose center is closest to it). Then, in the second step, the center of the k-th cluster is evaluated as

$$\boldsymbol{\mu}_k = \frac{\sum_{q=0}^{N_t-1} p_{q,k} \mathbf{r}_q}{\sum_{q=0}^{N_t-1} p_{q,k}}, \tag{5.99}$$

with $k = 0, 1, ..., K-1$. It is important to point out that:

[10] A detailed description of how a bi-plot is generated can be found in ref. [109, Sect. 5.3]

1. In principle, the initial values of the cluster centers can be arbitrarily chosen. In this case, however, the algorithm may require several iterations to reach convergence. A better initialization procedure consists in choosing the initial centers in a random fashion.

2. The sum appearing in the denominator of the RHS of the eq. (5.99) gives the overall number of points assigned to the k-th cluster; consequently, the cluster center evaluated on the basis of the same equation represents the mean of all the data points $\mathbf{r}_q$ assigned to the k-th cluster.

3. Iterations are stopped when there is no further change in the assignments of the data points to the K clusters or their overall number has reached a fixed threshold.

Let us analyse now an application of the K-means technique to the dataset $\mathcal{D}$ (5.90), where $N_t = 100$,

$$\mathbf{r}_q \triangleq \left[\hat{R}_q \, , \, \hat{\phi}_q\right]^T , \tag{5.100}$$

and $\hat{R}_q$ and $\hat{\phi}_q$ represent the estimates of the range and of the azimuth, respectively, of the single point target observed in the q-th trial; these estimates are generated by the algorithm illustrated in paragraph 2.5 and employed in a FCMW radar system equipped with the antenna array illustrated in Fig. 2.3-b) ($d = \lambda/4$ is assumed). Moreover, in generating the q-th observation of the dataset $\mathcal{D}$ (5.90), the following assumptions have been made:

a) The amplitude $a_q^{(v)}$ of the sinusoid observed on the v-th virtual antenna is uniformly distributed over the interval $[0.3, 1.0]$ V.

b) The random variable $a_q^{(v)}$ is independent of $a_p^{(u)}$ for any $u \neq v$ and/or $p \neq q$.

c) The overall number of time-domain samples (N) acquired from each of the four RX antennas is equal to 512 and the standard deviation σ_w of the noise affecting them is equal to 1.0 V (see eq. (2.7)).

d) The oversampling factor $M = 4$ and the threshold $P_{th} = 0.5\ \mathrm{V^2Hz^{-1}}$ are employed by the detection algorithm based on eqs. (2.55)-(2.56).

e) The range R_q of the target detected in the q-th trial is uniformly distributed over the interval $[R_m \, , \, R_M] = [1.0\ \mathrm{m}, 9.0\ \mathrm{m}]$, whereas its azimuth is randomly selected in the set of relative integers ranging from ϕ_m to ϕ_M, with $\phi_M = -\phi_m = 45^\circ$.

f) The parameters of the employed radar system take on the same values as those selected for the example illustrated for the SVM and K-NN methods in paragraph 5.2.3.

In this case, the K-means algorithm is employed to group the detected targets in three different clusters (consequently, $K = 3$ is selected) on the basis of their azimuth only; the points of the first (third) cluster are characterized by $\phi_q < -15^\circ$ ($\phi_q > 15^\circ$), whereas those of the second one by $|\phi_q| \leq 15^\circ$. The observations collected in the synthetically generated dataset and their partitioning into the clusters generated by the K-means technique are shown in Fig. 5.8, where circles of different colours are used to identify targets assigned to distinct classes.

From these results it is easily inferred that:

1. all the points are correctly classified on the basis of their azimuth, even if an unlabelled dataset is used;

2. each of the centroids is located in the middle of the corresponding cluster and its position is influenced by the distribution of the detected targets along the range dimension.

This is a quite ideal scenario in which the targets are equally distributed in the range azimuth space. K-means works particularly well in this case and, more in general, in all the cases where clusters have globular or spherical shapes. In all the other scenarios where clusters have different shapes K-means is likely to fail and the performance drops significantly. In these cases other clustering techniques can be preferred; an overview of other clustering techniques employed in real automotive scenarios are described in Par. 5.4.4.

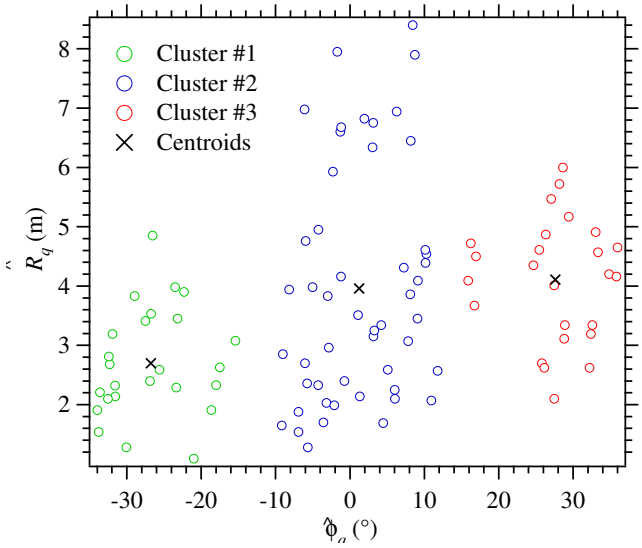


FIGURE 5.8: Representation of three clusters generated by the *K*-means algorithm. The green (red) circles refer to the targets detected on the left (right) of the considered radar system, whereas the blue circles to the targets detected in front of it. The black crosses identify the centroids of the clusters.

5.3 Data-driven approach based methods

In this section, after a description of some relevant differences between ML and DL techniques, the readers are introduced to deep neural networks by illustrating their architecture, their training and a specific application to an FMCW radar system. Finally, few fundamental DL methods employed in the field of MIMO radar systems are illustrated.

5.3.1 Relevant differences between ML and DL techniques

Machine learning techniques allow to achieve satisfying accuracy in various applications at the price of a reasonable computational complexity. Nevertheless, in pattern recognition problems, their capability is often limited by the features selected to learn common patterns and to detect them; in fact, in these cases, devising a transformation able to extract a suitable internal representation from the observed raw data requires good expertise and engineering skills [111]. A revolutionary data-driven approach to feature extraction is offered by DL methods. Despite the significant computational

complexity of these methods, in recent times their implementation has become possible thanks to the availability of low-cost powerful *graphic processing units* (GPUs), which make the exploitation of their inner parallelism possible.

Deep learning solves the problem of feature extraction by adopting a multilayer representation of raw data. This fundamental principle is exemplified by a *feedforward deep network*, also known as *multilayer perceptron* (MLP); such a network is able to represent a complicated mathematical function by composing multiple simpler functions, i.e. multiple *layers*. Generally speaking, a MLP consists of three different types of layers: an input layer, multiple hidden layers with learnable weights and an output layer. Its architecture can be represented through a *directed acyclic graph* (DAG), whose structure is exemplified in Fig. 5.9, that refers to the specific case of a fully connected MLP containing a single inner layer. The basic building block of each layer is the so called *neuron*.

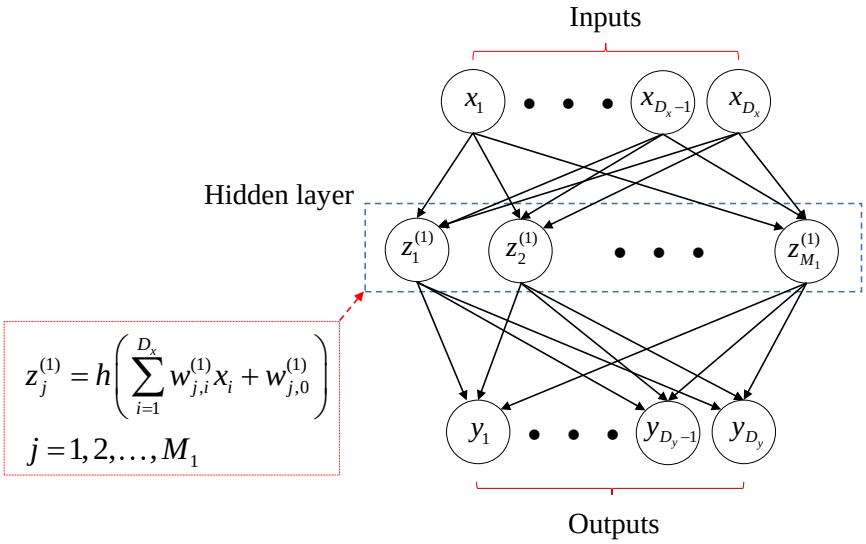


FIGURE 5.9: Directed acyclic graph describing the architecture of a fully-connected neural network. Variables are represented by circles (i.e., by nodes), whereas weights by the links between nodes. A single inner layer (i.e., $K = 1$) is assumed for simplicity.

In general, the output $z_j^{(k)}$ generated by the j-th neuron of the k-th layer can be expressed as

$$z_j^{(k)} = h\left(a_j^{(k)}\right), \tag{5.101}$$

with $j = 1, 2, ..., M_k$ and $k = 1, 2, ..., K$; here, M_k denotes the overall number of neurons in the k-th layer, K denotes the overall number of layers, $h(\cdot)$ is a differentiable nonlinear function (i.e., a sigmoid function, an hyperbolic tangent or rectifier linear unit) and the quantity

$$a_j^{(k)} \triangleq \sum_{i=1}^{M_{k-1}} w_{j,i}^{(k)} z_i^{(k-1)} + w_{j,0}^{(k)}, \tag{5.102}$$

known as *activation* function, is a linear combination of the neuron inputs $\{z_i^{(k-1)}; i = 1, 2, ..., M_{k-1}\}$ (whose *learnable weights* are the M_k parameters $\{w_{j,i}^{(k)}; i = 1, 2, ..., M_{k-1}\}$) and the *bias* $w_{j,0}^{(k)}$. The outputs of the neurons of the k-th layer are collected in the vector

$$\mathbf{z}^{(k)} \triangleq \left[z_1^{(k)}, z_2^{(k)}, ..., z_{M_k}^{(k)}\right], \tag{5.103}$$

that feeds the successive hidden layer. The input layer is fed by the D_x-dimensional input vector

$$\mathbf{x} \triangleq [x_1, x_2, ..., x_{D_x}], \tag{5.104}$$

whereas the output layer generates the D_y-dimensional output vector

$$\mathbf{y} \triangleq \left[y_1, y_2, ..., y_{D_y}\right], \tag{5.105}$$

on the basis of eqs. (5.101) and (5.102).

It is important to mention that: a) the learnable weights of the hidden layers can be interpreted as an encoded representation of the inputs; b) unlike ML methods, where a number of manually extracted features are chosen a priori, the considered neural network automatically extracts features through the use of non linear functions.

5.3.2 Training a deep neural network

Training a deep neural network is an art [108], even if in future years we will go from alchemy art to solid theory. Its objective is the same as that already illustrated for ML methods, i.e. the minimization of a *loss* or an *error function* (see paragraph 5.2.1). However, in a feedforward neural network, this result is achieved by using a local message passing scheme, according to which the internal representations of each neuron are sent forward, towards the output of the network, and the errors obtained by the computation of the loss are sent backward, from the output to the neurons, along the graph representing the network itself (e.g., see [86, Par. 5.3]). This scheme, known as *back-propagation* [112], operates as follows (batch processing is assumed here). For each pattern of the training set, the activations of the hidden and output layers of the considered network are computed through successive applications of eqs. (5.102) and (5.101), respectively; this process is known as *forward propagation*, since it proceeds from the input to the output of the network. The back-propagation algorithm, instead, allows to compute the gradient of the selected *error function*, denoted by E and corresponding to the *loss function* defined for ML methods (see eq. (5.8)), with respect to the weights appearing in each layer. The derivative of the error function E with respect to the weight $w_{j,i}^{(k)}$, referring to the i-th input of the j-th neuron in the k-th layer of the network, can be expressed as

$$\frac{\partial E}{\partial w_{j,i}^{(k)}} = \sum_q \frac{\partial E_q}{\partial w_{j,i}^{(k)}}, \tag{5.106}$$

with $j = 1, 2, ..., M_k$, $i = 1, 2, ..., M_{k-1}$ and $k = 1, 2, ..., K$; here, E_q represents the error associated with the q-th observation. Based on the chain rule, the partial derivative appearing in the RHS of eq. (5.106) can be evaluated as

$$\frac{\partial E_q}{\partial w_{j,i}^{(k)}} = \sigma_j^{(k)} z_i^{(k)}, \tag{5.107}$$

where $\sigma_j^{(k)} \triangleq \partial E_q / \partial a_j^{(k)}$, $z_i^{(k)} \triangleq \partial a_j^{(k)} / \partial w_{j,i}^{(k)}$ and $a_j^{(k)}$ is defined by eq. (5.102). Consequently, eq. (5.106) can be put in the form

$$\frac{\partial E}{\partial w_{j,i}^{(k)}} = \sum_q \sigma_j^{(k)} z_i^{(k)}. \tag{5.108}$$

The quantity $\sigma_j^{(k)}$ appearing in the last formula can be evaluated as follows. First, the quantity

$$\sigma_l^{(K)} \triangleq y_l - t_l \tag{5.109}$$

is computed for the l-th unit of the output layer, where t_l denotes its target. Then, the backpropagation formula

$$\sigma_j^{(k)} = h'\left(a_j^{(k)}\right) \sum_l w_{l,j}^{(k+1)} \sigma_l^{(k+1)}. \tag{5.110}$$

is applied for $k = K-1$, $K-2$, ..., 1 and, given k, for $j = 1$, 2, ..., M_k; here, $h'(\cdot)$ denotes the first derivative of the function $h(\cdot)$ appearing in eq. (5.101). This allows to recursively compute all the quantities $\{\sigma_l^{(k)}\}$ on the basis of the similar quantities $\{\sigma_l^{(k+1)}\}$ made available by all the units appearing in the $(k+1)$-th layer of the network.

It is worth noting that: a) the computational complexity of the network depends on the number of neurons in each hidden layer, since this determines the number of parameters to be tuned; b) overfitting may be observed in the presence of a large number of neurons. The last problem can be mitigated by including a *regularization term* in the considered error function (a similar strategy has been also proposed for ML methods; see eq. (5.52) in paragraph 5.2.2). An alternative to this approach is represented by the so called *early stopping* procedure, that consists in stopping network training when the error over a given validation dataset[11] is minimised.

Let us focus now on a neural network having the architecture illustrated in Fig. 5.9 and analyse its possible use in an FMCW radar system equipped with the antenna array shown in Fig. 2.8-b) ($d = \lambda/4$ is assumed). In the experiment at hand, the overall synthetically generated dataset includes $\hat{N}_t = 2500$ observations, all acquired in the presence of a single point target, whose range R_q and the azimuth ϕ_q are uniformly distributed over the intervals $[R_m, R_M] = [1\text{ m}, 7\text{ m}]$ and $[\phi_m, \phi_M] = [-60^\circ, 60^\circ]$, respectively, for any q. Moreover, the values selected for the parameters of the employed radar system are equal to those listed in the example of paragraph 5.2.2; the only difference is represented by the standard deviation of the noise affecting the received signal samples, that is $\sigma_w = \sqrt{2}/2$ V. The q-th observation and the associated label are[12]

$$\mathbf{r}_q \triangleq \left[r_{q,0}, r_{q,1}, r_{q,2}, r_{q,3}, r_{q,4}\right]^T = \left[\hat{\psi}_q^{(0)}, \hat{\psi}_q^{(1)}, \hat{\psi}_q^{(2)}, \hat{\psi}_q^{(3)}, \hat{f}_q\right]^T \tag{5.111}$$

and

$$\mathbf{t}_q \triangleq \left[t_{q,0}, t_{q,1}\right]^T = \left[R_q, \phi_q\right]^T, \tag{5.112}$$

respectively; here, $\hat{\psi}_q^{(v)} = \angle\hat{C}_q^{(v)}$ (with $v = 0$, 1, 2 and 3) and $\hat{C}_q^{(v)}$ is the complex amplitude measured on the v-th virtual element at the frequency $\hat{f}_q$ (2.57) (see eqs. (2.58) and (2.65)).

The aim of the neural network is predicting the position of the target (i.e., its azimuth and range) on the basis of a new observation. In this case, the network has 5 inputs two outputs, since $x_j = r_{q,j}$ (with $j = 0$, 1, ..., 4) and $y_k = t_{q,k}$ (with $k = 0$, 1). Moreover, a single hidden layer consisting of $M_1 = 10$ neurons is used; each of these neurons is connected to all the available inputs and employs the hyperbolic tangent transfer

[11] The *validation dataset* is a set of data on which the performance of the considered network is evaluated during its training.

[12] *Unwrapped* phases are employed in this case, since they ease network training

function

$$h(x) \triangleq \frac{\exp(2x)-1}{\exp(2x)+1} \tag{5.113}$$

in the evaluation of its output on the basis of eqs. (5.101)-(5.102). The predictions of the target range and azimuth are computed by the output layer, that contains two neurons.

The *scaled conjugate gradient* method [113] has been employed to train the network described above. The size of the training set $\mathcal{D}$ is $N_t = 2225$, since 85% of the overall dataset has been exploited for network training; the remaining part $\mathcal{D}_{ts}$ of the dataset, whose size is $\bar{N}_t = 375$, has been used as a test set. The simulation results have evidenced that the adopted network is able to accurately predict the position of a new target; in fact, the RMSEs evaluated for the range and the azimuth on the set $\mathcal{D}_{ts}$ are approximately equal to 4 cm and to 0.2°, respectively. Finally, it is worth noting that:

a) The use of the network described above does not require a specific expertise.

b) Unlike the regression methods illustrated in Par. 5.2.2, the employed network is able to predict both the azimuth and the range of a single point target; however, a by far larger dataset is used for its training.

In general, the main drawback of DL methods is represented by the size of the dataset, which is usually much larger than that needed by ML techniques; this results in a significant increase in the computational effort of the required training. Moreover, especially for radar application, the availability of a so large dataset is not guaranteed.

5.3.3 Autoencoders

An *autoencoder* (AE) is a neural network that, similarly as the PCA technique, is able to perform *dimensionality reduction* by learning an efficient representation of its input data in an unsupervised fashion. Since the goal of an AE is to approximate the identity function without learning it exactly, its D_y-dimensional output vector (5.105) can be expressed as

$$\mathbf{y} = \mathbf{h}_w(\mathbf{x}) \approx \mathbf{x}, \tag{5.114}$$

where $\mathbf{h}_w(\cdot)$ represents the transformation performed by the network on its D_x dimensional input vector $\mathbf{x}$.

The architecture of an *under-complete* AE based on a symmetric encoding-decoding structure is illustrated in [114]. In the following, only under-complete AEs are considered, since they are employed in various radar applications, as shown in paragraphs 5.4.1-5.4.4. It is also worth mentioning that, in such applications, autoencoding is often employed as pre-processing method preceding supervised classification; this allows to learn repetitive structures of input data when the training dataset is not so large. The last application of AEs will be analysed in paragraph 5.3.5 in more detail.

Let us now focus on a possible application of auto-encoding to an FMCW radar system equipped with the antenna array shown in Fig. 2.8-b) ($d = \lambda/4$ is assumed) and operating in the presence of at most a single point target. In this case, the set $\{\mathbf{x}_q^{(0)}, \mathbf{x}_q^{(1)}, \mathbf{x}_q^{(2)}, \mathbf{x}_q^{(3)}\}$ (e.g. see eq.(2.7)), consisting of four N-dimensional noisy vectors becomes available in the q-th trial, with $q = 0, 1, ..., N_t - 1$. Each of these vectors undergoes the DFT processing described in paragraph 2.5; this allows to compute the $4\hat{N}_0$-dimensional feature vector

$$\mathbf{R}_q = [R_{q,0}, R_{q,1}, ..., R_{q,4\hat{N}_0-1}]^T \triangleq \left[\left(\mathbf{Y}_q^{(0)}\right)^T, \left(\mathbf{Y}_q^{(1)}\right)^T, \left(\mathbf{Y}_q^{(2)}\right)^T, \left(\mathbf{Y}_q^{(3)}\right)^T\right]^T \tag{5.115}$$

for any q; here, for any v,

$$\mathbf{Y}_q^{(v)} = [Y_{q,0}^{(v)}, Y_{q,1}^{(v)}, ..., Y_{q,\hat{N}_0-1}^{(v)}]^T \triangleq M \left[\left| X_{b_m}^{(v)} \right|, \left| X_{b_m+1}^{(v)} \right|, ..., \left| X_{b_M}^{(v)} \right| \right]^T \tag{5.116}$$

is an $\hat{N}_0$-dimensional vector, $X_k^{(v)}$ is the k-th element of the N_0-dimensional vector $\mathbf{X}_q^{(v)}$ computed on the basis of eq. (2.50) (with $k = b_m, b_m + 1, ..., b_M$), M is the oversampling factor employed in DFT processing,

$$\hat{N}_0 \triangleq b_M - b_m + 1, \tag{5.117}$$

and b_m and b_M are integer parameters delimiting the portion of the received signal spectrum over which an amplitude peak, due to the presence of a possible target, is expected. Note that the pair (b_m, b_M) represents a form of a priori information and that, in general, the inequality $0 \leq b_m < b_M \leq N_0 - 1$ holds. Let assume now that the overall data set

$$\mathcal{D}_o = \{(\mathbf{R}_q, t_q); q = 0, 1, ..., \hat{N}_t - 1\}, \tag{5.118}$$

acquired in $\hat{N}_t = 2400$ independent trials, is available; here, the label $t_q = 1$ (-1) refers to the presence of a *real* (*false*) target detected on the basis of the deterministic strategy expressed by eq. (5.85). Moreover, the following assumptions are made to synthetically generate the set $\mathcal{D}_o$ (5.118):

a) Half of its data are associated with the detection of a real target, the remaining half with the detection of a false target.

b) The parameters of the employed radar system take on the same values as those selected for the example illustrated for the SVM and K–NN methods in Par. 5.2.3.

c) The stochastic models adopted for amplitude $a_q^{(v)}$ of the sinusoid observed on the v-th antenna in the presence of a real target, and for the range R_q and the azimuth ϕ_q of the target (if present) are the same as those defined in the example illustrated for the SVM and K–NN methods in paragraph 5.2.3.

d) The size $\hat{N}_0$ of the vector $\mathbf{X}_q^{(v)}$ is equal to 121, since (see eq. (5.117))

$$b_m = \left\lfloor \frac{2R_m \mu}{c} N_0 T_s \right\rfloor = 13 \tag{5.119}$$

and

$$b_M = \left\lfloor \frac{2R_M \mu}{c} N_0 T_s \right\rfloor = 133, \tag{5.120}$$

where $R_m = 0.5$ m ($R_M = 5.0$ m) represent the minimum (maximum) range expected for the target.

An AE is employed in the considered radar system to reduce the dimensionality of the feature vector $\mathbf{R}_q$ (5.115) (whose size is $4\hat{N}_0 = 484$); note that, unlike the deterministic approach described in paragraph 2.5 and based on a maximum search, an unsupervised data-driven method is exploited in this case. The adopted AE architecture includes only a single layer in its encoder and a single layer in its decoder, for simplicity. The compressed representation available at the output of the encoder layer is represented by the $\bar{M}$-dimensional vector

$$\mathbf{z}_q \triangleq \mathbf{h}_e \left(\mathbf{W}_e \mathbf{R}_q + \mathbf{b}_e \right) \tag{5.121}$$

collecting the hidden variables; here, $\mathbf{W}_e$ is a weight matrix of size $\bar{M} \times 4\,\hat{N}_0$, $\mathbf{b}_e$ is an $\bar{M}$-dimensional bias vector and $\mathbf{h}_e(\mathbf{x})$ is an $\bar{M}$-dimensional vector resulting from the element-by-element application of the positive saturating linear transfer function

$$h(x) = \begin{cases} 0 \text{ if } x \leq 0 \\ x \text{ if } 0 < x < 1 \\ 1 \text{ if } x \geq 1 \end{cases} \tag{5.122}$$

to the $4\hat{N}_0$-dimensional input vector $\mathbf{R}_q$ (5.115). The decoder maps the encoded representation $\mathbf{z}_q$ (5.121) back to the $4\hat{N}_0$-dimensional vector

$$\begin{aligned} \mathbf{y}_q &= [y_{q,0}, y_{q,1}, ..., y_{q,4\hat{N}_0-1}]^T \\ &\triangleq \mathbf{W}_d\,\mathbf{z}_q + \mathbf{b}_d, \end{aligned} \tag{5.123}$$

that represents an estimate of the original input vector; here, $\mathbf{W}_d$ is a $4\hat{N}_0 \times \bar{M}$ weight matrix and $\mathbf{b}_d$ is an $4\hat{N}_0$-dimensional bias vector. In simulations, $\bar{M} = 60$ has been selected; consequently, a 60-dimensional hidden vector is extracted from a 484-dimensional observation (i.e., roughly an eightfold dimensionality reduction is achieved). Moreover, the *scaled conjugate gradient* method [113] has been employed to train the AE. Training is based on the dataset $\mathcal{D}$, that contains 90% of the dataset $\mathcal{D}_o$ (5.118) and, consequently, involves $N_t = 2160$ observations; the remaining part of the dataset, whose size is $\bar{N}_t = 240$, forms the test set $\mathcal{D}_{ts}$. The effectiveness of the employed AE is exemplified by Fig. 5.10, where the output vector $\mathbf{y}_q$ generated by the autoencoder in response to a specific feature vector $\mathbf{R}_q$ of the test set is shown; this is also confirmed by the small RMSE evaluated over $\mathcal{D}_{ts}$: $RMSE = 0.1$ V is found in this case.

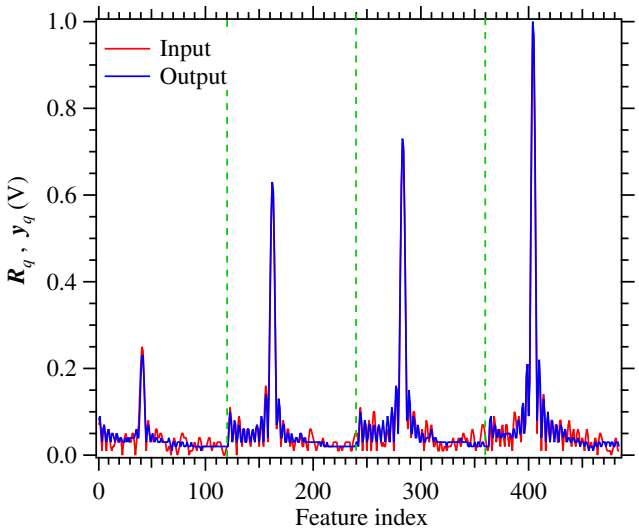


FIGURE 5.10: Example of a feature vector $\mathbf{R}_q$ (5.115) (red line) and of the corresponding output vector $\mathbf{y}_q$ (5.123) (blue line) predicted by the AE employed in the example of Par. 5.3.3. The contributions of the four vectors $\{\mathbf{Y}_q^{(v)};\ v = 0, 1, 2, 3\}$ which $\mathbf{R}_q$ is made of are delimited by green dashed lines.

This leads to the conclusion that the compressed representation computed by the AE and expressed by the vector $\mathbf{z}_q$ (5.121) is really able to capture all the relevant information conveyed by the input vector $\mathbf{R}_q$ (5.115). Finally, it is worth mentioning that the compressed representation $\mathbf{z}_q$ (5.121) can be exploited to train the linear SVM and K-NN

methods described in paragraph 5.2.3 and employed to discriminate between real and false targets. In the experiment at hand, these two supervised methods have trained on a dataset consisting of $N_t = 240$ observations ($K = 4$ has been selected for the K-NN method); half of them are associated with the detection of a real target, half with the detection of a false target. Computer simulations have evidenced that, despite the dimensionality reduction, a slightly better accuracy is achieved by the considered classification techniques; in fact, the obtained accuracies are equal to 93% and 97% for the K-NN and the linear SVM, respectively ($\bar{N}$-fold cross validation, with $\bar{N} = 5$, has been used) compared to 89% and 91% without dimensionality reduction.

5.3.4 Convolutional neural networks

Convolutional neural networks (CNNs) play an important role in DL applications, since they allow to exploit the spatio-temporal information available in a sequence of images [111], [114]; for this reason, they are trained using a labelled dataset. The processing performed by a CNN aims at capturing the local features of input images and is based on *spatially localized convolutional filtering*. Its typical architecture includes *convolutional*, *pooling*, *fully connected* layers, and is motivated by the fact that, in images, local groups of values may exhibit high correlation and local statistics are invariant to position. In fact, convolutional layers aim at detecting local features on the basis of the data originating from the previous layer, pooling layers at merging semantically similar features and fully connected layers at generating the final feature vector. The processing accomplished by the convolution and pooling operations on a greyscale image is outlined in Fig. 5.11.

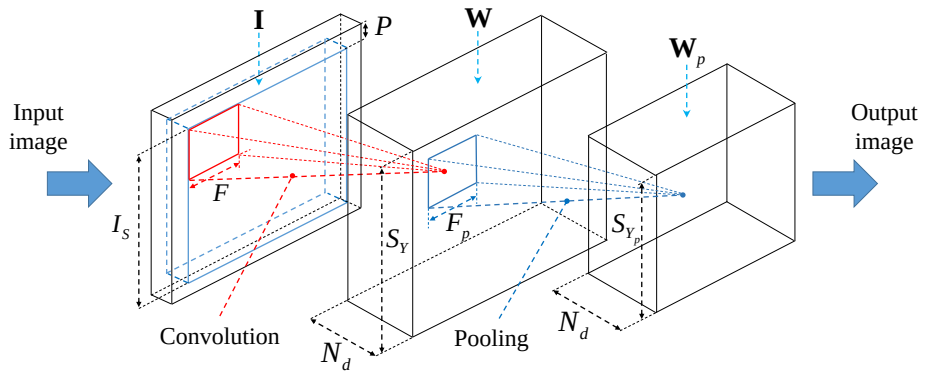


FIGURE 5.11: Representation of the *convolution* and *pooling* operations accomplished by a CNN on a greyscale image.

The output of the convolution depends on both its input, represented by a small portion of the image, and the adopted convolution *kernel*, denoted by $\{K[m,n]\}$; moreover, this operation is repeated on multiple disjoint portions until the whole input image is scanned. From a mathematical viewpoint, the convolution input is a matrix, consisting of $I_S \times I_S$ pixels and denoted by $\mathbf{I} = [I[i,j]]$, whereas the resulting output is a $S_Y \times S_Y$ matrix, called *activation* or *feature map* and denoted by $\mathbf{Y} = [Y[i,j]]$. The (i,j)-th element (i.e., unit) of the activation map is evaluated as[13]

$$Y[i,j] \triangleq \sigma\left(\sum_{m=-F/2}^{F/2-1}\sum_{n=-F/2}^{F/2-1} K[m,n]\, I[i-m, j-n]\right), \tag{5.124}$$

[13] In practical implementations, the convolution operations are substituted with the cross-correlations, since they are faster and more computationally efficient.

where F and $K[m,n]$ are the *size* of the convolutional filter (also known as *kernel size*) and its (m,n)-th weight, respectively, and $\sigma(\cdot)$ is a non linear *activation function*. Another relevant parameter of a convolutional layer is its *stride* S, that represents the number of pixels shifts over the input matrix when the kernel moves from a portion of the image to the next one; for instance, when the stride is one, the filter moves one pixel at a time. The area of the input image processed by the kernel can be also extended by adding a set of numbers (usually set at zero) to the border of image itself, as shown in Fig. 5.11; in that figure, the parameter P (dubbed *padding*) represents the number of zero columns and rows added to the input image. The stride, the padding and the kernel size of a convolutional layer influence the size S_Y of the output matrix; in fact, it can be shown that

$$S_Y = \frac{I_S - F + 2\,P}{S} + 1. \tag{5.125}$$

For this reason, the above mentioned parameters have to be jointly selected in a way that the RHS of last equation takes on an integer value.

Generally speaking, the convolution operation expressed by eq. (5.124) can be performed N_d times over the same image; in accomplishing this procedure, the parameters P and S do not change. This produces the output volume (i.e., matrix) $\mathbf{W}$ shown in Fig. 5.11 and having size $S_Y \times S_Y \times N_d$ (the parameter N_d is called *depth*); this matrix results from stacking N_d distinct activation maps, each representing a specific *slice*.

The convolutional layer represented in Fig. 5.11 feeds a *pooling layer*, whose task is reducing the dimensionality of each input slice and, consequently, the overall complexity of the considered CNN. The processing accomplished by the pooling layer can be easily described by referring to a single slice, denoted by $\mathbf{Y}$, of the output volume $\mathbf{W}$. Similarly as the convolution operation, the pooling operation is fed by a portion, having size $F_p \times F_p$, of the considered slice and generates the $S_{Y_p} \times S_{Y_p}$ output matrix $\mathbf{Y}_p = [Y_p[i,j]]$. The most popular layers of this type are known as *max pooling* and as *average pooling*. In the former case, the (i,j)-th pixel of the output matrix $\mathbf{Y}_p$ is computed as

$$Y_p[i,j] \triangleq \max_{p,q \in \mathcal{S}_{F_p}(i,j)} Y[p,q], \tag{5.126}$$

whereas in the latter one as

$$Y_p[i,j] \triangleq \frac{1}{F_p^2} \sum_{p,q \in \mathcal{S}_{F_p}(i,j)} Y[p,q], \tag{5.127}$$

where $\mathcal{S}_{F_p}(i,j) \triangleq \{(p,q) | -F_p/2 + i \leq p \leq i + F_p/2 - 1, -F_p/2 + j \leq p \leq F_p/2 - 1 + j)\}$ and the parameter F_p is called *pool size*. It can be shown that

$$S_{Y_p} = \frac{S_Y - F_p}{S_p} + 1, \tag{5.128}$$

where S_p is the *stride* of the pooling (its meaning is similar to that illustrated above for the parameter S). Note that the depth N_d of the final output volume $\mathbf{W}_p$ generated by pooling is the same as that of $\mathbf{W}$. Finally, it is important to point out that:

a) in CNN applications, a chain of pairs of convolutional and pooling layers is commonly used. Moreover, fully connected layers (FC) of different lengths are often added at the end of the cascade of convolutional/pooling layers; this allows to combine all the

extracted features in a 1D vector.

b) As shown in paragraphs 5.4.1-5.4.4, CNNs are employed in a number of radar applications ranging from human activity characterization to autonomous driving. Some experimental results about the use of CNNs in the classification of three different human activities CNN are illustrated in section 5.6.

5.3.5 Convolutional autoencoders

A *convolutional autoencoder* (CAE) may represent an appealing technical option in those applications in which a large amount of labelled data is unavailable. In fact, it combines the advantages offered by unsupervised learning techniques (i.e., by AEs) with the capability of CNNs to extract spatio-temporal information from a sequence of images. The architecture of a CAE is exemplified by Fig. 5.12; this network consists of an encoder side, combining convolutional and pooling (i.e., *downsampling*) layers, and of a decoder side, made of transposed convolutional (also called *deconvolutional*) and unpooling (i.e., *upsampling*) layers. Each transposed convolutional layer allows to upsample its input feature map with the aim of retrieving the original shape of the image available at the input of the first convolutional layer contained in the encoder side. In each unpooling layer, instead, an upsampling procedure exploiting the positions of the maxima stored in the corresponding max pooling operation executed at the encoder side is accomplished.

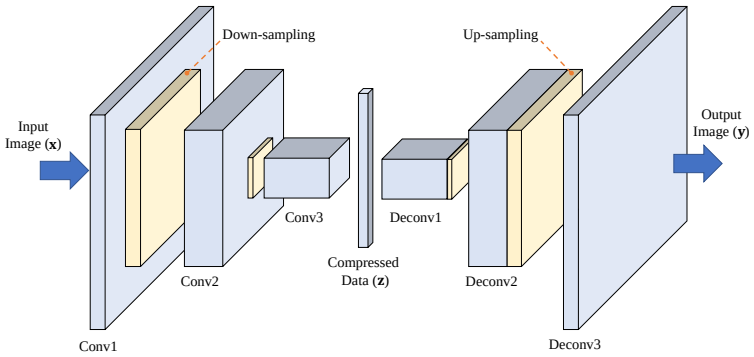


FIGURE 5.12: Example of CAE architecture.The acronym *ConvX* (with $X = 1, 2$ and 3) identifies the X-th convolutional and pooling layer, whereas *DeconvX* (with $X = 1, 2$ and 3) the transpose and unpooling layer.

5.3.6 Recurrent neural networks

In the neural networks treated so far, all the inputs and all the outputs are time independent from each other. Features related to the time evolution of the observed data can be extracted through a *recurrent neural network* (RNN) [115]. A well known example of RNN is the so called *Vanilla* RNN, whose architecture is represented in Fig 5.13-a). In this network, past information contribute to the computation of its output, since they are reinjected into the network itself and stored in its internal (i.e., hidden) state. Moreover, the following three distinct weight matrices are employed by this network: a) the $\bar{M} \times D_r$ matrix $\mathbf{U}$ employed in the mapping of the D_r-dimensional input vector $\mathbf{r}_q^{(t)}$ at time t to its $\bar{M}$-dimensional hidden state $\mathbf{h}^{(t)}$; b) the $\bar{M} \times \bar{M}$ square matrix $\mathbf{W}$ involved in the update of its internal state; c) the $D_r' \times \bar{M}$ matrix $\mathbf{V}$ employed to map $\mathbf{h}^{(t)}$ to its D_r'-dimensional output vector $\mathbf{o}^{(t)}$.

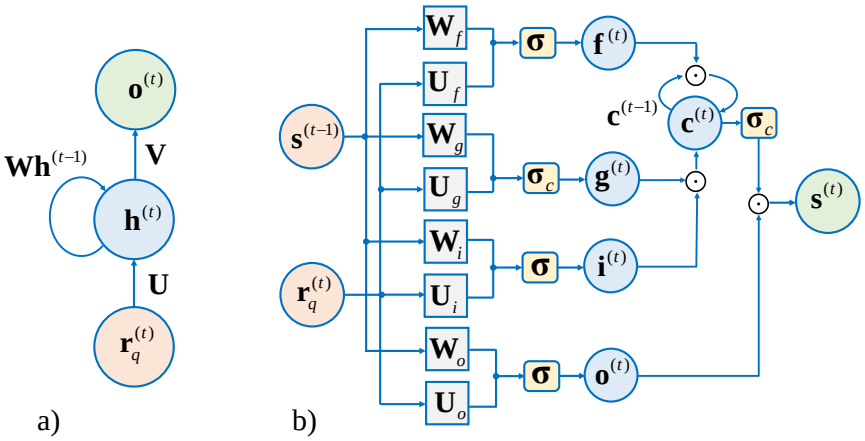


FIGURE 5.13: Architecture of: a) a Vanilla RNN; b) an LSTM neural network.

In fact, based on these matrices, the state update of the network and the computation of its output at time t can be expressed as

$$\mathbf{h}^{(t)} = \boldsymbol{\phi}(\mathbf{W}\,\mathbf{h}^{(t-1)} + \mathbf{U}\,\mathbf{r}_q^{(t)}) \tag{5.129}$$

and as

$$\mathbf{o}^{(t)} = \mathbf{V}\,\mathbf{h}^{(t)}, \tag{5.130}$$

respectively; here, $\boldsymbol{\phi}(\cdot)$ denotes a non-linear activation vector function. It is important to point out that:

a) A RNN can be thought as the result of the interconnection of multiple copies of the same network, each passing a message to a successor. In fact, *unrolling* it leads to a chain-like architecture, made of multiple replicas of the same module and such that each module passes a message to its successor.

b) The standard procedure for training a RNN is known as *backpropagation through time* (BPTT) [116]. Unluckily, it may not be so effective when training involves long time sequences, because of the so called *vanishing* and the *exploding gradient* problems [117]. The former problem refers to the exponential decrease observed in the norm of the gradient of the employed cost function during training, whereas the latter one concerns the opposite behaviour (more specifically, a large increase of the same gradient).

The problems mentioned in the last point can be circumvented by adopting a *long short term memory* (LSTM) neural network [118], whose architecture is illustrated in Fig. 5.13-b). This architecture consists of a *memory cell* and of three different *multiplicative gates*, namely an input gate, an output gate and a forget gate. The input gate, whose content at time t is denoted by $\mathbf{i}^{(t)}$, represents the input of the memory cell (whose content at time t is denoted by $\mathbf{c}^{(t)}$) and is employed to protect the content of this cell from perturbations due to irrelevant inputs. The output gate, whose content at time t is denoted by $\mathbf{o}^{(t)}$, protects the other units connected to the output of the memory cell from perturbations due to irrelevant memory contents. Finally, the forget gate, whose content at time t is denoted by $\mathbf{f}^{(t)}$, protects the contents stored in the vector $\mathbf{c}^{(t)}$ from the unwanted fluctuations of the memory at the previous instance (i.e., from $\mathbf{c}^{(t-1)}$).

In summary, the cell allows for long term memory storage, whereas the gates prevent memory contents from being perturbed by irrelevant inputs and outputs.

If $\mathbf{r}_q^{(t)}$ denotes the vector of input features at time t, the time evolution of the LSTM network shown in Fig. 5.13-b) is described by the equations

$$\mathbf{i}^{(t)} = \sigma(\mathbf{U}_i\,\mathbf{r}_q^{(t)} + \mathbf{W}_i\,\mathbf{s}^{(t-1)}), \tag{5.131}$$

$$\mathbf{f}^{(t)} = \sigma(\mathbf{U}_f\,\mathbf{r}_q^{(t)} + \mathbf{W}_f\,\mathbf{s}^{(t-1)}), \tag{5.132}$$

$$\mathbf{o}^{(t)} = \sigma(\mathbf{U}_o\,\mathbf{r}_q^{(t)} + \mathbf{W}_o\,\mathbf{s}^{(t-1)}), \tag{5.133}$$

$$\mathbf{g}^{(t)} = \sigma_c(\mathbf{U}_g\,\mathbf{r}_q^{(t)} + \mathbf{W}_g\,\mathbf{s}^{(t-1)}), \tag{5.134}$$

$$\mathbf{c}^{(t)} = \mathbf{c}^{(t-1)} \odot \mathbf{f}^{(t)} + \mathbf{g}^{(t)} \odot \mathbf{i}^{(t)} \tag{5.135}$$

and

$$\mathbf{s}^{(t)} = \sigma_c(\mathbf{c}^{(t)}) \odot \mathbf{o}^{(t)}; \tag{5.136}$$

here, $\sigma(\cdot)$ is a logistic sigmoid vector function, $\sigma_c(\cdot)$ is an hyperbolic tangent vector function, the operator $\odot$ denotes the Hadamard product, $\mathbf{s}^{(t)}$ is the output of the memory cell at time t, $\mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_o$ and $\mathbf{U}_g$ ($\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o$ and $\mathbf{W}_g$) are weight matrices characterizing the multiplicative gates and referring to the vector $\mathbf{r}_q^{(t)}$ ($\mathbf{s}^{(t-1)}$), and $\mathbf{g}^{(t)}$ can be interpreted as a candidate state, whose influence on the state $\mathbf{c}^{(t)}$ is controlled by the input gate through $\mathbf{i}^{(t)}$. From eqs. (5.131)-(5.136) it is easily inferred that: a) the contents of the input, output and forget gates at time t are proportional to a combination of both the vectors $\mathbf{r}_q^{(t)}$ and $\mathbf{s}^{(t-1)}$; b) the output state $\mathbf{s}^{(t)}$ depends not only on the cell content $\mathbf{c}^{(t)}$, but also on the content of the output gate (i.e. on $\mathbf{o}^{(t)}$).

Let us focus now on the application of an LSTM neural network to an FMCW radar system equipped with a single TX-RX pair and detecting a person that accomplishes specific activities and, in particular, runs or walks. In this case, each observation processed by the LSTM involves N_f consecutive frames, in each of which N_c chirps are transmitted (see paragraph 2.3). For this reason, the q-th observation processed by the considered network is extracted from $N_f\,N_c$ noisy vectors, acquired over N_f consecutive frames, (i.e., over $N_f\,N_c$ consecutive chirps). In the p-th frame (with $p = 0, 1, ..., N_f - 1$), the set of vectors $\{\mathbf{x}_{p,q}^{(0)}, \mathbf{x}_{p,q}^{(1)}, ..., \mathbf{x}_{p,q}^{(N_c-1)}\}$, each having size N, is available; here,

$$\mathbf{x}_{p,q}^{(k)} = \left[x_{p,q,0}^{(k)}, x_{p,q,1}^{(k)}, ..., x_{p,q,N-1}^{(k)}\right], \tag{5.137}$$

represents the vector of signal samples acquired in the k-th chirp interval of the p-th frame and its n-th sample $x_{p,q,n}^{(k)}$ is expressed by a formula similar to eq. (2.19) (with $n = 0, 1, ..., N - 1$). In the proposed experiment, the *Phased Array System* toolbox available in the MATLAB environment is employed to generate the useful signal component (i.e., the contribution of the detected person) to the vector $\mathbf{x}_{p,q}^{(k)}$ (5.137) [119]. This contribution is modelled as the superposition of L different echoes, each originating from a point-like target and associated with a different part of the body. Moreover, in the p-th frame contributing to the q-th observation, the l-th point target is characterized by its RCS $a_{p,l}^{(q)}$,

its range $R_{p,l}^{(q)}$ and its radial velocity $v_{p,l}^{(q)}$ (with $p = 0, 1, ..., N_f - 1$ and $l = 0, 1, ..., L - 1$). These parameters are assumed to be static over each frame; in addition, the values they take on in the p-th frame are automatically computed by the above mentioned toolbox on the basis of the height h_p of the person, its position and its RCS in the previous frame, the direction of its movement with respect to the radar system and its radial velocity v_q. The dataset processed by the network is

$$\mathcal{D}_o \triangleq \{(\mathbf{R}_q, t_q); q = 0, 1, ..., \hat{N}_t - 1\}, \tag{5.138}$$

where

$$\mathbf{R}_q \triangleq \left[\left(\mathbf{r}_0^{(q)}\right)^T, \left(\mathbf{r}_1^{(q)}\right)^T, ..., \left(\mathbf{r}_{N_f-1}^{(q)}\right)^T \right]^T, \tag{5.139}$$

is the q-th noisy observation, t_q its label,

$$\mathbf{r}_p^{(q)} \triangleq [\hat{R}_p^{(q)}, \hat{v}_p^{(q)}]^T, \tag{5.140}$$

and $\hat{R}_p^{(q)}$ and $\hat{v}_p^{(q)}$ are the estimates of the range $R_p^{(q)}$ and of the velocity $v_p^{(q)}$, respectively, of the considered person in the p-th frame (with $p = 0, 1, ..., N_f - 1$); moreover, it is assumed that $t_q = 1$ (-1) if the person is walking (running), i.e. if $|v_p^{(q)}| \leq v_{th}$ ($|v_p^{(q)}| > v_{th}$), being v_{th} a proper threshold.

In the proposed experiment, the dataset $\mathcal{D}_o$ (5.138) has been acquired in $\hat{N}_t = 400$ independent trials; half of the labels of this dataset are associated with a walker and the remaining half with a runner; moreover, the estimates $\hat{R}_p^{(q)}$ and $\hat{v}_p^{(q)}$ are computed by the algorithm consisting of the following two steps:

1. *Range Estimation* - In this step, the N-dimensional vector $\mathbf{x}_{p,q}^{(k)}$ (5.137) undergoes zero padding; this results in the N_0-dimensional vector $\mathbf{x}_{p,q,\mathrm{ZP}}^{(k)}$, with $N_0 \triangleq MN$ (here, the parameter M represents the selected oversampling factor). The last vector feeds a N_0-th order FFT, whose output is the N_0-dimensional vector $\mathbf{X}_{p,q}^{(k)} = [X_{p,q,0}^{(k)}, X_{p,q,1}^{(k)}, ..., X_{p,q,N_0-1}^{(k)}]^T$. Then, the average power spectrum

$$P_m^{(q)} \triangleq \frac{1}{N_c} \sum_{k=0}^{N_c-1} |X_{p,q,m}^{(k)}|^2, \tag{5.141}$$

is computed. Finally, $\hat{R}_p^{(q)}$ is evaluated as (see eqs. (2.55), (2.57) and (2.59))

$$\hat{R}_p^{(q)} = \frac{c}{2\mu} \hat{f}_p^{(q)}, \tag{5.142}$$

where $\hat{f}_p^{(q)} = \hat{m}_p^{(q)} / N_0 T_s$ and

$$\hat{m}_p^{(q)} = \arg \max_{m \in \{0,1,...,N_0/2\}} P_m^{(q)}. \tag{5.143}$$

2. *Velocity Estimation* - This step is based on the N_c-dimensional vector

$$\hat{\mathbf{C}}_{p,q} = [\hat{C}_{p,q}^{(0)}, \hat{C}_{p,q}^{(1)}, ..., \hat{C}_{p,q}^{(N_c-1)}]^T, \tag{5.144}$$

where

$$\hat{C}_{p,q}^{(k)} = M X_{p,q,\hat{m}_p^{(q)}}^{(k)} \tag{5.145}$$

and $\hat{m}_p^{(q)}$ is expressed by eq. (5.143). Applying zero padding to this vector produces the $N_0^{'}$-dimensional vector $\hat{\mathbf{C}}_{p,q}^{(\mathrm{ZP})}$, with $N_0^{'} \triangleq M_A N_c$ (here, the parameter M_A represents the selected oversampling factor); the last vector feeds a $N_0^{'}$-th order FFT, whose output is the $N_0^{'}$-dimensional vector

$$\mathbf{d}_{p,q} \triangleq \left[d_{p,q}^{(0)}, ..., d_{p,q}^{(N_0'/2)}, d_{p,q}^{(-N_0'/2+1)}, ..., d_{p,q}^{(-1)} \right]^T. \tag{5.146}$$

After solving the maximization problem

$$\hat{k}_p^{(q)} = \arg \max_{\tilde{k} \in \{-N_0'/2+1, -N_0'/2+2, ..., N_0'/2\}} \left| d_{p,q}^{(\tilde{k})} \right|, \tag{5.147}$$

the estimate (see eqs. (2.16) and (2.57))

$$\hat{v}_p^{(q)} = \frac{1}{2} f_p^{(q)} \lambda \tag{5.148}$$

of the person velocity is evaluated; here,

$$f_p^{(q)} \triangleq \frac{2\hat{k}_p^{(q)}}{N_0' T_0} \tag{5.149}$$

represents the Doppler frequency estimated in p-th frame.

For any q, in generating the sequence of pairs $\{(\hat{R}_p^{(q)}, \hat{v}_p^{(q)}); t = 0, 1, ..., N_f - 1\}$, the following assumptions have been made about the detected person:

a) its response to the signal radiated by the radar system consists of $L = 16$ echoes;

b) its height h_p is uniformly distributed over the interval $(1.70\,,\, 2.0)$ m;

c) its initial coordinates in a 3D space are $(x_0^{(q)}, y_0^{(q)}, z_0^{(q)}) = (0{,}10{,}0)$ m, whereas the coordinates of the employed radar device in the same reference system are $(x_r, y_r, z_r) = (0{,}0{,}1)$ m;

d) the angle ϕ_i representing the initial direction of its velocity is uniformly distributed over the domain $(60^\circ\,,\, -60^\circ) \cup (120^\circ, 180^\circ)$ (the reference line, with respect to which this angle is measured, is perpendicular to the array of the radar system).

e) the radial velocity $v_p^{(q)}$ is uniformly distributed over the interval $(0.1,\, 2.1)$ m/s $((-2.1,\, -0.1)$ m/s) if $\phi_i \in (60^\circ\,,\, -60^\circ)$ ($\phi_i \in (120^\circ,\, 180^\circ)$) for any p and, in each frame, changes in an independent fashion;

f) the initial amplitude $a_{0,l}^{(q)}$ is equal to 1V (with $l = 0,\, 1,\, ...,\, L-1$).

Moreover, the following choices have been made for the employed radar system:

a) the TX waveform has $\lambda = 4$ mm, $\mu = 1.5625 \cdot 10^{13}$ Hz s^{-1}, $T = 64\ \mu$s, $T_0 = 72\ \mu$s;

b) each frame consists of $N_c = 128$ chirps;

c) consecutive frames are separated by a time interval lasting $\Delta t = 40$ ms;

d) the sampling period employed at the receive side is $T_s = 12.5\,\mu$s;

e) the overall number of time-domain samples acquired in each chirp interval is $N = 1024$ and the standard deviation of the noise affecting each sample is $\sigma_w = 0.1$ V (see eq. (2.19));

f) the oversampling factors $M = 2$ and $M_A = 8$, and the threshold $v_{th} = 1.1$ m/s are selected for the range/estimation algorithm illustrated above;

g) each observation refers to $N_f = 30$ consecutive frames.

The representation, on a Cartesian plane, of two different feature vectors (see eqs. (5.139)-(5.140)), is provided in Fig. 5.14.

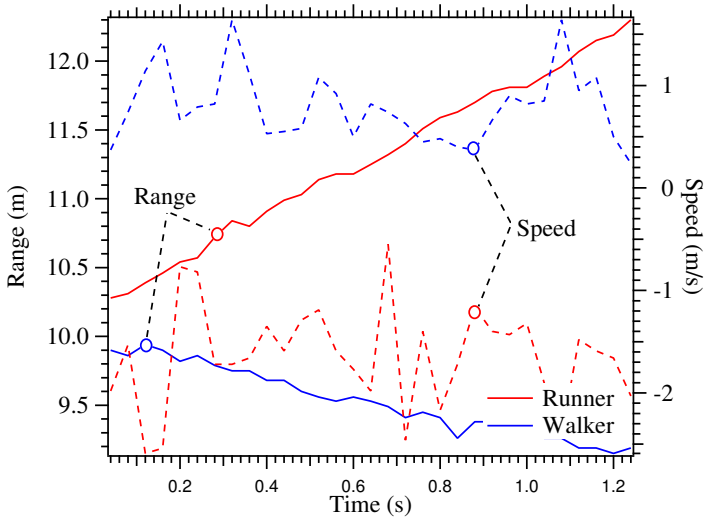


FIGURE 5.14: Representation of the elements of the two feature vectors $\mathbf{R}_0$ and $\mathbf{R}_1$; one refers to a runner (red lines), the other one to a walker (blue lines).

These vectors are denoted by $\mathbf{R}_0$ and $\mathbf{R}_1$; the former refers to a runner, whereas the latter to a walker. Note that the range difference $\Delta\hat{R}_0 \triangleq |\hat{R}^{(0)}_{N_f-1} - \hat{R}^{(0)}_0|$ referring to the runner is greater than the corresponding quantity (i.e., $\Delta\hat{R}_1 \triangleq |\hat{R}^{(1)}_{N_f-1} - \hat{R}^{(1)}_0|$) referring to the walker. In this case, the proposed LSTM network is employed to discriminate a walker from a runner. The core of its architecture is characterized by an LSTM layer, able to learn the long term dependencies between different frames. The behaviour of network is described by the block diagram shown in Fig. 5.13 and by eqs. (5.131)-(5.135) (the time index t corresponds to the frame index p in this case). Moreover, in this experiment, the following choices have been made:

1) the size of the input vector is $D_r = 2$, whereas that of the inner state is $\bar{M} = 10$;

2) the non-linear gate activation function $\sigma(x) = [1 + \exp(-x)]^{-1}$ is used;

3) the size of each of the weight matrices $\{\mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_g, \mathbf{U}_o\}$ is $\bar{M} \times D_r = 10 \times 2$, whereas that of the weight matrices $\{\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_g, \mathbf{W}_o\}$ is $\bar{M} \times \bar{M} = 10 \times 10$;

5) both the initial cell content $\mathbf{c}^{(0)}$ and the initial state $\mathbf{s}^{(0)}$ are independently chosen as random vectors of size $\bar{M} = 10$;

6) a fully connected layer and a softmax layer (see paragraph 5.3.8) have been added at the output of the LSTM layer to perform classification.

The *adaptive moment estimation* (briefly, *adam*) optimizer [120] has been exploited to train the proposed network (i.e., to tune all the above mentioned weighted matrices); the batch size, the (constant) learning rate and the number of epochs selected for this procedure are $N_S = 32$, $\gamma = 10^{-3}$ and $N_E = 50$, respectively (see eq. (5.51)). Moreover, a training set $\mathcal{D}$ of size $N_t = 300$, corresponding to 75% of the dataset $\mathcal{D}_o$ (5.138) has been employed for training; the remaining part $\mathcal{D}_{ts}$ of $\mathcal{D}_o$ has been used as a test set (collecting $\bar{N}_t = 100$ observations)). Using these parameters a 98% accuracy is achieved by the adopted LSTM network in distinguishing people that are running from others that are walking.

The obtained results suggest that:

a) Combining deterministic estimators with deep learning methods can result in classification techniques achieving excellent performance;

b) Merging range and velocity information can enhance the discrimination capability of the network;

c) Observing range/velocity evolution over time (i.e., over multiple consecutive frames) significantly contribute to improve network accuracy.

5.3.7 Generative adversarial networks

A *generative adversarial network* (GAN) is a probabilistic generative method consisting of two deep neural networks, called *generator* and *discriminator*, and competing against one another [121]; its architecture is shown in Fig. 5.15. The generator produces a sample[14] $\mathbf{x} = \mathbf{G}(\mathbf{z}, \boldsymbol{\theta}_g)$ from a pdf $f_g(\mathbf{x})$, starting from an input noise variable $\mathbf{z} \sim f_z(\mathbf{z})$; here, $\mathbf{G}(\cdot,\cdot)$ is called *generative model* and is typically implemented through a neural network, whereas $\boldsymbol{\theta}_g$ is the vector of training parameters.

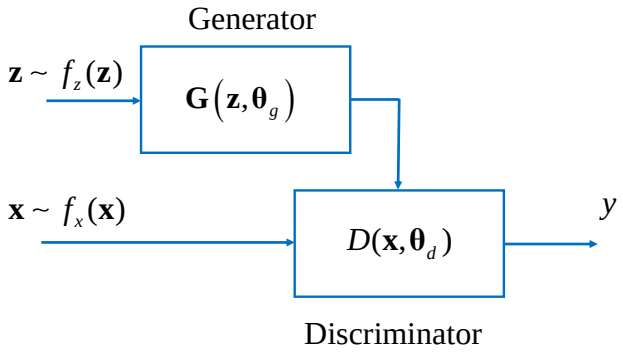


FIGURE 5.15: Architecture of a generative adversarial network.

The discriminator, instead, generates the output $y = D(\mathbf{x}, \boldsymbol{\theta}_d)$, that represents the probability that an input $\mathbf{x}$ originates from the training data (i.e., from their pdf $f_d(\mathbf{x})$) rather than from the generator pdf $f_g(\mathbf{x})$; here, $D(\cdot,\cdot)$ represents the *discriminative model* and $\boldsymbol{\theta}_d$ is the vector of the training parameters characterizing the network that implements the model itself. In practice, the generative model can be thought as a team of counterfeiters, trying to produce fake currency for fooling the police, while the discriminator, acting like the police, tries to detect the counterfeit currency. Competition in this game drives both teams to improve their methods. In fact, the objective of the training of the generative network is minimizing the accuracy of the discriminative network when the data generated by the former network are provided to the latter one; on the contrary, the objective of the discriminator is maximizing the probability of assigning the correct label to both the real data of the training set and the fake samples originating from the generator. For this reason, the interaction between the discriminator and the generator can be modelled as a *two-player minimax game*. This leads to formulating the optimal strategy of these networks as the solution of the minimax problem

$$\min_{\mathbf{G}} \max_{D} V(D, \mathbf{G}) = \min_{\boldsymbol{\theta}_g} \max_{\boldsymbol{\theta}_d} V(D, \mathbf{G}), \tag{5.150}$$

where

$$V(D, \mathbf{G}) \triangleq \mathrm{E}_{\mathbf{x}\sim f_d(\mathbf{x})} \left\{ \log D\left(\mathbf{x}, \boldsymbol{\theta}_d\right) \right\} + \mathrm{E}_{\mathbf{z}\sim f_z(\mathbf{z})} \left\{ \log\left(1 - D\left(\mathbf{G}\left(\mathbf{z}, \boldsymbol{\theta}_g\right), \boldsymbol{\theta}_d\right)\right) \right\}. \tag{5.151}$$

The backpropagation algorithm can be used for training a GAN; the training process allows the discriminator of the considered GAN to learn, through a proper feature

[14] Scalar variables are considered in this paragraph, for simplicity.

representation, how to identify real inputs among the generated data and, similarly, the generator how to generate realistic data.

Generative adversarial networks have the favourable property that a wide variety of functions can be incorporated into their model; these make them able to represent very sharp (and even degenerate) data distributions. However, their use require the availability of efficient tools to solve the minimax optimization problem (5.150). Moreover, a tight synchronization between the generator and the discriminator has to be guaranteed during training; in fact, if one of the two networks learns too quickly, the other one may fail to learn.

5.3.8 Softmax Classification Layer

Generally speaking, the DL methods illustrated above can be employed to extract the relevant features of an image. Once this result has been obtained, any multi-class problem referring to that image can be solved by adding a *softmax layer* to the employed network. If K classes are assumed, the target of this layer is generating the posterior probability

$$p_i = \frac{\exp(a_i(\mathbf{r}^{(L)}))}{\sum_{j=0}^{K-1} \exp(a_j(\mathbf{r}^{(L)}))} \tag{5.152}$$

for the i-th class, with $i = 0, 1, ..., K-1$; here,

$$a_j(\mathbf{r}^{(L)}) = \mathbf{w}_j^T \mathbf{r}^{(L)} + w_{j,0} \tag{5.153}$$

and $\mathbf{r}^{(L)}$ is an L-dimensional feature vector made available by the previous hidden (convolutional or LSTM) layer, and $\mathbf{w}_j$ and $w_{j,0}$ are an L-dimensional weight vector and a bias term, respectively, characterizing the softmax layer.

5.4 Applications of learning techniques to MIMO radars

In this section some applications of the learning methods illustrated in sections 5.2 and 5.3 to MIMO radar systems are proposed. More specifically, these methods are exploited in the following fields: a) human motion characterization; b) human gesture recognition (HGR); c) fall detection and health-care monitoring; d) autonomous driving.

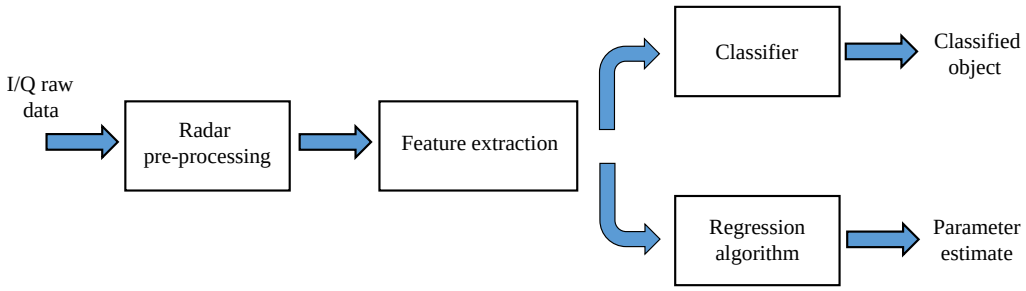


FIGURE 5.16: Block diagram representing the signal processing chain of a MIMO radar system that employs a learning method.

Before delving into the analysis of each application, it is worth pointing out that the typical processing accomplished at the receive side of any MIMO radar system employing a learning method for classification and/or regression is based on the block diagram shown in Fig. 5.16. First, the received signal undergoes frequency downconversion to

generate its in phase and quadrature components. Sampling these components produces a stream of raw data, which is pre-processed (e.g., it may undergo FFT processing; see paragraphs 2.3 and 2.5) before extracting relevant features from it. Finally, these features are processed by a classifier or by a regression algorithm; in the former case, a specific object class is selected, whereas, in the latter one, an estimate of the parameters of interest is evaluated. Feature extraction is based on the prior knowledge about the employed radar system if a ML method is exploited; on the contrary, features are automatically selected and extracted from pre-processed data if a DL method is adopted. In the following paragraphs, various details about the processing accomplished by the blocks appearing in Fig. 5.16 are provided.

5.4.1 Human motion characterization

Human motion characterization aims at recognizing and classifying different human activities on the basis of the *micro-movements* of the detected targets; such movements usually originate from mechanical vibrations or rotations (overlapping to a bulk translation) and may generate a frequency modulation in the received signal; this phenomenon is also known as *micro-Doppler*. These micro-Doppler fluctuations observed in the spectrograms of radar signals are known as *micro-Doppler signatures*[15]. Classifiers employed for this application aim at: a) identifying different types of human motion (e.g., walking, running and sitting) [122]; b) differentiating human motion from that of other living animals [123]; c) remotely identifying potential active shooters [124], [125].

In the technical literature about this application, the following two methods are exploited to extract relevant features from micro-Doppler signatures, also called spectrograms: a) manual extraction of *handcrafted* features; b) automatic extraction of features based on a *data-driven* approach. Machine learning methods exploiting manual extraction of features have been investigated in [125]–[133], whereas the automatic extraction of features from spectrograms through DL methods has been proposed in [122], [124], [134]–[137]. It is important to bear in mind that:

1. Machine learning methods relying on spectrogram information usually exploit: a) *physical features* related to the characteristics of the observed motion; b) *transform-based* features; c) *speech-inspired* features. Physical features include the frequency and the bandwidth of the received waveforms, the offset and the signal intensity of the associated signature (see paragraph 2.3). The first two physical features are strictly related to motion dynamics, whereas the other types of features to the RCS of the body of the observed person [123].

2. Transform-based features exploited by ML methods can be extracted from a received waveform by evaluating a) its spectral coefficients (e.g., its *discrete cosine transform* coefficients) or b) other signal-dependent coefficients. As far as point b) is concerned, the use of *linear predictive coding* (LPC) has been proposed in ref. [128] to transform a time-varying Doppler signal in a low dimensional set of prediction coefficients. A different approach, based on the computation of pseudo-Zernike moments, is illustrated in ref. [129, Sect. II-A, eqs. (10)-(12)]; this allows to extract relevant characteristics from micro-Doppler signatures, such as translational and scale invariance.

[15] Additional details regarding the detection of micro-Doppler signatures through the computation of spectrograms and cadence velocity diagrams are provided in Par. 5.6

3. In many cases, the dimensionality of the feature space can be substantially reduced (see paragraph 5.2.4). An interesting example of this approach is offered in ref. [130], where the use of a 1D standard PCA and of a robust PCA for extracting physical features from a Doppler radar signal is investigated.

Frequently used ML methods for the classification of human motion based on a set of handcrafted features include the *naive Bayes* (NB) [131], the non linear SVM [132] and the K-NN methods [133]. The use of a binary SVM classifier for multi-class problems in human motion characterization is investigated in ref. [126], where a classification procedure based on a *decision-tree* is proposed. This procedure is based on the idea of representing a classification problem involving multiple choices as a set of binary classification problems, each of which is solved through a binary SVM. This approach is exemplified in [126, Fig. 8], where a decision tree referring to the case of seven classes is illustrated. In practice, a binary SVM is employed for each node of the employed decision tree in order to separate the possible activities in two groups; if each of the two groups is further divided, another SVM classifier is used at an underlying node.

The most relevant problems emerging from the study of ML-based classification of human motion concern the processing methods to be employed for the extraction of hand-crafted features from raw micro-Doppler signals, the sensitivity of these methods to noise and clutter, and the impact of similarities among the considered classes on their performance. The ability of a *deep neural network* to learn the relevant features directly from the available raw data allows to solve the above mentioned problems. This consideration has motivated the investigation of *deep* CNNs (DCNNs; see paragraph 5.3.4) for the automatic extraction of features in human motion characterization. The use of a DCNN, fed by spectrograms (converted in *red green blue*, RGB, or greyscale images), and employing convolutional layers and pooling layers of small size, has been proposed in ref. [134]. A different DL method, based on the same principles as convolutional autoencoding (see paragraph 5.3.5), has been developed in ref. [135, Par. IV-C, Fig.8]. It combines the ability of a DCNN to capture local features of input images with that of an AE to directly learn features through an unsupervised pre-training procedure. In this case, after an initial and unsupervised pre-training stage, the decoder of a CAE is substituted by a few fully connected layers and a softmax classifier. This procedure allows the resulting DCNN to learn specific patterns from the processed signatures, so easing training for supervised classification. The performance results obtained in this case lead to the conclusion that a CAE not only is able to outperform conventional classification methods based on handcrafted features (e.g., SVM), but also a standard DCNN.

Finally, it is useful to mention that another important research problem investigated in the field considered in this paragraph is represented by the *de-noising of micro-Doppler spectra*. In this case, the training set includes two different types of images: a) perfectly clean spectrograms; b) the same spectrograms affected by background noise. The use of a deep GAN, based on a convolutional encoder-decoder structure, has been proposed in ref. [136] for this application. The performance results obtained in this case evidence that this network does not affect the relevant components of micro-Doppler spectra and is able to outperform other classic de-noising techniques commonly used for the suppression of background noise.

5.4.2 Human gesture recognition

The significant attention paid to HGR is due to its exploitation in advanced *human computer interfaces* (HCIs), that are employed in a number of control, infotainment and security applications. Relevant information about the dynamics of human gestures are typically contained in the micro-Doppler signatures acquired over consecutive transmitted frames. Therefore, similarly as human activity characterization, relevant physical features can be easily extracted from spectrograms. A commonly employed ML tool for classifying vectors of handcrafted features in HGR systems is represented by *hidden Markov modelling* [138]. This approach leads to classifying a new sequence of data, called observation, on the basis of a stochastic model, called *hidden Markov model* (HMM), which has been extracted from past observations and describes their generation. If an HMM of a given random phenomenon is available, the probability of observing a specific realization (e.g., a specific gesture), conditioned on a given sequence of hidden states, can be computed. In this case, model training aims at estimating the so called *transition* and *emission* probability matrices of the developed HMM; the former matrix collects the probabilities to move from a given state to another one, while the latter one the probabilities that a given observation is generated in each specific state.

The efficacy of a HMM-based classifier depends on the overall number of states characterizing the model; in general, a larger number of states allows to model a more complicated process and to improve prediction accuracy. However, a discrete state space of small size is often adopted in HGR applications to mitigate the overall complexity of the developed HGR system. This choice makes the resulting classifier unable to distinguish gestures characterized only by subtle differences in their spectrograms. For this reason, DCNNs are usually preferred.

One of the first important research activities focusing on the exploitation of this type of networks in HGR is the well known Google's *Soli project* [139], whose scope has been the development of a HGR mobile and wearable device based on a RF sensor. Various research results about this research field can be found in ref. [140], where it is shown that the accuracy of these deep classifiers gets worse if: a) the number of classes[16] increases; b) the incident angle and/or the distance between the gesture and the employed radar device get larger. The accuracy of a classifier based on a DCNN can be improved by extracting features not only from spectrograms, but also from range-Doppler maps [141]. Another DL architecture, specifically developed for RF HGR and combining the ability of a CNN network of capturing local features of input images with that of coping with time-varying signals, has been investigated in ref. [142]. This architecture consists of a 3D-CNN for spatial-temporal modelling of short consecutive frames, an LSTM for extracting global temporal features and a final classification layer (a detailed block diagram is illustrated in ref. [142, Fig. 7]). This architecture achieves a very high recognition accuracy, and outperforms other conventional ML and DL methods used in HGR applications, like HMMs or 2D-CNNs.

5.4.3 Fall detection and health-care monitoring

Human falls represent a worldwide health problem and are known to be one of the main causes of unintentional injury death in seniors; this motivates the recent interest in devising electronic systems able to detect their occurrence. Another important problem in the field of technology for human health concerns the development of non-invasive and

[16] The maximum number of distinct hand gestures considered in ref. [140] is equal to 10.

non-contact devices for monitoring human vital signs, such as breath and heart rates, and sleep quality. Various results in both research areas have evidenced that innovative solutions to both problems can be developed by exploiting ML and DL methods fed by the micro-Doppler signatures acquired through a radar system. In any case, when the overall number of classes to be identified increases and the degree of dissimilarity between the Doppler signatures characterizing them reduces, DL methods are preferred, since they achieve better accuracy.

An interesting study on the dynamics of human falls analysed through micro-Doppler signatures can be found in in ref. [143], where it is shown that fall accidents can be distinguished from normal activities on the basis of: a) the strength of the received echo (i.e., the RCS of the subject under test); b) the distance of the radar device from the body of the subject under test during a fall; c) the Doppler information acquired during the movement of the subject itself. Experimental results have evidenced that, when a subject starts falling, the observed Doppler frequency increases steeply; on the contrary, the RCS of the human subject gradually decreases since its tilt angle gets larger. In this case, ML and DL algorithms can be trained to detect a fall on the basis of the time variations of Doppler signatures. A specific DL classifier based on a stacked AE and exploiting a range-Doppler radar has been developed in ref. [144], where it is shown that the proposed solution is more accurate than PCA-based methods in detecting different actions, such as falling, walking, sitting and bending.

The use of learning techniques in the analysis of sleep stages has been investigated in ref. [145], where a solution based on a K-NN classifier has been proposed. The exploitation of learning techniques for heart and breath rate estimation represents a challenging problem, because a large and heterogeneous datasets for network training cannot be easily built and contactless systems for vital sign monitoring are strongly limited by body movements. Some interesting contributions to this field are provided by refs. [146], [147] and [148]. More specifically, a method based on a classical feed-forward NN for hearth rate estimation is proposed in ref. [146], whereas a DL method for body movement compensation is investigated in ref. [147]. Finally, a contactless breathing disorder recognition system using 2.4-GHz Doppler radar and based on a linear SVM classifier is developed in ref. [148].

The real-time implementation of radar sensing methods for HGR, health monitoring and fall detection can be computationally intensive. This problem becomes more relevant in all those applications in which multiple persons have to be monitored in the same environment; in fact, in such cases, the exploitation of the MIMO technology becomes mandatory, because of the need of localising multiple agents. This explains why an important technical challenge is represented by the exploitation of hardware platforms that support parallel computing (namely, FPGAs and GPUs), require a limited power consumption and can manage a large data rate at their inputs [149].

5.4.4 Autonomous driving

Automotive radar represents one of the key enabling technologies for autonomous driving. In a typical processing chain employed for target detection in a MIMO FMCW radar system for automotive applications the signals acquired through multiple receive antennas undergo multidimensional FFT processing; this allows to extract range, Doppler

and DoA information. The data generated by the FFT blocks are processed by a detection algorithm, whose objective is identifying the presence of multiple extended targets, and estimating their spatial coordinates and their radial velocity (i.e., the Doppler shift characterizing them). Each of the detected targets (e.g., pedestrians, cars or bicycles) usually appears as a cloud of point targets; the association of each point to a given extended target is called *clustering*. The simplest unsupervised strategy that can be adopted for target detection is *thresholding*; in this case, a target is detected if the amplitude spectrum of the acquired signals exceeds a fixed threshold, as illustrated in paragraphs 2.5 and 5.2.3 for specific FMCW radar systems. A more refined alternative is represented by the *constant false alarm rate* (CFAR) technique [150]. This method consists in estimating the level of interference in each cell in the range domain of interest and in exploiting these information for the detection of the presence of a target in each cell of a radar image. Unluckily, due to the high resolution achieved by automotive radars, a single target can occupy multiple adjacent cells; when this occurs, the CFAR technique undergoes performance degradation because of the contamination affecting the estimated interference level. Clustering techniques rely on the key idea that each cluster of points is a region containing a group of detected targets, whose center typically corresponds to the point target characterized by the strongest reflectivity (see Par. 5.2.4). This means that each cluster has a density (in terms of targets per region) which is considerably larger than that outside it; for this reason, a given point is expected to be part of a cluster if the number of its neighbours is greater than a proper threshold. Learning methods for unsupervised clustering include the *density based clustering algorithm* (DBSCAN) [151], [152], and the K-means algorithm [153]. The main difference between these two methods consists in the fact that the former method, unlike the latter one, does not require prior knowledge of the number of clusters and their shape.

All the techniques described above (namely, thresholding, CFAR and clustering) allow to detect multiple point targets and to cluster them. In general, learning methods can be adopted to improve detection performance. A number of technical problems have been identified in this area; most of them require the development of sophisticated signal processing algorithms. Specific contributions about the use of ML methods in target detection can be found in refs. [154]–[156]. In particular, a K-NN classifier is proposed as an alternative to robust CFAR detection in ref. [156], whereas the use of the SVM and PCA techniques for improving angular resolution is investigated in ref. [154]. The use of DL methods for target classification in a 2D space, instead, have been studied in refs. [102], [157]–[164]. It is worth mentioning that, in the literature, the first results about the use of DL methods in automotive radar systems appeared after 2015, when it was found that DCNNs were able to simultaneously detect, localize and classify multiple targets by simply analysing 2D range-azimuth (or range-Doppler) maps. Networks originally developed for computer vision applications, like *AlexNet* [165] or *ImageNet* [166], have inspired the architecture of various networks for automatically extracting features from automotive radar images [102], [157], [158]. Despite this, the CNNs usually devised for automotive applications are not as deep as those employed in computer vision. This difference is mainly due to the fact that: a) the information provided by range-azimuth or range-Doppler maps are not as rich as traditional RGB images; b) the employed inference procedure has to be as fast as possible [159]. These ideas are exemplified by the CNN proposed in ref. [160] for the classification of automotive targets, like motorcycles, cars, bicycles and pedestrians; its architecture, illustrated in ref. [160, Fig. 2], consists of three convolutional layers and filters of size 3×3 (whose depths are equal to 32, 64 and 128, respectively). Moreover, each convolutional layer is followed by a 2×2 average-pooling layer, two fully-connected layers and a softmax layer, which

is used at the end for classification. A relevant novelty introduced in this work (and in ref. [161] too) consists in considering a certain *region of interest* (ROI) around the desired targets in the analysed scene as prior information to be used during training, in order to improve the learning procedure.

Deep learning methods can be also employed to solve the problem of *scene understanding*, i.e. of correctly interpreting the events occurring around it (e.g., the event of a vehicle passing near a pedestrian that crosses a road). In this case, improving the prediction accuracy of the employed NN requires exploiting the information contained in the frames preceding and following the frame under test because of the high variability of the data provided by MIMO radar systems. An architecture based on the cascade of a LSTM module with a CNN has been proposed in ref. [162]; this exploits the temporal information provided by radar signals and is able to capture the dynamics of the surrounding scene.

Finally, it is worth mentioning that learning methods can be also employed to detect the fatigue of the driver's eyes [155] and to mitigate the interference orginating from the transmission of multiple MIMO radars in the same area. In general, the interference affecting a MIMO radar system can be due to the system itself (*self-interference*) or from other radar systems placed on the same vehicle or on other vehicles (*cross or mutual interference*); in both cases, this phenomenon results in an increase of the observed noise floor and, consequently, affects the detectability of targets. The use of RNNs for interference mitigation has been investigated in refs. [163] and [164].

5.5 Current trends in research on MIMO radars

In this section, a short description of three research trends in the field of DL techniques for MIMO radars is provided. More specifically, the focus will be on *transfer learning*, and recent DL methods for object detection and classification. Then, the role that *explainable artificial intelligence* (XAI) may play in the radar field during the next years will be analysed in the following.

5.5.1 Transfer learning

The minimization procedure accomplished by a deep NN trained from scratch (through random initialization) may lead to a local minimum which is far from the globally optimal solution if the involved cost function is highly non-convex. Moreover, if the dataset employed in network training is not large enough, the risk of over-fitting is quite high. These problems are likely to arise in radar applications, where the availability of a large and heterogeneous data set is quite rare. When this occurs, *transfer learning* could represent a tool to solve them; in fact, this method often allows to achieve a good generalization capability even if the available dataset is limited [167], [168]. Transfer learning is based on the idea of exploiting the knowledge gained from a different domain to solve other related classification problems. Two approaches to the exploitation of this method in radar applications have been recently proposed. The first approach, developed for the classification of human activities, is based on training an *unsupervised network*, characterized by an encoder-decoder structure and employed to learn specific patterns appearing in the available dataset [169]. When the decoder becomes able to reconstruct the input data with a reasonable accuracy, it is removed, and fully connected and softmax layers are added in cascade to the associated encoder. Finally, the resulting network is trained in a supervised manner with a smaller, but labelled dataset: this procedure is called

fine-tuning.

The second approach is based on the architecture developed in ref. [170]. In this case, a DCNN network trained on a large dataset of RGB images is combined with fully connected and softmax layers initialized from scratch; this results in a new network, which is fine-tuned on a small dataset. The decision about which type of transfer learning has to be preferred is based on the size of the available dataset and on the similarity of the last dataset with the one used for pre-training the selected network architecture. It has been shown that the final score of a DCNN-based classifier can be improved either by exploiting a pre-training procedure based on a simulated radar dataset [171] or by employing a pre-trained DCNN on a separate large scale RGB dataset [170].

5.5.2 Object detection and classification

The aims of object detection and classification are the labelling of all the objects appearing in a given image and the generation of a bounding box identifying their position. The *fast* R-CNN [172] and *faster* R-CNN [173] are examples of *region-based* CNNs for object detection based on bounding boxes. Another relevant solution of this type is represented by the so called *You only look once* (YOLO) network[17]. When building up a dataset for training this network, each detectable target is bounded with a box characterized by specific size and position in the whole image. If an object detection problem in which different targets can be associated to several (say, K) classes is considered, the YOLO network should be preferred to the other methods mentioned above because of: a) its ability to predict not only the size and the position of the bounding box associated with a given target, but also the probability that the target inside a given box belongs to a certain class; b) its architecture which, being based on a CNN, is simple and fast; c) its ability to learn very general representations of objects. The results illustrated in ref. [174] for various applications evidence that a YOLO network outperforms a R-CNN in terms of detection ability, since it produces a lower number of false negatives. However, it is important to underline that a YOLO network usually makes a significant number of localization errors and, consequently, achieves a limited accuracy. Better results are obtained if an improved architecture, known as YOLO v2 and originally proposed in ref. [175], is adopted. This new version of the YOLO network is still based on a convolutional architecture, but employs *anchor boxes*[18] in predicting the position of objects. The use of anchor boxes makes the learning procedure easy, since the network has only to adjust and refine their size in order to fit an object detected in the processed image. A specific application of the YOLO v2 network to a MIMO radar system is illustrated in paragraph 5.6.2.

A recent research topic in the field of target detection and classification is represented by the use of *semantic segmentation*, that represents a powerful technique adopted for classifying the pixels of an image (a fixed set of classes is assumed in this case). The state of the art in semantic segmentation for image processing is represented by: a) *fully convolutional networks* (FCNs) [176], in which a convolutional network endowed with a pixel classification layer (instead of a fully connected layer) is used; b) SegNet [177] and U-Net [178], both based on a symmetrical encoder-decoder architecture. A more complex method is represented by *instance segmentation*, whose aim is not only detecting and classifying all the objects appearing in an image, but also generating the segmentation

[17] The name of this network has been inspired by the human ability of looking once at an image and instantly recognizing the objects it contains.

[18] Anchor boxes are a set of predefined bounding boxes having certain height and width.

of each instance appearing in the bounding box associated with each detected object. To accomplish the last task, the Facebook AI research group has proposed a new method called, *Mask*-R-CNN, that extends a Faster R-CNN by adding a branch for the prediction of the segmentation mask in each ROI [179].

It is important to note that the application of the above mentioned DL techniques to object detection and localization in radar images is still at an early stage. Despite this, specific DL methods inspired by FCNs and U-Net have already been implemented for detecting and estimating the position of different targets (like cars and other automotive targets) on the basis of range-Doppler-azimuth radar maps [180]–[182]. Moreover, the use of semantic segmentation in the radar field has been already investigated for the classification and localization of 3D point clouds of automotive targets, like cars, tractors and pedestrians; various results referring to automotive MIMO radars that operate at 77 GHz can be found in refs. [183] and [184], [185] . The experimental results shown in these manuscripts evidence that the performance of the NNs employed for semantic segmentation substantially improves if radar data are fused with those one provided by others (typically optical) sensors. It should not be forgotten, for instance, that radar information can be augmented by a highly dense point cloud generated by a lidar device and that lidar data can be replaced by radar data in case of adverse weather or lighting conditions. An example of radar-centric automotive dataset based on radar, lidar and camera data for is described in ref. [186]; this dataset has been exploited in ref. [187] to test DL algorithms for 3D object detection.

5.5.3 Explainable artificial intelligence

Neural networks and sophisticated decision methods are currently employed in a number of applications to solve complicated tasks. The requirement of *transparency* is becoming more and more important in AI, especially when it is employed in autonomous systems. Unluckily, understanding which features are evaluated by a DNN in making decision is a complicated problem. Explainable artificial intelligence is a new branch of AI and concerns the problem of how the effectiveness of a deep network can be guaranteed [188]. An interesting method to improve the transparency of a DNN is based on the visualization of the features learned by each layer of the network [189]. The first layers of a DCNNs tested on radar images typically learn basic features, that depend on the size of their convolution filters. In fact, large (small) filters memorize general shapes (more specific properties), whereas some filters are also able to learn noise and clutter [135]. An alternative method to get some insight on the learning process of a CNN is based on the idea of identifying the parts of a radar image that are relevant for the classification of the object under test; such parts are also known as *spatial supports*. This approach allows to assess if a specific network is robust in making decision on the basis of a correct analysis of the given image. A specific technique, called *saliency extraction*, is based on this idea and, in particular, on the evaluation of the so called *saliency map*, as illustrated in ref. [190].

5.6 Experimental results

In this section it is shown how specific ML and DL methods can be employed in a commercial colocated MIMO radar system to: a) classify three different human activities; b) estimate the range and DOA (azimuth) of a single target in a 2D propagation scenario. In both cases, such methods are compared, in terms of accuracy and processing time; moreover, in case b), a comparison with deterministic methods is also carried out.

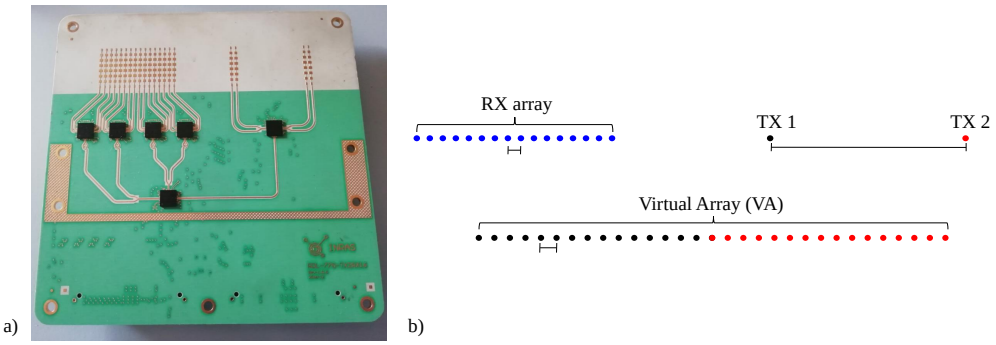


FIGURE 5.17: a) Colocated MIMO radar system and b) Geometry of the physical TX and RX arrays (top) and the corresponding virtual array (bottom).

It is worth stressing that, unlike the previous sections, the results illustrated below do not originate from a synthetically generated dataset. In fact, the following tools have been exploited to generate them[19]:

1. A colocated FMCW MIMO radar manufactured by *Inras GmbH* [191]. This radar device, shown in Fig. 5.17-a) and employed to acquire all the measurements, operates in the E-band (the center frequency of its transmitted signal is $f_0 = 77$ GHz) and is equipped with a TX ULA and an RX ULA, consisting of $N_T = 2$ and $N_R = 16$ antennas, respectively (see Fig. 5.17-a)); even if, in principle, $2 \cdot 16 = 32$ virtual channels are available, only $N_V = 31$ of them are exploited in this work, since two elements of the virtual array overlap (see Fig. 5.17-b)).

2. A *pico-flexx camera* manufactured by *PMD Technologies Inc.* [90]. This time-of-flight camera is employed as a reference sensor, being able to provide a depth map or, equivalently, a three-dimensional point-cloud of a small region of the observed environment. A more complete description of this sensor has been provided in Sect. 4.5.

3. A desktop computer equipped with a single i7 processor. All the software has been developed in the MATLAB and/or Python environment and run on this computer.

In the following two paragraphs, various details about the experiments accomplished for the two specific applications mentioned above and the most relevant obtained results are illustrated.

5.6.1 Human activity classification

The first experiment concerns the classification of following three different human activities: *walking*, *running* and *jumping*. The following choices have been made:

1. The person whose activity has to be classified is alone and is in front of the employed radar device (at the bore-sight respect to the radar system).

2. A single pair of TX-RX antennas is used (since angular information is not required).

3. The transmitted waveform is characterized by the following parameters: $N_c = 128$, $T = 128$ μs, $T_R = 32$ μs and $B = 1$ GHz (consequently, $\mu = 7.8 \cdot 10^{12}$ GHz/s; see eq. (2.6)).

[19] The datasets employed in these experiments are available at: https://www.sigcomm.unimore.it/downloads/.

4. At the receive side, analog-to-digital conversion is accomplished at the sampling frequency $f_s = 80$ MHz and $N = 1024$ samples are acquired over each chirp period and an oversampling factor $M = 4$ is considered.

Different classification methods have been tested for this application. First of all, the following five ML methods are taken into consideration: a linear SVM technique, the K-NN technique (with $K = 4$), an Adaboost classifier with decision stumps as weak learners (see paragraph 5.2.3), a *customised* double stage SVM binary classifier (CSVM) and a specific version of the Adaboost, called *Stagewise Additive Modeling using a Multiclass Exponential loss function* (SAMME) [192]. As far as DL methods are concerned, only a specific CNN has been taken into consideration, since, as shown below, the preprocessed data feeding it can be interpreted as 2D images.

All these methods are fed by the matrices $\mathbf{E}$ and $\mathbf{G}$, i.e. the *spectrogram* and the *cadence velocity diagram* (CVD), respectively. In a FMCW radar system equipped with a single TX antenna and a single RX antenna, these maps are generated as follows. Let assume that $\mathbf{x}^{(k)}$ denotes the N-dimensional (column) vector consisting of the real measurements acquired in the k-th chirp of a transmission frame, with $k = 0, 1, ..., N_c - 1$, where N_c is the overall number of chirps forming the frame itself. The N_c vectors $\{\mathbf{x}^{(k)}; k = 0, 1, ..., N_c - 1\}$ are collected in the matrix

$$\mathbf{R} = [\mathbf{x}^{(0)}\, \mathbf{x}^{(1)}\, \cdots\, \mathbf{x}^{(N_c-1)}]\,, \tag{5.154}$$

having size $N \times N_c$. This matrix undergoes zero-padding, that turns it into a matrix $\mathbf{R}_{\mathrm{ZP}}$ of size $N_0 \times N_0^{'}$. The last matrix feeds a $N_0 \times N_0^{'}$-th order FFT, that generates the *range-Doppler* (complex) *matrix*

$$\mathbf{D} = \left[d_{p,q}\right] \triangleq \mathrm{FFT}_{N_0 \times N_0^{'}}\left[\mathbf{R}\right], \tag{5.155}$$

where $\mathrm{FFT}_{X\times Y}[\cdot]$ denotes 2D FFT operator of size $X \times Y$; note that the index p (q) labelling the elements of the matrix $\mathbf{D}$ refers to the range (Doppler) domain. Representing, on a Cartesian plane, the absolute value of the elements of the matrix $\mathbf{D}$ generates the range-Doppler map.

Let us suppose now that N_f consecutive frames (each consisting of N_c chirps) are transmitted by the considered radar system and that the range-Doppler matrix $\mathbf{D}$ (5.155) is evaluated for each frame (the matrix referring to the m-th frame is denoted by $\mathbf{D}_m = [d_{p,q}^{(m)}]$, with $m = 0, 1, ..., N_f - 1$). The micro-Doppler *signatures*, characterizing a certain range interval, can be acquired through the real matrix $\mathbf{E} = \left[E_{m,q}\right]$, having size $N_f \times N_0^{'}$ and whose element on its m-th row and q-th column is evaluated as

$$E_{m,q} \triangleq \sum_{p=p_{\min}}^{p_{\max}} |d_{p,q}^{(m)}|^2 \tag{5.156}$$

with $m = 0, 1, ..., N_f - 1$ and $q = 0, 1, ..., N_0^{'} - 1$; here, $p_{\min}$ ($p_{\max}$) denotes the value of the index p associated with the minimum (maximum) range of interest. Representing the elements of the matrix $\mathbf{E}$ on a Cartesian plane produces the so called *spectrogram* [193], that shows the time evolution of the Doppler phenomenon. Additional information about the dynamical properties of a moving target can be acquired through the cadence velocity diagram. Its generation is based on the complex matrix $\mathbf{G} = [G_{l,q}]$, having size $N_f^{'} \times N_0^{'}$ and computed as the $N_f^{'} \times N_0^{'}$-th order FFT of the matrix $\mathbf{E}_{ZP} = [E_{m,q}^{(ZP)}]$, that

results from zero padding of the matrix **E** defined above; therefore, we have that

$$G_{l,q} \triangleq \frac{1}{N_f} \sum_{m=0}^{N_f'-1} E_{m,q}^{(ZP)} \exp\left(-j2\pi \frac{m}{N_f'} \bar{f}_l \, T_F\right) \tag{5.157}$$

with $l = 0, 1, ..., N_f' - 1$ and $q = 0, 1, ..., N_0' - 1$; here, T_F is the duration of a single transmission frame, $E_{m,q}^{(ZP)} = E_{m,q}$ for $m = 0, 1, ..., N_f - 1$ and $E_{m,q}^{(ZP)} = 0$ for $m > N_f - 1$ and

$$\bar{f}_l \triangleq \frac{l}{T_F} \tag{5.158}$$

is the l-th *cadence frequency*. The CVD results from representing, on a Cartesian plane, the absolute value of the elements of the matrix **G**. In all the experiments made for the considered application, $N_f = 143$, $N_0' = 256$ and $N_f' = 512$. Examples of the spectrograms associated with the three possible activities are shown in Figs. 5.18-(a), -(b) and -(c) (note that the same time scale is used in all these figures), whereas an example of CVD is illustrated in Fig. 5.19.

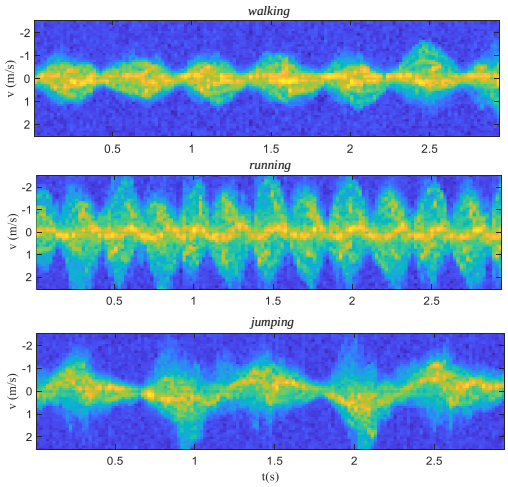


FIGURE 5.18: Spectrograms observed for the following three different activities: walking (top), running (center) and jumping (bottom).

Moreover, in the last figure, two additional plots, one referring to the cadence frequency of the observed motion (left), the other one to its velocity (bottom), are also given for completeness. From Figs. 5.18-5.19 it is easily inferred that:

a) The period of the spectrogram (i.e., the distance between its consecutive peaks) is inversely proportional to the speed of the observed motion.

b) The shape of the spectrogram is influenced by the type of motion.

c) The CVD diagram contains important information regarding the motion and it is strictly related to the shape of the spectrogram. The cadence frequencies indicate how frequently a specific velocity component repeats in the observation interval.

An experimental campaign has been accomplished to build up an experimental dataset, that collects $N_t = 150$ observations equally divided among the three classes. Each observation refers to N_f consecutive frames, each consisting of N_c chirps, and is acquired over an observation interval whose duration is $T_O = 3$ s (each frame lasts $T_F = T_O / N_f = 21$ ms).

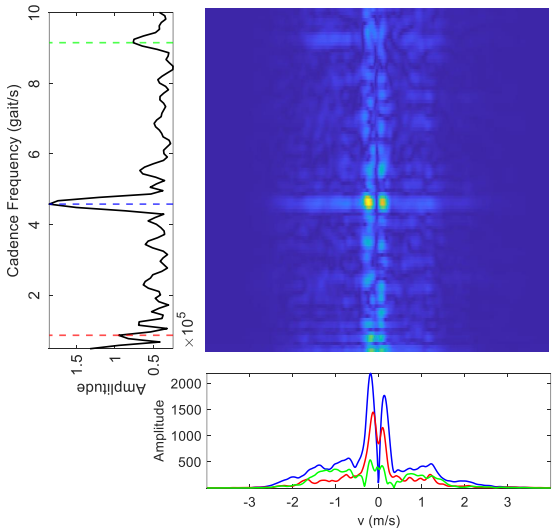


FIGURE 5.19: Representation of a CVD and of two diagrams extracted from it (one providing information about cadence frequencies, the other one about velocities). In the diagram appearing on the left, the three strongest frequency components are identified by blue, red and green dashed lines; each line is associated with the velocity profile shown in the other diagram and having the same colour.

Moreover, the q-th entry of the dataset $\mathcal{D}_o$ processed by the above mentioned ML methods is represented by the pair $(\mathbf{r}_q, t_q)$ (see eq. (5.2)), where

$$\mathbf{r}_q = \left[r_{q,0}, r_{q,1}, r_{q,2}, r_{q,3}\right] \tag{5.159}$$

is a 4D feature vector (so that $D_r = 4$) and t_q is a integer label identifying the specific activity which the vector $\mathbf{r}_q$ is associated with ($t_q = 0$, 1 and 2 if the observed person is walking, running or jumping, respectively). The first three elements of the vector $\mathbf{r}_q$ (5.159) depend on the value $\mathbf{G}_q = [G^{(q)}_{l,m}]$ of the matrix $\mathbf{G}$ computed for the q-th observation, since

$$r_{q,0} \triangleq \frac{\hat{l}_q}{N'_f \; T_F}, \tag{5.160}$$

$$r_{q,1} \triangleq \frac{1}{N'_0} \sum_{m=0}^{N'_0-1} \left(G^{(q)}_{\hat{l}_{q,0},m} - \mu^{(q)}_0 \right) \cdot \left(G^{(q)}_{\hat{l}_{q,1},m} - \mu^{(q)}_1 \right) \tag{5.161}$$

and

$$r_{q,2} \triangleq \frac{1}{N'_0} \sum_{m=0}^{N'_0-1} \left(\left(G^{(q)}_{\hat{l}_{q,0},m} \right)^2 + \left(G^{(q)}_{\hat{l}_{q,1},m} \right)^2 \right); \tag{5.162}$$

here,

$$\hat{l}_q \triangleq \arg \max_{l \in \{0,1,\ldots,N'_f-1\}} V^{(q)}_l, \tag{5.163}$$

$\hat{l}_{q,k}$ is the index identifying the k-th largest peak appearing in the sequence $\{V^{(q)}_l; l = 0, 1, \ldots, N'_f - 1\}$ (with $k = 0$ and 1),

$$V^{(q)}_l \triangleq \sum_{m=0}^{N'_0-1} G^{(q)}_{l,m}, \tag{5.164}$$

and $\mu_k^{(q)}$ is the mean of the elements of the $\hat{l}_{q,k}$-th row of the matrix $\mathbf{G}_q$, i.e. of the vector

$$\mathbf{G}_{\hat{l}_{q,k}}^{(q)} \triangleq \left[G_{\hat{l}_{q,k},0}^{(q)}, G_{\hat{l}_{q,k},1}^{(q)}, ..., G_{\hat{l}_{q,k},N_0'-1}^{(q)} \right]^T, \tag{5.165}$$

with $k = 0$ and 1. The last feature of $\mathbf{r}_q$ (5.159) (namely, the quantity $r_{q,3}$) depends on the value $\mathbf{E}_q$ of the matrix $\mathbf{E}$ computed for the q-th observation, since it represents the period of the spectrogram, i.e. the distance between two consecutive peaks observed along the time dimension. It is important to point out that:

1. The parameter $r_{q,0}$ (5.160) represents the strongest frequency component detected in the CVD diagram (see Fig. 5.19). The value of this parameter is expected to increase with the speed of the observed person.

2. The parameters $\hat{l}_{q,0}$ and $\hat{l}_{q,1}$ identify the two strongest frequencies (denoted by $\hat{f}_{q,0}$ and $\hat{f}_{q,1}$, respectively) detected in the CVD referring to the q-th observation; such frequencies are evaluated as

$$\hat{f}_{q,k} = \frac{\hat{l}_{q,k}}{N_f' \, T_F}, \tag{5.166}$$

   with $k = 0$ and 1.

3. The parameter $\mu_k^{(q)}$ is the mean of the *velocity profile* expressed by the N_0'-dimensional vector $\mathbf{G}_{\hat{l}_{q,k}}^{(q)}$ (5.165).

4. The parameter $r_{q,1}$ (5.161) represents the covariance between the velocity profiles $\mathbf{G}_{\hat{l}_{q,0}}^{(q)}$ and $\mathbf{G}_{\hat{l}_{q,1}}^{(q)}$, whereas $r_{q,2}$ is the overall energy associated with both profiles; this experimental data have evidenced that the value of $r_{q,1}$ ($r_{q,2}$) decreases (increases) as the speed of the observed person gets larger (smaller).

5. The value of $r_{q,3}$ is inversely proportional to the speed of the observed person, since an increase of the speed shortens the period of the spectrogram.

As far as the adopted ML methods are concerned, the following choices have been made:

a) The K-NN classifier is structured as illustrated in paragraph 5.2.3.

b) The classifiers based on the SVM and the Adaboost methods exploit the pairwise classification approach illustrated at the end of paragraph 5.2.3. For this reason, they combine $L = K(K-1)/2 = 3$ *identical* binary classifiers (i.e., base learners).

c) The CSVM method is obtained by cascading two linear SVM binary classifiers (whose behaviour is described in paragraph 5.2.3). The first SVM classifier (denoted by SVM #1) distinguishes jumping from the rest of the activities and is fed by the feature vector $\mathbf{r}_q' = \left[r_{q,0} \, , \, r_{q,1} \right]$ (in this case, the scalar labels $t_q' = 1$ and $t_q' = -1$ are associated with jumping, and with walking and running, respectively). The second classifier (SVM #2) processes the observations related to running and walking only and is fed by the feature vector $\mathbf{r}_q'' = \left[r_{q,2} \, , \, r_{q,3} \right]$ (in this case, the scalar labels $t_q'' = 1$ and $t_q'' = -1$ are associated with walking and running, respectively). The final predictions of the CSVM are generated on the basis of the SVM #1 (SVM #2) predictions for jumping (running and walking).

d) The employed version of the SAMME method is the one implemented in the Python library Scikit-learn [194] (namely, `sklearn.ensemble.AdaBoostClassifier`) and represents a specific version of the Adaboost technique for solving multi-class problems;

in practice, it is based on a decision tree classifier characterized by two nodes (instead of a simple decision stump). This methods outperforms a classical Adaboost technique by simply emphasizing the weights assigned to misclassified points.

In this experiment, a $\bar{N}$-fold cross-validation, with $\bar{N} = 5$, has been employed. The accuracy achieved by the considered ML methods and the processing time they have required for training and prediction are listed in Table 5.1.

	SVM	K-NN	ADA	CSVM	SAMME
Accuracy (%)	89	90	91	90	91
Training time (s)	0.1	0.03	4.5	0.06	0.45
Prediction time (s)	0.01	0.01	0.5	0.01	0.05

TABLE 5.1: Accuracy, training time and prediction time evaluated for each of the ML methods considered for human activity classification.

From these results, it is easily inferred that:

a) The accuracy is reasonably good in all cases (slightly above the 90%).

b) The Adaboost performs marginally better than the K-NN and SVM methods, at the price of substantially larger computation time.

c) The best trade-off in terms of performance and computation time is achieved by the K-NN technique.

d) The CSVM method requires a lower computational effort (especially in training) with respect to the method based on SVM and round-robin binarization. This is mainly due to the fact that the former approach employs only two learners, whereas the latter one three binary classifiers.

e) The SAMME algorithm achieves the same accuracy as the round-robin binarization of the classic Adaboost, even if its computation time (in both training and prediction) is approximately ten times smaller.

The ML methods tested in the first part of the experiment exploit a dataset of manually extracted features (see eqs. (5.159)-(5.162)). On the contrary, the CNN employed in the second part of the experiment is able to classify human activities by recognizing specific patterns directly in the matrix $\mathbf{E}$. A description of its architecture is provided in Table 5.2. The first three layers of the employed network are represented by three convolutional 2D filters, having size 15×5 and depths 4, 8 and 16; moreover, each filter feeds a linear rectifier, followed by a max pooling layer. Each max pooling layer allows to halve the size of the image made available by the previous layer, so that a significant dimensionality reduction is obtained. The first three layers are followed by another 2D convolutional filter with a batch normalization layer. The last layers are represented by a *fully-connected* (FC) and a *softmax* (Soft) layer transforming the residual 2D image in a vector of size 3, since three classes are considered. It is worth noting that the adoption of a CNN having a small depth is justified by the fact that spectrograms referring to the three activities are quite different, as exemplified by Fig. 5.18. The q-th entry of the dataset $\mathcal{D}_o$ processed by the employed CNN is represented by the pair $(\mathbf{r}_q, t_q)$, where, however, the observation $\mathbf{r}_q$ is represented by the value $\mathbf{E}_q$ taken on by matrix $\mathbf{E}$ in the q-th acquisition (the label t_q, instead, has the same meaning as in the ML case).

Layers	Filters	Size	Stride	Output
Convolutional + ReLu	4	15×5	1	$143 \times 53 \times 4$
Max pooling	-	15×5	2	$65 \times 25 \times 4$
Convolutional + ReLu	8	15×5	1	$65 \times 25 \times 8$
Max pooling	-	15×5	2	$26 \times 11 \times 8$
Convolutional + ReLu	16	15×5	1	$26 \times 11 \times 16$
Max pooling	-	3×3	2	$12 \times 5 \times 16$
Convolutional + BN + ReLu	3	3×3	1	$12 \times 5 \times 3$
FC + Soft	3	-	-	$1 \times 1 \times 3$

TABLE 5.2: Architecture of the CNN employed for the classification of three human activities.

Moreover, $N_f = 143$ and $\bar{N}_0' = 53$, and $\hat{N}_t = 150$ are assumed for the size of the matrix $\mathbf{E}$ and for the dataset $\mathcal{D}_o$, respectively. Network training is based on 60% of the whole dataset (the remaining part of the dataset has been equally divided to generate a validation set and a test set); moreover, it has been accomplished by an SGD minimization procedure, which is characterized by a subset S of 4 training data samples, a learning rate $\gamma^{(i)} = 10^{-3}$ for any i and an overall number of epochs $N_E = 50$ (see eq. (5.51)). A 96% classification accuracy has been achieved in this case; the most of misclassification events occurs between walking and running, especially when the speed during run is quite low and comparable to that one followed during walking. The proposed DL method achieves a substantially better generalization capability that the ML counterparts described above. We should not forget, however, that this result is achieved at the price of a training time of about 25 s; this is substantially larger than that required by the considered ML methods (see Table 5.1). Finally, it is important to mention that the computation time required by the employed CNN for evaluating a new prediction is about 0.03 s and, consequently, is reasonably short and comparable with the one characterizing the considered ML methods.

5.6.2 Estimation of the range and azimuth of a single target

The second proposed application concerns the detection of a specific target moving on a 2D multi-target scenario, and the estimation of its range and azimuth. In this experiment, the target to be detected is an omnidirectional reflector, obtained by merging eight corner reflectors[20]. This target is mounted, through a vertical carton support, on a Propeller Scribbler 3 mobile robot manufactured by Parallax Inc [195]. This robot has been programmed to move randomly inside a square white region delimited by four opaque black lines and whose side is equal to 2.5 m, as shown in Fig. 5.20; note that two corner reflectors have placed on the borders of this region in order to build a multi-target scenario. The following choices have been made in the acquisition of measurements:

1. The whole antenna array shown in Fig. 3.7-a) is exploited, so that $N_V = 31$ distinct virtual channels are available at the receive side.

2. The waveform radiated by each TX antenna is characterized by the following parameters: $N_c = 1$, $T = 64$ μs, $T_R = 32$ μs and $B = 2$ GHz (consequently, $\mu = 3.13 \cdot 10^{13}$ GHz/s; see eq. (2.6)).

[20] Its structure is inspired by the architecture of the *echo-master* corners used for maritime applications

3. At the receive side, analog-to-digital conversion is accomplished at the sampling frequency $f_s = 40$ MHz and $N = 2048$ samples are acquired over each chirp period.

4. The reference position of the target with respect to a three-dimensional reference system is evaluated by means of the pico-flexx camera. This sensor is aligned with the radar system, being mounted on the same plastic support of the radar device and at a fixed distance from it (about 10 cm) along the vertical direction.

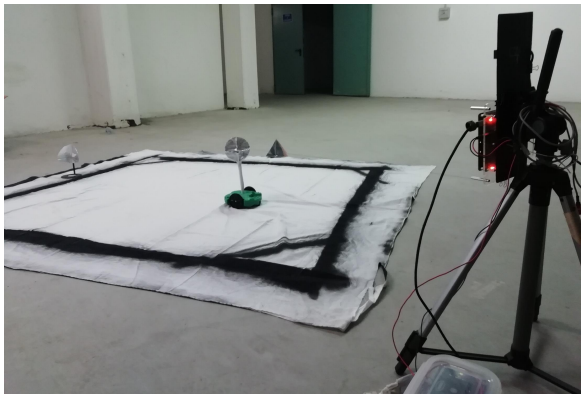


FIGURE 5.20: Experimental-setup developed for the second application. The region of interest is delimited by an opaque and black line; two coner reflectors are located on its border. A robot, equipped with corner reflectors, moves randomly inside that area. The employed radar system and pico-flexx camera are placed on the tripod visible on the right.

The following two supervised DL methods have been tested: a) a feed-forward NN exploiting some manually extracted features (further details about this method are provided below); b) a YOLO v2 NN for object detection (see paragraph 5.5.2). The q-th entry $(\mathbf{r}_q, \mathbf{t}_q)$ of the dataset $\mathcal{D}_o$ processed by the employed feed-forward NN is generated as follows. The label $\mathbf{t}_q$ associated with the q-th observation $\mathbf{r}_q$ is defined as

$$\mathbf{t}_q \triangleq [\hat{R}_q, \hat{\phi}_q] \tag{5.167}$$

where $\hat{R}_q$ and $\hat{\phi}_q$ represent the estimates of the target range R_q and azimuth ϕ_q, respectively, evaluated on the basis of the point-cloud made available by the pico-flexx camera. Such a camera generates the $N_p \times 3$ matrix

$$\mathbf{P} \triangleq [\mathbf{x}\ \mathbf{y}\ \mathbf{z}], \tag{5.168}$$

collecting the 3D coordinates of $N_p = 38304$ distinct points; here, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ are N_p-dimensional column vectors. The deterministic algorithm developed for the estimation of the target range and azimuth involves the computation of the estimates $(\hat{x}_q, \hat{y}_q, \hat{z}_q)$ of the target coordinates (x_q, y_q, z_q) in the q-th observation; note that z_q (i.e., the target height) is assumed to be approximately known ($z_q \cong 0.4$ m). This algorithm consists of the following three consecutive steps:

1. The size of the search space for the pair $(\hat{x}_q, \hat{y}_q)$ is reduced by extracting the set[21]

$$\mathcal{S}_q \triangleq \{(x_{q,n}, y_{q,n}, z_{q,n}) | z_{\min} \leq z_{q,n} \leq z_{\max}; n \in \Delta_q\}, \tag{5.169}$$

[21] Note that, in this step, a prior knowledge about the target height is exploited.

from the matrix $\mathbf{P}$ (5.168); here, Δ_q is a proper subset of the set of integers $\{0, 1, ..., N_p - 1\}$ and consists of $\bar{N}_q$ elements, whereas $z_{\min} = 0.3$ m and $z_{\max} = 0.5$ m represent two thresholds.

2. The estimates

$$\hat{x}_q = 1/\bar{N}_q \sum_{n=0}^{\bar{N}_q - 1} x_{q,n}, \tag{5.170}$$

$$\hat{y}_q = 1/\bar{N}_q \sum_{n=0}^{\bar{N}_q - 1} y_{q,n} \tag{5.171}$$

and

$$\hat{z}_q = 1/\bar{N}_q \sum_{n=0}^{\bar{N}_q - 1} z_{q,n} \tag{5.172}$$

are computed. The estimate $\hat{z}_q$ (5.172) is exploited only to check if the vector $(\hat{x}_q, \hat{y}_q, \hat{z}_q)$ is meaningful, i.e. if the condition $\hat{z}_q \approx 0.4$ m is satisfied; if this does not occur, the thresholds appearing in the RHS of eq. (5.169) should be properly adjusted (i.e., $z_{\min}$ should be increased and/or $z_{\max}$ reduced) in order to improve the obtained accuracy.

3. The estimates

$$\hat{R}_q = \sqrt{\hat{x}_q + \hat{y}_q} \tag{5.173}$$

and

$$\hat{\phi}_q = \arctan\left(\hat{y}_q / \hat{x}_q\right) \tag{5.174}$$

are evaluated.

The observation $\mathbf{r}_q$ labelled by $\mathbf{t}_q$ (5.167) is defined as[22]

$$\mathbf{r}_q \triangleq \left[\hat{\psi}_q^{(0)}, \hat{\psi}_q^{(1)}, ..., \hat{\psi}_q^{(N_V - 1)}, \hat{f}_q\right]^T, \tag{5.175}$$

where $\hat{f}_q$ is the frequency associated with the detected target (and estimated on the whole array) and $\hat{\psi}_q^{(v)}$ is the phase of the signal spectrum computed at the frequency $\hat{f}_q$ for the v-th virtual element (with $v = 0, 1, ..., N_V - 1$); note that the size of the vector $\mathbf{r}_q$ (5.175) is $D_r = N_V + 1 = 32$. The deterministic algorithm employed for the computation of the frequency $\hat{f}_q$ and the phases $\{\hat{\psi}_q^{(v)}\}$ forming $\mathbf{r}_q$ (5.175) consists of the following two steps:

1. *Coarse estimation of the target position* - The N-dimensional vector of the time domain samples acquired over the v-th virtual antenna (see eq. (2.48)) undergoes zero padding and FFT processing of order $N_0 = N \cdot M$ (in this experiment, $N_0 = 8192$, since $M = 4$). This produces the N_0-dimensional vector $\mathbf{X}_q^{(v)}$ (see eq. (2.50)), which is employed to compute the *power spectrum* $\mathbf{P}_q^{(v)} = [P_{q,0}^{(v)}, P_{q,1}^{(v)}, ..., P_{q,N_0-1}^{(v)}]$ on the basis of eq. (2.53). Then, given (see eq. (2.55))

$$\hat{l}_q^{(v)} \triangleq \arg \max_{\tilde{l} \in \{b_m, ..., b_M\}} P_{q,\tilde{l}}^{(v)}, \tag{5.176}$$

[22] *Unwrapped* phases are employed in this case, since they ease network training

a target is detected on the v-th antenna if $P^{(v)}_{q,\hat{l}_q^{(v)}} > P_d$, where P_d is a proper threshold; here, the integer parameter b_m (b_M) identifies the frequency bin corresponding to the minimum (maximum) measurable range R_m (R_M). In this experiment, $P_d = 0.9$, and

$$b_m = \left\lfloor \frac{2\mu N_0 T_s R_m}{c} \right\rfloor = 42 \tag{5.177}$$

and

$$b_M = \left\lfloor \frac{2\mu N_0 T_s R_M}{c} \right\rfloor = 147, \tag{5.178}$$

since $R_m = 1.0$ m and $R_M = 3.5$ m have been assumed. The procedure illustrated above is accomplished for each virtual channel (i.e., for $v = 0, 1, ..., N_V - 1$) and is employed to generate the set

$$\mathcal{S}_{\hat{l}} \triangleq \{\hat{l}_q^{(v_k)}; k = 0, 1, ..., N_A - 1\}, \tag{5.179}$$

with $v_k < v_{k+1}$ for any k; the size N_A of this set is usually smaller that N_V, since: a) the target may be missed on one or more virtual channels (this occurs when the condition (5.176) is not satisfied); b) the elements of $\mathcal{S}_{\hat{l}}$ are required to be distinct. The elements of $\mathcal{S}_{\hat{l}}$ are collected in the vector $\hat{\mathbf{l}}_q = [\hat{l}_q^{(v_0)}, \hat{l}_q^{(v_1)}, ..., \hat{l}_q^{(v_{N_A-1})}]^T$. Then, the following vectors are computed: a) the N_A-dimensional vector $\hat{\mathbf{f}}_q = [\hat{f}_q^{(v_0)}, \hat{f}_q^{(v_1)}, ..., \hat{f}_q^{(v_{N_A-1})}]^T$ and $\hat{\mathbf{R}}_q = [\hat{R}_q^{(v_0)}, \hat{R}_q^{(v_1)}, ..., \hat{R}_q^{(v_{N_A-1})}]^T$, that collect N_A estimates of the target frequency and range, respectively (these quantities computed on the basis of eqs. (2.57) and (2.59), respectively); b) the set of N_A vectors $\{\hat{\mathbf{C}}_q^{(v_k)}; k = 0, 1, ..., N_A - 1\}$, where $\hat{\mathbf{C}}_q^{(v_k)} = [\hat{C}_{q,0}^{(v_k)}, \hat{C}_{q,1}^{(v_k)}, ..., \hat{C}_{q,N_V-1}^{(v_k)}]^T$ is made of the complex amplitudes evaluated over the whole virtual array on the basis of eq. (2.58) under the assumption that $\hat{l} = \hat{l}_q^{(v_k)}$ for any k; c) the set of N_A vectors $\{\hat{\boldsymbol{\psi}}_q^{(v_k)}; k = 0, 1 , ..., N_A - 1\}$, where

$$\hat{\boldsymbol{\psi}}_q^{(v_k)} = [\hat{\psi}_{q,0}^{(v_k)}, \hat{\psi}_{q,1}^{(v_k)}, ..., \hat{\psi}_{q,N_V-1}^{(v_k)}]^T \tag{5.180}$$

and $\hat{\psi}_{q,l}^{(v_k)}$ is equal to the phase of the complex gain $\hat{C}_{q,l}^{(v_k)}$ for any k and l (see eq. (2.37)). Finally, each of the vectors $\{\hat{\mathbf{C}}_q^{(v_k)}\}$ undergoes zero padding, that increases their size to $\bar{N}_0 = 128$, and $\bar{N}_0$-th order FFT processing for azimuth estimation (see eqs. (2.67)-(2.69)). This produces the vector $\hat{\boldsymbol{\phi}}_q = [\hat{\phi}_q^{(v_0)}, \hat{\phi}_q^{(v_1)}, ..., \hat{\phi}_q^{(v_{N_A-1})}]^T$, collecting N_A different estimates of the target azimuth. Therefore, this step produces N_A distinct estimates $\{(\hat{f}_q^{(v_k)}, \hat{R}_q^{(v_k)}, \hat{\phi}_q^{(v_k)}); k = 0, 1 , ..., N_A - 1\}$ of the target frequency, range and azimuth, respectively.

2. *Fine estimation of the target position* - A *single* estimate of the target frequency, range and azimuth is evaluated in this step on the basis of the N_A estimates $\{(\hat{f}_q^{(v_k)}, \hat{R}_q^{(v_k)}, \hat{\phi}_q^{(v_k)})\}$ available at the end of the previous step. This estimate is computed as follows. First, the quantity

$$\hat{v}_q = \min_{\tilde{l} \in \mathcal{S}_{\hat{l}}} \left| \hat{\phi}_q - \hat{\phi}_q^{(v_{\tilde{l}})} \right|, \tag{5.181}$$

under the constraint

$$\left| \hat{R}_q - \hat{R}_q^{(v_{\tilde{l}})} \right| < R_{th}, \tag{5.182}$$

with $R_{th} = 0.3$ m is computed; here, the quantities $\hat{R}_q$ and $\hat{\phi}_q$ are expressed by eq. (5.173) and eq. (5.174), respectively, and $\mathcal{S}_{\hat{l}}$ is the set defined by eq. (5.179). Then, the vector $\mathbf{r}_q$ (5.175) is evaluated as

$$\mathbf{r}_q = [\hat{\boldsymbol{\psi}}_q^{(\hat{v}_q)}, \hat{f}_q^{(\hat{v}_q)}]^T \tag{5.183}$$

where the vector $\hat{\boldsymbol{\psi}}_q^{(\hat{v}_q)}$ is expressed by eq. (5.180) with $v_k = \hat{v}_q$.

The entire dataset is generated by accomplishing the feature selection procedure expressed by eqs. (5.181)-(5.182) for any q. It is important to stress that, in this experiment, only a specific target must be selected for each observation. In fact, in a multiple target scenario like the considered one, it is hard to understand which elements of the set $\mathcal{S}_{\hat{l}}$ (5.179) are associated with the target of interest. The vector $\mathbf{r}_q$ (5.183) generated by the deterministic procedure described above represents the input of the feed-forward NN, whose response is the bidimensional vector $\hat{\mathbf{t}}_q \triangleq [\hat{t}_{q,0}, \hat{t}_{q,1}]$; the elements of this vector represent the estimates of the range and the azimuth, respectively, of the target detected on the basis of the q-th observation (see eq. (5.167)). This network contains three hidden layers, consisting of $M_1 = 30$, $M_2 = 20$ and $M_3 = 10$ neurons (see Fig. 5.9). Each of them employs a ReLu, characterized by the transfer function

$$h(x) = x\,\mathrm{u}\,(x)\,, \tag{5.184}$$

where $\mathrm{u}\,(\cdot)$ denotes the *unit step function*. The estimates of the target range and azimuth are computed by the output layer, that contains two neurons only.

The size of the whole dataset acquired in this experiment is $\hat{N}_t = 1438$; 80% of it has been exploited for training the considered NN and the remaining part for its test (therefore, the size of the training set and that of the test are $N_t = 1150$ and $\bar{N}_t = 288$, respectively). Moreover, training has been accomplished by an *adam* optimizer; the batch size, the (constant) learning rate and the number of epochs selected for this procedure are $N_S = 4$, $\gamma = 10^{-3}$ and $N_E = 50$, respectively (see eq. (5.51)). The elements of the feature vector $\mathbf{r}_q$ (with $q = 0, 1, ..., N_t - 1$) have been scaled before applying it to the network (more specifically, a min-max normalization has been employed [196]); this ensures that the absolute value of such elements belongs to the interval $[0, 1]$ and makes the training procedure more effective. The accuracy achieved by the network over the test set has been assessed by evaluating the RMSEs

$$\hat{\varepsilon}_R = \frac{1}{\sqrt{\bar{N}_t}} \left\| \hat{\mathbf{R}} - \hat{\mathbf{R}}_{NN} \right\| \tag{5.185}$$

and

$$\hat{\varepsilon}_\phi = \frac{1}{\sqrt{\bar{N}_t}} \left\| \hat{\boldsymbol{\phi}} - \hat{\boldsymbol{\phi}}_{NN} \right\|, \tag{5.186}$$

where $\hat{\mathbf{R}}$ ($\hat{\boldsymbol{\phi}}$) is the $\bar{N}_t$-dimensional vector collecting the values of the target range (azimuth) estimated by means of the pico-flexx camera over the test set and $\hat{\mathbf{R}}_{NN}$ ($\hat{\boldsymbol{\phi}}_{NN}$) is the corresponding prediction computed by the NN ($||\boldsymbol{x}||$ denotes the Euclidean norm of the vector $\boldsymbol{x}$). The network performance has been also assessed by evaluating its *detection score*

$$A_c = \frac{N_C}{N_C + N_W}, \tag{5.187}$$

where N_C (N_W) is the number of trials in the test set in which both target azimuth and

range have been correctly (wrongly) estimated (note that $N_C + N_W = \bar{N}_t$). In the q-th trial, estimation is deemed correct if $|\hat{R}_q - \hat{t}_{q,0}| \leq \Delta R$ and $|\hat{\phi}_q - \hat{t}_{q,1}| \leq \Delta\phi$, where $\Delta R = 20$ cm and $\Delta\phi = 5.5^\circ$. It is worth pointing out that the values selected for the parameters ΔR and $\Delta\phi$ account for the limited resolution of the employed camera and radar system. Actually, the value selected for ΔR may look larger than expected, because of the high resolution that can be potentially achieved by both the employed radar device and pico-flexx camera. However, readers should not forget that the algorithm employed for the computation of $\hat{R}_q$ is not error free (see eq. (5.173) and (5.174)), especially when the cluster of points Δ_q (5.169) is not so dense or when the size $\bar{N}_q$ in eq. (5.172) is large. A low density in the set $\mathcal{S}_q$ could be observed when, for instance, the robot reaches the corners of the delimited area or in presence of optical disturbances.

Methods	$\hat{\varepsilon}_R$ (m)	$\hat{\varepsilon}_\theta$ ($^\circ$)	*AC* (%)	*Training* (sec)	*Prediction* (msec)
FFT based	0.09	3.0	88	-	5
ANN	0.07	3.5	92	8	10
YOLO v2	0.03	1.5	98	398	20

TABLE 5.3: Accuracy, detection score, training and prediction time of a deterministic estimation algorithm, a feed-forward NN and a YOLO v2 network.

The estimated accuracy and precision achieved by the adopted NN together with the time required for its training and testing are listed in Table 5.3. In the same table, the values of the same parameters evaluated on the basis of the deterministic algorithm employed for feature extraction are also provided; note that, for any q, this algorithm can be exploited to generate the estimates $\hat{R}_{\hat{v}_q}^{(q)}$ and $\hat{\phi}_{\hat{v}_q}^{(q)}$ of the target range and azimuth, on the basis of $\hat{v}_q$ (5.181) (note that $\hat{R}_{\hat{v}_q}^{(q)}$ and $\hat{\phi}_{\hat{v}_q}^{(q)}$ represent the $\hat{v}_q$-the element of the vectors $\hat{\mathbf{R}}_q$ and $\hat{\boldsymbol{\phi}}_q$, respectively). From these results it is easily inferred that: a) the NN is able to accurately predict the position of the target; b) it outperforms the deterministic algorithm in terms of both accuracy and precision; c) its prediction time is comparable with the computation time required by the deterministic algorithm.

In general, feed-forward NNs require a clever selection of their feature vector; for this reason, some expertise in radar systems is desirable when applying them to target detection and estimation. This problem can be circumvented by applying the YOLO v2 network (see paragraph 5.5.2). Let us illustrate now how this network can be employed to solve the target detection and estimation problem taken into consideration in this paragraph. The q-the element of the collected dataset

$$\mathcal{D}_o \triangleq \{(\mathbf{r}_q, \mathbf{b}_q, t_q)\,; q = 0, 1, ..., \hat{N}_t - 1\} \tag{5.188}$$

consists of the following three components:

1. The noisy observation $\mathbf{r}_q = \mathbf{J}_q$, where $\mathbf{J}_q = [J_{l,p}^{(q)}]$ is a *range-azimuth matrix* having size $N_0 \times \bar{N}_0$ and computed on the basis of the measurements acquired in the q-th trial. The element on the l-th row and the m-th column of $\mathbf{J}_q$ is defined as

$$J_{l,p}^{(q)} \triangleq \frac{1}{N_0\,\bar{N}_0} \Big| \sum_{v=0}^{\bar{N}_0-1} \sum_{n=0}^{N_0-1} S_{l,p} \Big|, \tag{5.189}$$

where

$$S_{l,p} = x_{n,ZP}^{(v)} \exp\left(-j2\pi n\, f_l T_s\right) \exp\left(-j2\pi v \frac{d}{\lambda} s_p\right) \tag{5.190}$$

with $l = 0, 1, ..., N_0 - 1$ and $p = 0, 1, ..., \bar{N}_0 - 1$; here, $x_{n,ZP}^{(v)}$ (see eq. (2.33)) is the n-th element of the N_0-dimensional vector $\mathbf{x}_{ZP}^{(v)}$ (see eqs. (2.48)-(2.49)) that results from zero padding N-dimensional vector of time domain signal samples acquired over the v-th virtual antenna, T_s the sampling period, $f_l \triangleq l/(N_0 T_s)$ is the center frequency of the l-th frequency bin and $s_p \triangleq 2\,(p - \bar{N}_0/2)/\bar{N}_0$ is the p-th *normalized spatial frequency* (see eq. (2.67)). Note that the matrix $\mathbf{J}_q$ can be computed through a $N_0 \times \bar{N}_0$-order 2D FFT, and that the range and azimuth associated with $J_{l,p}^{(q)}$ (5.189) are (see eqs. (2.59) and (2.69), respectively)

$$\bar{R}_l = f_l \frac{c}{2\mu} \tag{5.191}$$

and

$$\bar{\phi}_p = \arcsin s_p, \tag{5.192}$$

respectively.

2. The vector

$$\mathbf{b}_q = \left[l_q, p_q, w_q, h_q\right] \tag{5.193}$$

describing the bounding box associated with the detected target; here, the couple of integers (l_q, p_q) identifies the frequency bin and the normalised spatial frequency, respectively, corresponding to the center of the box and w_q (h_q) represents the width (height) of the box itself.

3. The label t_q; this equal to 1 (-1) if a target is detected (absent).

In the experiment at hand, the values $N_0 = 8192$ and $\bar{N}_0 = 128$ in the computation of the elements of the matrix $\mathbf{J}_q$ have been selected. However, since $R_m = 1.0$ m ($R_M = 3.5$ m) and $\phi_m = -55^\circ$ ($\phi_M = 55^\circ$) have been assumed for the minimum (maximum) range, the $N_0 \times \bar{N}_0$ matrix $\mathbf{J}$ has been resized to an $\bar{N}_l \times \bar{N}_p$ matrix, where $\bar{N}_l = b_M - b_m + 1 = 106$ (the values of the parameters b_m and b_M are expressed by eqs. (5.177) and (5.178), respectively), $\bar{N}_p = d_M - d_m + 1 = 106$, with

$$d_m = \left\lfloor \frac{\bar{N}_0}{2}\left(s_p + 1\right) \right\rfloor = 11 \tag{5.194}$$

and

$$b_m = 42. \tag{5.195}$$

In addition, a square shape with $w_q = h_q = 12$ has been always assumed for the bounding box; its parameters l_q and p_q have been computed as

$$l_q = \arg \min_{b_m \le \tilde{l} \le b_M} \left|\hat{R}_q - R_{\tilde{l}}\right| \tag{5.196}$$

and as

$$p_q = \arg \min_{d_m \le \tilde{p} \le d_M} \left|\hat{\phi}_q - \phi_{\tilde{p}}\right| \tag{5.197}$$

where $\hat{R}_q$ ($\hat{\phi}_q$) is expressed by eq. (5.173) (eq. (5.174)) and $R_{\tilde{l}}$ ($\phi_{\tilde{p}}$) by eq. (5.191) (eq. (5.192)) with $l = \tilde{l}$ ($p = \tilde{p}$). The size of the dataset $\mathcal{D}_o$ (5.188) is $\hat{N}_t = 1438$; 80% of

its elements are used for training and the remaining part for testing; consequently, the sizes of the training set and the test set are $N_t = 1150$ and $\bar{N}_t = 288$, respectively. *Data augmentation* has been performed on the training and test set in order to reduce network overfitting, since their sizes are not so large; this procedure consists in randomly flipping and scaling the input image and the associated box.

Layers	Filters	Size	Stride	Output
Convolutional + BN + ReLu	16	5×5	2	$52 \times 52 \times 16$
Max pooling	-	2×2	2	$26 \times 26 \times 16$
Convolutional + BN + ReLu	32	5×5	2	$12 \times 12 \times 32$
Max pooling	-	2×2	2	$6 \times 6 \times 32$
Convolutional + BN + ReLu	64	3×3	1	$6 \times 6 \times 64$
Max pooling	-	2×2	2	$3 \times 3 \times 64$
Convolutional + BN + ReLu	128	3×3	1	$3 \times 3 \times 128$
Convolutional + BN + ReLu	256	3×3	1	$3 \times 3 \times 256$
Convolutional + BN + ReLu	512	3×3	1	$3 \times 3 \times 512$
Convolutional	6	1×1	1	$1 \times 1 \times 6$

TABLE 5.4: Architecture of the CNN employed for target detection and estimation.

The architecture of the employed network is summarized in Table 5.4. It consists of a cascade of 22 layers and is fed by a normalized version of the resized range-azimuth matrix generated through the procedure illustrated above and having size $\bar{N}_l \times \bar{N}_p = 106 \times 106$. Each of its first two convolutional layers has stride $S = 2$ and is followed by a max pooling layer for dimensionality reduction. The use of a *batch normalization* (BN) layer after each convolutional layer allows to avoid overfitting, since the dataset size is not so large; consequently, other forms of regularization (as dropout) are not required. The activation function at the end of each convolutional layer is ReLu (see eq. (5.184)). The filter depth in the last convolutional layer must be proportional to $\bar{N}_A \cdot (N_{P_A} + K)$, where $\bar{N}_A$ is number of anchor boxes, N_{P_A} is the number of predictions per each anchor and K is the number of classes (see refs. [174] and [175]). Since, in the considered test, $\bar{N}_A = 1$, $N_{P_A} = 5$ and $K = 1$ (if the background is ignored), the selected filter depth is equal to 6. A *transform* layer and an *output* layer are also included in the architecture of the adopted network. The former layer improves network stability in predicting the possible locations for the bounding box, whereas the latter one refines the estimate of the bounding box location. If the N_K candidate boxes $\{\mathbf{b}_q[k] = [l_q[k], p_q[k], w_q[k], h_q[k]]^T; k = 0, 1, ..., N_K - 1\}$ (all labelled by $t_q = 1$) are identified by network, the index $\hat{k}_q$ of the bounding box

$$\hat{\mathbf{b}}_q = \left[l_q[\hat{k}_q], p_q[\hat{k}_q], w_q[\hat{k}_q], h_q[\hat{k}_q] \right] \tag{5.198}$$

best fitting the ground truth box is evaluated as

$$\hat{k}_q = \arg \max_{\tilde{k} \in \{0,1,...,N_K-1\}} I_{\tilde{k}}^{(q)}, \tag{5.199}$$

where

$$I_{\tilde{k}}^{(q)} = \frac{A_{BG}^{(q)} \cap A_{BP}^{(q)}}{A_{BG}^{(q)} \cup A_{BP}^{(q)}} \tag{5.200}$$

is the *intersection over union* (IOU) associated with the k-th candidate box; here, $A_{BG}^{(q)}$ ($A_{BP}^{(q)}$) represents the surface of the ground truth (predicted) bounding box referring to the q-th observation. In the proposed experiment, a target is detected if $I_{\tilde{k}}^{(q)} > I_{th}$, where $I_{th} = 0.1$ is a properly selected threshold. Once the predicted bounding box $\hat{\mathbf{b}}_q$ (5.198) is known, the estimate of the target range (azimuth angle) is evaluated by setting $l = l_q[\hat{k}_q]$ ($p = p_q[\hat{k}_q]$) in eq. (5.191) (eq. (5.192)); note that the values selected for the parameters l and p identify the center of the predicted bounding box. The training procedure of the adopted network has been carried out through the SGD algorithm; a batch size $N_S = 10$, a learning rate $\gamma^{(i)} = 10^{-3}$ and a number of epochs $N_E = 25$ have been assumed (see eq. (5.51)). The testing procedure has evidenced that this network is able to predict the bounding boxes characterized by $I_{\tilde{k}}^{(q)} > 0.1$ over 98% of the test set. A realization of range-azimuth map associated with the matrix $\mathbf{J}$ and of the associated ground truth and the predicted bounding boxes around the detected target is illustrated in Fig. 5.21, where the position of the two corner reflectors placed on the border of the area of interest is also identified (see Fig. 5.20).

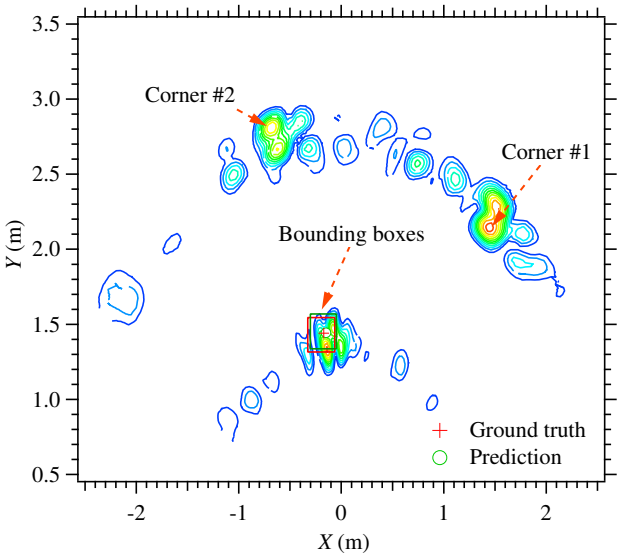


FIGURE 5.21: Range-azimuth map referring to the scenario illustrated in Fig. 5.20. The ground truth bounding box and the position of the target are identified by a red square and a red cross, respectively. The prediction of the network, together with the estimated bounding box, are identified by a green circle and a green square, respectively.

These results deserve the following comments:

a) The network is able to detect the target on the basis of the value of range and azimuth obtained through the pico-flexx camera.

b) In the considered case, the IOU between the ground truth bounding box (red line) and the predicted one (green line) is quite large, being equal to 0.73. Consequently, the estimate of the position of the target (green circle) is very accurate and certainly

much better than the one used as reference (red cross) (note that $|\hat{R}_q - R_{l_q[\hat{k}_q]}| = 0.001$ m and $|\hat{\phi}_q - \phi_{p_q[\hat{k}_q]}| = 0.9^\circ$ in this case).

The values of the achieved accuracy (evaluated in terms of the RMSEs $\hat{\varepsilon}_R$ (5.185) and $\hat{\varepsilon}_\phi$ (5.186)), the detection score (5.187), and the computational time required for training and testing are listed in Table 5.3. From these results it is easily inferred that:

a) The YOLO network outperforms the considered (deterministic) FFT-based method in target detection and estimation.

b) The value of the YOLO detection score A_c (5.187) is really high and better than that provided by the feed-forward NN.

c) The YOLO RMSE $\hat{\varepsilon}_R$ ($\hat{\varepsilon}_\phi$) is smaller than (close to) the one characterizing the feed-forward NN.

These results lead to the conclusion that the YOLO network is more robust than the feed-forward NN. Note also that, even if the complexity of this network is higher than those of the other two methods, the time it employs for computing its prediction is not too long, being in the order of few milliseconds. Since the YOLO v2 network tries to solve also a binary classification problem, other two important parameters for evaluating its performance are its *precision*

$$P = \frac{T_P}{T_P + T_N} \tag{5.201}$$

and its *recall*

$$R = \frac{T_P}{T_P + F_N}, \tag{5.202}$$

where T_P (T_N) represents the overall number of *true positives* (*true negatives*), i.e. the number of targets (false targets) classified correctly, and F_N is the overall number of false targets classified as targets.

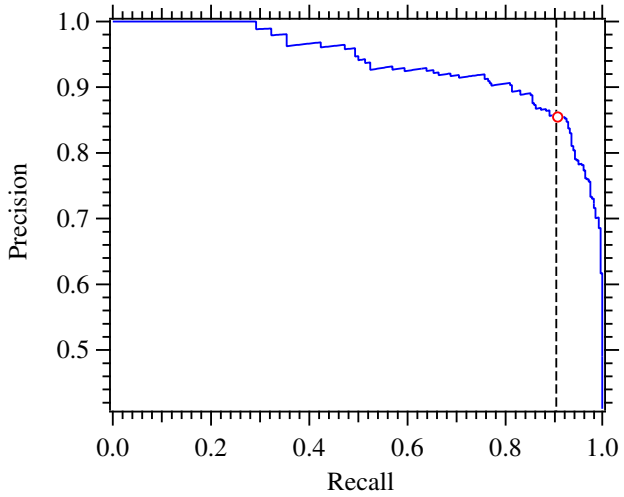


FIGURE 5.22: Representation of the precision versus recall plot referring to the YOLO v2 network employed in the second application. Note that, if the recall exceeds the threshold identified by the vertical dashed line, the precision decreases steeply.

The *precision versus recall* plot evaluated in the considered experiment is shown in Fig. 5.22. These results lead to the conclusion that, in this case, the precision remains

high for large values of the recall and drops steeply only when the recall exceeds 0.9. The area under the curve shown in Fig. 5.22 represents the so called *mean average precision* (mAP); in this case, it has found that $mAP = 93\%$ (note that the value of this parameter is expressed as a percentage since the precision P (5.201) and the recall R (5.202) are defined in the range $[0\,,\,1]$).

5.7 Conclusions

Machine learning and deep learning techniques are able to extract relevant information from the available data in the absence of an accurate mathematical description of the behaviour of radar devices and of the mechanisms of electromagnetic propagation. Even if important steps have been made in this field during the last years, significant research efforts are still required to make the adoption of these techniques in commercial systems a reality. In this chapter it has been shown how some learning methods can be exploited to solve simple classification and regression problems in FMCW radar systems operating in a 2D propagation scenario and in the presence of point targets. Then, various applications of learning methods to specific technical problems have been illustrated and relevant trends in research on MIMO radars have been identified. Finally, the application of machine learning and deep learning methods to two specific problems, namely human activity classification and range azimuth estimation, has been investigated. The obtained numerical results, based on experimental datasets acquired through a colocated MIMO device operating at 77 GHz, allow the readers to understand the high potential of these methods to solve real world problems. A pervasive use of such methods should be expected in the near future, as understanding of the learning methods described in this work, and MIMO technology is continuously evolving.

6 Colocated MIMO radars for smart agriculture

Future tractors, like cars, will be intelligent and possibly driver-less systems equipped with many sensors, like lidars, gps, odometers, inertial units and RGB or IR cameras. However, with respect to classical automotive scenarios, tractors are used in dirty environments and sensor data may be affected by its mechanical vibrations. Moreover, the beam transmitted by optical sensors may be blocked by the presence of vegetation; for the same reason, the signal-to-noise ratio of the GPS signal may become quite low, affecting the accuracy with which the position of the tractor is measured. Radars, on the contrary, are employed thanks to their ability to see in any weather and lighting conditions; through radar systems, in fact, it is possible to estimate the position and the velocity of a tractor with respect to soil or a plantation and, finally, to detect the presence of obstacle in the field. These advantages are particularly useful whenever a vehicle has to move between the rows of vineyards, where the term "row" indicates an alignment of poles suitable to support the growth of the plants. In fact, the vineyards may change during the seasons and plants along the rows may grow in an unpredictable way and with different shapes; therefore, currently, the vehicles that move between the rows of the vineyards are still driven by human operators. My contribution to smart agriculture have mainly regarded the development of techniques for autonomous driving of a tractor through the rows of a plantation; in particular, the innovations provided by my work consist in:

a) the development of a row recognition technique based on radar data and a preliminary knowledge of the distance between the vehicle and the rows;

b) the development of imaging techniques for the detection of multiple targets and estimation of their positions based on MIMO radar operating in a slowly-variable, two-dimensional or three-dimensional scenario.

6.1 Autonomous driving system through the rows of a plantation

The basic idea for detecting the rows of a plantation is based on the combination of a preliminary knowledge of the approximate distance between rows and the radar data. The device tested for this kind of application is an FMCW MIMO radar (like the one shown in Fig. 5.17) operating in E-band (with a center frequency of 77 GHz) and installed on an agricultural vehicle positioned approximately in the middle between two rows of the vineyard, as shown in Fig. 6.1-a). Let us assume that the radar system is equipped with an ULA composed by N_V virtual channels and that the horizontal spacing between the adjacent antenna of the array is equal to d. The n-th sample of the received radar signal $x_{r,n}^{(v)}$, acquired in correspondence of the v-th virtual channel and in

the presence of L targets can be expressed by eq. (2.7), where the delay $\tau_l^{(v)}$ is expressed by eq. (2.11), $l = 0, 1, 2..., L-1$, $n = 0, 1, 2, ..., N-1$ and $v = 0, 1, 2, ..., N_v - 1$. The technique for surveying the vineyard profile includes the following steps:

Step-1) *Calculation of a bi-dimensional cost function*, also known as range-azimuth map on the basis of a canonical 2D DFT procedure already shown in Par. 2.5 and in Par. 5.6. In practice, a one dimensional DFT of order N_0 (with $N_0 \geq N$) is evaluated for each antenna v of the array and the obtained l-th spectral coefficient can be written as

$$\bar{X}_l^{(v)} = \frac{1}{N} \sum_{n=0}^{N_0-1} x_{r,n}^{(v)} \exp\left(-j2\pi n f_l T_s\right) \tag{6.1}$$

where $l = 0, 1, 2, ..., N_0 - 1$ and $f_l \triangleq \frac{l}{N_0 Ts}$ (see eq. (2.52)). Then, another 1D DFT of order $\bar{N}_0$ (with $\bar{N}_0 \geq N_V$) is computed as

$$J[l, p] = \frac{1}{N_V} \sum_{v=0}^{N_V-1} \bar{X}_l^{(v)} \exp\left(-j2\pi v \frac{d}{\lambda} s_p\right) \tag{6.2}$$

where $l = 0, 1, 2, ..., N_0 - 1$, $p \in \left[-\frac{\bar{N}_0}{2}\,;\, \frac{\bar{N}_0}{2} - 1\right]$ and s_p is the normalised spatial frequency $s_p = 2\frac{p}{\bar{N}_0}$ (see eqs. (5.189) and (2.67)). Knowing $J[l, p]$ the cost function $S[l, p] \triangleq |J[l, p]|^2$ is evaluated. In particular, the index l is related to the target distance (see eqs. (2.57)-(2.59)) and the index p to its azimuth (see eq. (2.69)).

Step-2) *Thresholding and clustering* - In this part, all the significant values of the cost function **S** are found; this allows to identify all the meaningful values of the pair (l, p), i.e. a set of points $\{(l_k, p_k)\}$ on which there is a significant reflection:

$$S(l_k, p_k) > T \tag{6.3}$$

for any k, where T is a proper threshold. For each set of points $\{(l_k, p_k)\}$, another pair of points in Cartesian coordinates $\{(x_k, y_k)\}$ can be evaluated by mean of $x_k = R_k \cdot \cos(\theta_k)$ and $y_k = R_k \cdot \sin(\theta_k)$. This new set of coordinates is divided in two different groups on the basis of the geometry of the vineyard known a priori, selecting only those points that fall within a predetermined region of the 2D map, that correspond to a well defined region of the space, where the vineyard is expected to be found. In a more analytical way, if the two rectangular region are indicated by $\mathcal{S}_g$, with $g = 1, 2$, a generic pair (x_k, y_k) is contained in $\mathcal{S}_g$ if:

$$\begin{cases} x_{l_g} \leq x_l \leq x_{u_g} \\ y_{l_g} \leq y_l \leq y_{u_g} \end{cases} \tag{6.4}$$

where x_{l_g} (x_{u_g}) represents the lower (upper) bound of the region $\mathcal{S}_g$ along the x-axis and y_{l_g} (y_{u_g}) represents the lower (upper) bound of the region $\mathcal{S}_g$ along the y-axis.

Step-3) *Linear Interpolation* - This step relies on the assumption that the estimated vineyard profile should look like two sequences of vertical lines (one on the left, the other one on the right). In principle, the data belonging to each cluster $\mathcal{S}_g$ are associated with a single vertical line and the geometric parameters of each line can be estimated by applying a *least-squares* (LS) approach in order to linearly interpolate such data.

At this point, from the value of the parameters of the estimated line the distance between the vehicle and the rows can be inferred; moreover, on the basis of this distance the intelligent vehicle is able to pick the best decision in order to avoid damages and proceed along the chosen direction. An example of the range-azimuth map obtained by moving along the rows of a vineyard is shown in Fig. 6.1-b). On top of the map also two small red rectangles representing the regions $\mathcal{S}_g$ obtained on the basis of a priori information of of the distance between the rows and the vehicle are shown. In this map the strongest reflections are given by the pillow on the left and on the right side, but also small reflections coming from plants are detected; this allows to increase the reliability of the driving based on radars. However, when the machine is approaching to a bend, data coming from other sensors like GPS or Lidar may help the machine to bend correctly.

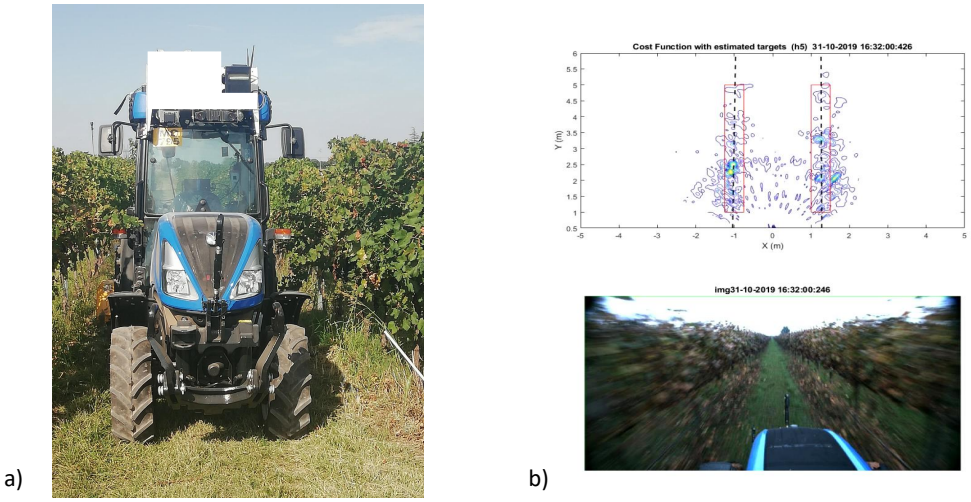


FIGURE 6.1: Representation of a) the measurement system set-up; b) the range-azimuth map obtained by moving along the rows of a vineyard , where the two small red rectangles represents the regions $\mathcal{S}_g$ obtained on the basis of a priori information of the distance between the rows and the vehicle.

6.2 Three-dimensional imaging based on a MIMO radar

The work presented in the previous section allows to detect the vineyard profile by smartly looking at the 2D range-azimuth map. However, a possible improvement of that method consists into directly look at the point-cloud that is obtained by the radar. In this section, the novel method based on an approximate maximum likelihood approach described in Par. 4.2 is proposed for vineyard profile detection. This method is also patent pending. The device employed for this kind of applications is an FMCW MIMO radar (like the one shown in Fig. 6.2-a)) operating in E-band with a center frequency of 77 GHz and it has to be associated with an agricultural vehicle positioned approximately in the middle between two rows of the vineyard, as shown in Fig. 6.1-a). It is endowed with a custom designed planar array made of $N_T = 16$ TX antennas and $N_R = 16$ RX antennas, each consisting of an array of six patch elements. The resulting virtual array, consisting of $N_V = 16 \cdot 16 = 256$ VAs, allows us to achieve a range resolution $\Delta R = 7.5$ cm, and azimuthal and elevation resolutions of $\Delta\theta_2 = 7.6^\circ$ and $\Delta\phi_2 = 3.8^\circ$, respectively (see eq. (2.64)). The received radar signal model $x_{r,n}^{(v)}$ considered in correspondence of

the n-th sample ($n = 0, 1, 2, ..., N-1$) and the v-th virtual channel can be described by eq. (2.7) and eq. (2.33).

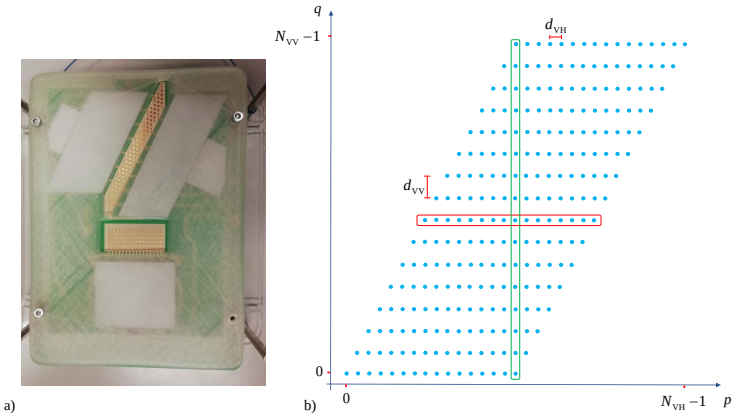


FIGURE 6.2: Representation of: a) the physical array of the Inras FMCW radar; b) the corresponding virtual array.

The main changes that need to be introduced in the RASCA described in Par. 4.2 to adapt it to the considered MIMO FMCW radar system consists on the following steps.

T1 *FFT processing* In this step:

a) the vector $\{\mathbf{x}_c^{(v)}\}$ (2.25) is substituted with the real one $\{\mathbf{x}_r^{(v)}\}$ (2.7);

b) each IFFT operation is replaced by an FFT. This means that the spectra $\{\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)}; v = 0, 1, ..., N_V - 1\}$ provide frequency domain information instead of time. For this reason, delay bins are replaced by bins in the frequency domain.

T2-S2) *Target detection and range estimation* – In this step, that aims at detecting the most relevant targets on each of the N_A antennas, a minor change is required in the cancellation procedure with respect to its counterpart employed in the MIMO SFCW radar system. This is due to the fact that the noisy measurements processed in a MIMO FMCW radar system are always represented by *real* (time-domain) sequences.

The initialization of the STDREC algorithm remains unchanged: the triad $(\mathbf{X}_0^{(v_k)}[0], \mathbf{X}_1^{(v_k)}[0], \mathbf{X}_2^{(v_k)}[0])$ is defined according to eq. (4.102) (with $k = 0, 1, ..., N_A - 1$). In its i-th iteration, this algorithm accomplishes the three steps described below (these are denoted by **STDREC-Sp** in the following, with $p = 1, 2$ and 3).

STDREC-S1) *Detection of a new target and estimation of its parameters* – In this step, the normalised frequency $F_i^{(v_k)}$ and the complex amplitude $C_i^{(v_k)}$ associated with the i-th target are estimated. Note that, generally speaking, the normalised frequency $F_i^{(v_k)}$ is not a multiple of of the *fundamental frequency* $F_{\text{DFT}} = 1/N_0$ that characterizes the output of the FFT processing executed in task **T1**; for this reason, it can be expressed as

$$F_i^{(v_k)} = F_{c,i}^{(v_k)} + \delta_i^{(v_k)} F_{\text{DFT}}, \tag{6.5}$$

where $F_{c,i}^{(v_k)}$ is a given *coarse estimate* of $F_i^{(v_k)}$ and $\delta_i^{(v_k)}$ is the *residual*. The processing accomplished in this step is executed by an algorithm dubbed *single frequency estimator* (SFE) that can be derived as the dual version of the CSDE algorithm proposed in Par. 4.1.1. Let us suppose to have the N_0-dimensional vectors $\mathbf{X}_m^{(v)} = \left[X_{m,0}^{(v)}, X_{m,1}^{(v)}, ..., X_{m,N_0-1}^{(v)}\right]^T \triangleq$

$\mathrm{DFT}_{N_0}\left[\mathbf{x}_{m,\mathrm{ZP}}^{(v)}\right]$, with $m = 0,\ 1,\ 2$, are computed for any v-th channel; here, $\mathrm{DFT}_{N_0}[\mathbf{x}]$ denotes, up to a scale factor $(1/N)$, the N_0 order *inverse discrete Fourier transform* of the N_0-dimensional vector $\{\mathbf{x}_r^{(v)}\}$. More specifically, we assume that

$$X_{m,k}^{(v)} \triangleq \frac{1}{N}\sum_{n=0}^{N-1} n^m\, x_{r,n}^{(v)} \exp\left(-j\frac{2\pi nk}{N_0}\right), \tag{6.6}$$

with $k = 0, 1, ..., N_0 - 1$ and $m = 0,\ 1,\ 2$. For every selected v_k-th channel[1], let us assume that:

a) a single target is present (the index i can be omitted);

b) the initial coarse estimate $\hat{F}_c^{(0)}$ of F is set to

$$\hat{F}_c^{(0)} = \hat{\alpha}\, F_{\mathrm{DFT}}, \tag{6.7}$$

where the integer $\hat{\alpha}$ is computed as

$$\hat{\alpha} = \arg \max_{\tilde{\alpha}\in\{0,1,\ldots,N_0/2\}} \left|X_{0,\tilde{\alpha}}^{(v_k)}\right|; \tag{6.8}$$

c) the quantity

$$\hat{\rho}^{(0)} \triangleq \frac{\hat{F}_c^{(0)}}{F_{\mathrm{DFT}}} = \hat{\alpha}; \tag{6.9}$$

d) the initial estimate $\hat{C}^{(0)}$ of C as

$$\hat{C}^{(0)} = G\big(\hat{F}_c^{(0)}\big) \tag{6.10}$$

where for a generic $\tilde{F}$ (at the beginning $\tilde{F} = \hat{F}_c^{(0)}$)

$$G\big(\tilde{F}\big) \triangleq \frac{\bar{X}(\tilde{F}) - \bar{X}^*(\tilde{F})g(\tilde{F})}{1 - \left|g(\tilde{F})\right|^2} \tag{6.11}$$

$$g(\tilde{F}) = \frac{1}{N}\frac{\exp(-j4\pi N\tilde{F}) - 1}{\exp(-j4\pi\tilde{F}) - 1}, \tag{6.12}$$

and

$$\bar{X}(\tilde{F}) \triangleq \frac{1}{N}\sum_{n=0}^{N-1} x_{r,n}^{(v_k)} \exp\left(-j2\pi n\tilde{F}\right); \tag{6.13}$$

e) the spectral coefficients $X_{1,\hat{\alpha}} = X_{1,\hat{\alpha}}^{(v_k)}$, $X_{2,\hat{\alpha}} = X_{2,\hat{\alpha}}^{(v_k)}$ (see eq. (6.6) for $m = 1, 2$), and the coefficients $\{K_p(2\hat{\alpha});\ p = 1, 2\}$ and $\{b(\hat{\alpha}),\ c(\hat{\alpha})\}$ are computed on the basis of the definitions

$$K_p\,(x) \triangleq \frac{1}{N}\sum_{n=0}^{N-1} g_p\,[n] \exp\left(-j\frac{2\pi nx}{N_0}\right), \tag{6.14}$$

$$b\,(\rho) \triangleq -\Re\left\{\hat{C}^*\bar{X}_{2,\rho}\right\} + 2\Re\left\{\left(\hat{C}^*\right)^2 K_2\,(2\rho)\right\} \tag{6.15}$$

and

$$c\,(\rho) \triangleq \Im\left\{\hat{C}^*\bar{X}_{1,\rho}\right\} - \Im\left\{\left(\hat{C}^*\right)^2 K_1\,(2\rho)\right\} \tag{6.16}$$

[1] the index v_k is omitted in this steps for simplicity.

respectively, considering $g_p[n] \triangleq n^p$ and $g_0[n] \triangleq 1$;

f) the initial estimate $\hat{\Delta}^{(0)}$ of Δ as

$$\hat{\Delta}^{(0)} = -\frac{c\left(\hat{\rho}^{(0)}\right)}{b\left(\hat{\rho}^{(0)}\right)} \tag{6.17}$$

g) the first fine estimate $\hat{F}^{(0)}$ of F as

$$\hat{F}^{(0)} = \hat{F}_c^{(0)} + \frac{\hat{\Delta}^{(0)}}{2\pi} \tag{6.18}$$

During the iterative procedure, the r-th iteration is fed by the estimates $\hat{F}^{(r-1)}$ and $\hat{C}^{(r-1)}$ of F and C, respectively, and produces the new estimates $\hat{F}^{(r)}$ and $\hat{C}^{(r)}$ of the same quantities (with $r = 1, 2, ..., N_{\text{SFE}}$, where N_{SFE} is the overall number of iterations); the procedure employed for the evaluation of $\hat{F}^{(r)}$ and $\hat{C}^{(r)}$ consists of the two steps described below (the p-th step is denoted by **SFE-Sp**).

SFE-S1 - The new estimate $\hat{\Delta}^{(r)}$ of Δ is computed on the basis of eq. (6.17); in the evaluation of the coefficients $\{b(\rho), c(\rho)\}$ appearing in the RHS of eq. (6.17), $\hat{C} = \hat{C}^{(r-1)}$ and

$$\rho = \hat{\rho}^{(r-1)} \triangleq \hat{F}^{(r-1)}/F_{\text{DFT}} \tag{6.19}$$

are assumed. Then,

$$\hat{F}^{(r)} = \hat{F}^{(r-1)} + \hat{\Delta}^{(i)}/(2\pi) \tag{6.20}$$

is evaluated.

SFE-S2 - The new estimate $\hat{C}^{(r)}$ of $\hat{C}$ is evaluated as $\hat{C}^{(r)} = G(\hat{F}^{(r)})$ (see eqs. (6.10)–(6.11), assuming that $\tilde{F} = \hat{F}^{(r)}$). Moreover, the index r is incremented by one before starting the next iteration. At the end of the last (i.e., of the N_{SFE}-th) iteration, the fine estimates $\hat{F}_i^{(v_k)} = \hat{F}^{(N_{\text{SFE}})}$ and $\hat{C}_i^{(v_k)} = \hat{C}^{(N_{\text{SFE}})}$ of F and C, respectively, become available[2].

STDREC-S2) *Cancellation of the new target* – The cancellation procedure requires the computation of the contributions

$$\mathbf{C}_{X_0}^{(v_k)}[i] = \hat{C}_i^{(v_k)}\, \bar{\mathbf{W}}_0^{(v_k)}[i] + \left(\hat{C}_i^{(v_k)}\right)^* \left(\bar{\mathbf{W}}_{0,c}^{(v_k)}[i]\right), \tag{6.21}$$

$$\mathbf{C}_{X_1}^{(v_k)}[i] = \hat{C}_i^{(v_k)}\, \bar{\mathbf{W}}_1^{(v_k)}[i] + \left(\hat{C}_i^{(v_k)}\right)^* \left(\bar{\mathbf{W}}_{1,c}^{(v_k)}[i]\right) \tag{6.22}$$

and

$$\mathbf{C}_{X_2}^{(v_k)}[i] = \hat{C}_i^{(v_k)}\ \ \bar{\mathbf{W}}_2^{(v_k)}[i] + \left(\hat{C}_i^{(v_k)}\right)^* \left(\bar{\mathbf{W}}_{2,c}^{(v_k)}[i]\right) \tag{6.23}$$

given by the i-th (i.e., by the last) target detected on the v_k-th VA; here, $\bar{\mathbf{W}}_m^{(v_k)}[i]$ denotes, up to a scale factor, the N_0-th order DFT of the vector

$$\bar{\mathbf{w}}_m^{(v_k)}[i] \triangleq \left[0, 1^m \cdot \bar{w}_i^{(v_k)}, 2^m \cdot \left(\bar{w}_i^{(v_k)}\right)^2, \ldots,\ (N-1)^m \cdot \left(\bar{w}_i^{(v_k)}\right)^{N-1}, 0, \ldots, 0\right]^T, \tag{6.24}$$

[2]here the channel index v_k and the target index i are used again, for better clarity.

with $m = 0$, 1 and 2, $\bar{\mathbf{W}}_{m,c}^{(v_k)}[i]$ the N_0-th order DFT of the vector $(\bar{\mathbf{w}}_m^{(v_k)}[i])^*$,

$$\bar{w}_i^{(v_k)} \triangleq \exp(j2\pi\hat{F}_i^{(v_k)}) \tag{6.25}$$

and

$$\bar{F}_i^{(v_k)} \triangleq \hat{f}_i^{(v_k)} T_s \tag{6.26}$$

is the normalised frequency associated with $\hat{f}_i^{(v_k)}$, considering T_s the sampling time. It is important to point out that an efficient method can be used for the computation of the vectors $\bar{\mathbf{W}}_m^{(v_k)}[i]$ and $\bar{\mathbf{W}}_{m,c}^{(v_k)}[i]$ appearing in the RHS of eqs. (6.21)–(6.23) (with $m = 0$, 1 and 2); note that, for any k, these vectors represent, up to a scale factor, the N_0-th order DFTs of the sequences $\{n^m\,(\bar{w}_i^{(v_k)})^n;\, n = 0, 1, ..., N-1\}$ and $\{n^m\,((\bar{w}_i^{(v_k)})^*)^n;\, n = 0, 1, ..., N-1\}$, respectively. In fact, the l-th element of the vectors $\bar{\mathbf{W}}_m^{(v_k)}[i]$ and $\bar{\mathbf{W}}_{m,c}^{(v_k)}[i]$ is given by

$$\bar{W}_m^{(v_k)}[i,l] = \frac{1}{N}\sum_{n=0}^{N-1} n^m \left(\bar{w}_i^{(v_k)}\right)^n \exp\left(-j\frac{2\pi l}{N_0}n\right) = \frac{1}{N}\sum_{n=0}^{N-1} n^m (q[l])^n \tag{6.27}$$

and

$$\bar{W}_{m,c}^{(v_k)}[i,l] = \frac{1}{N}\sum_{n=0}^{N-1} n^m \left(\left(\bar{w}_i^{(v_k)}\right)^*\right)^n \exp\left(-j\frac{2\pi l}{N_0}n\right) = \frac{1}{N}\sum_{n=0}^{N-1} n^m (q_c[l])^n, \tag{6.28}$$

respectively, where

$$q[l] \triangleq \exp\left(j2\pi\left(\hat{F}_i^{(v_k)} - \frac{l}{N_0}\right)\right) \tag{6.29}$$

and

$$q_c[l] \triangleq \exp\left(j2\pi\left(-\hat{F}_i^{(v_k)} - \frac{l}{N_0}\right)\right). \tag{6.30}$$

Therefore, the identities (4.76)-(4.78) can be exploited for an efficient computation of the RHSs of eqs. (6.27) and (6.28).

STDREC-S3) *Computation of the residual energy in the time domain* – The energy of the residual time-domain vector $\mathbf{X}_0^{(v_k)}[i+1]$ is computed (see eq. (4.116)) and compared with the positive threshold T_{STDREC}. If this energy is below the threshold, the STDREC algorithm stops and $L_k = i$ relevant targets are detected on the v_k-th VA; otherwise, the recursion index i is increased by one and a new recursion is started (i.e., go back to **STDREC-S1**).

T2-S3) *Range information fusion* – In this case, the set $\mathcal{A}_b$ (see eq. (4.118)) collects L_b relevant time bins $\{\hat{\alpha}_l;\, l = 0, 1, \ldots, L_b - 1\}$, whereas the set $\{E_{b,l}\}$ contains the energy associated with each of them (see eq. (4.119)). The average energy $E_{b,l}$ associated with the $\hat{\alpha}_l$-th frequency bin is evaluated as

$$E_{b,l} = \frac{1}{N_{b,l}}\sum_{k=0}^{N_A-1}\sum_{i=0}^{L_k-1}\left|\hat{C}_i^{(v_k)}\right|^2 \delta\left[\hat{\alpha}_i^{(v_k)} - \hat{\alpha}_l\right], \tag{6.31}$$

with $l = 0, 1, \ldots, L_b - 1$; here, $N_{b,l}$ is defined by eq. (4.120). The final output of the information fusion is represented by the set $\mathcal{S}_{\text{RPE}}$ (4.86).

The processing accomplished in **T3** (i.e., by the SPE) has the same structure and

interpretation as that illustrated for the SFCW radar system since it aims at estimating the angular parameters of the targets on the basis of the discrete range profile computed in **T2**. However, delay bins are now replaced by frequency bins and, consequently, the normalised horizontal and vertical delays $F_{\mathrm{H},i}$ and $F_{\mathrm{V},i}$ by their normalised frequency counterparts. Note also that, in step **T3-S1**, the matrix $\mathbf{X}[l]$ is still defined on the basis of eq. (4.122), but the complex amplitude $C_i[p,q,l]$ appearing in eq. (4.88) is replaced by

$$C_i[p,q,l] = C_i[l] \exp\left[j\frac{4\pi}{\lambda} \left[d_{\mathrm{VH}} \left(p - p_R \right) \cos(\phi_i[l]) \sin(\theta_i[l]) + d_{\mathrm{VV}} \left(q - q_R \right) \sin(\phi_i[l]) \right] \right] \tag{6.32}$$

where $C_i[l]$ is the complex amplitude observed for the i-th target in the l-th time bin on the reference VA. This leads to some relevant differences in the formulas employed by the STDAE algorithm in the estimation of the parameters $\{\theta_i[l],\ \phi_i[l],\ F_i[l],\ C_i[l]\}$ (see **STDAEC-S1**)) with respect to the case of a MIMO SFCW radar system. The STDAE processing can be summarised as follows.

STDAE-S1) FFT *processing on the reference* VULA – In this step, the vector

$$\mathbf{s}^{(i)}_{\mathrm{VULA},k}[l] = \left[s^{(i)}_{k,0}[l], s^{(i)}_{k,1}[l], ..., s^{(i)}_{k,\bar{N}_0-1}[l] \right]^T \triangleq \mathrm{DFT}_{\bar{N}_0} \left[\bar{\mathbf{S}}^{(i)}_{\mathrm{VULA},k}[l] \right], \tag{6.33}$$

is evaluated for $k = 0, 1$ and 2; note that

$$s^{(i)}_{k,m}[l] \triangleq \frac{1}{N_{\mathrm{VULA}}} \sum_{p=0}^{N_{\mathrm{VULA}}-1} S^{(i)}_{k,p}[l] \exp\left(-j\frac{2\pi pm}{\bar{N}_0} \right), \tag{6.34}$$

with $m = 0, 1, ..., \bar{N}_0 - 1$, and that the quantity $S^{(i)}_{k,p}[l]$ is defined by eq. (4.125).

STDAE-S2) *Single angle estimation* – The normalised vertical frequency $F_{\mathrm{V},i}[l]$ is still expressed by eq. (4.90); in this case, the quantities $\hat{C}_{\mathrm{V},i}[l]$ and $\hat{F}_{\mathrm{V},i}[l]$ are computed by the CSFE algorithm. The description of the CSFE algorithm is similar to that illustrated for the SFE; the only differences being represented by the fact that:

$$\tilde{C} = \hat{C} = \bar{X}(\hat{F}), \tag{6.35}$$

where $\bar{X}(\hat{F})$ is computed as (6.13) by replacing $x^{(v_k)}_{r,n}$ with $s^{(i)}_{k,m}[l]$, eqs. (6.15) and (6.16) are replaced by

$$b(\rho) \triangleq \Re\{\hat{C}^* \bar{X}_{2,\rho}\} \tag{6.36}$$

and

$$c(\rho) \triangleq -\Im\{\hat{C}^* \bar{X}_{1,\rho}\}. \tag{6.37}$$

and $\bar{X}_{1,\rho}$, $\bar{X}_{2,\rho}$ can be computed on the basis of eq. (6.6) considering $m = 1, 2$, $k = \rho$ and replacing $x^{(v)}_{r,n}$ by $s^{(i)}_{k,m}[l]$.

STDAE-S3) *Vertical folding* – Vertical folding is employed to compensate for the phase differences between the considered HULAs. However, the *phase rotation factor* appearing in eq. (4.132) is computed as

$$R^{(\mathrm{VF})}_i[l,q] \triangleq \left[\exp\left(-j2\pi \hat{F}_{\mathrm{V},i}[l] \right) \right]^{q-q_R}, \tag{6.38}$$

and $\hat{F}_{V,i}[l]$ is an estimate of the normalised vertical frequency $F_{V,i}[l]$ (evaluated in **STDAE-S2**).

STDAE-S4) FFT *processing and horizontal frequency estimation* – In this step, the CSFE algorithm is exploited to compute the estimates $\hat{F}_{\mathrm{H},i}[l]$ and $\hat{C}_i[l]$ of the normalised horizontal frequency $F_{\mathrm{H},i}[l]$ and of the complex amplitude $C_i[l]$, respectively; both are associated with the i-th target. The normalised frequency $F_{\mathrm{H},i}[l]$ is still expressed by eq. (4.89), but its estimate $\hat{F}_{\mathrm{H},i}[l]$ and the estimate $\hat{C}_{\mathrm{H},i}[l]$ of $C_i[l]$ are computed by the CSFE algorithm.

STDAE-S5) *Overall folding and frequency/amplitude estimation* – In this step, the angular information (i.e., the spatial frequencies $\hat{F}_{\mathrm{V},i}[l]$ and $\hat{F}_{\mathrm{H},i}[l]$) computed in the previous steps are exploited to apply folding to the whole receive antenna array or to a portion of it. Similarly as the case of a MIMO SFCW radar, overall folding requires the computation of the N_0-dimensional vectors $\mathbf{X}_{0,\mathrm{OF}}[l]$, $\mathbf{X}_{1,\mathrm{OF}}[l]$ and $\mathbf{X}_{2,\mathrm{OF}}[l]$ (see eq. (4.134)); the horizontal phase rotation $R_i^{(\mathrm{HF})}[l,p]$ appearing on the RHS of eq. (4.135) is defined as

$$R_i^{(\mathrm{HF})}[l,p] \triangleq \left[\exp\left(-j2\pi\hat{F}_{\mathrm{H},i}[l]\right)\right]^{p-p_R}. \tag{6.39}$$

If a peak is detected in the sequence made of the absolute values of the elements of $\mathbf{X}_{0,\mathrm{OF}}[l]$, the CSFE algorithm is run to estimate, on the basis of the vectors $\mathbf{X}_{0,\mathrm{OF}}[l]$, $\mathbf{X}_{1,\mathrm{OF}}[l]$ and $\mathbf{X}_{2,\mathrm{OF}}[l]$, the final estimates $\hat{F}_i[l]$ and $\hat{C}_i[l]$ of the parameters $F_i[l]$ and $C_i[l]$ characterizing the i-th target detected in the l-th delay bin. Then, if the estimated frequency $\hat{F}_i[l]$ (see eq. (6.5)) is close to $\hat{\alpha}_l$ and the quantity $\hat{\alpha}_i[l]$ appears in one of the pairs of set $\mathcal{S}_{\mathrm{RPE}}$ (4.86), it is discarded. Otherwise, the new couple $(\hat{\alpha}_i[l], E_{b,L_b})$ (see eq. (4.140)) is added to the set $\mathcal{S}_b$ and the number of its elements (denoted by L_b) is increased by one.

STDAEC-S2) *Target cancellation* – The contribution $\mathbf{C}_{X_0}^{(i)}[l]$, given by the i-th target detected in the l-th time bin, to the vector $\mathbf{X}^{(i)}[l]$ is evaluated as

$$C_{X_0}^{(i)}[p,q,l] = \hat{C}_i[l]\ \exp\left\{j2\pi\left[(p-p_R)\hat{F}_{\mathrm{H},i}[l] + (q-q_R)\hat{F}_{\mathrm{V},i}[l]\right]\right\}, \tag{6.40}$$

for any VA (i.e., for any p and q) and it is cancelled from $\mathbf{X}^{(i)}[l]$. Cancellation consists in the evaluation of the new residual according to eq. (4.143).

STDAEC-S3) *Residual energy test* – In this step, the energy $E^{(i+1)}[l]$ of the residual $\mathbf{X}^{(i+1)}[l]$ evaluated in the previous step is computed on the basis of eq. (4.144)) and is compared with a positive threshold; if this energy is below the threshold, the STDAEC algorithm stops, otherwise the recursion index i is increased by one and a new iteration is started by going back to **STDAEC-S1**. All the target information acquired from the $\hat{\alpha}_l$-th frequency bin are collected in the set

$$\mathcal{T}_l \triangleq \left\{\left(\hat{C}_i[l], \hat{F}_i[l], \hat{\alpha}_i[l], \hat{F}_{V,i}[l], \hat{F}_{H,i}[l]\right); i = 0,1,\ldots,D[l]-1\right\} \tag{6.41}$$

with $l = 0, 1, \ldots, L_b - 1$; here, $D[l]$ denotes the overall number of targets detected in the considered bin.

In **T3-S2** the evaluation of spatial coordinates $(\hat{\phi}_i[l], \hat{\theta}_i[l])$ is accomplished in the same way as in the SFCW case, i.e. on the basis of eqs. (4.146)–(4.147), while

$$\hat{R}_i[l] = \frac{c}{2\mu T_s}\hat{F}_i[l], \tag{6.42}$$

where μ and T_s are defined in (2.6)-(2.8). Finally, the available information are merged

to generate the overall set $\mathcal{I}_t$ (4.148), which, generally speaking, can be represented as a cloud of $\hat{L}$ points; note that $\mathcal{I}_t$ results of the union of the sets $\{\mathcal{I}_t^{(l)}\}$ and that the set $\mathcal{I}_t^{(l)}$ is still defined according to eq. (4.149). An example of the whole point-cloud derived by the proposed RASCA method is shown in Fig. 6.3 (the data were collected in a vineyard through the radar shown in Fig. 6.2). From a qualitative point of view, the reconstructed point-cloud provides a nice and complete description of the two rows of the vineyard (see Fig. 6.1-b)). Other and more quantitative results cannot be provided here for trade secret.

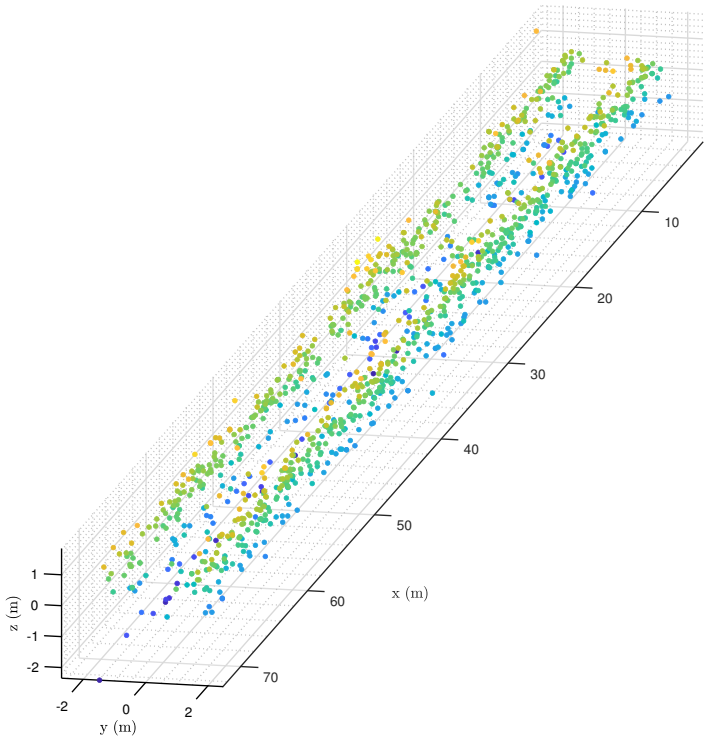


FIGURE 6.3: Representation of the point-cloud obtained by the proposed imaging RASCA technique for the scenario proposed in Fig. 6.1.

7 | Conclusions and future works

In this thesis, an overview of various signal processing techniques developed for colocated MIMO radars operating at mm-waves has been provided. Firstly, its has been shown that the CLEAN algorithm and the MWLA, proposed for the first time for processing the measurements provided by commercial radars operating in *C* and *E* band, allow to achieve a good resolution in the estimation of range, DOA and amplitude of multiple targets in both 2D and a 3D propagation scenarios.

Nevertheless, the need of a more computationally efficient technique for generating radar point-clouds has led to the development of: a) the CSDE (CSDEC) method for the detection and the estimation of range and complex amplitude of single (multiple) target(s); b) the RASCA technique for detecting multiple targets, estimating their range, DOAs and amplitude in both 2D and a 3D scenarios. These methods have been experimentally tested on the measurements acquired from a state of the art SFCW MIMO radar, operating in *E* band. The experimental results have led to the conclusion that a sufficiently good accuracy is achieved by all the proposed methods at the price of a reasonable computational complexity.
During my doctorate studies, I have employed deterministic algorithms especially for range and angular estimation; however, in future works it could be very interesting to apply these methods also along the Doppler domain for a more fined velocity estimation respect traditional methods.

The use of various machine learning and deep learning techniques within the field of colocated MIMO radars has been analysed in detail to assess the potentialities provided by colocated MIMO radars in complicated highly dynamic scenarios. In particular, five machine learning methods and a CNN were successfully employed for classifying three different human activities on the basis of a custom designed dataset; this has been generated by extracting multiple features from the measurements collected though a compact mm-wave FMCW radar. Additionally, two data driven approaches have been proposed for detecting a single target and estimating its position in a bidimensional scenario in presence of disturbances. Even in this case, the availability of a custom designed dataset composed by the radar features obtained from data collected though a compact mm-wave FMCW radar and a TOF camera has allowed me to show that the selected network can detect the considered target and estimate its position with a reasonably good accuracy. However, learning based techniques applied to data collected with MIMO radars may be used in future works for solving different problems like preventing or detecting the fall of elderly people and estimating their vital signs or recognizing different human gestures; this last application is very interested for modern and smart infotainment systems. Moreover, as far as obstacle detection and localization is concerned, other scenarios and targets constituted by different materials are needed for testing the proposed network. In particular, with a more heterogeneous dataset, it is

possible to implement also a new network for both classifying and detecting the obstacles.

Finally, a possible application of colocated MIMO radars to autonomous driving in smart agriculture has been described. In particular, it has been shown that a regression algorithm based on radar data and the RASCA technique can be employed to reconstruct the profile of a vineyard. Also in this case some improvements are possible in future. In particular, future directions may concern with the implementation of filtering techniques and learning based methods able to cope with point-clouds obtained from multiple sensors for estimating the trajectory of the vehicle.

All these applications demonstrate how modern and compact MIMO radars may be employed in a number of applications, not only in the automotive market; in fact, they allow to detect anomalies, gestures, people, falls and, more generally, everything that may happen on a particular scenario and in the presence of different light and environment conditions. However, their future seems to be strictly related to that of automotive field, because of the huge efforts devoted to the development of the first, autonomous and driver-less car.

Bibliography

[1] H. Essen, S. Stanko, R. Sommer, *et al.*, "A high performance 220-ghz broadband experimental radar," in *2008 33rd International Conference on Infrared, Millimeter and Terahertz Waves*, 2008, pp. 1–1.

[2] J. Winters, "On the Capacity of Radio Communication Systems with Diversity in a Rayleigh Fading Environment," *IEEE J. Sel. Areas Commun.*, vol. 5, no. 5, pp. 871–878, 1987.

[3] G. J. Foschini, "Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multi-Element Antennas," *Bell Labs Tech. J.*, pp. 41–59, 1996.

[4] G. Foschini and M. Gans, "On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas," *Wireless Pers. Commun.*, vol. 6, no. 3, pp. 311–335, 1998, ISSN: 1572-834X.

[5] E. Telatar, "Capacity of multi-antenna gaussian channels," *Eur. Trans. Telecommun.*, vol. 10, no. 6, pp. 585–595, 1999.

[6] G. Vitetta, D. Taylor, G. Colavolpe, F. Pancaldi, and P. Martin, ***Wireless Communications: Algorithmic Techniques***. New York: Wiley, 2013, ISBN: 978-0470512395.

[7] E. Fishler, A. Haimovich, R. Blum, D. Chizhik, L. Cimini, and R. Valenzuela, "MIMO radar: an idea whose time has come," in *Proc. 2004 IEEE Radar Conf.*, 2004, pp. 71–78.

[8] D. W. Bliss and K. W. Forsythe, "Multiple-input multiple-output (MIMO) radar and imaging: degrees of freedom and resolution," in *Proc. 37th Asilomar Conf. Signals, Sys. Comput.*, vol. 1, 2003, pp. 54–59.

[9] J. Li and P. Stoica, "MIMO Radar with Colocated Antennas," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 106–114, 2007.

[10] A. M. Haimovich, R. S. Blum, and L. J. Cimini, "MIMO Radar with Widely Separated Antennas," *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 116–129, 2008.

[11] E. Fishler, A. Haimovich, R. Blum, R. Cimini, D. Chizhik, and R. Valenzuela, "Performance of MIMO radar systems: advantages of angular diversity," in *Conf. Rec. 38th Asilomar Conf. Signals, Sys. Comput.*, vol. 1, 2004, pp. 305–309.

[12] M. A. Richards, ***Fundamentals of radar signal processing***. New York: McGraw-Hill Education, 2005.

[13] C. Pfeffer, R. Feger, C. Wagner, and A. Stelzer, "FMCW MIMO Radar System for Frequency-Division Multiple TX-Beamforming," vol. 61, no. 12, pp. 4262–4274, 2013.

[14] R. Feger, C. Pfeffer, and A. Stelzer, "A frequency-division MIMO FMCW radar system using delta-sigma-based transmitters," in *Proc. 2014 IEEE MTT-S Int. Microw. Symp. (IMS)*, 2014, pp. 1–4.

[15] D. Schindler, B. Schweizer, C. Knill, J. Hasch, and C. Waldschmidt, "MIMO-OFDM Radar Using a Linear Frequency Modulated Carrier to Reduce Sampling Requirements," vol. 66, no. 7, pp. 3511–3520, 2018, ISSN: 1557-9670.

[16] H. Griffiths, P. Knott, and W. Koch, "Christian Hülsmeyer: Invention and Demonstration of Radar, 1904," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 9, pp. 56–60, 2019.

[17] I. C. Society, ***A Brief History of Communications**: IEEE Communications Society - a Fifty-year Foundation for the Future*. The Society, 2002, ISBN: 9780780398252.

[18] N. Wiener, ***Extrapolation, Interpolation, and Smoothing of Stationary Time Series: with Engineering Applications***. Technology Press of the Massachusetts Institute of Technology, 1949.

[19] A. Fenn, D. Temme, W. Delaney, and W. Courtney, "The Development of Phased-Array Radar Technology," *Lincoln Lab. J.*, vol. 12, 2000.

[20] P. Barton, "Digital beam forming for radar," *IEE Proc. F - Commun., Radar Signal Process.*, vol. 127, no. 4, pp. 266–277, 1980.

[21] S. H. Talisa, K. W. O'Haver, T. M. Comberiate, M. D. Sharp, and O. F. Somerlock, "Benefits of Digital Phased Array Radars," *Proc. IEEE*, vol. 104, no. 3, pp. 530–543, 2016.

[22] J. Li and P. Stoica, ***MIMO Radar Signal Processing***. New York: Wiley, 2008, pp. 1–448.

[23] Luzhou Xu, Jian Li, and P. Stoica, "Radar imaging via adaptive MIMO techniques," in *Proc. 14th Eur. Signal Process. Conf.*, 2006, pp. 1–5.

[24] M. Kronauge and H. Rohling, "New chirp sequence radar waveform," vol. 50, no. 4, pp. 2870–2877, 2014.

[25] T. Spreng, U. Prechtel, B. Schönlinner, V. Ziegler, A. Meusling, and U. Siart, "UWB near-field MIMO radar: Calibration, measurements and image reconstruction," in *Proc. 2013 Eur. Radar Conf. (EuRAD)*, 2013, pp. 33–36.

[26] C. Sturm and W. Wiesbeck, "Waveform Design and Signal Processing Aspects for Fusion of Wireless Communications and Radar Sensing," *Proc. IEEE*, vol. 99, no. 7, pp. 1236–1259, 2011.

[27] C. Pfeffer, R. Feger, and A. Stelzer, "A stepped-carrier 77-GHz OFDM MIMO radar system with 4 GHz bandwidth," in *Proc. 2015 Eur. Radar Conf. (EuRAD)*, 2015, pp. 97–100.

[28] A. Bourdoux, U. Ahmad, D. Guermandi, S. Brebels, A. Dewilde, and W. Van Thillo, "PMCW waveform and MIMO technique for a 79 GHz CMOS automotive radar," in *Proc. 2016 IEEE Radar Conf. (RadarConf)*, 2016, pp. 1–5.

[29] D. E. Dudgeon, "Fundamentals of digital array processing," *Proc. IEEE*, vol. 65, no. 6, pp. 898–904, 1977.

[30] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[31] P. Stoica, Zhisong Wang, and Jian Li, "Robust Capon beamforming," *IEEE Signal Process. Lett.*, vol. 10, no. 6, pp. 172–175, 2003.

[32] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, 1986, ISSN: 1558-2221.

[33] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," vol. 37, no. 7, pp. 984–995, 1989, ISSN: 0096-3518.

[34] J. Li and R. T. Compton, "Angle and polarization estimation using ESPRIT with a polarization sensitive array," vol. 39, no. 9, pp. 1376–1383, 1991.

[35] F. C. Robey, S. Coutts, D. Weikle, J. C. McHarg, and K. Cuomo, "MIMO radar theory and experimental results," in *Conf. Rec. 38th Asilomar Conf. Signals, Sys. Comput.*, vol. 1, 2004, pp. 300–304.

[36] C. Duo-fang, C. Bai-xiao, and Q. Guo-dong, "Angle estimation using ESPRIT in MIMO radar," in *Electron. Lett.*, vol. 44, 2008, pp. 770–771.

[37] D. Ciuonzo, G. Romano, and R. Solimene, "Performance Analysis of Time-Reversal MUSIC," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2650–2662, 2015.

[38] F. Belfiori, W. van Rossum, and P. Hoogeboom, "Application of 2D MUSIC algorithm to range-azimuth FMCW radar data," in *Proc. 9th Eur. Radar Conf.*, 2012, pp. 242–245.

[39] D. Cohen and Y. C. Eldar, "Sub-Nyquist Radar Systems: Temporal, Spectral, and Spatial Compression," *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 35–58, 2018.

[40] L. Zhao, L. Wang, L. Yang, A. M. Zoubir, and G. Bi, "The Race to Improve Radar Imagery: An overview of recent progress in statistical sparsity-based techniques," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 85–102, 2016.

[41] S. Fortunati, R. Grasso, F. Gini, M. Greco, and K. Lepage, "Single-snapshot DOA estimation by using Compressed Sensing," *EURASIP J. Adv. Signal Process. (JASP)*, 2014.

[42] G. Hueber and A. M. Niknejad, ***Millimeter-Wave Circuits for 5G and Radar***, ser. The Cambridge RF and Microwave Engineering Series. Cambridge University Press, 2019.

[43] J. Bock, H. Schafer, K. Aufinger, *et al.*, "SiGe bipolar technology for automotive radar applications," in *Bipolar/BiCMOS Circuits and Technol., Proc. 2004 Meeting*, 2004, pp. 84–87.

[44] A. Hajimiri, H. Hashemi, A. Natarajan, Xiang Guan, and A. Komijani, "Integrated Phased Array Systems in Silicon," *Proc. IEEE*, vol. 93, no. 9, pp. 1637–1655, 2005.

[45] K. Koh and G. M. Rebeiz, "An X- and Ku-Band 8-Element Phased-Array Receiver in 0.18-μm SiGe BiCMOS Technology," *IEEE J. Solid-State Circuits*, vol. 43, no. 6, pp. 1360–1371, 2008.

[46] H. P. Forstner, H. Knapp, H. Jager, *et al.*, "A 77GHz 4-channel automotive radar transceiver in SiGe," in *Proc. 2008 IEEE Radio Freq. Integr. Circuits Symp.*, 2008, pp. 233–236.

[47] D. Freundt and B. Lucas, "Long Range Radar Sensor for High-Volume Driver Assistance Systems Market," in *SAE Techn. Paper*, SAE International, 2008.

[48] C. M. Schmid, R. Feger, C. Wagner, and A. Stelzer, "Design of a linear non-uniform antenna array for a 77-GHz MIMO FMCW radar," in *Proc. 2009 IEEE MTT-S Int. Microw. Workshop Wireless Sens., Local Positioning, RFID*, 2009, pp. 1–4.

[49] Z. Tong, A. Stelzer, and E. Kolmhofer, "77 GHz center-fed differential microstrip antenna array," in *Proc. 5th Eur. Conf. Antennas Propag. (EUCAP)*, 2011, pp. 583–586.

[50] J. Lee, Y. Li, M. Hung, and S. Huang, "A Fully-Integrated 77-GHz FMCW Radar Transceiver in 65-nm CMOS Technology," *IEEE J. of Solid-State Circuits*, vol. 45, no. 12, pp. 2746–2756, 2010.

[51] L. Zheng, M. Lops, Y. C. Eldar, and X. Wang, "Radar and Communication Coexistence: An Overview: A Review of Recent Methods," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 85–99, 2019.

[52] C. Wagner, A. Stelzer, and H. Jager, "PLL Architecture for 77-GHz FMCW Radar Systems with Highly-Linear Ultra-Wideband Frequency Sweeps," in *2006 IEEE MTT-S International Microwave Symposium Digest*, 2006, pp. 399–402.

[53] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, "Millimeter Wave Technology for Automotive Radar Sensors in the 77 GHz Frequency Band," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 3, pp. 845–860, 2012.

[54] S. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive Radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, pp. 22–35, 2017.

[55] C. A. Balanis, ***Antenna theory: analysis and design***. Wiley-Interscience, 2005.

[56] M. Harter, T. Mahler, T. Schipper, A. Ziroff, and T. Zwick, "2-D antenna array geometries for MIMO radar imaging by Digital Beamforming," in *2013 European Radar Conference*, 2013, pp. 383–386.

[57] J. Gamba, ***Radar Signal Processing for Autonomous Driving***. Springer Publishing Company Inc., 2019, ISBN: 9789811391927.

[58] I. Bilik, O. Longman, S. Villeval, and J. Tabrikian, "The Rise of Radar for Autonomous Vehicles: Signal Processing Solutions and Future Research Directions," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 20–31, 2019, ISSN: 1558-0792.

[59] F. Engels, P. Heidenreich, A. M. Zoubir, F. K. Jondral, and M. Wintermantel, "Advances in Automotive Radar: A framework on computationally efficient high-resolution frequency estimation," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 36–46, 2017.

[60] J. Selva, "ML Estimation and Detection of Multiple Frequencies Through Periodogram Estimate Refinement," vol. 24, no. 3, pp. 249–253, 2017.

[61] E. Sirignano, A. Davoli, G. M. Vitetta, and F. Viappiani, "A Comparative Analysis of Deterministic Detection and Estimation Techniques for MIMO SFCW Radars," *IEEE Access*, vol. 7, pp. 129 848–129 861, 2019.

[62] *Vayyar*. [Online]. Available: https://vayyar.com/.

[63] B. Kim, S. Kim, and J. Lee, "A Novel DFT-Based DOA Estimation by a Virtual Array Extension Using Simple Multiplications for FMCW Radar," 2018.

[64] S. Shirodkar, P. Barua, D. Anuradha, and R. Kuloor, "Heart-beat detection and ranging through a wall using ultra wide band radar," in *2011 Int. Conf. on Comm. and Sig. Proc.*, Calicut, India, 2011, pp. 579–583.

[65] H. Lv, T. Jiao, Y. Zhang, F. Liang, F. Qi, and J. Wang, "A Novel Method for Breath Detection via Stepped-Frequency Continuous Wave Ultra-Wideband (SFCW UWB) Radars Based on Operational Bandwidth Segmentation," *Sensors*, vol. 18, p. 3873, Nov. 2018.

[66] F. Gumbmann and A. Schiessl, "Short-Range Imaging System With a Nonuniform SFCW approach," *IEEE Trans. Microw. Theory Tech.*, vol. 65, no. 4, pp. 1345–1354, 2017.

[67] F. Belfiori, W. v. Rossum, and P. Hoogeboom, "Coherent MUSIC technique for range/angle information retrieval: Application to a frequency-modulated continuous wave MIMO radar," *IET Radar, Sonar Navigation*, vol. 8, no. 2, pp. 75–83, 2014, ISSN: 1751-8784.

[68] J. A. Högbom, "Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines," *Astron. Astrophys. Suppl.*, vol. 15, p. 417, 1974.

[69] D. Zankl, S. Schuster, R. Feger, and A. Stelzer, "What a Blast!: A Massive MIMO Radar System for Monitoring the Surface in Steel Industry Blast Furnaces," *IEEE Microw. Mag.*, vol. 18, pp. 52–69, 2017.

[70] M. Wax and A. Leshem, "Joint estimation of time delays and directions of arrival of multiple reflections of a known signal," *IEEE Trans. Sig. Proc.*, vol. 45, pp. 2477–2484, 1997.

[71] I. Ziskind and M. Wax, "Maximum likelihood localization of multiple sources by alternating projection," *IEEE Trans. Acoust., Speech, and Sig. Process.*, vol. 36, no. 10, pp. 1553–1560, 1988, ISSN: 0096-3518.

[72] T. K. Moon, "The expectation-maximization algorithm," *IEEE Sig. Proc. Mag.*, vol. 13, no. 6, pp. 47–60, 1996, ISSN: 1053-5888.

[73] M. Wax and I. Ziskind, "Detection of the number of coherent signals by the MDL principle," *IEEE Trans. Acoust., Speech and Signal Process.*, vol. 37, pp. 1190–1196, 1989.

[74] H. Akaike, "Information theory and an Extension of the Maximum Likelihood Principle," *Proc. of 2nd Int. Symp. on Information Theory*, pp. 267–281, 1973.

[75] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Trans. Acoust., Speech and Signal Process.*, vol. 36, pp. 477–489, 1988.

[76] J. Selva, "Efficient Wideband DOA Estimation Through Function Evaluation Techniques," *IEEE Trans. Sig. Proc.*, vol. 66, no. 12, pp. 3112–3123, 2018, ISSN: 1053-587X.

[77] B. Ait-El-Fquih and I. Hoteit, "A Variational Bayesian Multiple Particle Filtering Scheme for Large-Dimensional Systems," *IEEE Trans. Sig. Proc.*, vol. 64, no. 20, pp. 5409–5422, 2016, ISSN: 1053-587X.

[78] J. Selva, "An efficient Newton-type method for the computation of ML estimators in a uniform linear array," *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 2036–2045, 2005.

[79] D. Rife and R. Boorstyn, "Single Tone Parameter Estimation from Discrete-Time Observations," *IEEE Trans. Inf. Theory*, vol. 20, no. 5, pp. 591–598, Sep. 1974.

[80] E. Aboutanios and B. Mulgrew, "Iterative Frequency Estimation by Interpolation on Fourier Coefficients," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1237–1242, Apr. 2005.

[81] P. Gough, "A fast spectral estimation algorithm based on the FFT," *IEEE Trans. Signal Process.*, vol. 42, no. 6, pp. 1317–1322, Jun. 1994.

[82] J. Li and P. Stoica, "Efficient Mixed-Spectrum Estimation with Applications to Target Feature Extraction," *IEEE Trans. Signal Process.*, vol. 44, no. 2, pp. 281–295, Feb. 1996.

[83] M. Macleod, "Fast nearly ML estimation of the parameters of real or complex single tones or resolved multiple tones," *IEEE Trans. Signal Process.*, vol. 46, no. 1, pp. 141–148, Jan. 1998.

[84] S. Ye and E. Aboutanios, "An Algorithm for the Parameter Estimation of Multiple Superimposed Exponentials in Noise," in *Proc. of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 3457–3461.

[85] ——, "Rapid Accurate Frequency Estimation of Multiple Resolved Exponentials in Noise," *Signal Processing*, vol. 132, pp. 29–39, Mar. 2017.

[86] C. M. Bishop, ***Pattern Recognition and Machine Learning (Information Science and Statistics)***. Berlin, Heidelberg: Springer-Verlag, 2006, ISBN: 0387310738.

[87] O. Simeone, "A Very Brief Introduction to Machine Learning With Applications to Communication Systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, 2018.

[88] *Imaging radar using cascaded mmwave sensor reference design - tidep-01012*. [Online]. Available: https://www.ti.com/tool/TIDEP-01012.

[89] T.-T. V. Cao, J. Palmer, and P. E. Berry, "False alarm control of CFAR algorithms with experimental bistatic radar data," in *2010 IEEE Radar Conference*, 2010, pp. 156–161.

[90] *Pico-flexx*. [Online]. Available: https://pmdtec.com/picofamily/flexx/.

[91] G. M. Vitetta, F. Pancaldi, D. P. Taylor, and P. Martin, ***Wireless Communications: Algorithmic Techniques***. John Wiley & Sons, to appear.

[92] A. A. Zaidi, R. Baldemair, V. Moles-Cases, N. He, K. Werner, and A. Cedergren, "OFDM Numerology Design for 5G New Radio to Support IoT, eMBB, and MBSFN," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 78–83, 2018.

[93] S. Srikanth, P. A. Murugesa Pandian, and X. Fernando, "Orthogonal frequency division multiple access in WiMAX and LTE: a comparison," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 153–161, 2012.

[94] X. Wang and S. B. Wicker, "Channel estimation and feedback with continuous time domain parameters," in *Proc. 2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 4306–4312.

[95] H. Van Trees, ***Optimum Array Processing – Part IV of Detection, Estimation, and Modulation Therory***. New York: Wiley, 2002.

[96] G. Hakobyan and B. Yang, "High-Performance Automotive Radar: A Review of Signal Processing Algorithms and Modulation Schemes," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 32–44, 2019.

[97] S. Saponara, M. S. Greco, and F. Gini, "Radar-on-Chip/in-Package in Autonomous Driving Vehicles and Intelligent Transport Systems: Opportunities and Challenges," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 71–84, 2019.

[98] J. Yu and J. Krolik, "MIMO adaptive beamforming for nonseparable multipath clutter mitigation," vol. 50, no. 4, pp. 2604–2618, 2014.

[99] N. Shlezinger, R. Fu, and Y. C. Eldar, "Deep Soft Interference Cancellation for MIMO Detection," in *Proc. 2020 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 8881–8885.

[100] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[101] S. Z. Gurbuz and M. G. Amin, "Radar-Based Human-Motion Recognition With Deep Learning: Promising applications for indoor monitoring," *IEEE Signal Process. Mag.*, vol. 36, no. 4, pp. 16–28, 2019, ISSN: 1053-5888.

[102] J. Lombacher, M. Hahn, J. Dickmann, and C. Wöhler, "Potential of radar for static object classification using deep learning methods," in *Proc. 2016 IEEE MTT-S Int. Conf. Microw. Intell. Mobility (ICMIM)*, 2016, pp. 1–4.

[103] T. Cover and P. Hart, "Nearest neighbor pattern classification," vol. 13, no. 1, pp. 21–27, 1967, ISSN: 1557-9654.

[104] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[105] R. E. Schapire, "A Brief Introduction to Boosting," in *Proc. 16th Int. Joint Conf. Artif. Intell. (IJCAI'99)*, vol. 2, 1999, 1401–1406.

[106] J. Fürnkranz, "Round Robin Classification," *J. Mach. Learn. Res.*, vol. 2, pp. 721–747, 2002.

[107] S. Escalera, O. Pujol, and P. Radeva, "On the Decoding Process in Ternary Error-Correcting Output Codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 120–34, 2010.

[108] O. Simeone, "**A Brief Introduction to Machine Learning for Engineers**," in Now Foundations and Trends, 2018.

[109] I. Jolliffe, ***Principal Component Analysis***, M. Lovric, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[110] S. Lloyd, "Least squares quantization in PCM," vol. 28, no. 2, pp. 129–137, 1982.

[111] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–44, 2015.

[112] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[113] M. Møller, "A Scaled Conjugate Gradient Algorithm For Fast Supervised Learning," *Neural Networks*, vol. 6, pp. 525–533, 1993.

[114] M. Gori, ***Machine Learning: A Constraint-Based Approach***, 1st. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2017, ISBN: 0081006594.

[115] E. Mason, B. Yonel, and B. Yazici, "Deep learning for radar," in *Proc. 2017 IEEE Radar Conf.*, 2017, pp. 1703–1708.

[116] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990, ISSN: 1558-2256.

[117] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training Recurrent Neural Networks," *Proc. 30th Int. Conf. Mach. Learn. (ICML 2013)*, 2012.

[118] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural comput.*, vol. 9, pp. 1735–80, 1997.

[119] It.mathworks.com, *Phased Array System Toolbox*, 2020. [Online]. Available: https://it.mathworks.com/products/phased-array.html.

[120] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. Int. Conf. Learn. Representations*, 2014.

[121] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative Adversarial Networks," *Adv. in Neural Inf. Process. Sys.*, vol. 3, 2014.

[122] B. Vandersmissen, N. Knudde, A. Jalalvand, *et al.*, "Indoor Person Identification Using a Low-Power FMCW Radar," en, vol. 56, no. 7, pp. 3941–3952, 2018.

[123] Youngwook Kim, Sungjae Ha, and Jihoon Kwon, "Human Detection Using Doppler Radar Based on Physical Characteristics of Targets," en, vol. 12, no. 2, pp. 289–293, 2015.

[124] Y. Li, Z. Peng, R. Pal, and C. Li, "Potential Active Shooter Detection Based on Radar Micro-Doppler and Range-Doppler Analysis Using Artificial Neural Network," en, *IEEE Sensors J.*, vol. 19, no. 3, pp. 1052–1063, 2019.

[125] F. Fioranelli, M. Ritchie, and H. Griffiths, "Classification of Unarmed/Armed Personnel Using the NetRAD Multistatic Radar for Micro-Doppler and Singular Value Decomposition Features," vol. 12, no. 9, pp. 1933–1937, 2015.

[126] Y. Kim and H. Ling, "Human Activity Classification Based on Micro-Doppler Signatures Using a Support Vector Machine," vol. 47, no. 5, pp. 1328–1337, 2009.

[127] P. Molchanov, J. Astola, K. Egiazarian, and A. Totsky, "Ground moving target classification by using DCT coefficients extracted from micro-Doppler radar signatures and artificial neuron network," en, in *Proc. 2011 Microw., Radar, Remote Sens. Symp.*, 2011, pp. 173–176.

[128] R. J. Javier and Y. Kim, "Application of Linear Predictive Coding for Human Activity Classification Based on Micro-Doppler Signatures," vol. 11, no. 10, pp. 1831–1834, 2014, ISSN: 1558-0571.

[129] C. Clemente, L. Pallotta, A. De Maio, J. J. Soraghan, and A. Farina, "A novel algorithm for radar classification based on doppler characteristics exploiting orthogonal Pseudo-Zernike polynomials," vol. 51, no. 1, pp. 417–430, 2015, ISSN: 2371-9877.

[130] J. Zabalza, C. Clemente, G. Di Caterina, Jinchang Ren, J. J. Soraghan, and S. Marshall, "Robust PCA micro-doppler classification using SVM on embedded systems," vol. 50, no. 3, pp. 2304–2310, 2014, ISSN: 2371-9877.

[131] G. E. Smith, K. Woodbridge, and C. J. Baker, "Naive Bayesian radar micro-doppler recognition," in *Proc. 2008 Int. Radar Conf.*, 2008, pp. 111–116.

[132] S. Björklund, T. Johansson, and H. Petersson, "Evaluation of a micro-Doppler classification method on mm-wave data," in *Proc. 2012 IEEE Radar Conf.*, 2012, pp. 0934–0939.

[133] S. Abdulatif, Q. Wei, F. Aziz, B. Kleiner, and U. Schneider, "Micro-Doppler Based Human-Robot Classification Using Ensemble and Deep Learning Approaches," en, *Proc. 2018 IEEE Radar Conf. (RadarConf18)*, pp. 1043–1048, 2018.

[134] Y. Kim and T. Moon, "Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks," vol. 13, no. 1, pp. 8–12, 2016, ISSN: 1545-598X.

[135] M. S. Seyfioğlu, A. M. Özbayoğlu, and S. Z. Gürbüz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," vol. 54, no. 4, pp. 1709–1723, 2018, ISSN: 2371-9877.

[136] S. Abdulatif, K. Armanious, F. Aziz, U. Schneider, and B. Yang, "Towards Adversarial Denoising of Radar Micro-Doppler Signatures," in *Proc. 2019 Int. Radar Conf.*, 2019, pp. 1–6.

[137] S. Abdulatif, F. Aziz, K. Armanious, B. Kleiner, B. Yang, and U. Schneider, "Person Identification and Body Mass Index: A Deep Learning-Based Study on Micro-Dopplers," in *Proc. 2019 IEEE Radar Conf. (RadarConf)*, 2019, pp. 1–6.

[138] G. Malysa, D. Wang, L. Netsch, and M. Ali, "Hidden Markov model-based gesture recognition with FMCW radar," in *Proc. 2016 IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, 2016, pp. 1017–1021.

[139] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with Soli: Exploring Fine-Grained Dynamic Gesture Recognition in the Radio-Frequency Spectrum," in *Proc. 29th Annu. Symp. User Interface Softw. Technol. (UIST '16)*, 2016, pp. 851–860.

[140] Y. Kim and B. Toomajian, "Hand Gesture Recognition Using Micro-Doppler Signatures With Convolutional Neural Network," en, *IEEE Access*, vol. 4, pp. 7125–7130, 2016, ISSN: 2169-3536.

[141] Z. Peng, C. Li, J.-M. Munoz-Ferreras, and R. Gomez-Garcia, "An FMCW radar sensor for human gesture recognition in the presence of multiple targets," in *Proc. 1st IEEE MTT-S Int. Microw. Bio Conf. (IMBIOC)*, 2017, pp. 1–3.

[142] Z. Zhang, Z. Tian, and M. Zhou, "Latern: Dynamic Continuous Hand Gesture Recognition Using FMCW Radar Sensor," *IEEE Sensors J.*, vol. 18, pp. 3278–3289, 2018.

[143] Z. Peng, J.-M. Munoz-Ferreras, R. Gomez-Garcia, and C. Li, "FMCW radar fall detection based on ISAR processing utilizing the properties of RCS, range, and Doppler," in *Proc. 2016 IEEE MTT-S Int. Microw. Symp. (IMS)*, 2016, pp. 1–3.

[144] B. Jokanović and M. Amin, "Fall detection using deep learning in range-doppler radars," vol. 54, no. 1, pp. 180–189, 2018.

[145] H. Hong, L. Zhang, C. Gu, Y. Li, G. Zhou, and X. Zhu, "Noncontact Sleep Stage Estimation Using a CW Doppler Radar," *IEEE Trans. Emerg. Sel. Topics in Circuits and Syst.*, vol. 8, no. 2, pp. 260–270, 2018.

[146] J. Saluja, J. Casanova, and J. Lin, "A Supervised Machine Learning Algorithm for Heart-Rate Detection Using Doppler Motion-Sensing Radar," *IEEE J. Electromagn., RF and Microw. in Medicine and Biol.*, vol. 4, no. 1, pp. 45–51, 2020.

[147] C. Gu, J. Wang, and J. Lien, "Deep Neural Network based Body Movement Cancellation for Doppler Radar Vital Sign Detection," in *Proc. 2019 IEEE MTT-S Int. Wireless Symp. (IWS)*, 2019, pp. 1–3.

[148] H. Zhao, H. Hong, D. Miao, *et al.*, "A Noncontact Breathing Disorder Recognition System Using 2.4-GHz Digital-IF Doppler Radar," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 1, pp. 208–217, 2019.

[149] J. Le Kernec, F. Fioranelli, C. Ding, *et al.*, "Radar Signal Processing for Sensing in Assisted Living: The challenges associated with real-time implementation of emerging algorithms," *IEEE Signal Process. Mag.*, vol. 36, no. 4, pp. 29–41, 2019, ISSN: 1053-5888.

[150] H. Rohling, "Radar CFAR Thresholding in Clutter and Multiple Target Situations," pp. 608–621, 1983.

[151] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. 2nd Int. Conf. Knowl. Discovery, Data Mining*, Portland, Oregon, 1996, 226–231.

[152] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Wheel extraction based on micro doppler distribution using high-resolution radar," in *Proc. 2015 IEEE MTT-S Int. Conf. Microw. Intell. Mobility (ICMIM)*, 2015.

[153] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probability*, vol. 1, 1967, pp. 281–297.

[154] A. El Gonnouni, M. Martinez-Ramon, J. L. Rojo-Alvarez, G. Camps-Valls, A. R. Figueiras-Vidal, and C. G. Christodoulou, "A Support Vector Machine MUSIC Algorithm," vol. 60, no. 10, pp. 4901–4910, 2012, ISSN: 1558-2221.

[155] Y. Kim, "Detection of Eye Blinking Using Doppler Sensor With Principal Component Analysis," en, *IEEE Antennas Wireless Propag. Lett.*, vol. 14, pp. 123–126, 2015, ISSN: 1536-1225, 1548-5757.

[156] A. Coluccia, A. Fascista, and G. Ricci, "Robust CFAR Radar Detection Using a K-nearest Neighbors Rule," in *Proc. 2020 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 4692–4696.

[157] S. Capobianco, L. Facheris, F. Cuccoli, and S. Marinai, "Vehicle Classification Based on Convolutional Networks Applied to FMCW Radar Signals," in *Traffic Mining Appl. to Police Activities - Proc. 1st Italian Conf. Traffic Police (TRAP-2017)*, vol. 728, 2017, pp. 115–128.

[158] T. Giese, J. Klappstein, J. Dickmann, and C. Wöhler, "Road course estimation using deep learning on radar data," in *Proc. 18th Int. Radar Symp. (IRS 2017)*, 2017, pp. 1–7.

[159] H. Dbouk, H. Geng, C. M. Vineyard, and N. R. Shanbhag, "Low-Complexity Fixed-Point Convolutional Neural Networks For Automatic Target Recognition," in *Proc. 2020 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 1598–1602.

[160] K. Patel, K. Rambach, T. Visentin, D. Rusev, M. Pfeiffer, and B. Yang, "Deep Learning-based Object Classification on Automotive Radar Spectra," in *Proc. 2019 IEEE Radar Conf. (RadarConf)*, 2019.

[161] J. M. García, D. Zoeke, and M. Vossiek, "MIMO-FMCW Radar-Based Parking Monitoring Application With a Modified Convolutional Neural Network With Spatial Priors," *IEEE Access*, vol. 6, pp. 41 391–41 398, 2018.

[162] B. Major, D. Fontijne, A. Ansari, *et al.*, "Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors," in *Proc. 2019 IEEE/CVF Int. Conf. Comput. Vision Workshop (ICCVW)*, 2019, pp. 924–932.

[163] J. Mun, H. Kim, and J. Lee, "A Deep Learning Approach for Automotive Radar Interference Mitigation," in *Proc. 2018 IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, 2018, pp. 1–5.

[164] J. Mun, S. Ha, and J. Lee, "Automotive Radar Signal Interference Mitigation Using RNN with Self Attention," in *Proc. 2020 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 3802–3806.

[165] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Neural Inf. Process. Syst.*, vol. 25, 2012.

[166] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. 2009 IEEE Conf. Comput. Vision Pattern Recogn.*, 2009, pp. 248–255.

[167] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[168] Z. Zheng, T. Ruan, Y. Wei, and Y. Yang, "VehicleNet: Learning Robust Feature Representation for Vehicle Re-identification," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recogn. (CVPR)*, 2019.

[169] M. S. Seyfioğlu and S. Z. Gürbüz, "Deep Neural Network Initialization Methods for Micro-Doppler Classification With Low Training Sample Support," vol. 14, no. 12, pp. 2462–2466, 2017, ISSN: 1558-0571.

[170] J. Park, R. Javier, T. Moon, and Y. Kim, "Micro-Doppler Based Classification of Human Aquatic Activities via Transfer Learning of Convolutional Neural Networks," en, *Sensors*, vol. 16, no. 12, p. 1990, 2016, ISSN: 1424-8220.

[171] M. S. Seyfioglu, B. Erol, S. Z. Gurbuz, and M. G. Amin, "Diversified radar micro-Doppler simulations as training data for deep residual neural networks," in *Proc. 2018 IEEE Radar Conf. (RadarConf18)*, 2018, pp. 0612–0617.

[172] R. Girshick, "Fast R-CNN," in *Proc. 2015 IEEE Int. Conf. Comput. Vision (ICCV)*, 2015, pp. 1440–1448.

[173] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, ISSN: 1939-3539.

[174] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. 2016 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, 2016, pp. 779–788.

[175] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proc. 2017 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, 2017, pp. 6517–6525.

[176] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. 2015 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, 2015, pp. 3431–3440.

[177] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017, ISSN: 1939-3539.

[178] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proc. Med. Image Comput. and Comput.-Assisted Intervention (MICCAI 2015)*, 2015, pp. 234–241.

[179] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. 2017 IEEE Int. Conf. Comput. Vision (ICCV)*, 2017, pp. 2980–2988.

[180] G. Zhang, H. Li, and F. Wenger, "Object Detection and 3d Estimation Via an FMCW Radar Using a Fully Convolutional Network," in *Proc. 2020 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 4487–4491.

[181] O. Bialer, D. Shapiro, and A. Jonas, "Object Surface Estimation from Radar Images," in *Proc. 2020 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 4132–4136.

[182] S. Gasperini, M. Paschali, C. Hopke, D. Wittmann, and N. Navab, "Signal Clustering With Class-Independent Segmentation," in *Proc. 2020 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 3982–3986.

[183] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, "Semantic Segmentation on Radar Point Clouds," *Proc. 21st Int. Conf. Inf. Fusion (FUSION)*, pp. 2179–2186, 2018.

[184] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proc. 2017 IEEE Conf. Comput. Vision Pattern Recogn. (CVPR)*, 2017, pp. 77–85.

[185] J. Pegoraro and M. Rossi, "Real-Time People Tracking and Identification From Sparse mm-Wave Radar Point-Clouds," *IEEE Access*, vol. 9, pp. 78 504–78 520, 2021.

[186] M. Meyer and G. Kuschk, "Automotive Radar Dataset for Deep Learning Based 3D Object Detection," in *Proc. 16th Eur. Radar Conf. (EuRAD)*, 2019, pp. 129–132.

[187] M. Meyer and G. Kuschk, "Deep Learning Based 3D Object Detection for Automotive Radar and Camera," in *Proc. 16th Eur. Radar Conf. (EuRAD)*, 2019, pp. 133–136.

[188] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82 –115, 2020, ISSN: 1566-2535.

[189] M. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Proc. Eur. Conf. Comput. Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8689, 2013.

[190] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," in *Proc. Workshop at Int. Conf. Learn. Representations*, 2014.

[191] *Inras*, 2020. [Online]. Available: http://www.inras.at/.

[192] J. Zhu, S. Rosset, H. Zou, and T. Hastie, "Multi-class AdaBoost," *Statist. and its interface*, vol. 2, 2006.

[193] V. C. Chen, F. Li, S. Ho, and H. Wechsler, "Micro-Doppler effect in radar: phenomenon, model, and simulation study," vol. 42, no. 1, pp. 2–21, 2006, ISSN: 2371-9877.

[194] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[195] *Parallax inc*. [Online]. Available: https://www.parallax.com/.

[196] L. Al Shalabi and Z. Shaaban, "Normalization as a Preprocessing Engine for Data Mining and the Approach of Preference Matrix," *Proc. 2006 Int. Conf. Dependability Comput. Sys.*, pp. 207–214, 2006.