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

COLOCATED MULTIPLE-INPUT MULTIPLE-OUTPUT RADARS FOR SMART MOBILITY

Presentata da: Pasquale Di Viesti

Coordinatore Dottorato

Nicolò Cavina

Supervisore

Giorgio Matteo Vitetta

Co-supervisore

Paolo Pavan

Esame finale anno 2022

Abstract

In recent years, radars have been used in many applications such as *precision agriculture* and *advanced driver assistant systems*. Optimal techniques for the estimation of the overall number of targets and of their spatial coordinates cannot be employed in real world radars, 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. This has motivated the development of various sub-optimal estimation techniques able to achieve good estimation accuracy at a manageable computational cost. Another fundamental technical issue in radar systems for advanced driver assistant systems is the tracking of multiple targets. Even if various filtering techniques have been developed to solve this problem in the past, new computationally efficient and robust algorithms for target tracking can be devised by exploiting a recent probabilistic approach, based on the use of the factor graph and the *sum-product algorithm*.

The two key contributions provided by this dissertation are the investigation of the filtering and smoothing problems from a factor graph perspective and the development of efficient algorithms for two-dimensional and three-dimensional radar imaging. As far as the first contribution is concerned, a new factor graph for the filtering problem is derived and the sum-product rule is applied to this graphical model; this allows to interpret known algorithms and to develop new filtering techniques. Then, a general method, based on graphical modelling, is proposed to derive filtering algorithms that involve a network of interconnected Bayesian filters; moreover, this method is exemplified by devising a new filtering method. Finally, the proposed graphical approach is exploited to devise a new smoothing algorithm. Numerical results for specific dynamic systems evidence that our algorithms can achieve a better complexity-accuracy tradeoff and tracking capability than other filtering and smoothing techniques appeared in the literature. Regarding radar imaging, various algorithms are developed for frequency modulated continuous wave radars; all these algorithms rely on novel and efficient methods for the detection of multiple superimposed tones in noise and the estimation of their parameters. The accuracy achieved by these algorithms in the presence of multiple closely spaced targets is assessed on the basis of both synthetically generated data and of the measurements acquired through two different commercial multiple-input multiple-output radars.

Sommario

Negli ultimi anni, i radar sono stati utilizzati in molte applicazioni come l'*agricoltura di precisione* e i sistemi avanzati di sistemi di assistenza alla guida (*advanced driver assistant systems*). Tecniche ottimali per la stima del numero complessivo di bersagli e delle loro coordinate spaziali non possono essere impiegate nei sistemi radar reali, poiché richiedono la risoluzione di complicati problemi di ottimizzazione multidimensionale e, di conseguenza, comportano un enorme sforzo computazionale, anche in presenza di un piccolo numero di bersagli. Questo ha motivato lo sviluppo di varie tecniche di stima sub-ottima in grado di ottenere una buona precisione di stima ad un costo computazionale gestibile. Un'altra questione tecnica fondamentale nei sistemi radar per sistemi avanzati di assistenza alla guida è l'inseguimento di bersagli multipli. Nonostante varie tecniche di filtraggio siano state sviluppate per risolvere questo problema in passato, nuovi algoritmi computazionalmente efficienti e robusti per l'inseguimento dei bersagli possono essere sviluppati sfruttando un recente approccio probabilistico basato sull'uso dei grafi di fattore e sull'algoritmo somma-prodotto (*sum-product algorithm*).

I due contributi chiave forniti da questa dissertazione sono l'indagine dei problemi di filtraggio e smoothing utilizzando i grafi di fattore e lo sviluppo di algoritmi efficienti per la generazione di immagini radar bidimensionali e tridimensionali. Per quanto riguarda il primo contributo, un nuovo grafo di fattori è stato derivato per il problema del filtraggio e la regola del sum-product è stata applicata a questo modello grafico; questo permette di interpretare algoritmi noti e di sviluppare nuove tecniche di filtraggio. In aggiunta, viene proposto un metodo generale, basato sulla modellazione grafica, per derivare algoritmi di filtraggio che coinvolgono una rete di filtri Bayesiani interconnessi; inoltre, questo metodo è esemplificato dall'ideazione di un nuovo metodo di filtraggio. Infine, l'approccio grafico proposto viene sfruttato per ideare un nuovo algoritmo di smoothing. I risultati numerici per specifici sistemi dinamici dimostrano che i nostri algoritmi possono raggiungere un migliore compromesso complessità-precisione e capacità di tracciamento rispetto ad altre tecniche di filtraggio e smoothing apparse in letteratura. Per quanto riguarda l'imaging radar, vari algoritmi sono stati sviluppati per i radar a onda continua modulati in frequenza (*frequency modulated continuous wave radars*); tutti questi algoritmi si basano su metodi nuovi ed efficienti per il rilevamento di toni multipli sovrapposti al rumore e la stima dei loro parametri. La precisione raggiunta da questi algoritmi in presenza di più bersagli ravvicinati è valutata sulla base sia di dati generati sinteticamente che delle misure acquisite attraverso due diversi radar commerciali a ingresso e uscita multipli.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor Professor Giorgio Matteo Vitetta for his invaluable advices, unparalleled support, profound belief in my work and patience during my PhD studies. I wish also to thank my co-supervisor Professor Paolo Pavan for the opportunity to pursue this PhD course and for his support.

I would like to express my appreciation to the reviewers of this dissertation for their valuable suggestions.

I would also like to extend my gratitude to CNH Industrial and in particular to Eng. Luca Ferrari and Eng. Bart Lenaerts for their valuable collaboration and helpful advices.

Many thanks to all my PhD colleagues and laboratory mates for the very nice time spent together at the university supporting each other.

I am extremely grateful to my family: without you this would not have been possible! Finally, thanks to my loving and caring Francesca.

In primo luogo, vorrei esprimere la mia più profonda gratitudine al mio supervisore, il professor Giorgio Matteo Vitetta, per i suoi preziosi consigli, l'impareggiabile sostegno, la profonda fiducia nel mio lavoro e la pazienza dimostrata durante questo percorso di dottorato. Desidero anche ringraziare il mio co-supervisore, il professor Paolo Pavan, per l'opportunità di seguire questo corso di dottorato e per il suo sostegno.

Vorrei esprimere un sentito ringraziamento ai revisori di questa tesi per i loro preziosi suggerimenti.

Vorrei anche estendere la mia gratitudine a CNH Industrial e in particolare all'Ing. Luca Ferrari e all'Ing. Bart Lenaerts per la loro preziosa collaborazione e gli utili consigli.

Molte grazie a tutti i miei colleghi di dottorato e compagni di laboratorio per aver condiviso con me questi anni all'università sostenendoci a vicenda.

Sono estremamente grato alla mia famiglia: senza di voi questo non sarebbe stato possibile! Infine, grazie alla mia amorevole e premurosa Francesca.

Contents

List of Figures

List of Tables

List of Acronyms

AM	alternating minimization
AWGN	additive white Gaussian noise
BIF	backward information filtering
BP	belief propagation
CLG	conditionally linear Gaussian
CRLB	Cramér-Rao lower bound
CSFE	complex single frequency estimator
CSFEC	complex single frequency estimator & cancellation
DBF	double Bayesian filtering
DBIF	double backward information filtering
DBS	double Bayesian smoothing
DBSA	double Bayesian smoothing algorithm
DDBSA	disjoint double Bayesian smoothing algorithm
DFT	discrete Fourier transform
DMPF	dual marginalized particle filtering
DOA	direction of arrival
EKF	extended Kalman filter
EM	expectation maximization
ET	execution time
FFT	Fast Fourier Transform
FMCW	frequency modulated continuous wave
FG	factor graph
GM	Gaussian mixture
HCSFEC	hybrid complex single frequency estimator & cancellation
HSFEC	hybrid single frequency estimator & cancellation
HSTDREC	hybrid single target detection range estimation & cancellation

HULA horizontal uniform linear array

iid independent and identically distributed

KF Kalman filtering

LLR log-likelihood ratio

LNA low noise amplifier

MBF multiple Bayesian filtering

MBFA multiple Bayesian filtering algorithm

MIMO multiple-input multiple-output

ML maximum likelihood

MPF marginalized particle filtering (Chapter 1)

MPF multiple particle filtering (Chapter 2)

MSE mean square error

MU measurement update

MUSIC multiple signal classification

pdf probability density function

PF particle filtering

PM pseudo measurement

RASCA range & angle serial cancellation algorithm

RBPF Rao-Blackwellized particle filtering

RBPS Rao-Blackwellized particle smoothing

RBSS Rao-Blackwellized serial smoothing

RHS right hand side

RMSE root mean square error

RPE range profile estimator

rvs random variables

RX receive

SDBF simplified double Bayesian filtering

SDBSA simplified double Bayesian smoothing algorithm

SDDBSA simplified disjoint double Bayesian smoothing algorithm

SDMPF simplified dual marginalized particle filtering

SFE single frequency estimator

SFEC single frequency estimator & cancellation

SIC serial interference cancellation

SIR	sequential importance resampling
SISO	soft-in soft-out
SMC	sequential monte carlo
SMPF	simplified marginalized particle filtering
SNR	signal to noise ratio
SPA	sum product algorithm
SPE	spatial estimator
SPS	serial particle smoothing
SSM	state space model
STDAE	single target detection angular estimation
STDAEC	single target detection angular estimation & cancellation
STDREC	single target detection range estimation & cancellation
TDM	time division multiplexing
TRP	target range profile
TU	time update
TX	transmit
ULA	uniform linear array
URA	uniform rectangular array
VA	virtual antenna
VULA	vertical uniform linear array

"The formulation of a problem is often more essential than its solution, which may be merely a matter of mathematical or experimental skill. To raise new questions, new possibilities, to regard old problems from a new angle, requires creative imagination and marks real advance in science."

— A. Einstein, *The Evolution Of Physics*

Introduction

In this chapter, the reference scenario which my PhD work refers to is defined first. Then, the scope of this dissertation is illustrated and an outline of its contents is provided. Finally, some notations employed throughout this document are defined.

Reference Scenario

The research work illustrated in this thesis concerns the development of various algorithms that can be employed in modern systems for *radio detection and ranging* (briefly, radar systems). Since its invention, radar has found a number of applications, in both civilian and military fields. The first experimental work on radars, mainly focused on military applications, dates back to the 1930s, although the basic principles can be traced back to experiments on electromagnetic radiation conducted by the German physicist Heinrich Hertz in the 1880s. In 1904, the principles demonstrated by Hertz were put into practice by Christian Huelsmeyer in the realization of an anticollision naval system that was patented in the same year [1]; however, this invention had little interest because of its limitations on the maximum achievable range. Since the 1930s, however, due to the increasing interest in military applications, different countries, including the United States and most of the European countries, began testing radar systems that operated in the *very high frequency* (VHF) band. The experiments carried out by the Radiation Laboratory of the *Massachusetts Institute of Technology* (MIT) on the magnetron paved the way to the use of radar systems operating in the S-band (2–4 GHz) during World War II. In the years following the end of that war, the advances in electronic technology and signal processing led to the development of pulse Doppler radar, monopulse radar, phased array radar and *synthetic aperture radar* (SAR). In the last decades of the 1900s, major efforts have been made to achieve significant advancements in civilian applications (e.g., SAR and weather surveillance radars for ground and meteorological observations).

In the last years, radar systems have been used in different fields [2], including civilian aviation, navigation, Earth observation, meteorology, medicine, precision agriculture and *advanced driver assistant systems* (ADAS). In particular, as far as the last two fields are concerned, such systems can substantially benefit from the availability of the antenna arrays, i.e. of the so called *multiple-input multiple-output* (MIMO) technology, whose commercial use is now possible thanks to the recent advances in millimeter-wave semiconductor technology and to the development of novel signal processing techniques [3]. Radar systems equipped with antenna arrays can be divided in *statistical* MIMO radars [4, 5] and *colocated* MIMO radars [6] on the basis of the distance between their transmit and receive arrays; in the first case, transmit and receive antennas are widely separated, whereas, in the second one, they are closely spaced and, in particular, they are usually placed on the same shield. In this dissertation, we focus on colocated MIMO radars operating at mmWave; such systems play an important role in a number of applications, because of their limited cost, their small size and their ability to detect the presence of multiple targets. However, it is important to keep in mind that the performance achieved by any colocated MIMO radar system depends not only on some important characteristics of its hardware (e.g., the operating

frequency, the number of transmit and receive antennas, the configuration of the transmit and receive arrays, etc.), but also on the techniques that are employed in the generation of its radiated waveforms and in the processing of the measurements acquired through its receive array.

As far as the last issue is concerned, substantial research efforts have been devoted to the investigation of signal processing methods that, based on the measurements acquired from the whole array in a single snapshot (briefly, acquired in a single frame), can estimate the overall number of targets and their spatial coordinates. Unluckily, *optimal* (i.e., *maximum likelihood*, ML) techniques cannot be employed in this case, 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 [2]. This has motivated the development of various *sub-optimal* estimation techniques able to achieve good estimation accuracy at a manageable computational cost. For instance, a well known sub-optimal technique that can be employed in real world radar systems is the one described in ref. [7] for *frequency modulated continuous wave* (FMCW) radar systems. It 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-elevation domain or a range-azimuth domain in *three-dimensional* (3D) and *two-dimensional* (2D) imaging, respectively. Despite the practical importance of this technique, the computational effort it requires is still significant, since it involves multidimensional spectral analysis of the acquired signals. Moreover, it suffers from the following relevant drawback: it can miss 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* [8].

When multiple frames are available, the information extracted from each of them are merged to estimate the trajectory of the detected targets. Information fusion can be accomplished through a proper *filtering technique* if online processing of the available data is required to achieve *target tracking* [9, 10]. However, more accurate results can be obtained if data are processed offline; in this case, a *smoothing technique* can be employed [11]. In the past, substantial research efforts have been devoted to the use of filtering methods for *multi target tracking* (MTT). Common approaches to MTT include a) the *multiple hypothesis tracking* (MHT) or the *joint probabilistic data association* (JPDA) methods (e.g., see [12, Section II] and [13], respectively); b) the exploitation of *factor graphs* (FGs) and the *sum-of-products* (SPA) algorithm [14]. In most cases, a *point target* model has been assumed in the development of such methods. More recently, various researchers have focused on the problem of tracking objects whose physical extent is significantly larger than the resolution of the employed radar sensor; such targets may lead to the generation of multiple measurements [15]. The algorithms available in the technical literature for *extended object tracking* (EOT) are based on:

a) Adaptations of the JPDA or the MHT methods (e.g., see [16] and [12], respectively).

b) Random finite set approaches such as the *probability hypothesis density* (PHD) (e.g., see [17]).

c) The adoption of different models for the representation of extended objects; these models include simple geometrical shapes (e.g. circle, rectangle, or ellipse) [18], more flexible representations known as *random hyper-surface models* (RHM) for star-convex objects [19] and point cloud representations [20].

d) The exploitation of factor graph and the SPA [21, 22].

Although the available methods can be employed in real world applications, they may fail in complicated environments (e.g., in the presence of weak targets in crowded scenarios). In such

cases, provided that an estimation delay is tolerable, more accurate results can be obtained by replacing filtering with smoothing. Various technical solutions are available in the literature; for instance, the use of the *Rauch–Tung–Striebel* (RTS) smoother in conjunction with the MHT has been proposed in ref. [23]. However, such solutions are computationally demanding, especially in the presence of a significant number of targets [24].

Thesis Scope and Outline

My PhD studies have focused on the following two research problems:

1. The development of detection/estimation algorithms for 2D and 3D radar imaging. These algorithms should be able to detect multiple targets and estimate theirs parameters (namely, their range, azimuth and elevation) at the price of a limited computational cost.

2. The exploitation of a FG approach to derive filtering and smoothing techniques that involve a network of interconnected Bayesian filters.

The solutions devised for these problems are new and appealing from a computational perspective. Unluckily, because of time limitations, the application of the developed filtering and smoothing techniques to the MTT problem mentioned in the previous section has not been investigated.

This dissertation is organized as follows. In Part I, a factor graph approach to Bayesian filtering and smoothing is illustrated. In particular, in Chapter 1, this approach is employed to tackle the Bayesian filtering problem for conditionally linear Gaussian state-space models; more specifically, it is shown that: a) marginalized particle filtering can be interpreted as a form of forward only message passing over the devised graph; b) novel filtering methods can be easily developed by exploiting the graph structure and/or simplifying probabilistic messages. Then, in Chapter 2, a general method for deriving filtering algorithms that involve a network of interconnected Bayesian filters is proposed and new filtering techniques, based on this method, are devised. Moreover, it is shown that these algorithms can achieve a better complexity-accuracy tradeoff than marginalized particle filtering and multiple particle filtering. Finally, in Chapter 3, the conceptual approach proposed in Chapter 2 is exploited to devise new Bayesian smoothing algorithms that can achieve a better complexity-accuracy tradeoff and tracking capability than other related algorithms recently appeared in the literature.

In Part II, instead, novel algorithms for 2D and 3D radar imaging are derived. In particular, in Chapter 4, novel methods for the detection of multiple superimposed tones in noise and the estimation of their parameters are derived; these methods can achieve a substantially better complexity-accuracy tradeoff than various related techniques in the presence of multiple closely spaced tones. Finally, in Chapter 5, the problem of detecting multiple targets and estimating their spatial coordinates in a colocated MIMO radar system is investigated. Various solutions are developed for *frequency modulated continuous wave* (FMCW) radars and the accuracy achieved by these algorithms in the presence of multiple targets is assessed on the basis of both synthetically generated data and of the measurements acquired through two commercial MIMO radars.

In both parts, different appendices containing additional results and proofs are provided. Some conclusions and suggestions for future research are given in Chapter 6.

Notations

In Part I the following notations are adopted: a) the *probability density function* (pdf) of a random vector $\mathbf{R}$ evaluated at point $\mathbf{r}$ is denoted $f(\mathbf{r})$; b) $\mathcal{N}(\mathbf{r};\eta_r,\mathbf{C}_r)$ represents the pdf of a Gaussian random vector $\mathbf{R}$ characterized by the mean η_r and covariance matrix $\mathbf{C}_r$ evaluated at point $\mathbf{r}$; c) the precision (or weight) matrix associated with the covariance matrix $\mathbf{C}_r$ is denoted $\mathbf{W}_r$, whereas the transformed mean vector $\mathbf{W}_r\eta_r$ is denoted $\mathbf{w}_r$.

Bibliography

[1] C. Huelsmeyer, "Télémobiloscope," Patent FR343 846A.

[2] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive Radars: A Review of Signal Processing Techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017.

[3] J. Li and P. Stoica, Eds., *MIMO Radar Signal Processing.* Hoboken, NJ: J. Wiley & Sons, 2009.

[4] A. M. Haimovich, R. S. Blum, and L. J. Cimini, "MIMO Radar with Widely Separated Antennas," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 116–129, 2008.

[5] E. Fishler, A. Haimovich, R. Blum, R. Cimini, D. Chizhik, and R. Valenzuela, "Performance of MIMO radar systems: advantages of angular diversity," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, 2004, pp. 305–309 Vol.1.

[6] M. A. Richards, *Fundamentals of Radar Signal Processing*, 2005.

[7] S. Rao, "MIMO Radar," *Texas Instruments - Application Report SWRA554A*, July 2018.

[8] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, "Overview of Bayesian Sequential Monte Carlo Methods for Group and Extended Object Tracking," *Digital Signal Processing*, vol. 25, pp. 1–16, Feb. 2014.

[9] S. Blackman and R. Popoli, *Design and analysis of modern tracking systems.* Norwood, MA: Artech House., 1999.

[10] Y. Bar-Shalom and W. Blair, *Multitarget-Multisensor Tracking: Applications and Advances, Vol. III.* Norwood, MA: Artech House., 2000.

[11] A. Mahalanabis, B. Zhou, and N. Bose, "Improved multi-target tracking in clutter by PDA smoothing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 1, pp. 113–121, 1990.

[12] S. P. Coraluppi and C. A. Carthel, "Multiple-Hypothesis Tracking for Targets Producing Multiple Measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 3, pp. 1485–1498, Jun. 2018.

[13] Y. Bar-Shalom, F. Daum, and J. Huang, "The Probabilistic Data Association Filter," *IEEE Control Syst. Mag.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.

[14] F. Meyer, T. Kropfreiter, J. L. Williams, R. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message Passing Algorithms for Scalable Multitarget Tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[15] J. Gunnarsson, L. Svensson, L. Danielsson, and F. Bengtsson, "Tracking Vehicles Using Radar Detections," in *2007 IEEE Intelligent Vehicles Symposium*, Jun. 2007, pp. 296–302.

[16] G. Vivone and P. Braca, "Joint Probabilistic Data Association Tracker for Extended Target Tracking Applied to X-Band Marine Radar Data," *IEEE Journal of Oceanic Engineering*, vol. 41, no. 4, pp. 1007–1019, 2016.

[17] K. Granstrom, C. Lundquist, and O. Orguner, "Extended Target Tracking Using a Gaussian-Mixture PHD Filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3268–3286, 2012.

[18] L. Hammarstrand, L. Svensson, F. Sandblom, and J. Sorstedt, "Extended Object Tracking Using a Radar Resolution Model," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2371–2386, Jul. 2012.

[19] N. Wahlström and E. Özkan, "Extended Target Tracking Using Gaussian Processes," *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4165–4178, Aug. 2015.

[20] M. Kumru and E. Ozkan, "Three-Dimensional Extended Object Tracking and Shape Learning Using Gaussian Processes," *IEEE Trans. Aerosp. Electron. Syst.*, pp. 1–1, 2021.

[21] F. Meyer and J. L. Williams, "Scalable Detection and Tracking of Extended Objects," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8916–8920.

[22] F. Meyer and J. Williams, "Scalable Detection and Tracking of Geometric Extended Objects," *IEEE Transactions on Signal Processing*, pp. 1–1, 2021.

[23] W. Koch, "Fixed-interval retrodiction approach to Bayesian IMM-MHT for maneuvering multiple targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 2–14, 2000.

[24] S. L. Yeung, S. Tager, P. Wilson, R. Tharmarasa, W. Armour, and J. Thiyagalingam, "A Parallel Retrodiction Algorithm for Large-Scale Multitarget Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 5–21, 2021.

Part I

A Factor Graph approach to Filtering and Smoothing

One

Marginalized Particle Filtering and Related Filtering Techniques as Message Passing

In this chapter a factor graph approach is employed to investigate the recursive filtering problem for conditionally linear Gaussian state-space models. First, we derive a new factor graph for the considered filtering problem; then, we show that applying the sum-product rule to our graphical model results in both known and novel filtering techniques. In particular, we prove that: a) marginalized particle filtering can be interpreted as a form of forward only message passing over the devised graph; b) novel filtering methods can be easily developed by exploiting the graph structure and/or simplifying probabilistic messages.

Part of this Chapter has been published in [1].

1.1 Introduction

The nonlinear filtering problem consists in inferring the posterior distribution of the hidden state of a nonlinear dynamic system from a set of past and present measurements [2]. It is well known that, if a nonlinear dynamic system can be described by a *state-space model* (SSM), a general sequential procedure, based on the Bayes' rule and known as *Bayesian filtering*, can be easily derived for recursively computing the above mentioned posterior distribution [2]. Unluckily, the general formulas describing the Bayesian filtering recursion admit closed form solutions for *linear Gaussian* and *linear Gaussian mixture* SSMs only [3]. On the contrary, *approximate* solutions are available for general *nonlinear* models; these are based on *sequential Monte Carlo* (SMC) techniques (also known as *particle filtering* methods) which represent a powerful tool for numerical approximations (e.g. see [4]–[5] and references therein). While SMC filtering methods can be directly applied to an arbitrary nonlinear SSM, it has been recognized that their estimation accuracy can be improved in the case of *conditionally linear Gaussian* (CLG) SSMs [6], [7]. In fact, the linear substructure of such models can be marginalised, so reducing the dimension of their sample space [6], [8]. This idea has led to the development of the so called *Rao-Blackwellized particle filtering* (also dubbed *marginalized particle filtering*, MPF) [6], [9] and other filtering methods originating from it [10]–[11]. These methods play a fundamental role, since CLG models are suitable to represent the behavior of dynamic systems in a number of fields, including econometrics [12], positioning and navigation [13], magnetic resonance imaging [14], human motion tracking [15] and acoustic source localization [16].

Recently, it has been shown that the linear/nonlinear filtering problem and the related *linear/nonlinear smoothing problems* can be revisited from a *factor graph* (FG) perspective. More specifically, on the one hand, it has been proved that Kalman filtering and PF can be interpreted as specific instances of the so called *sum-product algorithm* (SPA) [17], [18] over graphical models (see [17], [19] and [20], [21], respectively) and, consequently, as *message passing procedures on* FGs. On the other hand, it has been shown that the FG approach represents a powerful tool to develop new Rao-Blackwellized algorithms for filtering [22] and smoothing [23].

The content of this chapter is based on our recent work [22], and aims at providing a FG

perspective on MPF and related filtering methods. In the technical literature the derivation of MPF always relies on the factorization of a specific posterior pdf (e.g., [6, see eq. (7)]). In this chapter, instead, after developing a novel FG representation of the filtering problem for a CLG SSM, we prove that MPF can be seen as an application of the SPA to this graphical model. This approach not only sheds new light on a known filtering algorithm, but allows to analyse the problems of developing new filtering algorithms based on the same graphical model as MPF from a message passing perspective. In particular, as far as the last issue is concerned, we focus on the problems of: a) developing a new filtering method, called *dual* MPF (DMPF); b) deriving simplified versions of MPF/DMPF. We believe that these contributions can be of great interest for all the readers interested in the applications of FGs to statistical signal processing and, more specifically, in Rao-Blackwellized filtering techniques.

The remaining part of this chapter is organized as follows. A mathematical description of the considered SSMs is illustrated in Section 1.2, whereas a FG-based representation of the related filtering problem is provided in Section 1.3. Then, in Section 1.4, it is shown that applying the SPA and proper message scheduling to a portion of the FG developed for a CLG SSM leads to MPF. The development of the DMPF technique and the problem of simplifying MPF are illustrated in Sections 1.5 and 1.6, respectively. The filtering methods analysed in this chapter are compared, in terms of accuracy and computational effort, in Section 1.7. Finally, some conclusions are offered in Section 1.8.

1.2 Model Description

In this chapter we focus on a discrete-time SSM whose D-dimensional *hidden state* in the k-th interval is denoted $\mathbf{x}_k \triangleq [x_{1,k}, x_{1,k}, ..., x_{D,k}]^T$, and whose *state update* and *measurement models* are expressed by

$$\mathbf{x}_{k+1} = \mathbf{f}_k\big(\mathbf{x}_k\big) + \mathbf{w}_k \tag{1.1}$$

and

$$\begin{aligned}\mathbf{y}_k &\triangleq [y_{1,k}, y_{1,k}, ..., y_{P,k}]^T \\ &= \mathbf{h}_k\big(\mathbf{x}_k\big) + \mathbf{e}_k\end{aligned} \tag{1.2}$$

respectively. Here, $\mathbf{f}_k\big(\mathbf{x}_k\big)$ ($\mathbf{h}_k\big(\mathbf{x}_k\big)$) is a time-varying D-dimensional (P-dimensional) real function and $\mathbf{w}_k$ ($\mathbf{e}_k$) is the k-th element of the process (measurement) noise sequence $\big\{\mathbf{w}_k\big\}$ ($\big\{\mathbf{e}_k\big\}$); this sequence consists of D-dimensional (P-dimensional) *independent and identically distributed* (iid) Gaussian noise vectors, each characterized by a zero mean and a covariance matrix $\mathbf{C}_w$ ($\mathbf{C}_e$). Moreover, statistical independence between $\big\{\mathbf{e}_k\big\}$ and $\{\mathbf{w}_k\}$ is also assumed.

The general models (1.1)-(1.2) can be rewritten in a different way if the considered SSM is CLG [6], [7], [23], [24]. In fact, under this assumption, the state vector in the k-th interval can be partitioned as $\mathbf{x}_k = [(\mathbf{x}_k^{(L)})^T, (\mathbf{x}_k^{(N)})^T]^T$, where, $\mathbf{x}_k^{(L)} \triangleq [x_{1,k}^{(L)}, x_{2,k}^{(L)}, ..., x_{D_L,k}^{(L)}]^T$ ($\mathbf{x}_k^{(N)} \triangleq [x_{1,k}^{(N)}, x_{2,k}^{(N)}, ..., x_{D_N,k}^{(N)}]^T$) is the so called *linear* (*nonlinear*) *component* of $\mathbf{x}_k$, with $D_L < D$ ($D_N = D - D_L$). Moreover, the models[1] (e.g., see [23, Sec. II] and [24, Sec. 2])

$$\mathbf{x}_{k+1}^{(Z)} = \mathbf{A}_k^{(Z)}\big(\mathbf{x}_k^{(N)}\big)\,\mathbf{x}_k^{(L)} + \mathbf{f}_k^{(Z)}\big(\mathbf{x}_k^{(N)}\big) + \mathbf{w}_k^{(Z)} \tag{1.3}$$

and

$$\mathbf{y}_k = \mathbf{g}_k\big(\mathbf{x}_k^{(N)}\big) + \mathbf{B}_k\big(\mathbf{x}_k^{(N)}\big)\,\mathbf{x}_k^{(L)} + \mathbf{e}_k \tag{1.4}$$

can be adopted for the update of the *linear* ($Z = L$) and *nonlinear* ($Z = N$) components, and for the measurement vector, respectively. In the state update model (1.3), $\mathbf{f}_k^{(Z)}(\mathbf{x}_k^{(N)})$ ($\mathbf{A}_k^{(Z)}(\mathbf{x}_k^{(N)})$)

[1] Note that, unlike [6] and [7], the dependence of the noise terms on the nonlinear state component is not accounted for in the following models for simplicity. However, including this dependence does not substantially modify the derivation of the message passing algorithms illustrated in Sections 1.4–1.6, since, as it will become clearer later, this simply implies that the covariance matrices associated with noise terms depend on the particle representing the nonlinear state component.

is a time-varying D_Z-dimensional real function ($D_Z \times D_L$ real matrix) and $\mathbf{w}_k^{(Z)}$ consists of the first D_L (last D_N) elements of $\mathbf{w}_k$ if $Z = L$ ($Z = N$); independence between $\{\mathbf{w}_k^{(L)}\}$ and $\{\mathbf{w}_k^{(N)}\}$ is also assumed for simplicity and the covariance matrix $\mathbf{w}_k^{(L)}$ ($\mathbf{w}_k^{(N)}$) is denoted $\mathbf{C}_w^{(L)}$ ($\mathbf{C}_w^{(N)}$). In the measurement model (1.4), instead, $\mathbf{g}_k(\mathbf{x}_k^{(N)})$ ($\mathbf{B}_k(\mathbf{x}_k^{(N)})$) is a time-varying P-dimensional real function ($P \times D_L$ real matrix).

In the following Section we tackle the so-called *filtering problem*, that concerns the evaluation of the posterior pdf $f(\mathbf{x}_t|\mathbf{y}_{1:t})$ at an instant $t > 1$, given a) the initial pdf $f(\mathbf{x}_1)$ and b) the $t \cdot P$-dimensional *measurement* vector

$$\mathbf{y}_{1:t} = \left[\mathbf{y}_1^T, \mathbf{y}_2^T, ..., \mathbf{y}_t^T\right]^T. \tag{1.5}$$

1.3 Representation of the Filtering Problem via Factor Graphs

From a statistical viewpoint, a complete description of the SSM described by Eqs. (1.1)–(1.2) is provided by the *Markov model* $f(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and the *observation model* $f(\mathbf{y}_k|\mathbf{x}_k)$ for any k. If the pdf $f(\mathbf{x}_1)$ is known, the computation of the posterior (i.e., filtered) pdf $f(\mathbf{x}_t|\mathbf{y}_{1:t})$ for $t \geq 1$ can be accomplished by means of an exact *Bayesian recursive procedure*, consisting of a *measurement update* (MU) step followed by a *time update* (TU) step. Following [17, Sec. II, p. 1297] and [23, Sec. III], the k-th recursion of this procedure (with $k = 1, 2, ..., t$) is formulated with reference to the joint pdf $f(\mathbf{x}_t, \mathbf{y}_{1:t})$ (in place of the associated a posteriori pdf $f(\mathbf{x}_t|\mathbf{y}_{1:t})$), since this leads more easily to its representation as a *message passing algorithm* over a proper Forney-style FG [21]. In practice, in the MU of the k-th recursion, the joint pdf (providing a statistical description of the *forward estimate* of $\mathbf{x}_k$)

$$f\big(\mathbf{x}_k, \mathbf{y}_{1:k}\big) = f\big(\mathbf{x}_k, \mathbf{y}_{1:k-1}\big) f\big(\mathbf{y}_k\big|\mathbf{x}_k\big), \tag{1.6}$$

is computed on the basis of pdf $f(\mathbf{x}_k, \mathbf{y}_{1:k-1})$ (evaluated in the TU of the previous recursion) and the present measurement vector $\mathbf{y}_k$. Then, in the TU of the same recursion, $f(\mathbf{x}_k, \mathbf{y}_{1:k})$ (1.6) is exploited to compute the pdf

$$f\big(\mathbf{x}_{k+1}, \mathbf{y}_{1:k}\big) = \int f\big(\mathbf{x}_{k+1}\big|\mathbf{x}_k\big) f\big(\mathbf{x}_k, \mathbf{y}_{1:k}\big) d\mathbf{x}_k, \tag{1.7}$$

which represents a one-step *forward prediction* about the future state $\mathbf{x}_{k+1}$. Since Eqs. (1.6) and (1.7) involve only *products of pdfs* and a *sum* (i.e., integration) *of products*, their evaluation can be represented as a *forward only* message passing over the cycle free FG shown in Fig. 1.1-a) (all the rules adopted in the development of this and in the following graphical models are illustrated in Appendix A, where the SPA is also described). In fact, if the input message[2] $\vec{m}_{\mathrm{fp}}\big(\mathbf{x}_k\big) = f(\mathbf{x}_k, \mathbf{y}_{1:k-1})$ enters this FG, the message going out of the *equality node* is given by $\vec{m}_{\mathrm{fe}}\big(\mathbf{x}_k\big) = \vec{m}_{\mathrm{fp}}\big(\mathbf{x}_k\big) f\big(\mathbf{y}_k\big|\mathbf{x}_k\big)$, so that $\vec{m}_{\mathrm{fe}}\big(\mathbf{x}_k\big) = f(\mathbf{x}_k, \mathbf{y}_{1:k})$ (see Eq. (1.6)); then, the message emerging from the *function node* referring to the pdf $f(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is expressed by

$$\vec{m}_{\mathrm{fp}}\big(\mathbf{x}_{k+1}\big) = \int f\big(\mathbf{x}_{k+1}\big|\mathbf{x}_k\big) \vec{m}_{\mathrm{fe}}\big(\mathbf{x}_k\big) d\mathbf{x}_k, \tag{1.8}$$

so that $\vec{m}_{\mathrm{fp}}\big(\mathbf{x}_{k+1}\big) = f(\mathbf{x}_{k+1}, \mathbf{y}_{1:k})$ (see Eq. (1.7)).

In [23, Sec. III] it has been shown that the FG shown in Fig. 1.1-a) can be used to devise a new graphical model for the CLG SSM described by Eqs. (1.3)–(1.4). The development of this model can be summarized as follows. If the nonlinear portion $\mathbf{x}_k^{(N)}$ is *known* for any k, the computation of the filtered pdf $f(\mathbf{x}_k^{(L)}|\mathbf{y}_{1:k})$ for the *linear* state component can benefit not only from the knowledge of $\mathbf{y}_k$ (1.4), but also from that of the quantity (see Eq. (1.3) with $Z = N$)

$$\mathbf{z}_k^{(L)} \triangleq \mathbf{x}_{k+1}^{(N)} - \mathbf{f}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big) = \mathbf{A}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big)\mathbf{x}_k^{(L)} + \mathbf{w}_k^{(N)}, \tag{1.9}$$

[2] In the following the acronyms *fp* and *fe* are employed in the subscripts of various messages, so that readers can easily understand their meaning; in fact, the messages these acronyms refer to represent a form of *forward prediction* and *forward estimation*, respectively.

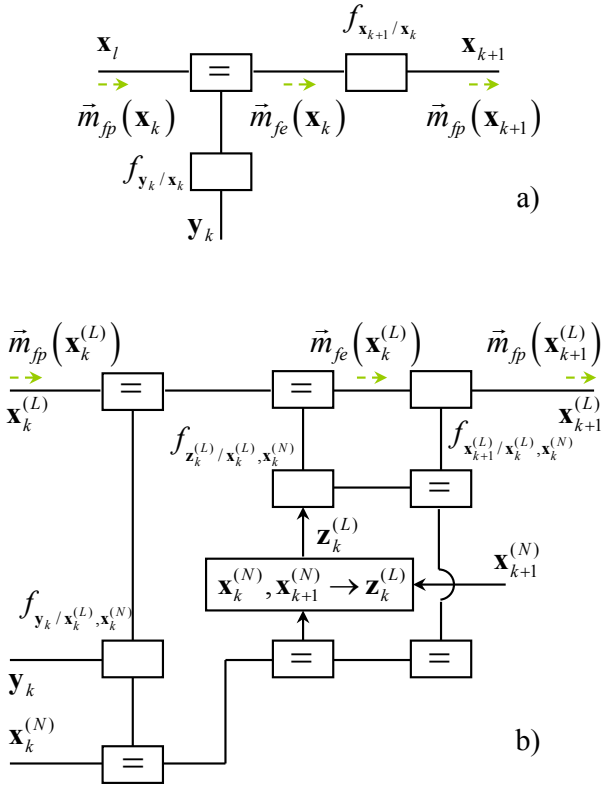


Figure 1.1: Factor graphs representing: a) the k-th recursion of Bayesian filtering for an SSM described by the *Markov model* $f(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and the *observation model* $f(\mathbf{y}_k|\mathbf{x}_k)$; b) the k-th recursion of Bayesian filtering for the linear state component of a CLG SSM. In both graphical models, the SPA message flow is indicated by green arrows.

which can be interpreted as a *pseudo-measurement* (PM) [6], since it does not originate from real measurements, but from the constraints expressed by the state equation (1.3). Then, the k-th recursion of Bayesian filtering for the linear state component, given $\mathbf{x}_k^{(N)}$ and $\mathbf{x}_{k+1}^{(N)}$, can be represented through a graphical model similar to that shown in Fig. 1.1-a). In fact, if the state $\mathbf{x}_k$, the Markov model $f\big(\mathbf{x}_{k+1}\big|\mathbf{x}_k\big)$ and the measurement model $f\big(\mathbf{y}_k\big|\mathbf{x}_k\big)$ appearing in that FG are replaced by their counterparts $\mathbf{x}_k^{(L)}$, $f(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(L)},\mathbf{x}_k^{(N)})$ and $f(\mathbf{y}_k|\mathbf{x}_k^{(L)},\mathbf{x}_k^{(N)})$, respectively, and the contribution due to the PM model $f(\mathbf{z}_k^{(L)}|\mathbf{x}_k^{(L)},\mathbf{x}_k^{(N)})$ is kept into account, the FG shown in Fig. 1.1-b) is obtained. It is important to point out that: 1) the new graph contains a node which does not refer to a density factorization (this peculiarity is also evidenced by the presence of an arrow on all the edges connected to such a node), but represents the transformation from the couple $(\mathbf{x}_k^{(N)},\mathbf{x}_{k+1}^{(N)})$ to $\mathbf{z}_k^{(L)}$ (see Eq. (1.9)); 2) the input and output messages appearing in this FG (denoted $\vec{m}_{\mathrm{fp}}(\mathbf{x}_k^{(L)})$ and $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(L)})$, respectively) and the forward estimate message $\vec{m}_{\mathrm{fe}}(\mathbf{x}_k^{(L)})$ refer to the linear state component only.

Actually, following the line of reasoning illustrated above for the linear state component, a *dual* graphical model can be devised to represent the evaluation of the filtered pdf $f(\mathbf{x}_k^{(N)}|\mathbf{y}_{1:k})$ for the *nonlinear* state component, provided that the linear component $\mathbf{x}_k^{(L)}$ is *known* for any k. In fact, in this case, the PM (see Eq. (1.3) with $Z = L$)

$$\mathbf{z}_k^{(N)} \triangleq \mathbf{x}_{k+1}^{(L)} - \mathbf{A}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)\,\mathbf{x}_k^{(L)} = \mathbf{f}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big) + \mathbf{w}_k^{(L)}, \tag{1.10}$$

described by the pdf $f(\mathbf{z}_k^{(N)}|\mathbf{x}_k^{(N)})$, becomes available. Consequently, a FG similar to the one shown in Fig. 1.1-b) can be obtained by simply replacing $\mathbf{x}_k^{(L)}$, $\mathbf{x}_{k+1}^{(L)}$, $f(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(L)},\mathbf{x}_k^{(N)})$ and $f(\mathbf{z}_k^{(L)}|\mathbf{x}_k^{(L)},\mathbf{x}_k^{(N)})$ with $\mathbf{x}_k^{(N)}$, $\mathbf{x}_{k+1}^{(N)}$, $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)})$ and $f(\mathbf{z}_k^{(N)}|\mathbf{x}_k^{(L)})$, respectively. Then, the

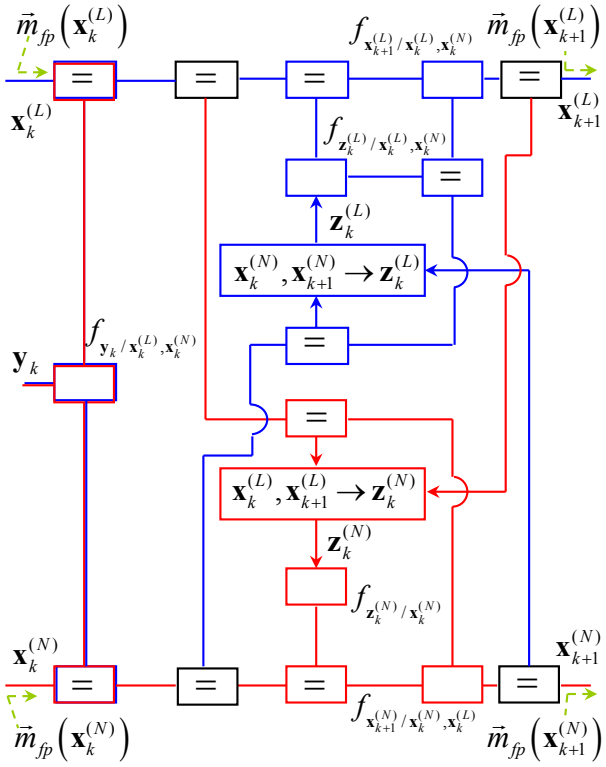


Figure 1.2: Overall factor graph resulting from the merge of two sub-graphs, one referring to filtering for $\mathbf{x}_k^{(L)}$ (in blue), the other one to that for $\mathbf{x}_k^{(N)}$ (in red). The equality constraint nodes introduced to connect these subgraphs are identified by black lines. The flow of the messages along the half edges $\mathbf{x}_k^{(L)}$ and $\mathbf{x}_k^{(N)}$ (input) and that of the messages along the half edges $\mathbf{x}_{k+1}^{(L)}$ and $\mathbf{x}_{k+1}^{(N)}$ (output) are indicated by green arrows.

FGs developed for $\mathbf{x}_k^{(L)}$ and $\mathbf{x}_k^{(N)}$ can be *merged* by simply adding some equality constraint nodes for the shared variables (namely, $\mathbf{x}_k^{(L)}$, $\mathbf{x}_k^{(N)}$, $\mathbf{x}_{k+1}^{(L)}$ and $\mathbf{x}_{k+1}^{(N)}$), as shown in Fig. 1.2. This merge, proposed for the first time in [22], can be intepreted as an instance of the more general concept of *concatenation* (and, more specifically, of *parallel concatenation* [25]) of graphical models. Note that this concept has been widely exploited in the field of channel coding and channel estimation (e.g., see [18], [26], [27] and references therein). However, as far as we know, the graphical model shown in Fig. 1.2 represents its first application to the field of filtering techniques; moreover, as shown in the next Sections, it can provide new insights into Bayesian filtering for CLG SSMs. Unluckily, the new FG, unlike the FGs represented in Fig. 1.1, is *not cycle-free*; this property can be related to the fact that, generally speaking, filtering for $\mathbf{x}_k^{(N)}$ is *not decouplable* from that for $\mathbf{x}_k^{(L)}$. Given the FG of Fig. 1.2 and its input messages $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(L)}) = f(\mathbf{x}_k^{(L)}, \mathbf{y}_{1:k-1})$ and $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(N)}) = f(\mathbf{x}_k^{(N)}, \mathbf{y}_{1:k-1})$ (entering the FG along the half edges associated with $\mathbf{x}_k^{(L)}$ and $\mathbf{x}_k^{(N)}$, respectively), we would like to derive a *forward only* message passing algorithm similar to the one illustrated in Fig. 1.1-a) and generating the output messages $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1}^{(L)}) = f(\mathbf{x}_{k+1}^{(L)}, \mathbf{y}_{1:k})$ and $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1}^{(N)}) = f(\mathbf{x}_{k+1}^{(N)}, \mathbf{y}_{1:k})$ (emerging from the FG along the half edges associated with $\mathbf{x}_{k+1}^{(L)}$ and $\mathbf{x}_{k+1}^{(N)}$, respectively). The computation of $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1}^{(L)})$ and $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1}^{(N)})$ on the basis of $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(L)})$ and $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(N)})$ and of the pdfs appearing in Fig. 1.2 requires *marginalization* with respect to $\mathbf{x}_k$. This can still be done, even if in an approximate fashion, by applying the SPA to the FG of Fig. 1.2, provided that a proper *scheduling strategy* is adopted in passing the involved probabilistic messages along the considered FG [18]. In the following Section, we show that following this approach leads to MPF.

1.4 Message Passing in Marginalized Particle Filtering

In this Section we prove that the MPF technique can be interpreted as a *forward only* message passing algorithm over the FG shown in Fig. 1.2 and that the equations describing its k-th recursion result from the application of the SPA to that FG. To begin, we note that MPF processing does not involve the full structure of our graphical model, since it does not exploit the PM $\mathbf{z}_k^{(N)}$ (1.10). For this reason, in the following we refer to the simplified FG shown in Fig. 1.3, which has been obtained from the one illustrated in Fig. 1.2 by removing the block representing the transformation from $(\mathbf{x}_k^{(L)}, \mathbf{x}_{k+1}^{(L)})$ to $\mathbf{z}_k^{(N)}$ and all the edges referring to the evaluation of the last vector. As far as the input and the output messages of this FG are concerned, we assume that the a priori information available about $\mathbf{x}_k^{(N)}$ at the beginning of the k-th recursion are represented by the set $S_{\text{fp},k}^{(N)} = \{\mathbf{x}_{\text{fp},k,j}^{(N)},\, j = 1, 2, ..., N_p\}$, collecting N_p *predicted particles*, and by their (uniform) weights $\{w_{\text{fp},k,j} = 1/N_p,\, j = 1, 2..., N_p\}$. Therefore, at the beginning of the considered recursion, our knowledge about $\mathbf{x}_k^{(N)}$ is condensed in the particle-dependent message

$$\vec{m}_{\text{fp},j}\big(\mathbf{x}_k^{(N)}\big) = \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{\text{fp},k,j}^{(N)}\big), \tag{1.11}$$

with $j = 1, 2, ..., N_p$. The a priori information available about $\mathbf{x}_k^{(L)}$, instead, is represented by a set of N_p Gaussian pdfs; the j-th Gaussian pdf is conveyed by the particle-dependent message

$$\vec{m}_{\text{fp},j}\big(\mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{x}_k^{(L)}; \eta_{\text{fp},k,j}^{(L)}, \mathbf{C}_{\text{fp},k,j}^{(L)}\big), \tag{1.12}$$

providing a statistical description of $\mathbf{x}_k^{(L)}$ conditioned on $\mathbf{x}_k^{(N)} = \mathbf{x}_{\text{fp},k,j}^{(N)}$. From the statistical representation of the state components illustrated above it can be easily inferred that, in developing a message passing algorithm over the considered FG, we can focus on: a) a *single particle* and, in particular, on the j-th particle $\mathbf{x}_{\text{fp},k,j}^{(N)}$; b) on the Gaussian pdf $\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(L)})$ associated with that particle. Moreover, this algorithm must generate the output messages $\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(N)})$ and $\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(L)})$, which are required to have the *same functional form* as $\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(N)})$ (1.11) and $\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(L)})$ (1.12), respectively. As far as message scheduling is concerned, our choice is based on the description of the MPF technique summarized in algorithm 1 of [6, Sec. II] and is represented in Fig. 1.3, where: a) Gaussian messages (non Gaussian messages) are associated with blue (red) arrows; b) the notation nL (nL'), nN ($n\,N'$), FPL (FPL'), FPN (FPN') and ZL is employed to represent the messages $\vec{m}_{n,j}(\mathbf{x}_k^{(L)})$ ($\vec{m}_{n,j}(\mathbf{x}_{k+1}^{(L)})$), $\vec{m}_{n,j}(\mathbf{x}_k^{(N)})$ ($\vec{m}_{n,j}(\mathbf{x}_{k+1}^{(N)})$), $\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(L)})$ ($\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(L)})$), $\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(N)})$ ($\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(N)})$) and $\vec{m}_{z,j}(\mathbf{z}_k^{(L)})$, respectively, passed along the considered FG. In practice, the computation of the passed messages is accomplished in five consecutive steps, according to the following order: 1) $\vec{m}_{1,j}(\mathbf{x}_k^{(N)})$, $\vec{m}_{2,j}(\mathbf{x}_k^{(N)})$; 2) $\vec{m}_{1,j}(\mathbf{x}_k^{(L)})$, $\vec{m}_{2,j}(\mathbf{x}_k^{(L)})$; 3) $\vec{m}_{3,j}(\mathbf{x}_{k+1}^{(N)})$, $\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(N)})$; 4) $\vec{m}_j(\mathbf{z}_k^{(L)})$, $\vec{m}_{3,j}(\mathbf{x}_k^{(L)})$, $\vec{m}_{4,j}(\mathbf{x}_k^{(L)})$; 5) $\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(L)})$. In the following, we illustrate the aim of each step and provide the expressions of the evaluated messages (additional mathematical details can be found in Appendix B).

1) MU for $\mathbf{x}_k^{(N)}$ - In this step, the weight of the j-th particle $\mathbf{x}_{\text{fp},k,j}^{(N)}$ is updated on the basis of $\mathbf{y}_k$. This requires computing

$$\vec{m}_{1,j}\big(\mathbf{x}_k^{(N)}\big) = \mathcal{N}\big(\mathbf{y}_k; \eta_{1,k,j}^{(N)}\big(\mathbf{x}_k^{(N)}\big), \mathbf{C}_{1,k,j}^{(N)}\big(\mathbf{x}_k^{(N)}\big)\big), \tag{1.13}$$

where $\eta_{1,k,j}^{(N)}(\mathbf{x}_k^{(N)}) \triangleq \mathbf{B}_k(\mathbf{x}_k^{(N)}), \eta_{\text{fp},k,j}^{(L)} + \mathbf{h}_k(\mathbf{x}_k^{(N)})$ and $\mathbf{C}_{1,k,j}^{(N)}(\mathbf{x}_k^{(N)}) \triangleq \mathbf{B}_k(\mathbf{x}_k^{(N)})\, \mathbf{C}_{\text{fp},k,j}^{(L)}\, \mathbf{B}_k^T(\mathbf{x}_k^{(N)}) + \mathbf{C}_e$ and

$$\vec{m}_{2,j}\big(\mathbf{x}_k^{(N)}\big) = \vec{m}_{\text{fp},j}\big(\mathbf{x}_k^{(N)}\big)\, \vec{m}_{1,j}\big(\mathbf{x}_k^{(N)}\big) \tag{1.14}$$

$$= w_{\text{fe},k,j}\, \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{\text{fp},k,j}^{(N)}\big); \tag{1.15}$$

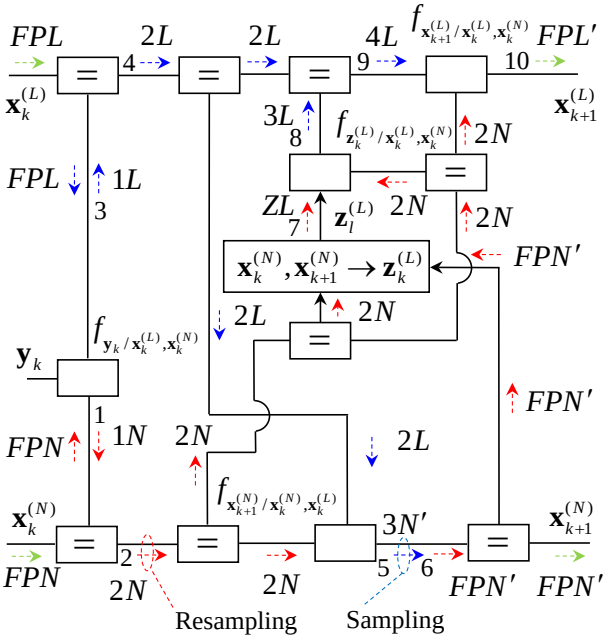


Figure 1.3: Overall factor graph for the representation of the k-th recursion of MPF. The message flow referring to the j-th particle is shown. Gaussian messages (non Gaussian) messages computed in the considered recursion are associated with blue (red) arrows, whereas input and output messages with green arrows; the integers $1-10$ are used to specify the adopted message scheduling.

in the last equation, the quantity[3]

$$w_{\mathrm{fe},k,j} \triangleq \mathcal{N}\big(\mathbf{y}_k; \eta^{(N)}_{1,k,j}, \mathbf{C}^{(N)}_{1,k,j}\big) \tag{1.16}$$

represents the new particle weight combining the a priori information about $\mathbf{x}^{(N)}_k$ with the information provided by the new measurement; moreover,

$$\eta^{(N)}_{1,k,j} \triangleq \eta^{(N)}_{1,k,j}\big(\mathbf{x}^{(N)}_{\mathrm{fp},k,j}\big) = \mathbf{B}_{k,j}\, \eta^{(L)}_{\mathrm{fp},k,j} + \mathbf{h}_{k,j} \tag{1.17}$$

and

$$\mathbf{C}^{(N)}_{1,k,j} \triangleq \mathbf{C}^{(N)}_{1,k,j}\big(\mathbf{x}^{(N)}_{\mathrm{fp},k,j}\big) = \mathbf{B}_{k,j}\, \mathbf{C}^{(L)}_{\mathrm{fp},k,j}\, \mathbf{B}^{T}_{k,j} + \mathbf{C}_e, \tag{1.18}$$

with $\mathbf{h}_{k,j} \triangleq \mathbf{h}_k(\mathbf{x}^{(N)}_{\mathrm{fp},k,j})$ and $\mathbf{B}_{k,j} \triangleq \mathbf{B}_k(\mathbf{x}^{(N)}_{\mathrm{fp},k,j})$. The N_p particle weights $\{w_{\mathrm{fe},k,j}\}$ undergo normalization; this produces the new weight

$$W_{\mathrm{fe},k,j} \triangleq w_{\mathrm{fe},k,j}/\Lambda^{(w)}_k \tag{1.19}$$

with $j = 1, 2, ..., N_p$, where $\Lambda^{(w)}_k \triangleq \sum_{j=1}^{N_p} w_{\mathrm{fe},k,j}$. Then, particle *resampling* with replacement is accomplished over the set $S^{(N)}_{\mathrm{fp},k}$ on the basis of the weights $\{W_{\mathrm{fe},k,j}\}$. Note that, even if resampling does not emerge from the application of SPA to the considered graph, its use, as it will become clearer at the end of this Section, plays an important role in the generation of the new particles for $\mathbf{x}^{(N)}_{k+1}$. Moreover, it can be easily incorporated in our message passing; in fact, resampling simply entails that the N_p particles $\{\mathbf{x}^{(N)}_{\mathrm{fp},k,j}\}$ and their associated weights $\{W_{\mathrm{fe},k,j}\}$ are replaced by the new particles $\{\mathbf{x}^{(N)}_{\mathrm{fe},k,j}\}$ (forming the new set $S^{(N)}_{\mathrm{fe},k}$) and their uniform weights $\{\tilde{W}_{\mathrm{fe},k,j} = 1/N_p\}$,

[3] In evaluating this weight, the factor $[\det(\mathbf{C}^{(N)}_{1,k,j})]^{-P/2}$ appearing in the expression of the involved Gaussian pdf is usually neglected, since this entails a negligible loss in estimation accuracy.

respectively. Consequently, after resampling, $\vec{m}_2(\mathbf{x}_k^{(N)})$ (1.15) is replaced by

$$\vec{m}_{2,j}\big(\mathbf{x}_k^{(N)}\big) = \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{\mathrm{fe},k,j}^{(N)}\big). \tag{1.20}$$

and the set of Gaussian messages $\{\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(L)})\}$ (see (1.12)) is properly reordered; this ensures that, after reordering, the message $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(L)})$ is associated with $\mathbf{x}_{\mathrm{fe},k,j}^{(N)}$, and that the messages belonging to that set and referring to all the discarded particles are not propagated to the next steps.

2) First MU for $\mathbf{x}_k^{(L)}$ - In this step our statistical knowlege about $\mathbf{x}_k^{(L)}$ is updated on the basis of $\mathbf{y}_k$; this requires the computation of the messages

$$\vec{m}_{1,j}\big(\mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{x}_k^{(L)}; \eta_{1,k,j}^{(L)}, \mathbf{C}_{1,k,j}^{(L)}\big) \tag{1.21}$$

and

$$\vec{m}_{2,j}\big(\mathbf{x}_k^{(L)}\big) = \vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(L)})\, \vec{m}_{1,j}(\mathbf{x}_k^{(L)}) \tag{1.22}$$

$$= \mathcal{N}\big(\mathbf{x}_k^{(L)}; \eta_{2,k,j}^{(L)}, \mathbf{C}_{2,k,j}^{(L)}\big). \tag{1.23}$$

The covariance matrix $\mathbf{C}_{p,k,j}^{(L)}$ and the mean vector $\eta_{p,k,j}^{(L)}$ (with $p = 1$ and 2) are evaluated on the basis of the associated precision matrices

$$\mathbf{W}_{1,k,j}^{(L)} \triangleq \big(\mathbf{C}_{1,k,j}^{(L)}\big)^{-1} = \mathbf{B}_{k,j}^T\, \mathbf{W}_e\, \mathbf{B}_{k,j} \tag{1.24}$$

and

$$\mathbf{W}_{2,k,j}^{(L)} \triangleq \big(\mathbf{C}_{2,k,j}^{(L)}\big)^{-1} = \mathbf{W}_{\mathrm{fp},k,j}^{(L)} + \mathbf{W}_{1,k,j}^{(L)}, \tag{1.25}$$

and of the associated transformed mean vectors

$$\mathbf{w}_{1,k,j}^{(L)} \triangleq \mathbf{W}_{1,k,j}^{(L)}\, \eta_{1,k,j}^{(L)} = \mathbf{B}_{k,j}^T\, \mathbf{W}_e\, \big(\mathbf{y}_k - \mathbf{h}_{k,j}\big) \tag{1.26}$$

and

$$\mathbf{w}_{2,k,j}^{(L)} \triangleq \mathbf{W}_{2,k,j}^{(L)}\, \eta_{2,k,j}^{(L)} = \mathbf{w}_{\mathrm{fp},k,j}^{(L)} + \mathbf{w}_{1,k,j}^{(L)}, \tag{1.27}$$

respectively; here, $\mathbf{W}_e \triangleq \mathbf{C}_e^{-1}$, $\mathbf{W}_{\mathrm{fp},k,j}^{(L)} \triangleq (\mathbf{C}_{\mathrm{fp},k,j}^{(L)})^{-1}$ and $\mathbf{w}_{\mathrm{fp},k,j}^{(L)} \triangleq \mathbf{W}_{\mathrm{fp},k,j}^{(L)} \eta_{\mathrm{fp},k,j}^{(L)}$.

3) TU for $\mathbf{x}_k^{(N)}$ - This step aims at generating the j-th particle for $\mathbf{x}_{k+1}^{(N)}$ and its associated weight; both these information are conveyed by the message (see Fig. 1.3)

$$\vec{m}_{3,j}\big(\mathbf{x}_{k+1}^{(N)}\big) = \int\int f\big(\mathbf{x}_{k+1}^{(N)} | \mathbf{x}_k^{(L)}, \mathbf{x}_k^{(N)}\big) \cdot \vec{m}_{2,j}\big(\mathbf{x}_k^{(L)}\big) \vec{m}_{2,j}\big(\mathbf{x}_k^{(N)}\big) d\mathbf{x}_k^{(L)} d\mathbf{x}_k^{(N)} \tag{1.28}$$

$$= \mathcal{N}\big(\mathbf{x}_{k+1}^{(N)}; \eta_{3,k,j}^{(N)}, \mathbf{C}_{3,k,j}^{(N)}\big), \tag{1.29}$$

where

$$\eta_{3,k,j}^{(N)} \triangleq \mathbf{A}_{k,j}^{(N)}\, \eta_{2,k,j}^{(L)} + \mathbf{f}_{k,j}^{(N)}, \tag{1.30}$$

$$\mathbf{C}_{3,k,j}^{(N)} \triangleq \mathbf{C}_w^{(N)} + \mathbf{A}_{k,j}^{(N)}\, \mathbf{C}_{2,k,j}^{(L)} \big(\mathbf{A}_{k,j}^{(N)}\big)^T, \tag{1.31}$$

$\mathbf{A}_{k,j}^{(N)} \triangleq \mathbf{A}_k^{(N)}(\mathbf{x}_{\mathrm{fe},k,j}^{(N)})$ and $\mathbf{f}_{k,j}^{(N)} \triangleq \mathbf{f}_k^{(N)}(\mathbf{x}_{\mathrm{fe},k,j}^{(N)})$. Note that, in principle, we should set $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_{k+1}^{(N)}) = \vec{m}_{3,j}(\mathbf{x}_{k+1}^{(N)})$ (see Fig. 1.3). However, as already mentioned above, the output message $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_{k+1}^{(N)})$ is required to have the same functional form as $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(N)})$ (1.11). This result can be achieved by a) *sampling*[4] the Gaussian function $\mathcal{N}(\mathbf{x}_{k+1}^{(N)}; \eta_{3,k,j}^{(N)}, \mathbf{C}_{3,k,j}^{(N)})$ (see Eq. (1.29)), that is drawing a sample $\mathbf{x}_{\mathrm{fp},k+1,j}^{(N)}$ from it and b) assigning to the new particle $\mathbf{x}_{\mathrm{fp},k+1,j}^{(N)}$ a probability $w_{\mathrm{fp},k+1,j}$ equal

[4] Artificial noise can be introduced in the generation of this particle in order to mitigate the so called *degeneracy problem* [2, 28].

to the weight $\tilde{W}_{\mathrm{fe},k,j} = 1/N_p$ (originating from resampling). Note that repeating this procedure for any j generates the new set $S^{(N)}_{\mathrm{fp},k+1} = \{\mathbf{x}^{(N)}_{\mathrm{fp},k+1,j}\}$; the j-th particle of this set is conveyed by the message

$$\vec{m}_{\mathrm{fp},j}\big(\mathbf{x}^{(N)}_{k+1}\big) = \delta\big(\mathbf{x}^{(N)}_{k+1} - \mathbf{x}^{(N)}_{\mathrm{fp},k+1,j}\big). \tag{1.32}$$

This message is also used in the TU for $\mathbf{x}^{(L)}_k$, as illustrated in the next step.

4) Second MU for $\mathbf{x}^{(L)}_k$ - This step aims at accomplishing an additional MU for $\mathbf{x}^{(L)}_k$ (on the basis of the PM $\mathbf{z}^{(L)}_k$ (1.9)) and, then, at generating the output message $\vec{m}_{\mathrm{fp},j}(\mathbf{x}^{(L)}_{k+1})$. In this step, the messages

$$\vec{m}_j\big(\mathbf{z}^{(L)}_k\big) = f\big(\mathbf{z}^{(L)}_k\big|\mathbf{x}^{(N)}_{\mathrm{fe},k,j}, \mathbf{x}^{(N)}_{\mathrm{fp},k+1,j}\big) = \delta\big(\mathbf{z}^{(L)}_k - \mathbf{z}^{(L)}_{k,j}\big), \tag{1.33}$$

with $\mathbf{z}^{(L)}_{k,j} \triangleq \mathbf{x}^{(N)}_{\mathrm{fp},k+1,j} - \mathbf{f}^{(N)}_{k,j}$,

$$\vec{m}_{3,j}\big(\mathbf{x}^{(L)}_k\big) = \mathcal{N}\big(\mathbf{x}^{(L)}_k; \eta^{(L)}_{3,k,j}, \mathbf{C}^{(L)}_{3,k,j}\big), \tag{1.34}$$

and

$$\vec{m}_{4,j}\big(\mathbf{x}^{(L)}_k\big) = \vec{m}_{2,j}(\mathbf{x}^{(L)}_k)\vec{m}_{3,j}(\mathbf{x}^{(L)}_k) \tag{1.35}$$

$$= \mathcal{N}\big(\mathbf{x}^{(L)}_k; \eta^{(L)}_{4,k,j}, \mathbf{C}^{(L)}_{4,k,j}\big), \tag{1.36}$$

are generated. The covariance matrix $\mathbf{C}^{(L)}_{p,k,j}$ and the mean vector $\eta^{(L)}_{p,k,j}$ (with $p = 3$ and 4) are evaluated on the basis of the associated precision matrices $\mathbf{W}^{(L)}_{3,k,j} = (\mathbf{A}^{(N)}_{k,j})^T\mathbf{W}^{(N)}_w\mathbf{A}^{(N)}_{k,j}$ and

$$\mathbf{W}^{(L)}_{4,k,j} \triangleq \big(\mathbf{C}^{(L)}_{4,k,j}\big)^{-1} = \mathbf{W}^{(L)}_{2,k,j} + \big(\mathbf{A}^{(N)}_{k,j}\big)^T\,\mathbf{W}^{(N)}_w\mathbf{A}^{(N)}_{k,j}, \tag{1.37}$$

and the associated transformed mean vectors $\mathbf{w}^{(L)}_{3,k,j} = (\mathbf{A}^{(N)}_{k,j})^T\,\mathbf{W}^{(N)}_w\mathbf{z}^{(L)}_{k,j}$ and

$$\mathbf{w}^{(L)}_{4,k,j} \triangleq \mathbf{W}^{(L)}_{4,k,j}\,\eta^{(L)}_{4,k,j} = \mathbf{w}^{(L)}_{2,k,j} + \big(\mathbf{A}^{(N)}_{k,j}\big)^T\,\mathbf{W}^{(N)}_w\mathbf{z}^{(L)}_{k,j}, \tag{1.38}$$

respectively; here, $\mathbf{W}^{(N)}_w \triangleq [\mathbf{C}^{(N)}_w]^{-1}$.

5) TU for $\mathbf{x}^{(L)}_k$ - In this step, the message

$$\vec{m}_{\mathrm{fp},j}\big(\mathbf{x}^{(L)}_{k+1}\big) = \int\int f\big(\mathbf{x}^{(L)}_{k+1}\big|\mathbf{x}^{(L)}_k, \mathbf{x}^{(N)}_k\big)\cdot\vec{m}_{2,j}(\mathbf{x}^{(N)}_k)\,\vec{m}_{4,j}\big(\mathbf{x}^{(L)}_k\big)\,d\mathbf{x}^{(L)}_k d\mathbf{x}^{(N)}_k \tag{1.39}$$

$$= \mathcal{N}\big(\mathbf{x}^{(L)}_{k+1}; \eta^{(L)}_{\mathrm{fp},k+1,j}, \mathbf{C}^{(L)}_{\mathrm{fp},k+1,j}\big), \tag{1.40}$$

is computed; here,

$$\eta^{(L)}_{\mathrm{fp},k+1,j} \triangleq \mathbf{A}^{(L)}_{k,j}\eta^{(L)}_{4,k,j} + \mathbf{f}^{(L)}_{k,j}, \tag{1.41}$$

$$\mathbf{C}^{(L)}_{\mathrm{fp},k+1,j} \triangleq \mathbf{C}^{(L)}_w + \mathbf{A}^{(L)}_{k,j}\,\mathbf{C}^{(L)}_{4,k,j}\,\big(\mathbf{A}^{(L)}_{k,j}\big)^T, \tag{1.42}$$

$\mathbf{f}^{(L)}_{k,j} \triangleq \mathbf{f}^{(L)}_k(\mathbf{x}^{(N)}_{\mathrm{fe},k,j})$ and $\mathbf{A}^{(L)}_{k,j} \triangleq \mathbf{A}^{(L)}_k(\mathbf{x}^{(N)}_{\mathrm{fe},k,j})$. Since the message $\vec{m}_{\mathrm{fp},j}(\mathbf{x}^{(L)}_{k+1})$ (1.40) is Gaussian and the weight associated with it is $\tilde{W}_{\mathrm{fe},k,j} = 1/N_p$, the statistical representation generated by the SPA for $\mathbf{x}^{(L)}_{k+1}$ is a N_p-component *Gaussian mixture* (GM); note that all its components have the same weight, since resampling is always used in step 1). In fact, if resampling was not accomplished in the k-th recursion, the weight of the j-th component of this GM would be proportional to $W_{\mathrm{fe},k,j}$ (1.16) (i.e., to the weight assigned to the j-th particle *before resampling*); this would unavoidably raise the problem of sampling a GM with unequally weighted components in generating the particle set $S^{(N)}_{\mathrm{fp},k+1}$ and that of properly handling the resulting PMs $\{\mathbf{z}^{(L)}_{k,j}\}$. These considerations motivate the use of resampling *in each recursion*, independently of the *effective sample size* [2] characterizing the particle set $S^{(N)}_{\mathrm{fp},k}$.

Step 5) concludes the message passing accomplished within the k-th recursion. Note that this procedure needs a proper *initialization*. In practice, before starting the first recursion (corresponding to $k = 1$), the set $S_{\mathrm{fp},1}^{(N)}$ is generated for $\mathbf{x}_1^{(N)}$ sampling the pdf $f(\mathbf{x}_1^{(N)}) = \int f(\mathbf{x}_1) d\mathbf{x}_1^{(L)}$ and the same weight $w_{\mathrm{fp},1} = 1/N_p$ and Gaussian model $\mathcal{N}(\mathbf{x}_1^{(L)}; \eta_{\mathrm{fp},1}^{(L)}, \mathbf{C}_{\mathrm{fp},1}^{(L)})$ for $\mathbf{x}_1^{(L)}$ are assigned to each particle.

The processing tasks accomplished in the message passing procedure derived above are summarized in Algorithm 1 (where T denotes the overall duration of the observation interval). Note that our FG-based formulation of MPF, unlike the one appearing in [6, Par. II-D], is mainly expressed in terms of precision matrices and transformed mean vectors; we believe that this makes it more compact and easier to interpret.

Algorithm 1: Marginalized particle filtering

1 **Initialisation:** For $j = 1$ to N_p: sample the pdf $f(\mathbf{x}_1^{(N)})$ to generate the particle $\mathbf{x}_{\mathrm{fp},1,j}^{(N)}$, and assign the weight $w_{\mathrm{fp},1} = 1/N_p$ and the Gaussian $\mathcal{N}(\mathbf{x}_1^{(L)}; \eta_{\mathrm{fp},1}^{(L)}, \mathbf{C}_{\mathrm{fp},1}^{(L)})$ to it (the generated particles are collected in the set $S_{\mathrm{fp},1}^{(N)}$).

2 **Filtering:** For $k = 1$ to T:

a- MU for $\mathbf{x}_k^{(N)}$: For $j = 1$ to N_p: compute $\eta_{1,k,j}^{(N)}$ (1.17) and $\mathbf{C}_{1,k,j}^{(N)}$ (1.18), and the weight $w_{\mathrm{fe},k,j}$ (1.16).

b- *Normalization of particle weights*: For $j = 1$ to N_p: compute the normalised particle weights $\{W_{\mathrm{fe},k,j}\}$ according to (1.19).

c- *Resampling with replacement*: For $j = 1$ to N_p: generate the new particle $\mathbf{x}_{\mathrm{fe},k,j}^{(N)}$ by resampling with replacement over the particle set $S_{\mathrm{fp},k}^{(N)}$ and assign the new weight $\tilde{W}_{\mathrm{fe},k,j} = 1/N_p$ to it. Then, reorder the associated set of Gaussian models for the linear state component accordingly.

d- First MU for $\mathbf{x}_k^{(L)}$: For $j = 1$ to N_p: compute $\mathbf{W}_{1,k,j}^{(L)}$ (1.24) and $\mathbf{w}_{1,k,j}^{(L)}$ (1.26) ; then, compute $\mathbf{W}_{2,k,j}^{(L)}$ (1.25), $\mathbf{w}_{2,k,j}^{(L)}$ (1.27), $\mathbf{C}_{2,k,j}^{(L)} = [\mathbf{W}_{2,k,j}^{(L)}]^{-1}$ and $\eta_{2,k,j}^{(L)} = \mathbf{C}_{2,k,j}^{(L)} \mathbf{w}_{2,k,j}^{(L)}$.

e- TU for $\mathbf{x}_k^{(N)}$ - For $j = 1$ to N_p: compute $\eta_{3,k,j}^{(N)}$ (1.30) and $\mathbf{C}_{3,k,j}^{(N)}$ (1.31); then, sample the pdf $\mathcal{N}(\mathbf{x}_k^{(N)}; \eta_{3,k,j}^{(N)}, \mathbf{C}_{3,l,j}^{(N)})$ to generate the new particle $\mathbf{x}_{\mathrm{fp},k+1,j}^{(N)}$ and assign the weight $w_{\mathrm{fp},k+1,j} = 1/N_p$ to it.

f- Second MU for $\mathbf{x}_k^{(L)}$ - For $j = 1$ to N_p: compute $\mathbf{z}_{k,j}^{(L)}$ (1.33), $\mathbf{W}_{4,k,j}^{(L)}$ (1.37) and $\mathbf{w}_{4,k,j}^{(L)}$ (1.38) ;

g- TU for $\mathbf{x}_k^{(L)}$ - For $j = 1$ to N_p: compute $\eta_{\mathrm{fp},k+1,j}^{(L)}$ (1.41) and $\mathbf{C}_{\mathrm{fp},k+1,j}^{(L)}$ (1.42).

Finally, it is worth mentioning that: 1) the *forward estimate* of $\mathbf{x}_k$ is expressed by the pdf

$$\tilde{f}(\mathbf{x}_k|\mathbf{y}_{1:k}) \triangleq \sum_{j=1}^{N_p} \tilde{W}_{\mathrm{fe},k,j}\, \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{\mathrm{fe},k,j}^{(N)}\big) \vec{m}_{4,j}\big(\mathbf{x}_k^{(L)}\big), \tag{1.43}$$

that represents an approximation of the filtered pdf $f(\mathbf{x}_k|\mathbf{y}_{1:k})$; in the k-th recursion, estimates of $\mathbf{x}_k^{(N)}$ and $\mathbf{x}_k^{(L)}$ can be evaluated as $\hat{\mathbf{x}}_k^{(N)} = \sum_{j=1}^{N_p} W_{\mathrm{fe},k,j}\, \mathbf{x}_{\mathrm{fp},k,j}^{(N)}$ (see Eqs. (1.15) and (1.19)) and as $\hat{\mathbf{x}}_k^{(L)} = \sum_{j=1}^{N_p} \tilde{W}_{\mathrm{fe},k,j} \eta_{4,k,j}^{(L)}$, respectively.

1.5 Message Passing in Dual Marginalized Particle Filtering

In MPF the estimation of the linear state component can benefit from the availability of PMs in particle form; the estimation of the nonlinear state component, instead, relies on real measurements only. However, thanks to the symmetric structure of the FG illustrated in Fig. 1.2, a different solution, based on the *dual* graphical model shown in Fig. 1.4, can be developed for the considered filtering problem. Readers can easily verify that:

1) In the new graphical model, PMs can be evaluated for the nonlinear state component only; on the contrary, a *single* MU can be accomplished for the linear state component.

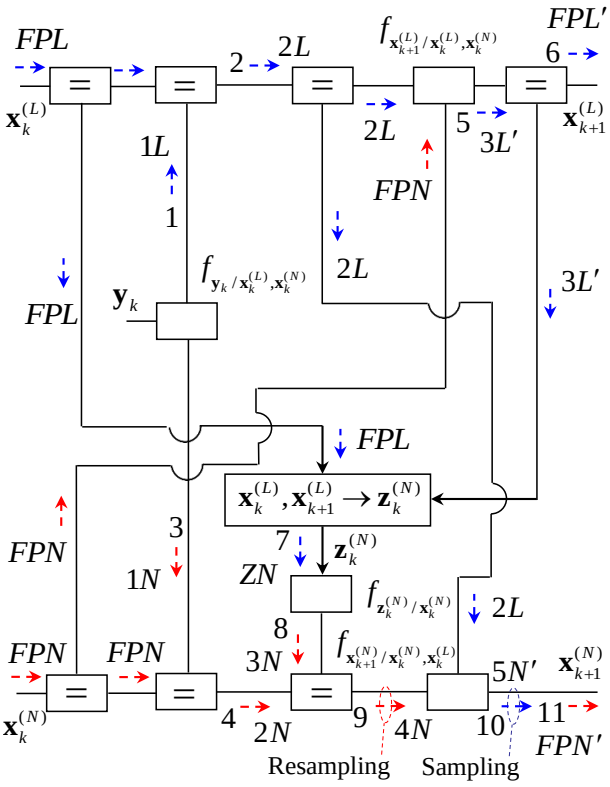


Figure 1.4: Overall factor graph for the representation of the k-th recursion of DMPF. The message flow referring to the j-th particle is shown. Gaussian messages (non Gaussian) messages computed in the considered recursion are associated with blue (red) arrows, whereas input and output messages with green arrows; the integers $1-9$ are used to specify the message scheduling adopted in the derivation of the algorithm.

2) In extracting the FG of Fig. 1.4 from that shown in Fig. 1.2, the position of the equality constraint nodes to which the function node $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ is connected has been exchanged with that of the equality constraint nodes appearing on their left; this modification is mainly motivated by the way the forward predictions $\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(L)})$ and $\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(N)})$, and the measurement $\mathbf{y}_k$ are employed in evaluation of the PM $\mathbf{z}_k^{(N)}$ (1.10).

Based on the FG shown in Fig. 1.4, a new filtering algorithm dubbed *dual* MPF (DMPF), can be developed by following the same approach as MPF. The message scheduling adopted in DMPF for the computation of the output messages $\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(N)})$ and $\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(L)})$ is illustrated in Fig. 1.4 (the notation employed for most of the involved messages is the same as Fig. 1.3; the only new acronym is ZN, representing the message $\vec{m}_{z,j}(\mathbf{z}_k^{(N)})$). In practice, the ordered SPA-based computation of the passed messages can be organized according to the following five steps: 1) $\vec{m}_{1,j}(\mathbf{x}_k^{(L)})$, $\vec{m}_{2,j}(\mathbf{x}_k^{(L)})$; 2) $\vec{m}_{1,j}(\mathbf{x}_k^{(N)})$, $\vec{m}_{2,j}(\mathbf{x}_k^{(N)})$; 3) $\vec{m}_{3,j}(\mathbf{x}_{k+1}^{(L)})$; 4) $\vec{m}_j(\mathbf{z}_k^{(N)})$, $\vec{m}_{3,j}(\mathbf{x}_k^{(N)})$, $\vec{m}_{4,j}(\mathbf{x}_k^{(N)})$, $\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(L)})$; 5) $\vec{m}_{5,j}(\mathbf{x}_{k+1}^{(N)})$, $\vec{m}_{\text{fp},j}(\mathbf{x}_{k+1}^{(N)})$. In the following, we illustrate the meaning of each step and provide the expressions of the computed messages; various mathematical details about the derivation of these expressions can be found in Appendix B.

1) MU for $\mathbf{x}_k^{(L)}$ - This step is the same as step 2) of MPF, and involves the computation of the messages $\vec{m}_{1,j}(\mathbf{x}_k^{(L)})$ and $\vec{m}_{2,j}(\mathbf{x}_k^{(L)})$; these messages are still expressed by Eqs. (1.21) and (1.23), respectively.

2) First MU for $\mathbf{x}_k^{(N)}$ - This step is the same as step 1) of MPF, and involves the computation of the messages $\vec{m}_{1,j}(\mathbf{x}_k^{(N)})$ and $\vec{m}_{2,j}(\mathbf{x}_k^{(N)})$, which are still expressed by Eqs. (1.13) and (1.14), respectively.

3) TU for $\mathbf{x}_k^{(L)}$ - This step aims at generating the message (see Fig. 1.4)

$$\vec{m}_{3,j}\big(\mathbf{x}_{k+1}^{(L)}\big) = \int f\big(\mathbf{x}_{k+1}^{(L)}\big|\mathbf{x}_k^{(L)},\mathbf{x}_k^{(N)}\big)\cdot\vec{m}_{\mathrm{fp},j}\big(\mathbf{x}_k^{(N)}\big)\;\vec{m}_{2,j}\big(\mathbf{x}_k^{(L)}\big)\,d\mathbf{x}_k^{(L)}\,d\mathbf{x}_k^{(N)} \tag{1.44}$$

$$= \mathcal{N}\big(\mathbf{x}_{k+1}^{(L)};\eta_{3,k,j}^{(L)},\mathbf{C}_{3,k,j}^{(L)}\big), \tag{1.45}$$

conveying a new prediction for $\mathbf{x}_k^{(L)}$; here,

$$\eta_{3,k,j}^{(L)} \triangleq \bar{\mathbf{A}}_{k,j}^{(L)}\;\eta_{2,k,j}^{(L)} + \bar{\mathbf{f}}_{k,j}^{(L)}, \tag{1.46}$$

$$\mathbf{C}_{3,k,j}^{(L)} \triangleq \mathbf{C}_w^{(L)} + \bar{\mathbf{A}}_{k,j}^{(L)}\;\mathbf{C}_{2,k,j}^{(L)}\big(\bar{\mathbf{A}}_{k,j}^{(L)}\big)^T, \tag{1.47}$$

$\bar{\mathbf{f}}_{k,j}^{(L)} \triangleq \mathbf{f}_k^{(L)}(\mathbf{x}_{\mathrm{fp},k,j}^{(N)})$ and $\bar{\mathbf{A}}_{k,j}^{(L)} \triangleq \mathbf{A}_k^{(L)}(\mathbf{x}_{\mathrm{fp},k,j}^{(N)})$.

4) Second MU for $\mathbf{x}_k^{(N)}$ - In this step a new weight, denoted $P_{\mathrm{fe},k,j}$, is evaluated for the j-th particle $\mathbf{x}_{\mathrm{fp},k,j}^{(N)}$; then, particle *resampling* with replacement is accomplished over the set $S_{\mathrm{fp},k}^{(N)}$ on the basis of the weights $\{P_{\mathrm{fe},k,j}\}$. In terms of message passing, this requires computing the messages $\vec{m}_j(\mathbf{z}_k^{(N)})$, $\vec{m}_{3,j}(\mathbf{x}_k^{(N)})$ and $\vec{m}_{4,j}(\mathbf{x}_k^{(N)})$. The message $\vec{m}_j(\mathbf{z}_k^{(N)})$ conveys the statistical information about the PM $\mathbf{z}_k^{(N)}$ and is given by

$$\vec{m}_j\big(\mathbf{z}_k^{(N)}\big) = \mathcal{N}\big(\mathbf{z}_k^{(N)};\eta_{\mathbf{z},k,j}^{(N)},\mathbf{C}_{\mathbf{z},k,j}^{(N)}\big), \tag{1.48}$$

where

$$\eta_{\mathbf{z},k,j}^{(N)} = \bar{\mathbf{f}}_{k,j}^{(L)} + \bar{\mathbf{A}}_{k,j}^{(L)}\big(\eta_{2,k,j}^{(L)} - \eta_{\mathrm{fp},k,j}^{(L)}\big), \tag{1.49}$$

and

$$\mathbf{C}_{\mathbf{z},k,j}^{(N)} = \mathbf{C}_w^{(L)} + \bar{\mathbf{A}}_{k,j}^{(L)}\big[\mathbf{C}_{2,k,j}^{(L)} - \mathbf{C}_{\mathrm{fp},k,j}^{(L)}\big]\big(\bar{\mathbf{A}}_{k,j}^{(L)}\big)^T. \tag{1.50}$$

The message $\vec{m}_{3,j}(\mathbf{x}_k^{(N)})$ is evaluated as[5] (see Fig. 1.4)

$$\vec{m}_{3,j}\big(\mathbf{x}_k^{(N)}\big) = \int\int \vec{m}_j\big(\mathbf{z}_k^{(N)}\big)\cdot\vec{m}_{\mathrm{fp},j}\big(\mathbf{x}_k^{(N)}\big)\,f\big(\mathbf{z}_k^{(N)}\big|\mathbf{x}_k^{(N)}\big)d\mathbf{z}_k^{(N)}d\mathbf{x}_k^{(N)} \tag{1.51}$$

$$= D_{1,k,j}^{(N)}\cdot\exp\big[\frac{1}{2}\big(\big(\eta_{1,k,j}^{(N)}\big)^T\mathbf{w}_{1,k,j}^{(N)} - \big(\eta_{\mathbf{z},k,j}^{(N)}\big)^T\mathbf{w}_{\mathbf{z},k,j}^{(N)} - \big(\bar{\mathbf{f}}_{k,j}^{(L)}\big)^T\mathbf{W}_w^{(L)}\,\bar{\mathbf{f}}_{k,j}^{(L)}\big)\big] \triangleq p_{\mathrm{fe},k,j} \tag{1.52}$$

and conveys the *unnormalized* weight $p_{\mathrm{fe},k,j}$ for the j-th particle $\mathbf{x}_{\mathrm{fp},k,j}^{(N)}$; here,

$$\mathbf{w}_{1,k,j}^{(N)} \triangleq \mathbf{W}_{1,k,j}^{(N)}\eta_{1,k,j}^{(N)} = \mathbf{w}_{\mathbf{z},k,j}^{(N)} + \mathbf{W}_w^{(L)}\;\bar{\mathbf{f}}_{k,j}^{(L)}, \tag{1.53}$$

$$\mathbf{W}_{1,k,j}^{(N)} \triangleq \big(\mathbf{C}_{1,k,j}^{(N)}\big)^{-1} = \mathbf{W}_{\mathbf{z},k,j}^{(N)} + \mathbf{W}_w^{(L)}, \tag{1.54}$$

$\mathbf{w}_{\mathbf{z},k,j}^{(N)} \triangleq \mathbf{W}_{\mathbf{z},k,j}^{(N)}\eta_{\mathbf{z},k,j}^{(N)}$, $\mathbf{W}_{\mathbf{z},k,j}^{(N)} \triangleq (\mathbf{C}_{\mathbf{z},k,j}^{(N)})^{-1}$, $\mathbf{W}_w^{(L)} \triangleq [\mathbf{C}_w^{(L)}]^{-1}$, $D_{1,k,j}^{(N)} \triangleq [\det(\tilde{\mathbf{C}}_{k,j}^{(N)})]^{-D_L/2}$ and $\tilde{\mathbf{C}}_{k,j}^{(N)} \triangleq \mathbf{C}_{\mathbf{z},k,j}^{(N)} + \mathbf{C}_w^{(L)}$. Note that the weight $p_{\mathrm{fe},k,j}$ (1.52) represents the *correlation* between the pdf $\vec{m}_j(\mathbf{z}_k^{(N)})$ evaluated on the basis of the definition of $\mathbf{z}_k^{(N)}$ (1.10) and the pdf originating from the fact that this quantity is expected to equal the random vector $\bar{\mathbf{f}}_{k,j}^{(L)} + \mathbf{w}_k^{(L)}$; for this reason, it expresses the *degree of similarity* between these two pfds.

Finally, the message $\vec{m}_{4,j}(\mathbf{x}_k^{(N)})$ is computed as

$$\vec{m}_{4,j}\big(\mathbf{x}_k^{(N)}\big) = \vec{m}_{2,j}\big(\mathbf{x}_k^{(N)}\big)\vec{m}_{3,j}\big(\mathbf{x}_k^{(N)}\big)$$

$$= \tilde{p}_{\mathrm{fe},k,j}\,\delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{\mathrm{fp},k,j}^{(N)}\big), \tag{1.55}$$

[5] In our computer simulations the factor $D_{1,k,j}^{(N)}$ appearing in the following formula has been always neglected, since it negligibly influences estimation accuracy.

where (see Eqs. (1.15) and (1.52))

$$\tilde{p}_{\mathrm{fe},k,j} = p_{\mathrm{fe},k,j} \cdot w_{\mathrm{fe},k,j}. \tag{1.56}$$

The particle weights $\{\tilde{p}_{\mathrm{fe},k,j}\}$ undergo normalization; this produces the weight

$$P_{\mathrm{fe},k,j} \triangleq \tilde{p}_{\mathrm{fe},k,j}/\Lambda_k^{(p)}, \tag{1.57}$$

with $j = 1, 2, ..., N_p$, where $\Lambda_k^{(p)} \triangleq \sum_{j=1}^{N_p} p_{\mathrm{fe},k,j}$. Particle resampling with replacement is accomplished now over the particle set $S_{\mathrm{fp},k}^{(N)} = \{\mathbf{x}_{\mathrm{fp},k,j}^{(N)}\}$ on the basis of the particle weights $\{P_{\mathrm{fe},k,j}\}$, so that the N_p predicted particles and their associated weights are replaced by the new particles $\{\mathbf{x}_{\mathrm{fe},k,j}^{(N)}\}$ and their weights $\{\tilde{P}_{\mathrm{fe},k,j} = 1/N_p\}$, respectively. After resampling $\vec{m}_4(\mathbf{x}_k^{(N)})$ (1.55) is replaced by

$$\vec{m}_{4,j}\big(\mathbf{x}_k^{(N)}\big) = \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{\mathrm{fe},k,j}^{(N)}\big) \tag{1.58}$$

and the set of messages $\{\vec{m}_{3,j}(\mathbf{x}_{k+1}^{(L)})\}$ (see Eq. (1.45)) is reordered accordingly, so providing the set of output messages $\{\vec{m}_{\mathrm{fp},j}(\mathbf{x}_{k+1}^{(L)})\}$.

5) TU for $\mathbf{x}_k^{(N)}$ - This step aims at generating the j-th particle for $\mathbf{x}_{k+1}^{(N)}$ and its associated weight. These information are conveyed by the message

$$\vec{m}_{5,j}\big(\mathbf{x}_{k+1}^{(N)}\big) = \int\int f\big(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(L)}, \mathbf{x}_k^{(N)}\big) \cdot \vec{m}_{2,j}\big(\mathbf{x}_k^{(N)}\big)\vec{m}_{2,j}\big(\mathbf{x}_k^{(L)}\big) d\mathbf{x}_k^{(L)} d\mathbf{x}_k^{(N)} \tag{1.59}$$

$$= \mathcal{N}\big(\mathbf{x}_{k+1}^{(N)}; \eta_{5,k,j}^{(N)}, \mathbf{C}_{5,k,j}^{(N)}\big) \tag{1.60}$$

in Gaussian form and by the message $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_{k+1}^{(N)})$ in particle form; here,

$$\eta_{5,k,j}^{(N)} \triangleq \mathbf{A}_{k,j}^{(N)} \eta_{2,k,j}^{(L)} + \mathbf{f}_{k,j}^{(N)}, \tag{1.61}$$

$$\mathbf{C}_{5,k,j}^{(N)} \triangleq \mathbf{C}_w^{(N)} + \mathbf{A}_{k,j}^{(N)}\, \mathbf{C}_{2,k,j}^{(L)}\, \big(\mathbf{A}_{k,j}^{(N)}\big)^T, \tag{1.62}$$

$\mathbf{A}_{k,j}^{(N)} \triangleq \mathbf{A}_k^{(N)}(\mathbf{x}_{\mathrm{fe},k,j}^{(N)})$ and $\mathbf{f}_{k,j}^{(N)} \triangleq \mathbf{f}_k^{(N)}(\mathbf{x}_{\mathrm{fe},k,j}^{(N)})$. Given the Gaussian function $\mathcal{N}(\mathbf{x}_{k+1}^{(N)}; \eta_{5,k,j}^{(N)}, \mathbf{C}_{5,k,j}^{(N)})$ (1.60), the new particle $\mathbf{x}_{\mathrm{fp},k+1,j}^{(N)}$ is generated by sampling it and the weight $w_{\mathrm{fp},k+1,j} = 1/N_p$ is assigned to this particle. Then, the message $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_{k+1}^{(N)})$ is computed as

$$\vec{m}_{\mathrm{fp},j}(\mathbf{x}_{k+1}^{(N)}) = \delta\big(\mathbf{x}_{k+1}^{(N)} - \mathbf{x}_{\mathrm{fp},k+1,j}^{(N)}\big). \tag{1.63}$$

This concludes the k-th recusion of DMPF. As far as its initialization is concerned, it can be accomplished exactly in the same way as in MPF.

The scheduling illustrated in the derivation of the DMPF algorithm mimicks the one adopted in MPF; however, it can be modified for the following reasons. From Eqs. (1.49)–(1.50) it is easily inferred that the computation of the message $\vec{m}_j(\mathbf{z}_k^{(N)})$ (1.48) (and, consequently, of the messages $\vec{m}_{3,j}(\mathbf{x}_k^{(N)})$ (1.52) and $\vec{m}_{4,j}(\mathbf{x}_k^{(N)})$ (1.55)) can be accomplished as soon as the message $\vec{m}_{2,j}(\mathbf{x}_k^{(L)})$ becomes available (i.e., as soon as step 1) is over). For this reason, in implementing the k-th recursion of DMPF, the involved messages can be computed according to following (alternative) order (different from the one indicated in Fig. 1.4): 1) $\vec{m}_{1,j}(\mathbf{x}_k^{(L)})$, $\vec{m}_{2,j}(\mathbf{x}_k^{(L)})$; 2) $\vec{m}_{1,j}(\mathbf{x}_k^{(N)})$, $\vec{m}_{2,j}(\mathbf{x}_k^{(N)})$; 3) $\vec{m}_j(\mathbf{z}_k^{(N)})$, $\vec{m}_{3,j}(\mathbf{x}_k^{(N)})$, $\vec{m}_{4,j}(\mathbf{x}_k^{(N)})$; 4) $\vec{m}_{3,j}(\mathbf{x}_{k+1}^{(L)})$, $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_{k+1}^{(L)})$; 5) $\vec{m}_{5,j}(\mathbf{x}_{k+1}^{(N)})$, $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_{k+1}^{(N)})$. Note also that, if this scheduling is adopted, the particle $\mathbf{x}_{\mathrm{fp},k,j}^{(N)}$ is replaced by its counterpart $\mathbf{x}_{\mathrm{fe},k,j}^{(N)}$ (available after particle resampling) in the evaluation of $\eta_{3,k,j}^{(L)}$ (1.46) and $\mathbf{C}_{3,k,j}^{(L)}$ (1.47). The DMPF technique based on the last scheduling is summarized in Algorithm 2.

Finally, it is interesting to point out that DMPF, unlike MPF, may not achieve accurate state estimation in the special case of CLG SSM investigated in [6, Par. III.B], i.e. when the

Algorithm 2: Dual marginalized particle filtering

Initialisation: same as step 1 of MPF.

Filtering: For $k = 1$ to T:

a- MU for $\mathbf{x}_k^{(L)}$: same as step 2d of Alg. 1.

b- First MU for $\mathbf{x}_k^{(N)}$: same as step 2a of Alg. 1.

c- Second MU for $\mathbf{x}_k^{(N)}$: For $j = 1$ to N_p: compute $\eta_{\mathbf{z},k,j}^{(N)}$ (1.49) and $\mathbf{C}_{\mathbf{z},k,j}^{(N)}$ (1.50), the weight $\tilde{p}_{\mathrm{fe},k,j}$ (1.52) and the overall particle weight $p_{\mathrm{fe},k,j}$ (1.56).

d- *Normalization of particle weights*: For $j = 1$ to N_p: compute the normalised particle weights $\{P_{\mathrm{fe},k,j}\}$ according to (1.57).

e- *Particle resampling*: For $j = 1$ to N_p: generate the new particle $\mathbf{x}_{\mathrm{fe},k,j}^{(N)}$ by resampling with replacement over the particle set $S_{\mathrm{fp},k}^{(N)}$ on the basis of the weights $\{P_{\mathrm{fe},k,j}\}$ and assign the new weight $w_{\mathrm{fe},k,j} = 1/N_p$ to it. Then, reorder the message set $\{\vec{m}_{2,j}(\mathbf{x}_k^{(L)})\}$ accordingly.

f- TU for $\mathbf{x}_k^{(L)}$: For $j = 1$ to N_p: compute $\eta_{3,k,j}^{(L)}$ (1.46) and $\mathbf{C}_{3,k,j}^{(L)}$ (1.47) (in doing so, replace $\overline{\mathbf{f}}_{k,j}^{(L)}$ and $\bar{\mathbf{A}}_{k,j}^{(L)}$ with $\mathbf{f}_{k,j}^{(L)} \triangleq \mathbf{f}_k^{(L)}(\mathbf{x}_{\mathrm{fe},k,j}^{(N)})$ and $\mathbf{A}_{k,j}^{(L)} \triangleq \mathbf{A}_k^{(L)}(\mathbf{x}_{\mathrm{fe},k,j}^{(N)})$, respectively); then, set $\eta_{\mathrm{fp},k+1,j}^{(L)} = \eta_{3,k,j}^{(L)}$ and $\mathbf{C}_{\mathrm{fp},k+1,j}^{(L)} = \mathbf{C}_{3,k,j}^{(L)}$.

g- TU for $\mathbf{x}_k^{(N)}$: For $j = 1$ to N_p: compute $\eta_{5,k,j}^{(N)}$ (1.61) and $\mathbf{C}_{5,k,j}^{(N)}$ (1.62); then, sample the pdf $\mathcal{N}(\mathbf{x}_k^{(L)}; \eta_{5,k,j}^{(N)}, \mathbf{C}_{5,k,j}^{(N)})$ to generate the new particle $\mathbf{x}_{\mathrm{fp},k+1,j}^{(N)}$ and assign the weight $w_{\mathrm{fp},k+1,j} = 1/N_p$ to it.

measurement equation (1.4) depends on the *nonlinear* state component only (in other words, when $\mathbf{B}_k(\mathbf{x}_k^{(N)}) = \mathbf{0}_{P,D_L}$ in Eq. (1.4)); in fact, in this case, new information about the linear state component can be acquired through the PM $\mathbf{z}_k^{(L)}$ (1.9) only and such a PM is unavailable in DMPF. Dually, MPF, unlike DMPF, may suffer from the same problem when the measurement equation depends on the *linear* state component only (i.e., when $\mathbf{h}_k(\mathbf{x}_k^{(N)}) = \mathbf{0}_P$ in Eq. (1.4)), so that the edge connecting $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ to $\mathbf{x}_k^{(N)}$ in Fig. 1.4 is missing (since $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)}) = f(\mathbf{y}_k|\,\mathbf{x}_k^{(L)})$). Note also that, in the last case, the first MU for $\mathbf{x}_k^{(N)}$ (i.e., step 2b of Algorithm 2) is not accomplished in DMPF; in practice, this means that $\vec{m}_{1,j}(\mathbf{x}_k^{(N)})$ is not computed and that $\vec{m}_{2,j}(\mathbf{x}_k^{(N)}) = \vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(N)})$.

1.6 Simplifying Message Passing in Filtering Algorithms

The message passing procedures illustrated in the previous two Sections show the inner structure of the processing accomplished by MPF and DMPF within each recursion. Therefore, they pave the way for the development of new filtering algorithms based on them. In this Section we first formulate some simple rules for simplifying the computation of messages in MPF and DMPF; then, we discuss where these rules can be employed. It is worth stressing that some methods for reducing MPF computational complexity [29] have been already proposed in the technical literature [10], [30], [11]. More specifically, the method proposed in [10] and [30] is based on representing the particle set for $\mathbf{x}_k^{(N)}$ as a *single* particle (that corresponds to the *center of mass* of the set itself), so that a *single* Kalman filter can be employed for the linear state component. Unluckily, this simplified MPF algorithm works only if the posterior distribution of $\mathbf{x}_k^{(N)}$ is *unimodal*. Its generalization to the case in which the posterior distribution of $\mathbf{x}_k^{(N)}$ is *multimodal* has been illustrated in [11]. The proposed technique is based on: a) partitioning the particles available in the k-th recursion into G_k groups or clusters (the parameter G_k is required to equal the number of *modes* of the posterior density of $\mathbf{x}_k^{(N)}$); b) representing each group through a single particle that corresponds to the *center of mass* of the group itself. This allows to reduce the overall number of Kalman filters from N_p to G_k. However, the implementation of this technique requires solving the following two specific problems: a) identifying the number of modes of the

posterior distribution of $\mathbf{x}_k^{(N)}$; b) partitioning the particles into clusters according to a grouping method in each recursion. Unluckily, solutions to these problems have not been proposed in [11].

We believe that, generally speaking, the following three simple rules can be exploited to simplify message passing in MPF and DMPF:

R1 - A set of N_p equal weight particles $\{\mathbf{x}_j^{(N)};\, j = 0,\, 1,\, ...,\, N_p - 1\}$ can be represented through its *center of mass* $\bar{\mathbf{x}}^{(N)} \triangleq \sum_{j=0}^{N_p-1} \mathbf{x}_j^{(N)}/N_p$ (this rule is also adopted in [30] and [11]).

R2 - A GM $f_{GM}\big(\mathbf{x}^{(L)}\big)$ having N_p equal weight components $\{\mathcal{N}(\mathbf{x}^{(L)}; \eta_j^{(L)}, \mathbf{C}_j^{(L)})\}$ and representing the statistical model for the linear state component can be approximated through its projection onto the (single) Gaussian pdf $f_G\big(\mathbf{x}^{(L)}\big) = \mathcal{N}\big(\mathbf{x}^{(L)}; \eta_G, \mathbf{C}_G\big)$, where[6] (e.g., see [31, Sec. IV])

$$\eta_G \triangleq \sum_{j=0}^{N_p-1} \eta_j^{(L)}/N_p \tag{1.64}$$

and

$$\mathbf{C}_G = \mathbf{C}_A - \eta_G\big(\eta_G\big)^T + \Sigma_\eta \tag{1.65}$$

with $\Sigma_\eta \triangleq (1/N_p)\sum_{j=0}^{N_p-1} \eta_j^{(L)}\big(\eta_j^{(L)}\big)^T$ and

$$\mathbf{C}_A = \sum_{j=0}^{N_p-1} \mathbf{C}_j^{(L)}/N_p. \tag{1.66}$$

R3 - The GM $f_{GM}\big(\mathbf{x}^{(L)}\big)$ defined in R2 can be also approximated through a N_p-component GM, whose Gaussian components have the same means as those of $f_{GM}\big(\mathbf{x}^{(L)}\big)$, but a *common covariance matrix*, expressed by $\mathbf{C}_A$ (1.66).

In practice, rule R1 can be employed in the MPF/DMPF formulas involving functions ($\mathbf{f}_k^{(L)}(\cdot)$, $\mathbf{f}_k^{(N)}(\cdot)$ or $\mathbf{h}_k(\cdot)$) and/or matrices ($\mathbf{A}_k^{(L)}(\cdot)$, $\mathbf{A}_k^{(N)}(\cdot)$ or $\mathbf{B}_k(\cdot)$) depending on $\mathbf{x}_k^{(N)}$, in order to make the contribution of such terms particle-independent. From a message passing viewpoint, this corresponds to replacing the set of equal weight messages $\{\vec{m}_j(\mathbf{x}^{(N)}) = \delta(\mathbf{x}^{(N)} - \mathbf{x}_j^{(N)})\}$ (e.g., see $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(N)})$ (1.11)) with the *single* particle-independent message $\vec{m}(\mathbf{x}^{(N)}) = \delta(\mathbf{x}^{(N)} - \bar{\mathbf{x}}^{(N)})$. Rules R2 and R3, instead, can be exploited to simplify the processing tasks involving $\mathbf{x}_k^{(L)}$, which is represented, in any step of MPF/DMPF, through a N_p- component GM. From a message passing viewpoint, R2 (R3) corresponds to replacing a set of Gaussian messages $\{\vec{m}_j(\mathbf{x}^{(L)}) = \mathcal{N}(\mathbf{x}^{(L)}; \eta_j^{(L)}, \mathbf{C}_j^{(L)})\}$ (associated with particles having the same weights; e.g., see $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(L)})$ (1.12)) with the (particle-independent) message $\vec{m}(\mathbf{x}^{(L)}) = \mathcal{N}(\mathbf{x}^{(L)}; \eta_G, \mathbf{C}_G)$ (with the structurally simpler set $\{\vec{m}_j(\mathbf{x}^{(L)}) = \mathcal{N}(\mathbf{x}^{(L)}; \eta_j^{(L)}, \mathbf{C}_A)\}$). Note also that, generally speaking, R2 should not be used if the pdf of $\mathbf{x}^{(L)}$ is multimodal.

In principle, a substantially complexity reduction can be achieved by reducing the overall number of *Cholesky decompositions* and *matrix inversions* required by MPF/DMPF in their k-recursion. Note that, on the one hand, the former are computed by MPF and DMPF in the generation of the new particle set $S_{\mathrm{fp},k+1}^{(N)}$ (and involve the N_p matrices $\{\mathbf{C}_{3,k,j}^{(L)}\}$ (1.31) and $\{\mathbf{C}_{5,k,j}^{(L)}\}$ (1.62), respectively). On the other hand, the latter are needed to compute: a) the N_p matrices $\{\mathbf{W}_{1,k,j}^{(N)}\}$ (employed in step 2-a of Alg. 1 and step 2-b of Alg. 2 for the computation of particle weights; see Eq. (1.16)); b) the N_p matrices $\{\mathbf{W}_{\mathrm{fp},k,j}^{(L)}\}$ (employed in step 2-d of Alg. 1 and step 2-a of Alg. 2; see Eqs. (1.25) and (1.27)); c) the N_p matrices $\{\mathbf{C}_{2,k,j}^{(L)}\}$ (employed in step 2-e of Alg. 1 and step 2-f of Alg. 2; see Eqs. (1.31) and (1.47), respectively); d) the N_p matrices $\{\mathbf{C}_{4,k,j}^{(L)}\}$ employed in step 2-g of Alg. 1 (see Eq. (1.42)); e) the N_p matrices $\{\mathbf{C}_{4,k,j}^{(L)}\}$ employed in step 2-g

[6] Note that the pdf transformations based on rules R2 and R3 preserve both the *mean* and the *covariance* of the considered GM.

of Alg. 1 (see Eq. (1.42)); f) the N_p matrices $\mathbf{W}_{\mathbf{z},k,j}^{(N)}$ and $\mathbf{C}_{1,k,j}^{(N)}$ required in the computation of the PM-based particle weights in step 2-c of Alg. 2 (see Eqs. (1.50), (1.54) and (1.52)).

Unluckily, no mathematical method is available in the technical literature for a priori assessing the impact of any simplification on estimation accuracy, because of the highly nonlinear behavior of the two considered algorithms; for this reason, for any SSM, the loss in estimation accuracy due to simplifications based on the rules R1-R3 can be assessed only via computer simulations. As a matter of fact, our simulation simulation results have evidenced that the performance degradation originating from the adoption of the three rules illustrated above is highly dependent on the structure of the considered SSM.

1.7 Numerical Results

In this Section, we first compare, in terms of accuracy and computational load, MPF, DMPF and simplified versions of both algorithms for a specific *unimodal* CLG SSM. Then, we consider a simple SSM characterized by a *bimodal* distribution of system state and analyse the impact of specific simplifications adopted in MPF/DMPF processing.

1.7.1 Unimodal state space model

The CLG SSM considered in this Section (and denoted SSM#1 in the following) refers to an agent moving on a plane and whose state $\mathbf{x}_k$ in the k-th observation interval is defined as $\mathbf{x}_k \triangleq [\mathbf{p}_k^T, \mathbf{v}_k^T]^T$, where $\mathbf{p}_k \triangleq [p_{x,k}, p_{y,k}]^T$ and $\mathbf{v}_k \triangleq [v_{x,k}, v_{y,k}]^T$ represent the agent position and its velocity, respectively (their components are expressed in m and in m/s, respectively). As far as the state update equations are concerned, we assume that the agent velocity is approximately constant within each sampling interval and the model describing its time evolution is obtained by including the contribution of a *position- and velocity-dependent force* in a first-order autoregressive model (characterized by the *forgetting factor* ρ, with $0 < \rho < 1$); therefore, the dynamic model

$$\mathbf{v}_{k+1} = \rho \mathbf{v}_k + \big(1-\rho\big)\mathbf{n}_{v,k} + T_s\, \mathbf{a}\big(\mathbf{p}_k\big), \tag{1.67}$$

is used for velocity; here, $\{\mathbf{n}_{v,k}\}$ is an *additive white Gaussian noise* (AWGN) process (whose elements are characterized by the covariance matrix $\mathbf{I}_2$), $\mathbf{a}\big(\mathbf{p}_k\big)$ is the acceleration resulting from the applied force and T is the sampling interval. Consequently, the dynamic model

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k \cdot T_s + \frac{1}{2} T_s^2\, \mathbf{a}\big(\mathbf{p}_k\big) + \mathbf{n}_{p,k} \tag{1.68}$$

can be employed for the position of the considered agent; here, $\{\mathbf{n}_{p,k}\}$ is an AWGN process (whose elements are characterized by the covariance matrix $\sigma_p^2 \mathbf{I}_2$), that accounts for model inaccuracy. We also assume that the position-dependent component of the force acting on the agent points towards the origin, whereas its velocity-dependent component represents a resistance to the motion of the agent; therefore, the resulting acceleration is expressed as

$$\mathbf{a}\big(\mathbf{p}_k\big) = -a_0 \mathbf{u}_{p,k}\, f_p\big(\big\|\mathbf{p}_k\big\|\big) - \tilde{a}_0 \mathbf{u}_{v,k}\, f_v\big(\big\|\mathbf{v}_k\big\|\big), \tag{1.69}$$

where a_0 and $\tilde{a}_0$ are scale factors (in m/s^2), $\mathbf{u}_{p,k} \triangleq \mathbf{p}_k / \big\|\mathbf{p}_k\big\|$ ($\mathbf{u}_{v,k} \triangleq \mathbf{v}_k / \big\|\mathbf{v}_k\big\|$) is the versor associated with $\mathbf{p}_k$ ($\mathbf{v}_k$) and $f_p\big(\mathbf{p}\big)$ ($f_v\big(\mathbf{v}\big)$) is a dimensionless function expressing the dependence of the given acceleration on the distance of the agent from the origin (on the intensity of its velocity). Moreover, the models $f_p(x) = x/d_0$ and $f_v(x) = (x/v_0)^3$ are adopted, where d_0 (v_0) is a *reference distance* (*reference velocity*); note that such models are continuous and differentiable, and contain a single parameter.

In our model, noisy and unbiased measurements are available for position only; therefore, the measurement vector $\mathbf{y}_k$ is expressed as

$$\mathbf{y}_k = \mathbf{p}_k + \mathbf{e}_k, \tag{1.70}$$

where $\{\mathbf{e}_k\}$ is an AWGN process, whose elements are characterized by the covariance matrix $\sigma_e^2\mathbf{I}_2$. Then, it is easy to show that, if we set $\mathbf{x}_k^{(L)} = \mathbf{p}_k$ and $\mathbf{x}_k^{(N)} = \mathbf{v}_k$, the state equation (1.67) ((1.68)) and the measurement equation (1.70) can been interpreted as instances of (1.3) with $Z = N$ ((1.3) with $Z = L$) and (1.4), respectively.

For SSM#1, we have run four filtering algorithms, namely MPF, a *simplified* version of MPF (called SMPF), DMPF and a *simplified* version of DMPF (called SDMPF). Both simplified algorithms are based on the approach illustrated in [30]; in other words, SMPF (SDMPF) results from the application of rule R1 only (see Section 1.6) to steps 2-**d**, 2-**f** and 2-**g** of Algorithm 1 (steps 2-**a** and 2-**f** of Algorithm 2). This means that the two MUs and the TU (the MU and the TU) referring to the linear state component are accomplished for a *single particle* in MPF (DMPF); in the k-th interval, this particle is evaluated as

$$\bar{\mathbf{x}}_{\mathrm{fx},k}^{(N)} = \sum_{j=0}^{N_p-1} \mathbf{x}_{\mathrm{fx},k,j}^{(N)}/N_p \tag{1.71}$$

and represents the center of mass the particle set $\{\mathbf{x}_{\mathrm{fx},k,j}^{(N)}\}$ (with $x = p$ or e); consequently, the particle dependent quantities $\mathbf{B}_{k,j}$, $\mathbf{h}_{k,j}$, $\mathbf{f}_{k,j}^{(N)}$, $\mathbf{A}_{k,j}^{(N)}$, $\mathbf{A}_{k,j}^{(L)}$ and $\mathbf{f}_{k,j}^{(L)}$ appearing in the formulas of both MPF and DMPF are all evaluated for $\mathbf{x}_k^{(N)} = \bar{\mathbf{x}}_{\mathrm{fx},k}^{(N)}$ only.

In our computer simulations, the estimation accuracy of the considered filtering methods has been assessed by evaluating the *root mean square error* (RMSE) of state estimates. More specifically, two RMSEs, namely RMSE_L(alg) and RMSE_N(alg) have been computed for any filtering technique (here, 'alg' denotes the algorithm these parameters refer to); in practice, RMSE_L(alg) (RMSE_N(alg)) represents the square root of the average *mean square error* (MSE) evaluated for the two elements of the linear (nonlinear) state component; this distinction is important since, as evidenced by our simulation results, the values taken on by these two RMSEs for a given SSM can be quite different. Our assessment of *computational requirements* is based, instead, on assessing the average *execution time* required over an observation interval lasting $T = 300\ T_s$ s (this quantity is denoted ET(alg) in the following).

In our computer simulations, the following values have been selected for the parameters of the considered SSM: $\rho = 0.99$, $T_s = 0.1$ s, $\sigma_p = 1 \cdot 10^{-3}$ m, $\sigma_e = 5 \cdot 10^{-2}$ m, $a_0 = 1.5$ m/s^2, $d_0 = 0.5$ m, $\tilde{a}_0 = 0.05$ m/s^2 and $v_0 = 1$ m/s. Moreover, the initial position $\mathbf{p}_0 \triangleq [p_{x,0}, p_{y,0}]^T$ and the initial velocity $\mathbf{v}_0 \triangleq [v_{x,0}, v_{y,0}]^T$ have been set to $[5\text{ m}, 8\text{ m}]^T$ and [4 m/s,4 m/s]^T.

Some numerical results showing the dependence of RMSE_L and RMSE_N on the number of particles (N_p) for MPF, SMPF, DMPF and SDMPF are illustrated in Fig. 1.5 (simulation results are indicated by markers, whereas continuous lines are drawn to fit them, so facilitating the interpretation of the available data). These results show that:

a) RMSE_L is significantly smaller than RMSE_N; this is mainly due to the fact that $\mathbf{y}_k$ (1.70) depends on $\mathbf{x}_k^{(L)}$ only.

b) No significant improvement in RMSE_N is achieved for $N_p \gtrsim 100$; on the contrary, the dependence of RMSE_L on N_p is really weak.

c) MPF is slightly outperformed by DMPF for $N_p < 100$.

d) On the one hand, SMPF is outperformed by MPF; on the other hand, surprisingly, SDMPF performs slightly better than the other three filtering algorithms.

Some numerical results showing the dependence of the ET on N_p for all the considered filtering algorithms are illustrated in Fig. 1.6. These results show that:

a) DMPF is slightly faster than MPF;

b) The simplifications adopted in both MPF and SMPF entail a substantial reduction of the computational effort; more specifically, ET(SMPF) (ET(SDMPF)) is about 68% (35%) smaller than ET(MPF) (ET(DMPF)).

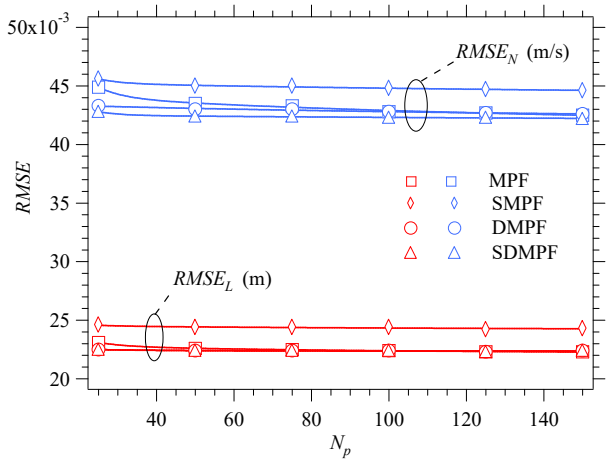


Figure 1.5: RMSE performance versus N_p for the linear component (RMSE_L) and the nonlinear component (RMSE_N) of system state; the CLG SSM described by eqs. (1.67)–(1.69) and four filtering techniques (MPF, SMPF, DMPF and SDMPF) are considered.

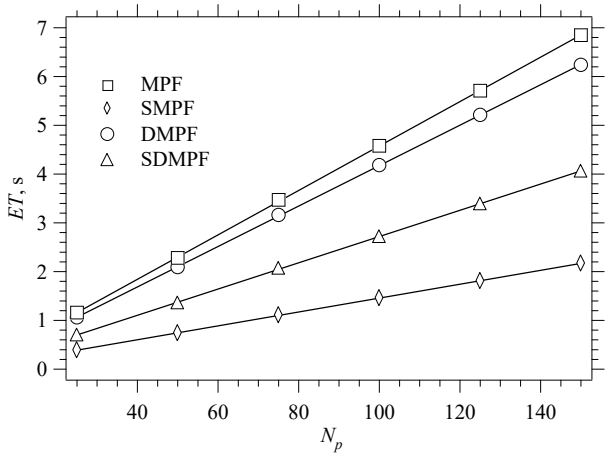


Figure 1.6: ET versus N_p for the MPF, SMPF, DMPF and SDMPF; the CLG SSM described by eqs. (1.67)–(1.69) is considered.

Therefore, from the results shown in Figs. 1.5 and 1.6 it is easily inferred that, in the considered scenario, the SDMPF algorithm achieves the best complexity-accuracy tradeoff.

1.7.2 Bimodal state space model

In this Paragraph a simple bimodal SSM (denoted SSM#2 in the following) inspired by [11] and described by the first-order autoregressive model

$$\mathbf{x}_{k+1} = \rho_x \mathbf{x}_k + \mathbf{u}\, f_u(\mathbf{x}_k) + \left(1 - \rho_x\right) \mathbf{n}_{x,k} \tag{1.72}$$

is considered; here, $\mathbf{x}_k \triangleq [x_{1,k}^{(L)}, x_{1,k}^{(N)}]^T$, ρ_x is a *forgetting factor* (with $0 < \rho_x < 1$) and $\{\mathbf{n}_{x,k}\}$ is an AWGN) process (whose elements are characterized by the covariance matrix $\mathbf{I}_2$), $\mathbf{u} = [1\,0]^T$ and $f_u(\mathbf{x}_k) = A \cdot \sin(x_{1,k}^{(N)})$. The measurement model is expressed by (see [11, Sec. 3, eqs. (2)-(3)])

$$y_{1,k} = \left(x_{1,k}^{(N)}\right)^2 + 2x_{1,k}^{(L)} + e_{1,k}, \tag{1.73}$$

$$y_{2,k} = \left(x_{1,k}^{(N)}\right)^2 - x_{1,k}^{(L)} + e_{2,k}; \tag{1.74}$$

the observation noise $\{\mathbf{e}_k\}$ (with $\mathbf{e}_k \triangleq [e_{1,k}, e_{2,k}]^T$) is modelled as an AWGN process having the same properties as its counterpart in Eq. (1.70).

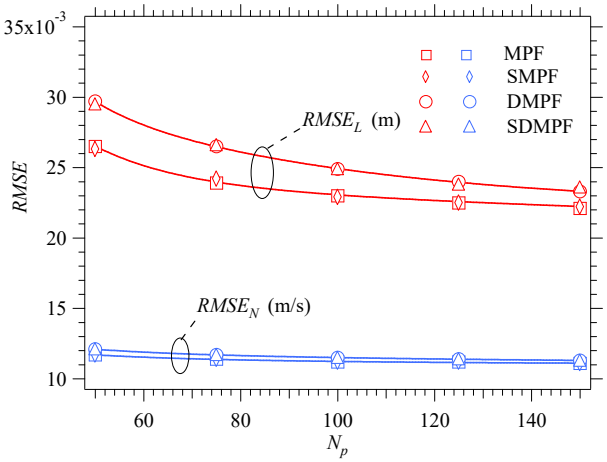


Figure 1.7: RMSE performance versus N_p for the linear component (RMSE_L) and the nonlinear component (RMSE_N) of system state; the CLG SSM described by eqs. (1.72)–(1.74) and four filtering techniques (MPF, SMPF, DMPF and SDMPF) are considered.

In the case of SSM#2, the SMPF and SDMPF algorithms described in the previous Paragraph and based on R1 only do not work, since the center of mass of any particle set provides a poor representation of the nonlinear state component [11]. However, in this new case, rule R3 (see Section 1.6) can be employed to develop new simplified versions of MPF and DMPF (dubbed again SMPF and SDMPF, respectively). More specifically, our simplified algorithms are based on the following two simplifications:

1. In step 2-**a** (2-**b**) of Algorithm 1 (Algorithm 2), the covariance matrices $\{\mathbf{C}_{1,k,j}^{(N)}\}$ (see Eq. (1.18)) have been merged into a single matrix $\mathbf{C}_{1,k}^{(N)}$ on the basis of Eq. (1.66) when computing the particle weights $\{w_{\text{fe},k,j}\}$ (see Eq. (1.16)); consequently, a single matrix inversion is required in this task.

2. In step 2-**e** of Algorithm 1 the covariance matrices $\{\mathbf{C}_{3,k,j}^{(N)}\}$ (see Eq. (1.31)) have been merged into a single matrix $\mathbf{C}_{3,k}^{(N)}$ according to Eq. (1.66) when computing the new particle set $\{\mathbf{x}_{\text{fp},k+1,j}^{(N)}\}$; consequently, a single Cholesky decomposition is accomplished in the generation of new particles. The same has been done in step 2-**g** of Algorithm 2, where the covariance matrices $\{\mathbf{C}_{5,k,j}^{(N)}\}$ (see Eq. (1.62)) have been merged into a single matrix $\mathbf{C}_{5,k}^{(N)}$.

In our computer simulations, the following values have been selected for the parameters of the considered SSM: $\rho_x = 0.99$, $A = 5$ and $\sigma_e = 5 \cdot 10^{-2}$. Moreover, the initial state $\mathbf{x}_0$ has been set to $[2, 0.5]^T$ and filtering algorithms have been run over 300 observation intervals. Some numerical results showing the dependence of RMSE_L and RMSE_N (referring to the estimation error on $|x_{1,k}^{(N)}|$) on the number of particles (N_p) for MPF, SMPF, DMPF and SDMPF are illustrated in Fig. 1.7. From these results it is easily inferred that:

a) The performance gap between MPF/DMPF and their simplified counterparts is negligible.

b) The DMPF (SDMPF) is outperformed by MPF (SMPF); for instance, RMSE_L(DMPF) and RMSE_N(DMPF) are roughly 1.08 and 1.03 times larger than RMSE_L(MPF) and RMSE_N(MPF), respectively, for $N_p = 100$.

As far the computational effort is concerned, the shorter execution time is provided by SDMPF (in particular, ET(SDMPF) $\cong$ 1.78 s for $N_p = 100$). The other algorithms are more computationally intensive, even if the computational gap between them is limited. In fact, in the considered case, ET(MPF), ET(DMPF) and ET(SMPF) are roughly 1.14, 1.15, 1.04 times larger than ET(SDMPF), respectively. Therefore, these results lead to the conclusion that, in the considered scenario, the SMPF technique achieves the best complexity-accuracy tradeoff.

1.8 Conclusions

In this chapter a FG approach has been employed to analyse the filtering problem for CLG SSMs. This has allowed us to: a) derive a new graphical model for representing the filtering problem as message passing; b) provide a new interpretation of MPF as a *forward only* message passing algorithm over the devised FG; c) develop a new filtering algorithm, called *dual* MPF (DMPF). Moreover, some rules for simplifying the computation of the passed messages in MPF and DMPF have been briefly illustrated. All the considered filtering techniques have been compared in terms of both accuracy and computational requirements for specific CLG SSMs. Our ongoing research activities in this area concern further applications of graphical models to the filtering and smoothing problems.

Two

Multiple Bayesian Filtering as Message Passing

In this chapter, a general method for deriving filtering algorithms that involve a network of interconnected Bayesian filters is proposed. This method is based on the idea that the processing accomplished inside each of the Bayesian filters and the interactions between them can be represented as message passing algorithms over a proper graphical model. The usefulness of our method is exemplified by developing new filtering techniques, based on the interconnection of a particle filter and an extended Kalman filter, for conditionally linear Gaussian systems. Numerical results for two specific dynamic systems evidence that the devised algorithms can achieve a better complexity-accuracy tradeoff than marginalized particle filtering and multiple particle filtering.

Part of this Chapter has been published in [32].

2.1 Introduction

It is well known that *Bayesian filtering* represents a general recursive solution to the *nonlinear filtering problem* (e.g., see [2, Sect. II, eqs. (3)–(5)]), that is to the problem of inferring the posterior distribution of the hidden state of a nonlinear *state-space model* (SSM). Unfortunately, this solution can be put in closed form only in a few cases [3]. For this reason, various filtering methods generating a functional approximation of the desired posterior pdf have been developed; these can be divided into *local* and *global* methods on the basis of the way the posterior *probability density function* (pdf) is approximated [10, 33, 34]. On the one hand, local techniques, like *extended Kalman filtering* (EKF) [3], are computationally efficient, but may suffer from error accumulation over time; on the other hand, global techniques, like *particle filtering* (PF) [4, 9], may achieve high accuracy at the price, however, of unacceptable complexity and numerical problems when the dimension of the state space becomes large [35–37]. These considerations have motivated the investigation of *various methods able to achieve high accuracy under given computational constraints*. Some of such solutions are based on the idea of *combining local and global methods*; relevant examples of this approach are represented by: 1) *Rao-Blackwellized particle filtering* (RBPF; also known as *marginalized particle filtering*) [6] and other techniques related to it (e.g., see [10]); 2) *cascaded architectures* based on the joint use of EKF and PF (e.g., see [38]). Note that, in the first case, the state vector is split into two disjoint components, namely, a *linear state component* and a *nonlinear state component*; moreover, these are estimated by a bank of Kalman filters and by a particle filter, respectively. In the second case, instead, an extended Kalman filter and a particle filter are run over partially overlapped state vectors. In both cases, however, two heterogeneous filtering methods are combined in a way that the resulting overall algorithm is forward only and, within each of its recursions, both methods are executed only once. Another class of solutions, known as *multiple particle filtering* (MPF), is based on the idea of partitioning the state vector into multiple substates and running multiple particle filters in parallel, one on each subspace [37, 39, 40]. The resulting network of particle filters requires the mutual exchange of statistical information (in the form of estimates/predictions of the tracked substates or parametric distributions), so that, within each filter, the unknown portion of the state vector can be integrated out in both weight computation and particle propagation. In

principle, the substates estimated by the particle filters forming the network are required to be separable in the state equation; however, recently, approximate MPF techniques have been also developed for the more general case of a non-separable state transition pdf [40]. Moreover, the technical literature about MPF has raised three interesting technical issues that have received limited attention until now. The first issue refers to the possibility of coupling an extended Kalman filter with each particle filter of the network; the former filter should provide the latter one with the statistical information required for integrating out the unknown portion of the state vector (see [41, Par. 3.2]). The second one concerns the use of filters having *partially overlapped* substates (see [42, Sec.1]). The third (and final) issue, instead, concerns the iterative exchange of statistical information among the interconnected filters of the network. Some work related to the first issue can be found in [43], [44] and [45]. In this case, however, the proposed solution is based on Rao-Blackwellisation; for this reason, each particle filter of the network is not coupled with a single extended Kalman filter, but with a bank of Kalman filters. The second issue has not been investigated at all, whereas limited attention has been paid to the third one; in fact, the last problem has been investigated only in [39], where a specific iterative method based on game theory has been developed. The need of employing iterative methods in MPF has been also explicitly recognized in [40], but no solution has been developed to meet it.

In this chapter, we first focus on the general problem of developing filtering algorithms that involve multiple interconnected Bayesian filters; these filters are run over distinct (but not necessarily disjoint) subspaces and can exploit iterative methods in their exchange of statistical information. The solution devised for this problem (and called *multiple Bayesian filtering*, MBF, since it represents a generalisation of the MPF approach) is based on previous work on the application of *factor graph theory* to the filtering and smoothing problems [1, 17]. More specifically, we show that: a) a *graphical model* can be developed for a network of Bayesian filters by combining multiple factor graphs, each referring to one of the involved filters; b) the pdfs computed by all these filters can be represented as messages passed on such a graphical model. This approach offers various important advantages. In fact, all the expressions of the passed messages can be derived by applying the same rule, namely the so called *sum-product algorithm* (SPA; also known as *belief propagation*) [17], [18], to the graphical model devised for the whole network. Moreover, iterative algorithms can be developed in a natural fashion once the cycles contained in this graphical model have been identified and the order according to which messages are passed on them (i.e., the *message scheduling*) has been established. The usefulness of our approach is exemplified by mainly illustrating its application to a network made of two Bayesian filters. More specifically, we investigate the interconnection of an extended Kalman filter with a particle filter, and develop two new filtering algorithms under the assumption that the considered SSM is *conditionally linear Gaussian* (CLG). Simulation results for two specific SSMs evidence that the devised algorithms perform similarly or better than RBPF and MPF, but require a smaller computational effort.

The remaining parts of this chapter are organized as follows. In Section 2.2, the filtering problem is analysed from a factor graph perspective for a network of multiple interconnected Bayesian filters. In Section 2.3, the tools illustrated in the previous section are applied to a network consisting of an extended Kalman filter interconnected with a particle filter, two new MBF algorithms are derived and their computational complexity is analysed in detail. The developed MBF algorithms are compared with EKF and RBPF, in terms of accuracy and execution time, in Section 2.4. Finally, some conclusions are offered in Section 2.5.

2.2 Graphical Modelling for Multiple Bayesian Filtering

In this paragraph, we illustrate some basic concepts about factor graphs and the computation of the messages passed over them. Then, we derive a graphical model for representing the overall processing accomplished by multiple interconnected Bayesian filters as a message passing on such a model.

2.2.1 Graphical Modelling for a Network of Bayesian Filters and Message Passing on it

In this chapter, we consider a discrete-time SSM whose $D-$dimensional *hidden state* at the k-th instant is denoted $\mathbf{x}_k \triangleq [x_{0,k}, x_{1,k}, ..., x_{D-1,k}]^T$, and whose *state update* and *measurement models* are expressed by

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k \tag{2.1}$$

and

$$\begin{aligned}\mathbf{y}_k &\triangleq [y_{0,k}, y_{1,k}, ..., y_{P-1,k}]^T \\ &= \mathbf{h}_k(\mathbf{x}_k) + \mathbf{e}_k,\end{aligned} \tag{2.2}$$

respectively. Here, $\mathbf{f}_k(\mathbf{x}_k)$ ($\mathbf{h}_k(\mathbf{x}_k)$) is a time-varying $D-$dimensional ($P-$dimensional) real function and $\mathbf{w}_k$ ($\mathbf{e}_k$) is the k-th element of the process (measurement) noise sequence $\{\mathbf{w}_k\}$ ($\{\mathbf{e}_k\}$); this sequence consists of $D-$dimensional ($P-$dimensional) *independent and identically distributed* (i.i.d.) Gaussian noise vectors, each characterized by a zero mean and a covariance matrix $\mathbf{C}_w$ ($\mathbf{C}_e$). Moreover, statistical independence between $\{\mathbf{e}_k\}$ and $\{\mathbf{w}_k\}$ is assumed for simplicity. Note that, from a statistical viewpoint, the SSM described by Eqs. (2.1)–(2.2) is characterized by the *Markov model* $f(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and the *observation model* $f(\mathbf{y}_k|\mathbf{x}_k)$ for any k.

In the following sections, we focus on the so-called *filtering problem*, which concerns the evaluation of the posterior pdf $f(\mathbf{x}_t|\mathbf{y}_{1:t})$ at an instant $t \geq 1$, given a) the initial pdf $f(\mathbf{x}_1)$ and b) the $t \cdot P$-dimensional measurement vector $\mathbf{y}_{1:t} = \left[\mathbf{y}_1^T, \mathbf{y}_2^T, ..., \mathbf{y}_t^T\right]^T$. It is well known that, if the pdf $f(\mathbf{x}_1)$ referring to the first observation instant is known, the computation of the posterior (i.e., filtered) pdf $f(\mathbf{x}_t|\mathbf{y}_{1:t})$ for $t \geq 1$ can be accomplished by means of an exact *Bayesian recursive procedure*, consisting of a *measurement update* step followed by a *time update* step. In [1, Sec. III], it is shown that, if this procedure is formulated with reference to the joint pdf $f(\mathbf{x}_t, \mathbf{y}_{1:t})$ (in place of the associated a posteriori pdf $f(\mathbf{x}_t|\mathbf{y}_{1:t})$), its k-th recursion (with $k = 1, 2, ..., t$) can be represented as a *forward only* message passing algorithm over the cycle free factor graph shown in Fig. 2.1. In the *measurement update*, the message $\vec{m}_{\text{fe}}(\mathbf{x}_k)$ going out of the equality node is computed as[1] (see Eq. (A.7))

$$\begin{aligned}\vec{m}_{\text{fe}}(\mathbf{x}_k) &= \vec{m}_{\text{fp}}(\mathbf{x}_k)\ \vec{m}_{\text{ms}}(\mathbf{x}_k) \\ &= f(\mathbf{x}_k, \mathbf{y}_{1:k}),\end{aligned} \tag{2.3}$$

where

$$\vec{m}_{\text{fp}}(\mathbf{x}_k) \triangleq f(\mathbf{x}_k, \mathbf{y}_{1:k-1}) \tag{2.4}$$

is the message feeding the considered graph and

$$\vec{m}_{\text{ms}}(\mathbf{x}_k) \triangleq f(\mathbf{y}_k|\mathbf{x}_k) \tag{2.5}$$

Note that the messages $\vec{m}_{\text{fp}}(\mathbf{x}_k)$ (2.4) and $\vec{m}_{\text{fe}}(\mathbf{x}_k)$ (2.3) convey the predicted pdf (i.e., the *forward prediction*) of $\mathbf{x}_k$ computed in the previous (i.e., in the $(k-1)$-th) recursion and the filtered pdf (i.e., the *forward estimate*) of $\mathbf{x}_k$ computed in the considered recursion, respectively, whereas the message $\vec{m}_{\text{ms}}(\mathbf{x}_k)$ (2.5) conveys the statistical information provided by the measurement $\mathbf{y}_k$ (2.2). In the *time update*, instead, the message that emerges from the function node referring to the pdf $f(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is evaluated as (see Eq. (A.8))

$$\int f(\mathbf{x}_{k+1}|\mathbf{x}_k)\ \vec{m}_{\text{fe}}(\mathbf{x}_k)\, d\mathbf{x}_k = f(\mathbf{x}_{k+1}, \mathbf{y}_{1:k}); \tag{2.6}$$

such a message is equal to $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1})$ (see Eq. (2.4)).

[1] In the following, the acronyms fp, fe, ms and pm are employed in the subscripts of various messages, so that readers can easily understand their meaning; in fact, the messages these acronyms refer to convey a *forward prediction*, a *forward estimate*, *measurement* information and *pseudo-measurement* information, respectively.

Let us take into consideration now a *network of N_F interconnected Bayesian filters*. In the following, we assume that:

a) All the filters of the network are fed by the same measurement vector (namely, $\mathbf{y}_k$ (2.2)), work in parallel and cooperate in order to estimate the state vector $\mathbf{x}_k$; in doing so, they can fully share their statistical information.

b) The i-th filter of the network (with $i = 1, 2, ..., N_F$), denoted F_i, works on a lower dimensional space and, in particular, estimates the portion $\mathbf{x}_k^{(i)}$ (having size D_i, with $D_i \leq D$) of the state vector $\mathbf{x}_k$; therefore, the substate $\bar{\mathbf{x}}_k^{(i)}$, representing the portion of $\mathbf{x}_k$ not included in $\mathbf{x}_k^{(i)}$, can be considered as a *nuisance* vector for F_i.

c) The set $\{\mathbf{x}_k^{(i)}\}$, collecting the substates estimated by all the filters of the network, covers $\mathbf{x}_k$, but does not necessarily represent a partition of it. In other words, some overlapping between the substates estimated by different filters is admitted. This means that the filtering algorithm running on the whole network may contain a form of *redundancy*, since one or more elements of the state vector can be independently estimated by different Bayesian filters.

We are interested in developing recursive filtering algorithms for the whole network of Bayesian filters. The approach we propose to solve this problem consists of the following three steps: S1) building N_F distinct factor graphs that allow us to represent the measurement and time updates accomplished by each filter of the network and its interactions with the other filters as message passing algorithms on them; S2) interconnecting the N_F factor graphs devised in the first step in order to generate a graphical model for the whole network; S3) deriving new filtering methods as message passing algorithms over the whole graphical model obtained in the second step.

Let us focus, now, on step S1. In developing a graphical model for filter F_i, the following considerations must be taken into account:

1) Since the portion $\bar{\mathbf{x}}_k^{(i)}$ of $\mathbf{x}_k$ is unknown to F_i (and, consequently, represents a *nuisance state*), an *estimate* of its pdf $f_k(\bar{\mathbf{x}}_k^{(i)})$ must be provided by the other filters of the network; this allows F_i to integrate out the dependence of its Markov model $f(\mathbf{x}_{k+1}^{(i)}|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)})$ and of its observation model $f(\mathbf{y}_k|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)})$ on $\bar{\mathbf{x}}_k^{(i)}$.

2) Filter F_i can benefit from the *pseudo-measurements* computed on the basis of the statistical information provided by the other filters of the network.

As far as the last point is concerned, it is worth pointing out that, in this chapter, any pseudo-measurement represents a *fictitious* measurement computed on the basis of the statistical information provided by a filtering algorithm different from the one benefiting from it; despite this, it can be processed as if it was a *real* measurement, provided that its statistical model is known. In practice, a pseudo-measurement $\mathbf{z}_k^{(i)}$ made available to the filter F_i is a P_i–dimensional random vector that, similarly as the real measurement $\mathbf{y}_k$ (2.2), can be modelled as[2]

$$\mathbf{z}_k^{(i)} = \tilde{\mathbf{h}}_k\big(\mathbf{x}_k^{(i)}\big) + \tilde{\mathbf{e}}_k^{(i)}, \tag{2.7}$$

where $\tilde{\mathbf{h}}_k\left(\mathbf{x}_k\right)$ is a time-varying P_i–dimensional function and $\tilde{\mathbf{e}}_k^{(i)}$ is a zero mean P_i-dimensional noise vector; if this notation is adopted, the pseudo-measurement information passed to the filter F_i is represented by pdf of $\mathbf{z}_k^{(i)}$ conditioned on $\mathbf{x}_k^{(i)}$. It is also worth mentioning that the evaluation of pseudo-measurements is often based on the *mathematical constraints* established by the Markov model of the considered SSM, as shown in the following section, where a specific network of filters is considered.

Based on the considerations illustrated above, the equations describing the measurement/time updates accomplished by F_i in the k-th recursion of the network can be formulated as follows. At the beginning of this recursion, F_i is fed by the forward prediction

$$\vec{m}_{\mathrm{fp}}\big(\mathbf{x}_k^{(i)}\big) = f(\mathbf{x}_k^{(i)}, \mathbf{y}_{1:k-1}), \tag{2.8}$$

[2]The possible dependence of the pseudo-measurement $\mathbf{z}_k^{(i)}$ (2.7) on the substate $\bar{\mathbf{x}}_k^{(i)}$ is ignored here, for simplicity.

originating from the previous recursion. In its first step (i.e., in its measurement update), it computes two filtered pdfs (i.e., two forward estimates), the first one based on the measurement $\mathbf{y}_k$ (2.2), the second one on the pseudo-measurement $\mathbf{z}_k^{(i)}$ (2.7). The first filtered pdf is evaluated as (see Eq. (2.3))

$$\vec{m}_{\mathrm{fe1}}\big(\mathbf{x}_k^{(i)}\big) = \vec{m}_{\mathrm{fp}}\big(\mathbf{x}_k^{(i)}\big)\, \vec{m}_{\mathrm{ms}}\big(\mathbf{x}_k^{(i)}\big), \tag{2.9}$$

where

$$m_{\mathrm{ms}}\big(\mathbf{x}_k^{(i)}\big) \triangleq \int f\big(\mathbf{y}_k|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)}\big)\, m_{\mathrm{mg1}}\big(\bar{\mathbf{x}}_k^{(i)}\big)\, d\bar{\mathbf{x}}_k^{(i)} \tag{2.10}$$

and $m_{\mathrm{mg1}}(\bar{\mathbf{x}}_k^{(i)})$ are the messages conveying *measurement* information and a filtered (or predicted) pdf of $\bar{\mathbf{x}}_k^{(i)}$ provided by the other filters, respectively. Similarly, the second filtered pdf is evaluated as (see Eq. (2.3))

$$\vec{m}_{\mathrm{fe2}}\big(\mathbf{x}_k^{(i)}\big) = \vec{m}_{\mathrm{fe1}}\big(\mathbf{x}_k^{(i)}\big)\, \vec{m}_{\mathrm{pm}}\big(\mathbf{x}_k^{(i)}\big), \tag{2.11}$$

where[3]

$$m_{\mathrm{pm}}\big(\mathbf{x}_k^{(i)}\big) \triangleq f\big(\mathbf{z}_k^{(i)}|\mathbf{x}_k^{(i)}\big) \tag{2.12}$$

is the message conveying *pseudo-measurement* information. Then, in its second step (i.e., in its time update), F_i computes the new forward prediction (see Eq. (2.6))

$$\vec{m}_{\mathrm{fp}}\big(\mathbf{x}_{k+1}^{(i)}\big) = \int\int f\big(\mathbf{x}_{k+1}^{(i)}|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)}\big)\, \vec{m}_{\mathrm{fe2}}\big(\mathbf{x}_k^{(i)}\big)\, m_{\mathrm{mg2}}\big(\bar{\mathbf{x}}_k^{(i)}\big)\, d\mathbf{x}_k\, d\bar{\mathbf{x}}_k^{(i)}, \tag{2.13}$$

where $m_{\mathrm{mg2}}(\bar{\mathbf{x}}_k^{(i)})$ has the same meaning as $m_{\mathrm{mg1}}(\bar{\mathbf{x}}_k^{(i)})$ (see Eq. (2.10)), but is not necessarily equal to it (since more refined information about $\bar{\mathbf{x}}_k^{(i)}$ could be made available by the other filters of the network after that the message $m_{\mathrm{ms}}(\mathbf{x}_k^{(i)})$ (2.10) has been computed).

Formulas (2.9)–(2.11) and (2.13) involve only products of pdfs and integrations of products; for this reason, their evaluation can be represented as a *forward only* message passing over the *cycle free* factor graph shown in Fig. 2.2. Note that, if this graph is compared with the one shown in Fig. 2.1, the following additional elements (identified by blue lines) are found:

1) Five equality nodes - Four of them allow to generate copies of the messages $\vec{m}_{\mathrm{fp}}(\mathbf{x}_k^{(i)})$, $\vec{m}_{\mathrm{fe1}}(\mathbf{x}_k^{(i)})$, $\vec{m}_{\mathrm{fe2}}(\mathbf{x}_k^{(i)})$ and $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(i)})$, to be shared with the other filters of the network, whereas the remaining one is involved in the second measurement update of F_i.

2) A block in which the predicted/filtered pdfs $\{\vec{m}_{\mathrm{fp}}(\mathbf{x}_k^{(l)})$, $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(l)})$, $\vec{m}_{\mathrm{fe}q}(\mathbf{x}_k^{(l)})$; $q = 1, 2$ and $l \neq i\}$ provided by the other filters of the network are processed - In this block, the messages $m_{\mathrm{mg}q}(\bar{\mathbf{x}}_k^{(i)})$ (with $q = 1$ and 2) and $m_{\mathrm{pm}}(\mathbf{x}_k^{(i)})$ are computed (see Eqs. (2.10), (2.12) and (2.13)); this block is connected to *oriented* edges only, that is to edges on which the flow of messages is unidirectional.

Given the graphical model represented in Fig. 2.2, step S2 can be accomplished by adopting the same conceptual approach as [1, Sec. III], where the factor graph on which RBPF and dual RBPF are based is devised by merging two sub-graphs, that refer to distinct substates. For this reason, a graphical model for the whole network of N_F Bayesian filters can be developed by interconnecting N_F distinct factor graphs, each structured like the one shown in that figure. For instance, if $N_F = 2$ is assumed for simplicity, this procedure results in the graphical model shown in Fig. 2.3. It is important to note that, in this case, if the substates $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$ estimated by F_1 and F_2, respectively, do not form a partition of the state vector $\mathbf{x}_k$, they share a portion of it; this consists of $N_{\mathrm{d}} \triangleq D_1 + D_2 - D$ state variables, that are separately estimated by the two Bayesian filters. The parameter N_{d} can be considered as the *degree of redundancy* characterizing the considered network of filters. The presence of redundancy in a filtering algorithm may result in an improvement of estimation accuracy and/or tracking capability; however, this is obtained

[3] If the pseudo-measurement $\mathbf{z}_k^{(i)}$ (2.7) depends also on $\bar{\mathbf{x}}_k^{(i)}$, marginalization with respect to this substate is required in the computation of the following message.

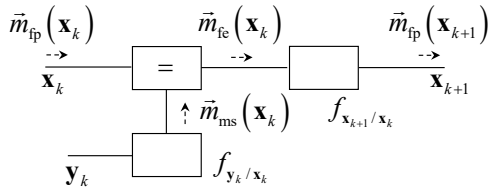


Figure 2.1: Message passing over the factor graph representing the k-th recursion of Bayesian filtering. An SSM characterized by the Markov model $f(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and the observation model $f(\mathbf{y}_k|\mathbf{x}_k)$ is considered.

at the price of an increased complexity with respect to the case in which F_1 and F_2 are run on disjoint substates.

Once the graphical model for the whole network has been developed, step S3 can be easily accomplished. In fact, recursive filtering algorithms for the considered network can be derived by systematically applying the SPA to its graphical model after that a proper *scheduling* has been established for the exchange of messages among its N_F Bayesian filters. Moreover, in developing a specific filtering algorithm to be run on a network of Bayesian filters, we must always keep in mind that:

1) Its k-th recursion is fed by the set of forward predictions $\{\vec{m}_{\text{fp}}(\mathbf{x}_k^{(i)}),\ i = 1,\ 2,\ ...,\ N_F\}$, and generates N_F couples of filtered densities $\{(\vec{m}_{\text{fe1}}(\mathbf{x}_k^{(i)}),\ \vec{m}_{\text{fe2}}(\mathbf{x}_k^{(i)})),\ i = 1,\ 2,\ ...,\ N_F\}$ and N_F new forward predictions $\{\vec{m}_{\text{fp}}(\mathbf{x}_{k+1}^{(i)}),\ i = 1,\ 2,\ ...,\ N_F\}$. Moreover, similarly as MPF, a *joint* filtered density for the whole state $\mathbf{x}_k$ is *unavailable* (unless the substate of one or more of the employed Bayesian filters coincides with $\mathbf{x}_k$) and multiple filtered/predicted pdfs are available for any substate shared by distinct filters.

2) Specific algorithms are needed to compute the pseudo-measurement and the nuisance substate pdfs in the $\{\mathrm{F}_l,\ l \neq i\}\rightarrow\mathrm{F}_i$ block appearing in Fig. 2.3. These algorithms depend on the considered SSM and on the selected message scheduling; for this reason, a general description of their structure cannot be provided.

3) The graphical model shown in Fig. 2.3, unlike the one illustrated in Fig. 2.1, is *not cycle free*; the presence of cycles is highlighted in the considered figure by showing the flow of messages along one of them. The presence of cycles raises the problems of a) identifying all the messages that can be *iteratively* refined and b) establishing the order according to which they are computed. Generally speaking, iterative message passing on the graphical model referring to a network of filters involves both the measurement updates and the time update accomplished by all the interconnected filters within each recursion of the network itself. In fact, this should allow each Bayesian filter to a) progressively refine the nuisance substate density employed in its measurement/time updates, and b) improve the quality of the pseudo-measurements exploited in its second measurement update. For this reason, if n_i iterations are run within each recursion of the whole network, the overall computational complexity associated with it is multiplied by n_i.

In the following section, a specific application of the general principles illustrated in this paragraph is analysed.

2.3 Filtering Algorithms Based on the Interconnection of an Extended Kalman Filter with a Particle Filter

In this section we focus on the development of two new filtering algorithms based on the interconnection of an extended Kalman filter with a particle filter. We first describe the graphical models on which these algorithms are based. Then, we provide a detailed description of the computed messages and their scheduling in a specific case. Finally, we provide a detailed analysis of the computational complexity of the devised algorithms.

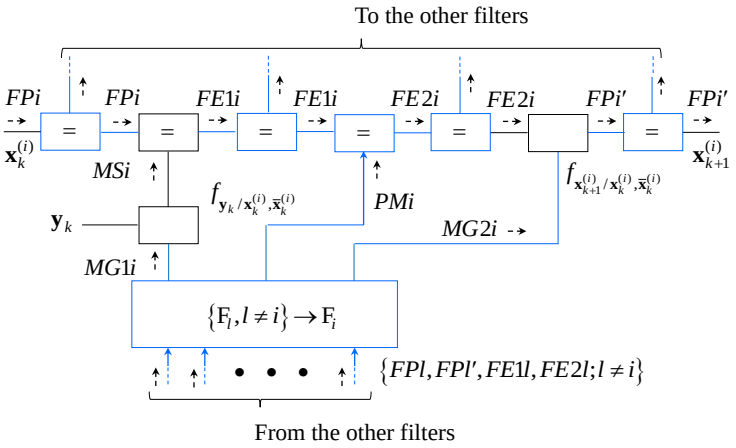


Figure 2.2: Message passing over the factor graph representing the couple of measurement updates and the time update accomplished by the i-th Bayesian filter in the k-th recursion of the network it belongs to. The messages $\vec{m}_{\mathrm{fp}}(\mathbf{x}_k^{(i)})$, $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(i)})$, $\vec{m}_{\mathrm{ms}}(\mathbf{x}_k^{(i)})$, $\vec{m}_{\mathrm{mg}p}(\bar{\mathbf{x}}_k^{(i)})$ (with $p = 1$ and 2), $\vec{m}_{\mathrm{pm}}(\mathbf{x}_k^{(i)})$ and $\vec{m}_{\mathrm{fe}q}(\mathbf{x}_k^{(i)})$ (with $q = 1$ and 2) are denoted FPi, FPi', MSi, $MGpi$, PMi and $FEqi$, respectively, to ease reading.

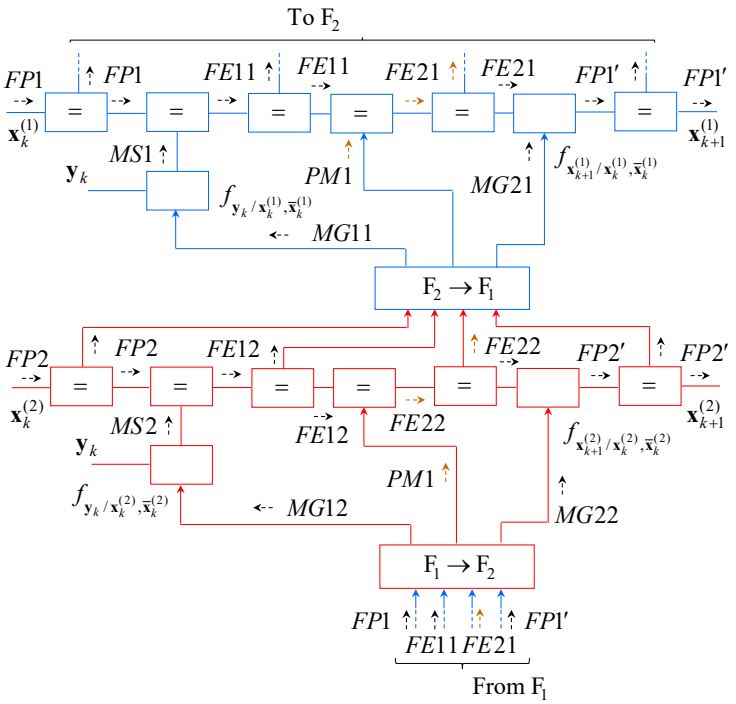


Figure 2.3: Graphical model based on the factor graph shown in Fig. 2.2 and referring to the interconnection of two Bayesian filters; the presence of a closed path (cycle) on which messages can be passed multiple times is highlighted by brown arrows.

2.3.1 Graphical Modelling

In this section, we develop new filtering algorithms for the class of CLG SSMs [6], [23], [1]; this allows us to partition the state vector at the k-th instant as $\mathbf{x}_k = [(\mathbf{x}_k^{(L)})^T, (\mathbf{x}_k^{(N)})^T]^T$, where $\mathbf{x}_k^{(L)} \triangleq [x_{0,k}^{(L)}, x_{1,k}^{(L)}, ..., x_{D_L-1,k}^{(L)}]^T$ ($\mathbf{x}_k^{(N)} \triangleq [x_{0,k}^{(N)}, x_{1,k}^{(N)}, ..., x_{D_N-1,k}^{(N)}]^T$) is its *linear* (*nonlinear*) *component* (with $D_N + D_L = D$). The devised algorithms rely on the following assumptions:

1) They involve two interconnected Bayesian filters, denoted F_1 and F_2.

2) Filter F_2 is a *particle filter*[4] employing N_p weighted particles and estimating the nonlinear state component only (so that $\mathbf{x}_k^{(2)} = \mathbf{x}_k^{(N)}$ and $\bar{\mathbf{x}}_k^{(2)} = \mathbf{x}_k^{(L)}$).

3) Filter F_1 is an *extended Kalman filter* and works on the *whole system state* or on the *linear state component* only. Consequently, in the first case (denoted **C.1** in the following), $\mathbf{x}_k^{(1)} = \mathbf{x}_k$ and $\bar{\mathbf{x}}_k^{(1)}$ is empty, and both the interconnected filters estimate the nonlinear state component (for this reason, the corresponding degree of redundancy is $N_\mathrm{d} = D_N$). In the second case (denoted **C.2** in the following), instead, $\mathbf{x}_k^{(1)} = \mathbf{x}_k^{(L)}$ and $\bar{\mathbf{x}}_k^{(1)} = \mathbf{x}_k^{(N)}$, and the two filters estimate disjoint substates (consequently, $N_\mathrm{d} = 0$).

This network configuration has been mainly inspired by RBPF. In fact, similarly as RBPF, the filtering techniques we develop are based on the idea of concatenating a local filtering method (EKF) with a global method (PF). However, unlike RBPF, a *single* extended Kalman filter is employed in place of a bank of Kalman filters. It is also worth remembering that, on the one hand, the use of a particle filter interconnected with an extended Kalman filter for tracking *disjoint* substates has been suggested in [41, Par. 3.2], where, however, no filtering algorithm based on this idea has been derived. On the other hand, a filtering scheme based on the interconnection of the same filters, but working on *partially overlapped substates*, has been derived in [46], where it has also been successfully applied to inertial navigation. Note also that, in the proposed filtering scheme, the two filters are aiding each other through the mutual exchange of statistical information. In fact, on the one hand, the particle filter helps the extended Kalman filter in its most difficult task, namely in the estimation of the nonlinear state component. On the other hand, the extended Kalman filter helps the particle filter by providing statistical information about the linear state component (required for marginalization) and a set of pseudo-measurements.

Based on the graphical model shown in Fig. 2.3, the factor graph illustrated in Fig. 2.4 can be drawn for case **C.1**. It is important to point out that:

1) Filter F_1 is based on *linearised* (and, consequently, *approximate*) Markov/measurement models of the considered SSM, whereas filter F_2 relies on *exact* models, as explained in more detail below.

2) Since the nuisance substate $\bar{\mathbf{x}}_k^{(1)}$ is empty, no marginalization is required in F_1; for this reason, the messages $\{\vec{m}_{\mathrm{mg}q}(\bar{\mathbf{x}}_k^{(1)});\ q = 1, 2\}$ (i.e., $MG11$ and $MG21$) visible in Fig. 2.3 do not appear in Fig. 2.4.

3) The new predicted pdf $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(2)}) = \vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(N)})$ and the second filtered pdf $\vec{m}_{\mathrm{fe2}}(\mathbf{x}_k^{(2)}) = \vec{m}_{\mathrm{fe2}}(\mathbf{x}_k^{(N)})$ computed by F_2 (i.e., the messages $FP2'$ and $FE22$, respectively) feed the $\mathrm{F}_2 \rightarrow \mathrm{F}_1$ block, where they are jointly processed to generate the pseudo-measurement message $\vec{m}_{\mathrm{pm}}(\mathbf{x}_k^{(1)}) = \vec{m}_{\mathrm{pm}}(\mathbf{x}_k)$ ($PM1$) made available to F_1. Similarly, as shown below, the computation of the pseudo-measurement message exploited by F_2 (i.e., of the message $\vec{m}_{\mathrm{pm}}(\mathbf{x}_k^{(2)}) = \vec{m}_{\mathrm{pm}}(\mathbf{x}_k^{(N)})$, $PM2$) requires the knowledge of a new predicted pdf that refers, however, to the *linear state component only*. In our graphical model, the computation of this prediction is accomplished by the $\mathrm{F}_1 \rightarrow \mathrm{F}_2$ block; this explains why the new predicted pdf $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(1)}) = \vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1})$ ($FP1'$) evaluated by F_1 and referring to the *whole state* of the considered SSM, does not feed the $\mathrm{F}_1 \rightarrow \mathrm{F}_2$ block.

4) Particle *resampling* with replacement has been included in the portion of the graphical model referring to filter F_2. This important task, accomplished after the second measurement update of this filter, does not emerge from the application of the SPA to our graphical model and ensures

[4] In particular, a *sequential importance resampling* filter is employed [2].

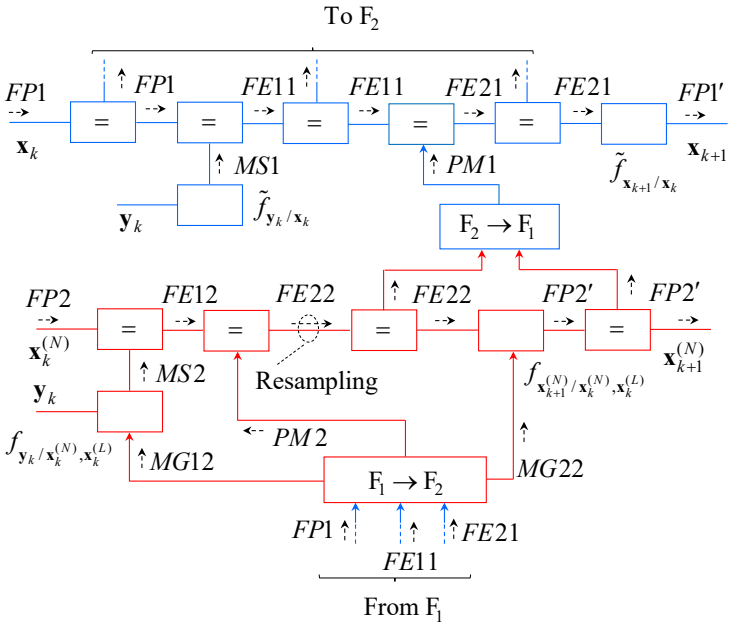


Figure 2.4: Graphical model based on the factor graph shown in Fig. 2.3 and referring to the interconnection of an extended Kalman filter (F_1) with a particle filter (F_2).

that the particles emerging from it are all equally likely. Note also that, because of the presence of particle resampling, two versions of the second filtered pdf $\vec{m}_{\mathrm{fe2}}(\mathbf{x}_k^{(2)}) = \vec{m}_{\mathrm{fe2}}(\mathbf{x}_k^{(N)})$ $(FE22)$ become available, one before resampling, the other one after it. As shown in the next paragraph, the second version of this message is exploited in the computation of the pseudo-measurement message $\vec{m}_{\mathrm{pm}}(\mathbf{x}_k^{(1)}) = \vec{m}_{\mathrm{pm}}(\mathbf{x}_k)$ $(PM1)$.

In the remaining part of this paragraph, we first provide various details about the filters F_1 and F_2, and the way pseudo-measurements are computed for each of them; then, we comment on how the factor graph shown in Fig. 2.4 should be modified if case **C.2** is considered.

Filter F_1 - Filter F_1 is based on the *linearised* versions of Eqs. (2.1) and (2.2), that is on the models (e.g., see [3, pp. 194–195])

$$\mathbf{x}_{k+1} = \mathbf{F}_k\,\mathbf{x}_k + \mathbf{u}_k + \mathbf{w}_k \tag{2.14}$$

and

$$\mathbf{y}_k = \mathbf{H}_k^T\,\mathbf{x}_k + \mathbf{v}_k + \mathbf{e}_k, \tag{2.15}$$

respectively; here, $\mathbf{F}_k \triangleq [\partial \mathbf{f}_k(\mathbf{x})/\partial \mathbf{x}]_{\mathbf{x}=\mathbf{x}_{\mathrm{fe},k}}$, $\mathbf{u}_k \triangleq \mathbf{f}_k(\mathbf{x}_{\mathrm{fe},k}) - \mathbf{F}_k\,\mathbf{x}_{\mathrm{fe},k}$, $\mathbf{H}_k^T \triangleq [\partial \mathbf{h}_k(\mathbf{x})/\partial \mathbf{x}]_{\mathbf{x}=\mathbf{x}_{\mathrm{fp},k}}$, $\mathbf{v}_k \triangleq \mathbf{h}_k(\mathbf{x}_{\mathrm{fp},k}) - \mathbf{H}_k^T\mathbf{x}_{\mathrm{fp},k}$ and $\mathbf{x}_{\mathrm{fp},k}$ $(\mathbf{x}_{\mathrm{fe},k})$ is the *forward prediction* (*forward estimate*) of $\mathbf{x}_k$ computed by F_1 in its $(k-1)$-th (k-th) recursion. Consequently, the approximate models

$$\tilde{f}(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{F}_k\,\mathbf{x}_k + \mathbf{u}_k, \mathbf{C}_w) \tag{2.16}$$

and

$$\tilde{f}(\mathbf{y}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{H}_k^T\,\mathbf{x}_k + \mathbf{v}_k, \mathbf{C}_e) \tag{2.17}$$

appear in the graphical model shown in Fig. 2.4.

Filter F_2 - In developing filter F_2, we assume that the portion of Eq. (2.1) referring to the nonlinear state component (i.e., the last D_N lines of the considered Markov model) and that the observation model (2.2) can be put in the form (e.g., see [1, eqs. (3)–(4)])

$$\mathbf{x}_{k+1}^{(N)} = \mathbf{A}_k^{(N)}(\mathbf{x}_k^{(N)})\,\mathbf{x}_k^{(L)} + \mathbf{f}_k^{(N)}(\mathbf{x}_k^{(N)}) + \mathbf{w}_k^{(N)} \tag{2.18}$$

and

$$\mathbf{y}_k = \mathbf{g}_k(\mathbf{x}_k^{(N)}) + \mathbf{B}_k(\mathbf{x}_k^{(N)})\,\mathbf{x}_k^{(L)} + \mathbf{e}_k, \tag{2.19}$$

respectively. In Eq. (2.18), $\mathbf{f}_k^{(N)}(\cdot)$ ($\mathbf{A}_k^{(N)}(\cdot)$) is a time-varying D_N-dimensional real function ($D_N \times D_L$ real matrix) and $\mathbf{w}_k^{(N)}$ consists of the last D_N elements of the noise term $\mathbf{w}_k$ appearing in Eq. (2.1) (the covariance matrix of $\mathbf{w}_k^{(N)}$ is denoted $\mathbf{C}_w^{(N)}$); moreover, in Eq. (2.19), $\mathbf{g}_k(\cdot)$ ($\mathbf{B}_k(\cdot)$) is a time-varying $P-$dimensional real function ($P \times D_L$ real matrix). This explains why filter F_2 is based on the exact pdfs

$$f\big(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)}\big)=\mathcal{N}\left(\mathbf{x}_k^{(N)};\mathbf{A}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big)\,\mathbf{x}_k^{(L)}+\mathbf{f}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big),\mathbf{C}_w^{(N)}\right) \tag{2.20}$$

and

$$f\big(\mathbf{y}_k|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)}\big)=\mathcal{N}\left(\mathbf{x}_k;\mathbf{g}_k\big(\mathbf{x}_k^{(N)}\big)+\mathbf{B}_k\big(\mathbf{x}_k^{(N)}\big)\,\mathbf{x}_k^{(L)},\mathbf{C}_e\right), \tag{2.21}$$

that appear in the graphical model shown in Fig. 2.4.

Computation of the pseudo-measurements for filter F_1 - Filter F_1 is fed by pseudo-measurement information about the *whole state* $\mathbf{x}_k$, that is about both the substates $\mathbf{x}_k^{(L)}$ and $\mathbf{x}_k^{(N)}$. On the one hand, N_p pseudo-measurements about the nonlinear state component are provided by the N_p particles contributing to the filtered pdf $\vec{m}_{\mathrm{fe2}}(\mathbf{x}_k^{(N)})$ $(FE22)$ available *after particle resampling*. On the other hand, N_p pseudo-measurements about the linear state component are evaluated by means of the same method employed by RBPF for this task. This method is based on the idea that the random vector (see [6, Par. II.D, p. 2283, eq. (24a)] and [1, Sec. III, p. 1524, eq. (9)])

$$\mathbf{z}_k^{(L)} \triangleq \mathbf{x}_{k+1}^{(N)}-\mathbf{f}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big), \tag{2.22}$$

depending on the *nonlinear state component only*, must equal the sum (see Eq. (2.18))

$$\mathbf{A}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big)\,\mathbf{x}_k^{(L)}+\mathbf{w}_k^{(N)}, \tag{2.23}$$

that depends on the *linear state component*. For this reason, N_p realizations of $\mathbf{z}_k^{(L)}$ (2.22) are computed in the $\mathrm{F}_2\rightarrow\mathrm{F}_1$ block on the basis of the messages $\vec{m}_{\mathrm{fe2}}(\mathbf{x}_k^{(N)})$ $(FE22)$ and $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(N)})$ $(FP2^{'})$ and are treated as measurements about $\mathbf{x}_k^{(L)}$.

Computation of the pseudo-measurements for filter F_2 - The messages feeding $\mathrm{F}_1\rightarrow\mathrm{F}_2$ block are employed for: a) generating a pdf of $\mathbf{x}_k^{(L)}$, so that the dependence of the state update and measurement models (i.e., of the densities $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)},\,\mathbf{x}_k^{(L)})$ (2.20) and $f(\mathbf{y}_k|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)})$ (2.21), respectively) on this substate can be integrated out; b) computing pseudo-measurement information about $\mathbf{x}_k^{(N)}$. As far as the last point is concerned, the approach we adopt is the same as that developed for *dual* RBPF in [1, Sec. V, pp. 1528–1529]. Such an approach relies on the Markov model

$$\mathbf{x}_{k+1}^{(L)}=\mathbf{A}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)\,\mathbf{x}_k^{(L)}+\mathbf{f}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)+\mathbf{w}_k^{(L)}, \tag{2.24}$$

referring to the *linear* state component [23], [1]; in the last expression, $\mathbf{f}_k^{(L)}(\mathbf{x}_k^{(N)})$ ($\mathbf{A}_k^{(L)}(\mathbf{x}_k^{(N)})$) is a time-varying D_L-dimensional real function ($D_L \times D_L$ real matrix), and $\mathbf{w}_k^{(N)}$ consists of the first D_L elements of the noise term $\mathbf{w}_k$ appearing in Eq. (2.1) (the covariance matrix of $\mathbf{w}_k^{(L)}$ is denoted $\mathbf{C}_w^{(L)}$, and independence between $\{\mathbf{w}_k^{(L)}\}$ and $\{\mathbf{w}_k^{(N)}\}$ is assumed for simplicity). From Eq. (2.24) it is easily inferred that the random vector

$$\mathbf{z}_k^{(N)} \triangleq \mathbf{x}_{k+1}^{(L)}-\mathbf{A}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)\,\mathbf{x}_k^{(L)}, \tag{2.25}$$

equals the sum

$$\mathbf{f}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)+\mathbf{w}_k^{(L)}, \tag{2.26}$$

that depends on $\mathbf{x}_k^{(N)}$ *only*; for this reason, $\mathbf{z}_k^{(N)}$ (2.25) can be interpreted as a pseudo-measurement about $\mathbf{x}_k^{(N)}$. In this case, the generation of pseudo-measurement information can be summarized as follows. First, N_p pdfs, one for each of the particles conveyed by the message $\vec{m}_{\mathrm{fe2}}(\mathbf{x}_k^{(N)})$ $(FE22)$,

are computed for the random vector $\mathbf{z}_k^{(N)}$ (2.25) by exploiting the statistical information about the linear state component made available by F_1. Then, each of these pdfs is *correlated* with the pdf obtained for $\mathbf{z}_k^{(N)}$ under the assumption that this vector is expressed by Eq. (2.26); this procedure results in a set of N_p particle weights, different from those computed on the basis of $\mathbf{y}_k$ (2.19) in the first measurement update of F_2.

A graphical model similar to the one shown in Fig. 2.4 can be easily derived from the general model appearing in Fig. 2.3 for case **C.2** too. The relevant differences with respect to case **C.1** can be summarized as follows:

1) Filters F_1 and F_2 estimate $\mathbf{x}_k^{(1)} = \mathbf{x}_k^{(L)}$ and $\mathbf{x}_k^{(2)} = \mathbf{x}_k^{(N)}$, respectively; consequently, their nuisance substates are $\bar{\mathbf{x}}_k^{(1)} = \mathbf{x}_k^{(N)}$ and $\bar{\mathbf{x}}_k^{(2)} = \mathbf{x}_k^{(L)}$, respectively.

2) The $F_2 \rightarrow F_1$ block is fed by the predicted/filtered pdfs computed by F_2; such pdfs are employed for: a) for providing F_1 with a pdf for $\mathbf{x}_k^{(N)}$, so that dependence of the Markov model (see Eq. (2.24))

$$f\left(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)}\right) = \mathcal{N}\left(\mathbf{x}_k^{(L)};\mathbf{A}_k^{(L)}\left(\mathbf{x}_k^{(N)}\right)\mathbf{x}_k^{(L)} + \mathbf{f}_k^{(L)}\left(\mathbf{x}_k^{(N)}\right),\mathbf{C}_w^{(L)}\right) \tag{2.27}$$

and of the measurement model $f(\mathbf{y}_k|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)})$ (2.21) on this substate can be integrated out; b) generating pseudo-measurement information about the substate $\mathbf{x}_k^{(L)}$ *only*. As far as point a) is concerned, it is also important to point out that the approximate model $\tilde{f}(\mathbf{y}_k|\mathbf{x}_k^{(L)})$ ($\tilde{f}(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(L)})$) on which F_1 is based can be derived from Eq. (2.21) (Eq. (2.27)) after setting $\mathbf{x}_k^{(N)} = \mathbf{x}_{\mathrm{fp},k}^{(N)}$ ($\mathbf{x}_k^{(N)} = \mathbf{x}_{\mathrm{fe},k}^{(N)}$); here, $\mathbf{x}_{\mathrm{fp},k}^{(N)}$ ($\mathbf{x}_{\mathrm{fe},k}^{(N)}$) denote the prediction (the estimate) of $\mathbf{x}_k^{(N)}$ evaluated on the basis of the message $\vec{m}_{\mathrm{fp}}(\mathbf{x}_k^{(N)})$ ($\vec{m}_{\mathrm{fe2}}(\mathbf{x}_k^{(N)})$) computed by F_2. Moreover, since Eqs. (2.19) and (2.24) exhibit a *linear* dependence on $\mathbf{x}_k^{(L)}$, F_1 becomes a standard Kalman filter.

The derivation of a specific filtering algorithm based on the graphical models described in this paragraph requires defining the scheduling of the messages passed on them and deriving mathematical expressions for such messages. These issues are investigated in detail in the following paragraph.

2.3.2 Message Scheduling and Computation

In this paragraph, a *recursive filtering technique*, called *double Bayesian filtering* (DBF) and based on the graphical model illustrated in Fig. 2.4, is developed. In each recursion of the DBF technique, F_1 is run before F_2; moreover, the presence of cycles in the graph on which it is based is accounted for by including a procedure for the iterative computation of the messages passed on them. Our description of the selected scheduling relies on Fig. 2.5, that refers to the k-th recursion and to the n-th iteration accomplished within this recursion (with $n = 1, 2, ..., n_i$, where n_i represents the overall number of iterations). It is important to point out that the following changes have been made in Fig. 2.5 with respect to Fig. 2.4:

1) A simpler notation has been adopted for the messages to ease reading. In particular, the symbols $FP2^{(n)}$, $FP2'^{(n)}$, q ($q^{(n)}$), qL ($qL^{(n)}$) and qN ($qN^{(n)}$) represent the messages $\vec{m}_{\mathrm{fp}}^{(n)}(\mathbf{x}_k^{(N)})$, $\vec{m}_{\mathrm{fp}}^{(n)}(\mathbf{x}_{k+1}^{(N)})$, $\vec{m}_q(\mathbf{x}_k)$ ($\vec{m}_q^{(n)}(\mathbf{x}_k)$), $\vec{m}_q(\mathbf{x}_k^{(L)})$ ($\vec{m}_q^{(n)}(\mathbf{x}_k^{(L)})$) and $\vec{m}_q(\mathbf{x}_k^{(N)})$ ($\vec{m}_q^{(n)}(\mathbf{x}_k^{(N)})$), respectively; moreover, the integer parameter n appearing in the superscript of some of them represents the iteration index.

2) Blue (red) arrows have been employed to identify Gaussian messages (messages in other forms).

3) The $F_1 \rightarrow F_2$ block is fed by the two filtered pdfs of $\mathbf{x}_k$ computed by F_1 (i.e., by the messages $\vec{m}_2(\mathbf{x}_k)$ and $\vec{m}_3^{(n)}(\mathbf{x}_k)$), but not by the predicted pdf $\vec{m}_{\mathrm{fp}}(\mathbf{x}_k)$, since the last message is useless.

3) The forward prediction $\vec{m}_{\mathrm{fp}}^{(n)}(\mathbf{x}_k^{(N)})$ feeding F_2 is involved in the proposed iterative procedure and may change from iteration to iteration because of resampling (in fact, this may lead to

discarding a portion of the particles conveyed by this message); for this reason, its dependence on the iteration index n has been explicitly indicated.

4) The same message (namely, $\vec{m}_1^{(n)}(\mathbf{x}_k^{(L)})$) is employed in F_2 for integrating out the dependence of the Markov model $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ (2.20) and of the measurement model $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ (2.21) on the linear component $\mathbf{x}_k^{(L)}$.

5) A memory cell (identified by the label 'D') has been added to store the last message evaluated in each iteration (i.e., the pseudo-measurement message $m_4^{(n)}(\mathbf{x}_k)$), so that it can be made available to F_1 at the beginning of the next iteration.

The DBF technique, at the beginning of its k-th recursion, is fed by the message

$$\vec{m}_{\text{fp}}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \eta_{\text{fp},k}, \mathbf{C}_{\text{fp},k}) \tag{2.28}$$

and

$$\vec{m}_{\text{fp}}\big(\mathbf{x}_k^{(N)}\big) = \sum_{j=1}^{N_p} \vec{m}_{\text{fp},j}\big(\mathbf{x}_k^{(N)}\big), \tag{2.29}$$

that corresponds to $FP2^{(1)}$ in Fig. 2.5; here,

$$\vec{m}_{\text{fp},j}\big(\mathbf{x}_k^{(N)}\big) = w_p\, \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)}\big) \tag{2.30}$$

is the j-th component of $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(N)})$, $\mathbf{x}_{k,j}^{(N)}$ is the j-th particle *predicted* in the previous (i.e., in the $(k-1)$-th) recursion and $w_p \triangleq 1/N_p$ is its weight. The DBF processes the messages $\vec{m}_{\text{fp}}(\mathbf{x}_k)$ (2.28) and $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(N)})$ (2.29), and the new measurement $\mathbf{y}_k$ (2.19), and generates: a) a couple of filtered densities for both $\mathbf{x}_k$ and $\mathbf{x}_k^{(N)}$; b) the output messages $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1})$ and $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1}^{(N)})$, having the *same functional form* as $\vec{m}_{\text{fp}}(\mathbf{x}_k)$ (2.28) and $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(N)})$ (2.29), respectively. The message passing accomplished to achieve these results can be divided in the three consecutive phases listed below.

I - In the first phase, filter F_1 accomplishes its first measurement update on the basis of the forward prediction $\vec{m}_{\text{fp}}(\mathbf{x}_k)$ and of the new measurement $\mathbf{y}_k$. This leads to the ordered computation of the messages $\vec{m}_1(\mathbf{x}_k)$ and $\vec{m}_2(\mathbf{x}_k)$.

II - In the second phase, an iterative procedure involving the first measurement update and the time update of F_2, and the computation of pseudo-measurements and their exploitation in the second measurement update of each filter is carried out. The n-th iteration of this procedure can be divided into six consecutive steps and leads to the ordered computation of the following messages: 1) $\vec{m}_3^{(n)}(\mathbf{x}_k)$, $\vec{m}_1^{(n)}(\mathbf{x}_k^{(L)})$; 2) $\vec{m}_{\text{fp}}^{(n)}(\mathbf{x}_k^{(N)})$, $\vec{m}_1^{(n)}(\mathbf{x}_k^{(N)})$, $\vec{m}_2^{(n)}(\mathbf{x}_k^{(N)})$; 3) $\vec{m}_3^{(n)}(\mathbf{x}_k^{(N)})$; 4) $\vec{m}_4^{(n)}(\mathbf{x}_k^{(N)})$; 5) $\vec{m}_{\text{fp}}^{(n)}(\mathbf{x}_{k+1}^{(N)})$; 6) $\vec{m}_4^{(n)}(\mathbf{x}_k)$.

III - In the third phase, the new predictions $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1})$ and $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1}^{(N)})$ are generated by F_1 and F_2, respectively. This involves the ordered computation of the following messages: $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1}^{(N)})$, $\vec{m}_3^{(n_i+1)}(\mathbf{x}_k)$ and $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1})$.

In the remaining part of this paragraph, the expressions of all the messages computed in each of the three phases described above are provided; the derivation of these expressions is sketched in Appendix C.

Phase I - In this phase, the forward prediction $\vec{m}_{\text{fp}}(\mathbf{x}_k)$ (2.28) feeding filter F_1 is merged with the message

$$\vec{m}_1(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \eta_{1,k}, \mathbf{C}_{1,k}), \tag{2.31}$$

conveying measurement information; the covariance matrix $\mathbf{C}_{1,k}$ and the mean vector $\eta_{1,k}$ of the last message are evaluated on the basis of the associated precision matrix

$$\mathbf{W}_{1,k} \triangleq (\mathbf{C}_{1,k})^{-1} = \mathbf{H}_k \mathbf{W}_e \mathbf{H}_k^T, \tag{2.32}$$

and of the associated transformed mean vector

$$\mathbf{w}_{1,k} \triangleq \mathbf{W}_{1,k}\, \eta_{1,k} = \mathbf{H}_k \mathbf{W}_e\, (\mathbf{y}_k - \mathbf{v}_k), \tag{2.33}$$

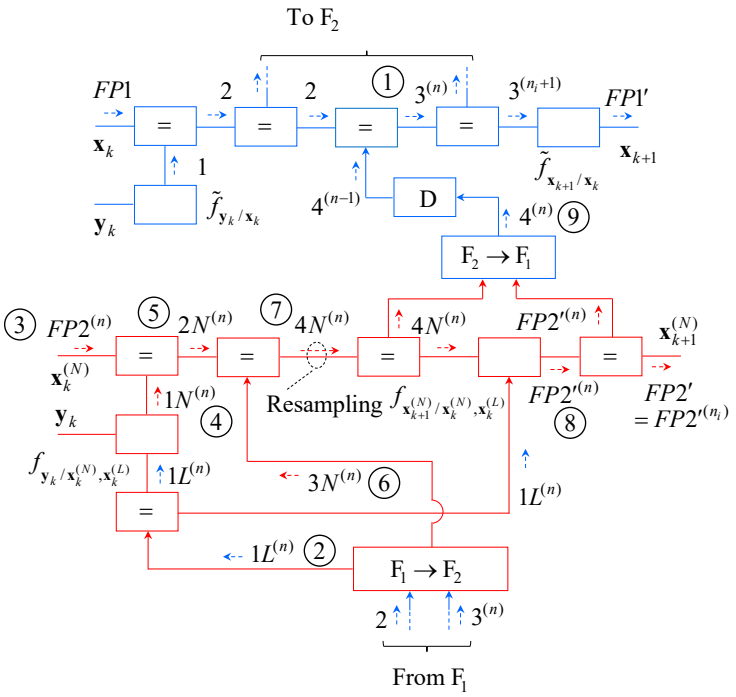


Figure 2.5: Message scheduling adopted in the k-th recursion of the DBF technique. The circled integers 1–9 specify the order according to which nine distinct messages are computed in the n-th iteration of phase II.

respectively, with $\mathbf{W}_e \triangleq \mathbf{C}_e^{-1}$. This results in the *first filtered pdf* (see Fig. 2.5)

$$\vec{m}_2(\mathbf{x}_k) = \vec{m}_{\text{fp}}(\mathbf{x}_k)\, \vec{m}_1(\mathbf{x}_k) \tag{2.34}$$

$$= \mathcal{N}(\mathbf{x}_k; \eta_{2,k}, \mathbf{C}_{2,k}) \tag{2.35}$$

computed by filter F_1; here, the covariance matrix $\mathbf{C}_{2,k}$ and the mean vector $\eta_{2,k}$ are evaluated on the basis of the associated precision matrix

$$\mathbf{W}_{2,k} \triangleq (\mathbf{C}_{2,k})^{-1} = \mathbf{W}_{\text{fp},k} + \mathbf{W}_{1,k}, \tag{2.36}$$

and of the associated transformed mean vector

$$\mathbf{w}_{2,k} \triangleq \mathbf{W}_{2,k}\, \eta_{2,k} = \mathbf{w}_{\text{fp},k} + \mathbf{w}_{1,k}, \tag{2.37}$$

respectively, $\mathbf{W}_{\text{fp},k} \triangleq (\mathbf{C}_{\text{fp},k})^{-1}$ and $\mathbf{w}_{\text{fp},k} \triangleq \mathbf{W}_{\text{fp},k}\, \eta_{\text{fp},k}$.

Phase II - A short description of the six steps accomplished in the n-th iteration of this phase is provided in the following. As shown below, the elements of the particle set processed by F_2 can change from iteration to iteration, even if its cardinality remains the same. In the following, the particle set available *at the beginning* of the n-th iteration is denoted $S_k[n] = \{\mathbf{x}_{k,j}^{(N)}[n];\ j = 1, 2, ..., N_p\}$; note that the initial particle set is $S_k[1] \triangleq \{\mathbf{x}_{k,j}^{(N)},\ j = 1, 2, ..., N_p\}$ (i.e., $\mathbf{x}_{k,j}^{(N)}[1] = \mathbf{x}_{k,j}^{(N)}$ for any j) and collects the N_p *predicted particles* conveyed by the message $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(N)})$ (2.29).

1) *Second measurement update in* F_1 - The *second filtered pdf* (see Fig. 2.5)

$$\vec{m}_3^{(n)}(\mathbf{x}_k) = \vec{m}_2(\mathbf{x}_k)\, \vec{m}_4^{(n-1)}(\mathbf{x}_k) \tag{2.38}$$

$$= \mathcal{N}\big(\mathbf{x}_k; \eta_{3,k}^{(n)}, \mathbf{C}_{3,k}^{(n)}\big) \tag{2.39}$$

is computed by F_1 in order to exploit the pseudo-measurement message $\vec{m}_4^{(n-1)}(\mathbf{x}_k)$ (evaluated in the previous iteration); since $\vec{m}_4^{(n-1)}(\mathbf{x}_k) = 1$ for $n = 1$ (note that F_1 cannot benefit from

pseudo-measurement information at the beginning of the first iteration) and $\vec{m}_4^{(n-1)}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \eta_{4,k}^{(n-1)}, \mathbf{C}_{4,k}^{(n-1)})$ for $n > 1$ (see Eq. (2.67)), it easy to show that

$$\mathbf{C}_{3,k}^{(n)} = \mathbf{C}_{2,k} \tag{2.40}$$

and

$$\eta_{3,k}^{(n)} = \eta_{2,k} \tag{2.41}$$

for $n = 1$, whereas

$$\mathbf{C}_{3,k}^{(n)} = \mathbf{W}_k^{(n-1)} \mathbf{C}_{4,k}^{(n-1)}, \tag{2.42}$$

and

$$\eta_{3,k}^{(n)} = \mathbf{W}_k^{(n-1)} \left[\mathbf{C}_{4,k}^{(n-1)} \mathbf{w}_{2,k} + \eta_{4,k}^{(n-1)} \right] \tag{2.43}$$

for $n > 1$; here, $\mathbf{W}_k^{(n-1)} \triangleq [\mathbf{C}_{4,k}^{(n-1)} \mathbf{W}_{2,k} + \mathbf{I}_D]^{-1}$. Then, the message $\vec{m}_3^{(n)}(\mathbf{x}_k)$ (2.39) is marginalized with respect to $\mathbf{x}_k^{(N)}$ in the $\mathrm{F}_1 \rightarrow \mathrm{F}_2$ block; this results in the message

$$\begin{aligned} \vec{m}_1^{(n)}\big(\mathbf{x}_k^{(L)}\big) &\triangleq \int \vec{m}_3^{(n)}(\mathbf{x}_k)\, d\mathbf{x}_k^{(N)} \\ &= \mathcal{N}(\mathbf{x}_k^{(L)}; \tilde{\eta}_{1,k}^{(n)}, \tilde{\mathbf{C}}_{1,k}^{(n)}), \end{aligned} \tag{2.44}$$

where $\tilde{\mathbf{C}}_{1,k}^{(n)}$ and $\tilde{\eta}_{1,k}^{(n)}$ are easily extracted from $\mathbf{C}_{3,k}^{(n)}$ (2.42) and $\eta_{3,k}^{(n)}$ (2.43) for $n > 1$ ($\mathbf{C}_{3,k}^{(n)}$ (2.40) and $\eta_{3,k}^{(n)}$ (2.41) for $n = 1$), respectively, since $\mathbf{x}_k^{(L)}$ consists of the first D_L elements of $\mathbf{x}_k$.

2) *First measurement update in* F_2 - This step concerns the computation of the message (see Fig. 2.5)

$$\vec{m}_2^{(n)}\big(\mathbf{x}_k^{(N)}\big) = \vec{m}_{\mathrm{fp}}^{(n)}\big(\mathbf{x}_k^{(N)}\big)\, \vec{m}_1^{(n)}\big(\mathbf{x}_k^{(N)}\big), \tag{2.45}$$

that represents the *first filtered pdf* computed by F_2. The message $\vec{m}_{\mathrm{fp}}^{(n)}(\mathbf{x}_k^{(N)})$ conveys a set of predicted particles; its j-th component is given by

$$\vec{m}_{\mathrm{fp},j}^{(n)}\big(\mathbf{x}_k^{(N)}\big) = w_p\, \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)}[n]\big) \tag{2.46}$$

and, consequently, coincides with $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(N)})$ (2.30) for $n = 1$ only; note also that the same weight is assigned to all the messages $\{\vec{m}_{\mathrm{fp},j}^{(n)}(\mathbf{x}_k^{(N)})\}$ for any n, since particle resampling is employed in each iteration of this phase (see step 4)). The message (see Fig. 2.5)

$$\vec{m}_1^{(n)}(\mathbf{x}_k^{(N)}) = \int f(\mathbf{y}_k | \mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})\, \vec{m}_1^{(n)}(\mathbf{x}_k^{(L)})\, d\mathbf{x}_k^{(L)}, \tag{2.47}$$

instead, conveys *measurement information*, that is the information about $\mathbf{x}_k^{(N)}$ provided by $\mathbf{y}_k$ (2.19). In particular, the value

$$w_{1,k,j}^{(n)} = \mathcal{N}\big(\mathbf{y}_k; \tilde{\eta}_{1,k,j}^{(n)}, \tilde{\mathbf{C}}_{1,k,j}^{(n)}\big) \tag{2.48}$$

taken on by the message $\vec{m}_1^{(n)}(\mathbf{x}_k^{(N)})$ (2.47) for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}[n]$ represents the measurement-based *weight* assigned to the j-th particle $\mathbf{x}_{k,j}^{(N)}[n]$; here,

$$\tilde{\eta}_{1,k,j}^{(n)} = \mathbf{B}_{k,j}[n]\, \tilde{\eta}_{1,k}^{(n)} + \mathbf{g}_{k,j}[n], \tag{2.49}$$

$$\tilde{\mathbf{C}}_{1,k,j}^{(n)} = \mathbf{B}_{k,j}[n]\, \tilde{\mathbf{C}}_{1,k}^{(n)}\, (\mathbf{B}_{k,j}[n])^T + \mathbf{C}_e, \tag{2.50}$$

$\mathbf{g}_{k,j}[n] \triangleq \mathbf{g}_k(\mathbf{x}_{k,j}^{(N)}[n])$ and $\mathbf{B}_{k,j}[n] \triangleq \mathbf{B}_k(\mathbf{x}_{k,j}^{(N)}[n])$. From (2.45), (2.46) and (2.48) it is easily inferred that $\vec{m}_2^{(n)}(\mathbf{x}_k^{(N)})$ (2.45) conveys the same set of particles as $\vec{m}_{\mathrm{fp}}^{(n)}(\mathbf{x}_k^{(N)})$ and that its j-th component is

$$\vec{m}_{2,j}^{(n)}\big(\mathbf{x}_k^{(N)}\big) = w_p\, w_{1,k,j}^{(n)}\, \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)}[n]\big). \tag{2.51}$$

3) *Computation of the pseudo-measurements for* F_2 - This step is accomplished in the $F_1 \rightarrow F_2$ block and aims at computing the message $\vec{m}_3^{(n)}(\mathbf{x}_k^{(N)})$; this conveys the statistical information about $\mathbf{x}_k^{(N)}$ that originates from the pseudo-measurement $\mathbf{z}_k^{(N)}$ (2.25) (further details about this message and its meaning are provided in Appendix C). Actually, what is really required in the next step is the value taken on by this message for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}[n]$ (with $j = 1, 2, ..., N_p$), because of the Dirac delta function conveyed by the message $\vec{m}_{2,j}^{(n)}(\mathbf{x}_k^{(N)})$ (2.51) and appearing in the *right-hand side* (RHS) of Eq. (2.58); such a value is

$$w_{3,k,j}^{(n)} = \breve{D}_{k,j}^{(n)} \cdot \exp\left[\frac{1}{2}\left(\left(\breve{\eta}_{3,k,j}^{(n)}\right)^T \breve{\mathbf{W}}_{3,k,j}^{(n)} \breve{\eta}_{3,k,j}^{(n)} - \left(\breve{\eta}_{z,k,j}^{(n)}\right)^T \cdot \breve{\mathbf{W}}_{z,k,j}^{(n)} \breve{\eta}_{z,k,j}^{(n)} - \left(\mathbf{f}_{k,j}^{(L)}[n]\right)^T \mathbf{W}_w^{(L)} \mathbf{f}_{k,j}^{(L)}[n]\right)\right], \tag{2.52}$$

and represents a new weight to be assigned to $\mathbf{x}_{k,j}^{(N)}[n]$, that is to the j-th particle of the set $S_k[n]$; here,

$$\breve{\mathbf{W}}_{3,k,j}^{(n)} \triangleq \left(\breve{\mathbf{C}}_{3,k,j}^{(n)}\right)^{-1} = \breve{\mathbf{W}}_{z,k,j}^{(n)} + \mathbf{W}_w^{(L)}, \tag{2.53}$$

$$\breve{\mathbf{w}}_{3,k,j}^{(n)} \triangleq \breve{\mathbf{W}}_{3,k,j}^{(n)} \breve{\eta}_{3,k,j}^{(n)} = \breve{\mathbf{w}}_{z,k,j}^{(n)} + \mathbf{W}_w^{(L)} \mathbf{f}_{k,j}^{(L)}[n], \tag{2.54}$$

$\mathbf{W}_w^{(L)} \triangleq [\mathbf{C}_w^{(L)}]^{-1}$, $\mathbf{f}_{k,j}^{(L)}[n] \triangleq \mathbf{f}_k^{(L)}(\mathbf{x}_{k,j}^{(N)}[n])$, $\breve{\mathbf{W}}_{z,k,j}^{(n)} \triangleq (\breve{\mathbf{C}}_{z,k,j}^{(n)})^{-1}$, $\breve{\mathbf{w}}_{z,k,j}^{(n)} \triangleq \breve{\mathbf{W}}_{z,k,j}^{(n)} \breve{\eta}_{z,k,j}^{(n)}$,

$$\breve{\mathbf{C}}_{z,k,j}^{(n)} = \mathbf{C}_w^{(L)} + \mathbf{A}_{k,j}^{(L)}[n] \left[\tilde{\mathbf{C}}_{3,k}^{(n)} - \tilde{\mathbf{C}}_{2,k}\right] \left(\mathbf{A}_{k,j}^{(L)}[n]\right)^T, \tag{2.55}$$

$$\breve{\eta}_{z,k,j}^{(n)} = \mathbf{A}_{k,j}^{(L)}[n] \left[\tilde{\eta}_{3,k}^{(n)} - \tilde{\eta}_{2,k}\right] + \mathbf{f}_{k,j}^{(L)}[n], \tag{2.56}$$

$\mathbf{A}_{k,j}^{(L)}[n] \triangleq \mathbf{A}_k^{(L)}(\mathbf{x}_{k,j}^{(N)}[n])$, $\breve{D}_{k,j}^{(n)} \triangleq [\det(\breve{\mathbf{C}}_{k,j}^{(n)})]^{-1/2}$, $\breve{\mathbf{C}}_{k,j}^{(n)} \triangleq \breve{\mathbf{C}}_{z,k,j}^{(n)} + \mathbf{C}_w^{(L)}$, and $\tilde{\eta}_{2,k}$ and $\tilde{\mathbf{C}}_{2,k}$ ($\tilde{\eta}_{3,k}^{(n)}$ and $\tilde{\mathbf{C}}_{3,k}^{(n)}$) are extracted from $\eta_{2,k}$ and $\mathbf{C}_{2,k}$ ($\eta_{3,k}^{(n)}$ and $\mathbf{C}_{3,k}^{(n)}$), respectively (see Eqs. (2.35) and (2.39)), since they refer to the first D_L elements of $\mathbf{x}_k$.

4) *Second measurement update in* F_2 - In this step, the weights of the particles forming the set $S_k[n]$ are updated on the basis of the weights $\{w_{3,k,j}^{(n)}\}$ computed in the previous step (see Eq. (2.52)). The new weight for the j-th particle $\mathbf{x}_{k,j}^{(N)}[n]$ is computed as

$$w_{4,k,j}^{(n)} \triangleq w_p \cdot w_{1,k,j}^{(n)} \cdot w_{3,k,j}^{(n)} \tag{2.57}$$

and combines the initial weight w_p (originating from $\vec{m}_{\mathrm{fp},j}^{(n)}(\mathbf{x}_k^{(N)})$ (2.46)) with the weights $w_{1,k,j}^{(n)}$ (2.48) and $w_{3,k,j}^{(n)}$ (2.52) related to the measurement $\mathbf{y}_k$ (2.19) and the pseudo-measurement $\mathbf{z}_k^{(N)}$ (2.25), respectively. Note also that the weight $w_{4,k,j}^{(n)}$ (2.57) is conveyed by the message (see Fig. 2.5)

$$\vec{m}_{4,j}^{(n)}\left(\mathbf{x}_k^{(N)}\right) = \vec{m}_{2,j}^{(n)}\left(\mathbf{x}_k^{(N)}\right) \vec{m}_3^{(n)}\left(\mathbf{x}_k^{(N)}\right) \tag{2.58}$$

$$= w_{4,k,j}^{(n)}\, \delta\left(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)}[n]\right), \tag{2.59}$$

that represents the j-th component of the message $\vec{m}_4^{(n)}(\mathbf{x}_k^{(N)})$ (with $j = 1, 2, ..., N_p$).

Once all the weights $\{w_{4,k,j}^{(n)}\}$ are available, their normalization is accomplished; this produces the normalized weights

$$W_{4,k,j}^{(n)} \triangleq C_k^{(n)}\, w_{4,k,j}^{(n)}, \tag{2.60}$$

where $C_k^{(n)} \triangleq 1/\sum_{j=1}^{N_p} w_{4,k,j}^{(n)}$. The particles $\{\mathbf{x}_{k,j}^{(N)}[n]\}$ and their weights $\{W_{4,k,j}^{(n)}\}$ represent the *second filtered pdf* of $\mathbf{x}_k^{(N)}$ computed by F_2 in the n-th iteration of the considered recursion; consequently, the *final* filtered pdf evaluated by F_2 is represented by the particles $\{\mathbf{x}_{k,j}^{(N)}[n_i]\}$ and their weights $\{W_{4,k,j}^{(n_i)}\}$ computed in the last iteration.

Resampling with replacement is now accomplished for the particle set $S_k[n]$ on the basis of the new weights $\{W_{4,k,j}^{(n)}\}$ (see Eq. (2.60)). This entails that the N_p particles $\{\mathbf{x}_{k,j}^{(N)}[n]\}$ and their associated weights $\{W_{4,k,j}^{(n)}\}$ are replaced by the new particles $\{\mathbf{x}_{k,j}^{(N)}[n+1]\}$, forming the set $S_k[n+1]$ and having identical weights (all equal to $w_p \triangleq 1/N_p$). Consequently, the effect of resampling can be represented as turning the message $\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(N)})$ (2.59) into the message

$$\vec{m}_{4,j}^{(n)}\big(\mathbf{x}_k^{(N)}\big) = w_p\,\delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)}[n+1]\big), \tag{2.61}$$

with $j = 1, 2, .., N_p$.

5) *Time update in* F_2 - In this step, the message $\vec{m}_{\mathrm{fp}}^{(n)}(\mathbf{x}_{k+1}^{(N)})$, conveying the predicted pdf of $\mathbf{x}_{k+1}^{(N)}$, is computed using the same method as RBPF (e.g., see [1, Sec. IV, p. 1526]). For this reason, for any j, the pdf (see Fig. 2.5)

$$\int\int f\big(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(L)}, \mathbf{x}_k^{(N)}\big)\cdot \vec{m}_{4,j}^{(n)}\big(\mathbf{x}_k^{(N)}\big)\,\vec{m}_1^{(n)}\big(\mathbf{x}_k^{(L)}\big)\,d\mathbf{x}_k^{(L)}\,d\mathbf{x}_k^{(N)} \tag{2.62}$$

$$= \mathcal{N}\big(\mathbf{x}_{k+1}^{(N)}; \eta_{3,k,j}^{(N)}, \mathbf{C}_{3,k,j}^{(N)}\big), \tag{2.63}$$

representing a prediction of $\mathbf{x}_{k+1}^{(N)}$ conditioned on $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}[n+1]$ is computed first; here,

$$\eta_{3,k,j}^{(N)} \triangleq \mathbf{A}_{k,j}^{(N)}\,[n+1]\;\tilde{\eta}_{1,k}^{(n)} + \mathbf{f}_{k,j}^{(N)}\,[n+1]\,, \tag{2.64}$$

$$\mathbf{C}_{3,k,j}^{(N)} \triangleq \mathbf{A}_{k,j}^{(N)}\,[n+1]\;\tilde{\mathbf{C}}_{1,k}^{(n)}\big(\mathbf{A}_{k,j}^{(N)}\,[n+1]\big)^T + \mathbf{C}_w^{(N)}, \tag{2.65}$$

$\mathbf{A}_{k,j}^{(N)}[n+1] \triangleq \mathbf{A}_k^{(N)}(\mathbf{x}_{k,j}^{(N)}[n+1])$ and $\mathbf{f}_{k,j}^{(N)}[n+1] \triangleq \mathbf{f}_k^{(N)}(\mathbf{x}_{k,j}^{(N)}[n+1])$. Then, the sample $\bar{\mathbf{x}}_{k+1,j}^{(N)}\,[n+1]$ is drawn from the Gaussian function (2.63) and the weight w_p is assigned to it; these information are conveyed by the j-th component

$$\vec{m}_{\mathrm{fp},j}^{(n)}\big(\mathbf{x}_{k+1}^{(N)}\big) = w_p\,\delta\big(\mathbf{x}_{k+1}^{(N)} - \bar{\mathbf{x}}_{k+1,j}^{(N)}\,[n+1]\big), \tag{2.66}$$

of the message $\vec{m}_{\mathrm{fp}}^{(n)}(\mathbf{x}_{k+1}^{(N)})$.

6) *Computation of the pseudo-measurements for* F_1 - This step is accomplished in the $\mathrm{F}_2 \rightarrow \mathrm{F}_1$ block and aims at generating the message (see Fig. 2.5)

$$\vec{m}_4^{(n)}\,(\mathbf{x}_k) = \mathcal{N}\big(\mathbf{x}_k; \eta_{4,k}^{(n)}, \mathbf{C}_{4,k}^{(n)}\big), \tag{2.67}$$

that conveys the pseudo-measurement information exploited by F_1 in its *second measurement update* of the next iteration. The mean vector $\eta_{4,k}^{(n)}$ is evaluated as

$$\eta_{4,k}^{(n)} = \Big[\big(\tilde{\eta}_{4,k}^{(n)}\big)^T, \big(\check{\eta}_{4,k}^{(n)}\big)^T\Big]^T, \tag{2.68}$$

where

$$\tilde{\eta}_{4,k}^{(n)} \triangleq \frac{1}{N_p}\sum_{j=1}^{N_p}\tilde{\eta}_{4,k,j}^{(n)} \tag{2.69}$$

and

$$\check{\eta}_{4,k}^{(n)} \triangleq \frac{1}{N_p}\sum_{j=1}^{N_p}\mathbf{x}_{k,j}^{(N)}[n] \tag{2.70}$$

are a D_L-dimensional mean vector and a D_N-dimensional mean vector, respectively. The covariance matrix $\mathbf{C}_{4,k}^{(n)}$, instead, is computed as

$$\mathbf{C}_{4,k}^{(n)} = \begin{bmatrix} \tilde{\mathbf{C}}_{4,k}^{(n)} & \dot{\mathbf{C}}_{4,k}^{(n)} \\ \big(\dot{\mathbf{C}}_{4,k}^{(n)}\big)^T & \check{\mathbf{C}}_{4,k}^{(n)} \end{bmatrix}, \tag{2.71}$$

where

$$\tilde{\mathbf{C}}_{4,k}^{(n)} \triangleq \frac{1}{N_p}\sum_{j=1}^{N_p}\tilde{\mathbf{r}}_{4,k,j}^{(n)} - \tilde{\eta}_{4,k}^{(n)}\left(\tilde{\eta}_{4,k}^{(n)}\right)^T, \tag{2.72}$$

is a $D_L \times D_L$ covariance matrix,

$$\check{\mathbf{C}}_{4,k}^{(n)} \triangleq \frac{1}{N_p}\sum_{j=1}^{N_p}\check{\mathbf{r}}_{4,k,j}^{(n)} - \check{\eta}_{4,k}^{(n)}\left(\check{\eta}_{4,k}^{(n)}\right)^T, \tag{2.73}$$

is a $D_N \times D_N$ covariance matrix and

$$\dot{\mathbf{C}}_{4,k}^{(n)} \triangleq \frac{1}{N_p}\sum_{j=1}^{N_p}\dot{\mathbf{r}}_{4,k,j}^{(n)} - \tilde{\eta}_{4,k}^{(n)}\left(\check{\eta}_{4,k}^{(n)}\right)^T, \tag{2.74}$$

is $D_L \times D_N$ cross-covariance matrix. Moreover, $\tilde{\mathbf{r}}_{4,k,j}^{(n)} \triangleq \tilde{\mathbf{C}}_{4,k,j}^{(n)} + \tilde{\eta}_{4,k,j}^{(n)}(\tilde{\eta}_{4,k,j}^{(n)})^T$, $\check{\mathbf{r}}_{4,k,j}^{(n)} \triangleq \mathbf{x}_{k,j}^{(N)}[n+1](\mathbf{x}_{k,j}^{(N)}[n+1])^T$ $\dot{\mathbf{r}}_{4,k,j}^{(n)} \triangleq \tilde{\eta}_{4,k,j}^{(n)}(\mathbf{x}_{k,j}^{(N)}[n+1])^T$, the covariance matrix $\tilde{\mathbf{C}}_{4,k,j}^{(n)}$ and the mean vector $\tilde{\eta}_{4,k,j}^{(n)}$ are computed on the basis of the associated precision matrix

$$\tilde{\mathbf{W}}_{4,k,j}^{(n)} \triangleq \left(\tilde{\mathbf{C}}_{4,k,j}^{(n)}\right)^{-1} = \left(\mathbf{A}_{k,j}^{(N)}[n+1]\right)^T \mathbf{W}_w^{(N)} \mathbf{A}_{k,j}^{(N)}[n+1] \tag{2.75}$$

and of the associated transformed mean vector

$$\tilde{\mathbf{w}}_{4,k,j}^{(n)} \triangleq \tilde{\mathbf{W}}_{4,k,j}^{(n)}\tilde{\eta}_{4,k,j}^{(n)} = \left(\mathbf{A}_{k,j}^{(N)}[n+1]\right)^T \mathbf{W}_w^{(N)} \mathbf{z}_{k,j}^{(L)}[n+1], \tag{2.76}$$

respectively, and

$$\mathbf{z}_{k,j}^{(L)}[n+1] \triangleq \bar{\mathbf{x}}_{k+1,j}^{(N)}[n+1] - \mathbf{f}_{k,j}^{(N)}[n+1]. \tag{2.77}$$

The computation of $\vec{m}_4^{(n)}(\mathbf{x}_k)$ (2.67) concludes step 6) and, consequently, the n-th iteration of phase II. Then, if the iteration index n is less than n_i, it is increased by one, so that a new iteration can be started by going back to step 1); otherwise, phase III is accomplished.

Phase III - In this phase, the message (see Fig. 2.5)

$$\vec{m}_3^{(n_i+1)}(\mathbf{x}_k) = \mathcal{N}\left(\mathbf{x}_k; \eta_{3,k}^{(n_i+1)}, \mathbf{C}_{3,k}^{(n_i+1)}\right), \tag{2.78}$$

conveying the final filtered pdf provided by F_1, is computed on the basis of Eqs. (2.38)–(2.43) as if a new iteration (corresponding to $n = n_i + 1$) was started. Then, if $k < t$, the output messages $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(N)})$ and $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1})$ (i.e., the new predicted densities) are computed; otherwise, DBF processing is over, since the final measurement has been processed. In the first case, the j-th component of $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}^{(N)})$ is generated by F_1 as (see Fig. 2.3)

$$\vec{m}_{\mathrm{fp},j}\left(\mathbf{x}_{k+1}^{(N)}\right) = \vec{m}_{\mathrm{fp},j}^{(n_i)}\left(\mathbf{x}_{k+1}^{(N)}\right) \tag{2.79}$$

for $j = 1, ..., N_p$ (see Eq. (2.46)); this means that the particle set $S_{k+1}[1]$ available at the beginning of the next recursion consists of the particles $\{\mathbf{x}_{k+1,j}^{(N)} = \bar{\mathbf{x}}_{k+1,j}^{(N)}[n_i+1];\ j = 1, 2, ..., N_p\}$. Then, the predicted pdf $\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1})$ is computed by F_1 as (see Fig. 2.5)

$$\vec{m}_{\mathrm{fp}}(\mathbf{x}_{k+1}) = \int \tilde{f}(\mathbf{x}_{k+1}|\mathbf{x}_k)\, \vec{m}_3^{(n_i+1)}(\mathbf{x}_k)\, d\mathbf{x}_k \tag{2.80}$$

$$= \mathcal{N}(\mathbf{x}_{k+1}; \eta_{\mathrm{fp},k+1}, \mathbf{C}_{\mathrm{fp},k+1}), \tag{2.81}$$

where

$$\eta_{\mathrm{fp},k+1} \triangleq \mathbf{F}_k\, \eta_{3,k}^{(n_i+1)} + \mathbf{u}_k, \tag{2.82}$$

and

$$\mathbf{C}_{\mathrm{fp},k+1} \triangleq \mathbf{C}_w + \mathbf{F}_k\, \mathbf{C}_{3,k}^{(n_i+1)} \mathbf{F}_k^T. \tag{2.83}$$

This concludes the k-th recursion of the DBF technique.

The algorithm described above needs a proper initialization. In our work, a (known) Gaussian pdf $f(\mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1; \eta_1, \mathbf{C}_1)$ is assumed for the initial $\mathbf{x}_1$; for this reason, DBF is initialized by setting $\vec{m}_{\text{fp}}(\mathbf{x}_1) = f(\mathbf{x}_1)$ for F_1 and by sampling the pdf $f(\mathbf{x}_1^{(N)})$ (that results from the marginalization of $f(\mathbf{x}_1)$ with respect to $\mathbf{x}_1^{(L)}$) N_p times in order to generate the initial particle set $S_1[1] = \{\mathbf{x}_{1,j}^{(N)},\, j = 1, ..., N_p\}$; then, the same weight ($w_p = 1/N_p$) is assigned to each particle.

All the processing tasks accomplished by the DBF technique are summarized in Algorithm 1. Note also that, at the end of the k-th recursion, estimates $\hat{\mathbf{x}}_{\text{fe},k}^{(N)}$ and $\hat{\mathbf{x}}_{\text{fe},k}^{(L)}$ of $\mathbf{x}_k^{(N)}$ and $\mathbf{x}_k^{(L)}$, respectively, can be evaluated as: a) $\hat{\mathbf{x}}_{\text{fe},k}^{(N)} = \sum_{j=1}^{N_p} W_{4,k,j}^{(n_i)}\, \mathbf{x}_{k,j}^{(N)}[n_i]$ (see our comments following Eq. (2.60)) or $\hat{\mathbf{x}}_{\text{fe},k}^{(N)} = \bar{\eta}_{3,k}^{(n_i+1)}$, where $\bar{\eta}_{3,k}^{(n_i+1)}$ consists of the *last* D_N elements of $\eta_{3,k}^{(n_i+1)}$ (see Eq. (2.78)); b) $\hat{\mathbf{x}}_{\text{fe},k}^{(L)} = \tilde{\eta}_{3,k}^{(n_i+1)}$, where $\tilde{\eta}_{3,k}^{(n_i+1)}$ consists of the *first* D_L elements of $\eta_{3,k}^{(n_i+1)}$.

Algorithm 3: Double Bayesian Filtering

Initialisation: For $j = 1$ to N_p: sample the pdf $f(\mathbf{x}_1^{(N)})$ to generate the particles $\mathbf{x}_{1,j}^{(N)}$ (forming the set $S_1[1]$), and assign the weight $w_p = 1N_p$ to each of them. Set $\mathbf{W}_{\text{fp},1} = \mathbf{W}_1 = [\mathbf{C}_1]^{-1}$, $\mathbf{w}_{\text{fp},1} = \mathbf{W}_1\eta_1$.

Filtering: For $k = 1$ to t:

a- *First measurement update in* F_1: Compute $\mathbf{W}_{2,k}$ (2.36) and $\mathbf{w}_{2,k}$ (2.37), $\mathbf{C}_{2,k} = [\mathbf{W}_{2,k}]^{-1}$ and $\eta_{2,k} = \mathbf{C}_{2,k}\mathbf{w}_{2,k}$. Then, extract $\tilde{\eta}_{2,k}$ and $\tilde{\mathbf{C}}_{2,k}$ from $\eta_{2,k}$ and $\mathbf{C}_{2,k}$, respectively, and set $\mathbf{W}_{4,k}^{(0)} = \mathbf{0}_{D,D}$ and $\mathbf{w}_{4,k}^{(0)} = \mathbf{0}_D$.

for $n = 1$ *to* n_i **do**

b- *Second measurement update in* F_1: Compute $\mathbf{C}_{3,k}^{(n)}$ and $\eta_{3,k}^{(n)}$ (see Eqs. (2.40)–(2.43); then, extract $\tilde{\eta}_{1,k}^{(n)}$ and $\tilde{\mathbf{C}}_{1,k}^{(n)}$ from $\eta_{3,k}^{(n)}$ and $\mathbf{C}_{3,k}^{(n)}$, respectively.

c- *Measurement updates in* F_2:

for $j = 1$ *to* N_p **do**

c1- *First measurement update*: compute $\bar{\eta}_{1,k,j}^{(n)}$ (2.49), $\tilde{\mathbf{C}}_{1,k,j}^{(n)}$ (2.50) and $w_{1,k,j}^{(n)}$ (2.48).

c2- *Computation of the pseudo-measurements for* F_2: compute $\check{\mathbf{C}}_{z,k,j}^{(n)}$ (2.55), $\check{\eta}_{z,k,j}^{(n)}$ (2.56), $\check{\mathbf{W}}_{z,k,j}^{(n)} = [\check{\mathbf{C}}_{z,k,j}^{(n)}]^{-1}$ and $\check{\mathbf{w}}_{z,k,j}^{(n)} = \check{\mathbf{W}}_{z,k,j}^{(n)}\check{\eta}_{z,k,j}^{(n)}$. Then, compute $\check{\mathbf{W}}_{3,k,j}^{(n)}$ (2.53), $\check{\mathbf{w}}_{3,k,j}^{(n)}$ (2.54), $\check{\mathbf{C}}_{3,k,j}^{(n)} = [\check{\mathbf{W}}_{3,k,j}^{(n)}]^{-1}$ and $\check{\eta}_{3,k,j}^{(n)} = \check{\mathbf{C}}_{3,k,j}^{(n)}\check{\mathbf{w}}_{3,k,j}^{(n)}$. Finally, compute $w_{3,k,j}^{(n)}$ (2.52).

c3- *Second measurement update*: compute $w_{4,k,j}^{(n)}$ (2.57).

end

d- *Normalization of particle weights*: compute the normalized weights $\{W_{4,k,j}^{(n)}\}$ according to Eq. (2.60).

e- *Resampling with replacement*: generate the new particle set $S_k[n+1] = \{\mathbf{x}_{k,j}^{(N)}[n+1]\}$ by resampling $S_k[n]$ on the basis of the weights $\{W_{4,k,j}^{(n)}\}$.

f- *Time update in* F_2: For $j = 1$ to N_p: Compute $\eta_{3,k,j}^{(N)}$ (2.64) and $\mathbf{C}_{3,k,j}^{(N)}$ (2.65), and sample the pdf $\mathcal{N}(\mathbf{x}_{k+1}^{(N)}; \eta_{3,k,j}^{(N)}, \mathbf{C}_{3,k,j}^{(N)})$ to generate the new particle $\mathbf{x}_{k+1,j}^{(N)}[n+1]$.

g- *Computation of the pseudo-measurements for* F_1: For $j = 1$ to N_p: Compute $\mathbf{z}_{k,j}^{(L)}[n+1]$ (2.77), $\tilde{\mathbf{W}}_{4,k,j}^{(n)}$ (2.75) and $\tilde{\mathbf{w}}_{4,k,j}^{(n)}$ (2.76), $\tilde{\mathbf{C}}_{4,k,j}^{(n)} = [\tilde{\mathbf{W}}_{4,k,j}^{(n)}]^{-1}$ and $\tilde{\eta}_{4,k,j}^{(n)} = \tilde{\mathbf{C}}_{4,k,j}^{(n)}\tilde{\mathbf{w}}_{4,k,j}^{(n)}$. Finally, compute $\eta_{4,k}^{(n)}$ (2.68) and $\mathbf{C}_{4,k}^{(n)}$ (2.71) (according to Eqs. (2.69)–(2.70) and (2.72)–(2.74), respectively).

end

h- *Compute forward predictions* (if $k < t$): For $j = 1$ to N_p: set $\mathbf{x}_{k+1,j}^{(N)} = \bar{\mathbf{x}}_{k+1,j}^{(N)}[n_i]$ (these particles form the set $S_{k+1}[1]$). Then, compute $\mathbf{C}_{3,k}^{(n_i+1)}$ (2.42) and $\eta_{3,k}^{(n_i+1)}$ (2.43). Finally, compute $\eta_{\text{fp},k+1}$ (2.82), $\mathbf{C}_{\text{fp},k+1}$ (2.83), $\mathbf{W}_{\text{fp},k+1} = [\mathbf{C}_{\text{fp},k+1}]^{-1}$ and $\mathbf{w}_{\text{fp},k+1} = \mathbf{W}_{\text{fp},k+1}\eta_{\text{fp},k+1}$.

Following the same line of reasoning, a filtering method similar to DBF can be developed for case **C.2**, that is for the second case considered in the previous paragraph. Details are omitted

for space limitations; however, the relevant differences between this method (called *simplified* DBF, SDBF, in the following) and the DBF technique can be summarized as follows:

1) In phase I, $\mathbf{x}_k^{(N)} = \hat{\mathbf{x}}_{\mathrm{fp},k}^{(N)}$ is assumed in computing the *first filtered pdf of* of $\mathbf{x}_k^{(L)}$, where $\hat{\mathbf{x}}_{\mathrm{fp},k}^{(N)}$ denotes the prediction of $\mathbf{x}_k^{(N)}$ evaluated on the basis of the message $\vec{m}_{\mathrm{fp}}(\mathbf{x}_k^{(N)})$ (2.29) provided by F_2.

2) In phase II, the message $\vec{m}_4^{(n)}(\mathbf{x}_k)$ (2.67) is replaced by

$$\vec{m}_4^{(n)}\big(\mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{x}_k^{(L)}; \tilde{\eta}_{4,k}^{(n)}, \tilde{\mathbf{C}}_{4,k}^{(n)}\big), \tag{2.84}$$

since the pseudo-measurements computed in the $\mathrm{F}_2 \rightarrow \mathrm{F}_1$ block refer to the linear state component only; here, $\tilde{\eta}_{4,k}^{(n)}$ and $\tilde{\mathbf{C}}_{4,k}^{(n)}$ are given by Eqs. (2.69) and (2.72), respectively.

3) In phase III, $\mathbf{x}_k^{(N)} = \hat{\mathbf{x}}_{\mathrm{fe},k}^{(N)}$ is assumed in computing the *prediction* of $\mathbf{x}_{k+1}^{(L)}$, where $\hat{\mathbf{x}}_{\mathrm{fe},k}^{(N)}$ denotes the estimate of $\mathbf{x}_k^{(N)}$ evaluated on the basis of the *final* filtered pdf computed by F_2.

2.3.3 Computational complexity

The computational cost of the DBF and SDBF techniques has been carefully assessed in terms of number of *floating operations* (flops) to be executed in each of their recursions. The general criteria adopted in estimating the computational cost of an algorithm are the same as those illustrated in [40, App. A, p. 5420] and are not repeated here for space limitations. A detailed analysis of the cost required by each task accomplished by the DBF and the SDBF techniques is provided in Appendix E.1. Our analysis leads to the conclusion that the overall computational cost of the DBF and of the SDBF are approximately of order $\mathcal{O}(N_{\mathrm{DBF}})$ and $\mathcal{O}(N_{\mathrm{SDBF}})$, respectively, with

$$\begin{aligned} N_{\mathrm{DBF}} = {} & 2PD^2 + 4P^2D + 16D^3/3 + 14n_iD^3/3 \\ & + n_i \cdot N_p(2PD_L^2 + 2P^2D_L + 2P^3/3 \\ & + 6D_L^3 + 6D_LD_N^2 + 4D_L^2D_N + D_N^3/3) \end{aligned} \tag{2.85}$$

and

$$\begin{aligned} N_{\mathrm{SDBF}} = {} & 2PD_L^2 + 4P^2D_L + 16D_L^3/3 + 14n_iD_L^3/3 \\ & + n_i \cdot N_p(2PD_L^2 + 2P^2D_L + 2P^3/3 \\ & + 6D_L^3 + 6D_LD_N^2 + 4D_L^2D_N + D_N^3/3). \end{aligned} \tag{2.86}$$

Each of the last two expressions has been derived as follows. First, the costs of all the tasks identified in Appendix E.1 for both the interconnected filters have been summed (see Eqs. (E.1)–(E.8)); then, the resulting expression has been simplified, keeping only the dominant contributions due to matrix inversions, matrix products and Cholesky decompositions and discarding all the contributions that originate from the evaluation of the matrices $\mathbf{A}_k^{(Z)}(\mathbf{x}_k^{(N)})$ (with $Z = L$ and N), $\mathbf{F}_k$, $\mathbf{H}_k$ and $\mathbf{B}_k$ and the functions $\mathbf{f}_k^{(Z)}(\mathbf{x}_k^{(N)})$ (with $Z = L$ and N), $\mathbf{f}_k(\mathbf{x}_k)$ and $\mathbf{g}_k(\mathbf{x}_k^{(N)})$. Moreover, the complexity of particle resampling has been ignored. A similar approach has been followed for EKF, for RBPF and for the MPF technique described in [37]; their complexities are approximately of order $\mathcal{O}(N_{\mathrm{EKF}})$, $\mathcal{O}(N_{\mathrm{RBBF}})$ and $\mathcal{O}(N_{\mathrm{MPF}})$, respectively, with

$$N_{\mathrm{EKF}} = 2PD^2 + 2P^2D + 2P^3/3 + 6D^3, \tag{2.87}$$

$$\begin{aligned} N_{\mathrm{RBPF}} = {} & N_p(4PD_L^2 + 6P^2D_L + 2P^3/3 + 6D_L^3 \\ & + 4D_L^2D_N + 6D_LD_N^2 + D_N^3/3) \end{aligned} \tag{2.88}$$

and

$$N_{\mathrm{MPF}} = n(2\,M\,L\,d_y^3/3 + M\,d_{x,i}^3/3); \tag{2.89}$$

note that the symbols appearing in the last formula are the same as those defined in ref. [37].

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 of the corresponding algorithms. In fact, distinct filtering techniques may have substantially smaller memory requirements and, as evidenced by our numerical results, this may influence their overall execution speed. For instance, the DBF/SDBF techniques need to store the state estimates and predictions generated by a single extended Kalman filter, whereas RBPF needs to memorise those computed by a bank of N_p Kalman filters running in parallel. Finally, it is worth stressing that N_{DBF} (2.85) and N_{SDBF} (2.86) exhibit a linear dependence on the parameter n_i. Actually, in our computer simulations, $n_i = 1$ has been always selected, since marginal improvements have been obtained by increasing n_i beyond unity.

2.4 Numerical Results

In this section we first compare, in terms of accuracy and execution time, the DBF and SDBF techniques with an extended Kalman filter (corresponding to F_1 of the DBF technique) and the RBPF technique (corresponding to the combination of F_2 of the DBF technique with a bank of N_p Kalman filters) for a specific CLG SSM, denoted SSM #1. This SSM is very similar to the dynamic model described in [1, Par. VII-A, p. 1531], and refers to an agent moving on a plane and whose state is defined as $\mathbf{x}_k \triangleq [\mathbf{p}_k^T, \mathbf{v}_k^T]^T$; here, $\mathbf{v}_k \triangleq [v_{x,k}, v_{y,k}]^T$ and $\mathbf{p}_k \triangleq [p_{x,k}, p_{y,k}]^T$ (corresponding to $\mathbf{x}_k^{(N)}$ and $\mathbf{x}_k^{(L)}$, respectively) represent the agent *velocity* and its *position*, respectively (their components are expressed in m/s and in m, respectively). The dynamic models (see [1, eqs. (67)–(68), p. 1531])

$$\mathbf{v}_{k+1} = \rho \mathbf{v}_k + (1-\rho)\ \mathbf{n}_{v,k} + \mathbf{a}\left(\mathbf{p}_k, \mathbf{v}_k\right) T_s, \tag{2.90}$$

and

$$\mathbf{p}_{k+1} = \mathbf{p}_k + \mathbf{v}_k T_s + \frac{1}{2}\mathbf{a}\left(\mathbf{p}_k, \mathbf{v}_k\right) T_s^2 + \mathbf{n}_{p,k} \tag{2.91}$$

are adopted for the agent velocity and position, respectively; here, ρ is a *forgetting factor* ($0 < \rho < 1$), T_s is the sampling interval, $\{\mathbf{n}_{v,k}\}$ and $\{\mathbf{n}_{p,k}\}$ are mutually independent *additive white Gaussian noise* (AWGN) processes (whose elements are characterized by the covariance matrices $\mathbf{I}_2$ and $\sigma_p^2 \mathbf{I}_2$, respectively),

$$\mathbf{a}\left(\mathbf{p}_k, \mathbf{v}_k\right) = -(a_0/d_0)\, \mathbf{p}_k - \tilde{a}_0\, f_v\left(\|\mathbf{v}_k\|\right)\ \mathbf{u}_{v,k}. \tag{2.92}$$

is the acceleration associated with position/velocity-dependent forces, a_0 and $\tilde{a}_0$ are scale factors (both expressed in $\mathrm{m/s^2}$), d_0 is a *reference distance,* $\mathbf{u}_{v,k} \triangleq \mathbf{v}_k / \|\mathbf{v}_k\|$ is the versor (i.e., the vector of unit norm) associated with $\mathbf{v}_k$ and $f_v(x) = (x/v_0)^3$ is a continuous, differentiable and dimensionless function (the parameter v_0 represents a *reference velocity*). Moreover, the measurement model

$$\mathbf{y}_k = [\mathbf{p}_k^T\ \|\mathbf{v}_k\|]^T + \mathbf{e}_k, \tag{2.93}$$

is adopted; here, $\{\mathbf{e}_k\}$ is an AWGN process, whose elements are characterized by the covariance matrix $\mathbf{C}_e = \mathrm{diag}(\sigma_{e,p}^2, \sigma_{e,p}^2, \sigma_{e,v}^2)$.

In our computer simulations, the estimation accuracy of the considered filtering techniques for SSM#1 has been assessed by evaluating two *root mean square errors* (RMSEs), one for the linear state component, the other for the nonlinear one, over an observation interval lasting $T = 300\ T_s$; these are denoted $\mathrm{RMSE}_L(\mathrm{alg})$ (m) and $\mathrm{RMSE}_N(\mathrm{alg})$ (m/s) respectively, where 'alg' denotes the algorithm these parameters refer to (note also that $\mathrm{RMSE}_N(\mathrm{DBF})$ is computed on the basis of the estimate of $\mathbf{v}_k$ generated by F_2, since this was found to be slightly more accurate than that evaluated by F_1). Our assessment of *computational requirements* is based, instead, on comparing N_{DBF} (2.85), N_{SDBF} (2.86), N_{EKF} (2.87) and N_{RBPF} (2.88), and on assessing the average *execution time* required by each algorithm over the whole observation interval. Moreover, the following values have been selected for the parameters of SSM#1: $\rho = 0.99$, $T_s = 0.1$ s, σ_p

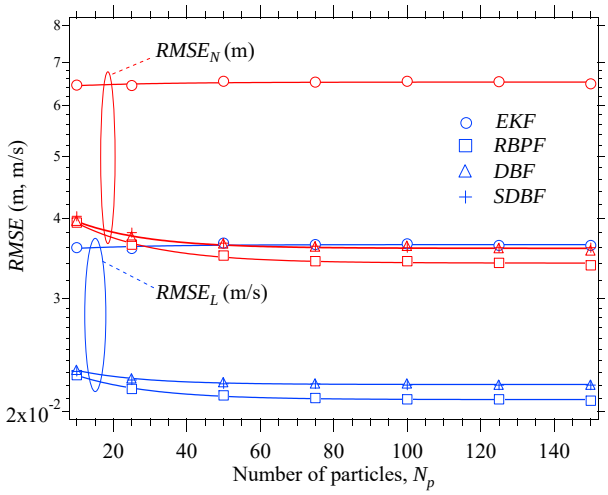


Figure 2.6: RMSE performance versus N_p for the linear component (RMSE_L; blue curves) and the nonlinear component (RMSE_N; red curves) of system state (SSM#1); EKF, RBPF, DBF and SDBF are considered.

$= 0.01$ m, $\sigma_{e,p} = 5 \cdot 10^{-2}$ m, $\sigma_{e,v} = 5 \cdot 10^{-2}$ m/s, $a_0 = 1.5$ m/s^2, $d_0 = 0.5$ m, $\tilde{a}_0 = 0.05$ m/s^2 and $v_0 = 1$ m/s (the initial position $\mathbf{p}_0 \triangleq [p_{x,0}, p_{y,0}]^T$ and the initial velocity $\mathbf{v}_0 \triangleq [v_{x,0}, v_{y,0}]^T$ have been set to $[5 \text{ m}, 8 \text{ m}]^T$ and $[4 \text{ m/s}, 4 \text{ m/s}]^T$, respectively). These values ensure that: a) the two components of the position vector are represented by fast and damped oscillations in the observation interval; b) the time variations of the state vector can be accurately tracked by RBPF.

Some numerical results showing the dependence of RMSE_L and RMSE_N on the number of particles (N_p) for RBPF, EKF, DBF and SDBF are illustrated in Fig. 2.6 (simulation results are indicated by markers, whereas continuous lines are drawn to fit them, so facilitating the interpretation of the available data); in this case, $n_i = 1$ has been selected for DBF/SDBF and the range $[10, 150]$ has been considered for N_p. These results show that:

1) The EKF technique is outperformed by the other three filtering algorithms in terms of both RMSE_L and RMSE_N for any value of N_p; for instance, RMSE_L(EKF) (RMSE_N(EKF)) is about 1.65 (1.80) times larger than RMSE_L(DBF) (RMSE_N(DBF)) for $N_p = 100$.

2) DBF/SDBF perform slightly worse than RBPF for the same value of N_p (for instance, RMSE_L(DBF) and RMSE_N(DBF) are about 5% larger than the corresponding quantities evaluated for RBPF).

3) No real improvement in terms of RMSE_L and RMSE_N is found for $N_p \gtrsim 100$, if RBPF, DBF or SDBF are employed.

4) The SDBF performs very similarly as DBF; for this reason, in this specific case, the presence of redundancy in the DBF does not allow to achieve a better estimation accuracy.

Despite their similar accuracies, RBPF, DBF and SDBF are characterized by different computational complexities and execution times. This is evidenced by the numerical results appearing in Fig. 2.7 and showing the dependence of the execution time and the computational complexity on N_p for the considered filtering algorithms. For instance, from these results it is easily inferred that the DBF complexity is about 39% smaller that of RBPF for $N_p = 100$; however, the gap in terms of execution time is even larger mainly for the reasons illustrated at the end of Paragraph 2.3.3 (in particular, the execution time for the DBF is approximately 0.61 times smaller than that required by RBPF). Moreover, the results shown in Figs. 2.6–2.7 lead to the conclusion that, in the considered scenario, DBF/SDBF achieve a better accuracy-complexity tradeoff than RBPF.

The second SSM considered in this work has been inspired by refs. [39] and [41]. In fact, it refers to a sensor network employing P sensors placed on the vertices of a square grid (partitioning a square area having side equal to l m) and receiving the reference signals radiated, at the same

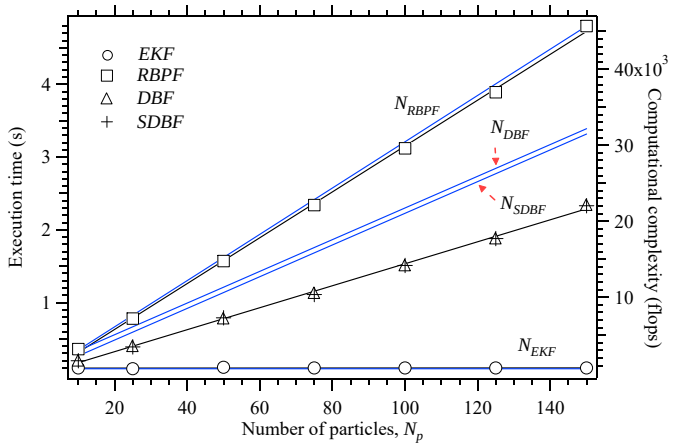


Figure 2.7: Execution time (black curves and markers) and computational complexity (blue curves) versus N_p for EKF, RBPF, DBF and SDBF; SSM#1 is considered.

power level and at the same frequency, by N independent targets moving on a plane. Each target evolves according to the motion model described by

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mathbf{n}_{v,k} \tag{2.94}$$

and Eq. (2.91) with $\mathbf{a}(\mathbf{p}_k, \mathbf{v}_k) = 0$ for any k. In this case, the considered SSM (denoted SSM#2 in the following) refers to the whole set of targets and its state vector $\mathbf{x}_k$ results from the ordered concatenation of the vectors $\{\mathbf{x}_k^{(i)};\ i = 1,\ 2,\ ...,\ N\}$, where $\mathbf{x}_k^{(i)} \triangleq [(\mathbf{v}_k^{(i)})^T, (\mathbf{p}_k^{(i)})^T]^T$, and $\mathbf{v}_k^{(i)}$ and $\mathbf{p}_k^{(i)}$ represent the i-th target *velocity* and the *position*, respectively. Moreover, the following additional assumptions have been made about this SSM: 1) the process noises $\mathbf{n}_{p,k}^{(i)}$ and $\mathbf{n}_{v,k}^{(i)}$, affecting the i-th target position and velocity, respectively, are given by $\mathbf{n}_{p,k}^{(i)} = (T_s^2/2)\,\mathbf{n}_{a,k}^{(i)}$ and $\mathbf{n}_{v,k}^{(i)} = T_s\,\mathbf{n}_{a,k}^{(i)}$, where $\{\mathbf{n}_{a,k}^{(i)}\}$ is two-dimensional AWGN, representing a random acceleration and having covariance matrix $\sigma_a^2\,\mathbf{I}_2$ (with $i = 1,\ 2,\ ...,\ N$); 2) the measurement acquired by the q-th sensor (with $q = 1,\ 2,\ ...,\ P$) in the k-th observation instant is given by

$$y_{q,k} = 10\log_{10}\Big(\Psi \sum_{i=1}^{N} d_0^2\,\|\mathbf{s}_q - \mathbf{p}_k^{(i)}\|^{-2}\Big) + e_k, \tag{2.95}$$

where the measurement noise $\{e_k\}$ is AWGN having zero mean and variance σ_e^2, Ψ denotes the normalized power received by each sensor from any target at a distance d_0 from the sensor itself and $\mathbf{s}_q$ is the position of the considered sensor; 3) the overall measurement vector $\mathbf{y}_k$ results from the ordered concatenation of the measurements $\{y_{q,k};\ q = 1,\ 2,\ ...,\ P\}$ and, consequently, provides information about the position only; 4) the initial position $\mathbf{p}_0^{(i)} \triangleq [p_{x,0}^{(i)}, p_{y,0}^{(i)}]^T$ and the initial velocity $\mathbf{v}_0^{(i)} \triangleq [v_{x,0}^{(i)}, v_{y,0}^{(i)}]^T$ of the i-th target have been randomly selected (with $i = 1,\ 2,\ ...,\ N$). As far as the last point is concerned, it is important to mention that, in our computer simulations, the region covered by the sensors has been partitioned into a number of squares (the overall number of squares is not smaller than N) and distinct targets have been placed in different squares in a random fashion; moreover, the initial velocity of each target has been randomly selected within the interval $(v_{\min}, v_{\max})$ in order to ensure that the trajectories of distinct targets do not cross each other in the observation interval. The following values have been selected for the parameters of SSM#2: $P = 25$, $l = 10^3$ m, $T_s = 1$ s, $\sigma_a^2 = 0.1$ m/s^2, $\sigma_e^2 = -35$ dB, $\Psi = 1$, $d_0 = 1$ m, $v_{\min} = 0$ m/s and $v_{\min} = 0.1$ m/s. Moreover, a number N of targets ranging from 1 to 5 has been observed for $T = 120\ T_s$ s.

Our computer simulations for SSM#2 have aimed at evaluating: a) the accuracy achieved by different filtering algorithms in tracking the position of N targets; b) the probability that each

filtering algorithm *diverges* in the considered observation interval (this parameter is denoted P_{FD} in the following). In practice, the accuracy achieved in position tracking has been assessed by estimating the RMSE characterizing the whole set $\{\mathbf{p}_k^{(i)};\ i = 1,\ 2,\ ...,\ N\}$ over each instant of the considered observation interval; note that, if the i-th target is considered, its position $\mathbf{p}_k^{(i)}$ represents the *nonlinear* component of the associated substate $\mathbf{x}_k^{(i)}$, because of the nonlinear dependence of $\mathbf{y}_k$ on it (see Eq. (2.95)). On the other hand, the probability P_{FD} has been assessed by carefully identifying all the simulation runs in which the tracking of at least one of the N targets fails. Moreover, the tracking accuracy and the probability of divergence have been evaluated for the following six filtering techniques: 1) EKF; 2) RBPF; 3) the MPF technique developed in [37] and based on the interconnection of N identical particle filters (one for each target); 4) DBF; 5) SDBF; 6) a novel filtering algorithm based on the interconnection of $N_F = N + 1$ filters and dubbed MBF *algorithm* (MBFA). The last algorithm involves the interconnection of an extended Kalman filter with N particle filters, each representing the filtered/predicted pdfs of a two-dimensional vector through $\tilde{N}_p$ weighted particles. More specifically, the i-th particle filter estimates the position $\mathbf{p}_k^{(i)}$ of the i-th target (with $i = 1,\ 2,\ ...,\ N$); consequently, the degree of redundancy of the MBFA is $N_{\mathrm{d}} = 2N$, that is the same as DBF. The computation of the messages passed in the k-th recursion of the MFBA is based on the same equations as those derived for DBF; the only modifications are due to the fact that:

1) The measurement update accomplished by the i-th particle filter of the MBFA requires integrating out the dependence of the measurement vector $\mathbf{y}_k$ on the $(N-1)$ positions $\{\mathbf{p}_k^{(j)};\ j \neq i\}$. This marginalization is accomplished by exploiting the pdfs of the positions $\{\mathbf{p}_k^{(j)};\ j \neq i\}$ *predicted* by the other $(N-1)$ particle filters. Moreover, the computation of particle weights requires drawing L particles from the predicted pdfs of the other filters (see [37, eq. (7), p. 354]).

2) The computation of the pseudo-measurements for the extended Kalman filter requires a particle representation for the whole vector $\mathbf{p}_k$, that results from the ordered concatenation of the vectors $\{\mathbf{p}_k^{(i)};\ i = 1,\ 2,\ ...,\ N\}$. In the MBFA, the j-th particle for $\mathbf{p}_k$ is generated by: a) taking the j-th element of the particle set made available, after resampling, by each of the N particle filters (with $j = 1,\ 2,\ ...,\ \tilde{N}_p$); b) concatenating the N particles obtained in this way.

In our computer simulations, $N_p = 500$ has been selected for RBPF, DBF and SDBF. Moreover, in the MPF technique and in the MBFA, each of N particle filters makes use of $\tilde{N}_p = \lfloor N_p/N \rfloor$ particles, where $N_p = 500$. Note also that: a) the parameter $\tilde{N}_p$ corresponds to the parameter M of [37, Sec. III], since $J = 1$ is set in MPF (where J denotes the number of children generated in the time update step); b) in our simulations, the ratio $L/\tilde{N}_p$ is always close to 1/3 for both the MPF technique and the MBFA. The choices illustrated above ensure that all the algorithms involving PF have comparable execution times; for instance, the execution time of RBPF, MPF and DBF is approximately 21.4%, 3.4% and 0.9% larger, respectively, than that of the MBFA for $N = 5$ targets. Despite this, these techniques exhibit different behaviours. In fact, our computer simulations have evidenced that, on the one hand, EKF and SDBF quickly diverge after their initialization and, therefore, are useless in the considered scenario. On the other hand, the RBPF, the MPF and the DBF techniques, and the MBFA achieve similar accuracies in tracking conditions, but are characterized by different probabilities of divergence. This is evidenced by Fig. 2.8, that shows the dependence of the probability P_{FD} on the overall number of targets. From these results it is easily inferred that, as the number of target increases, the RBPF and the MPF techniques are substantially outperformed by the DBF technique and the MBFA. These results lead to the conclusion that the property of redundancy can play a key role in some applications, since it can substantially reduce the probability of divergence of a filtering algorithm.

2.5 Conclusions

In this chapter, the problem of developing filtering algorithms that involve multiple interconnected Bayesian filters running in parallel has been investigated. The devised solution, called *multiple*

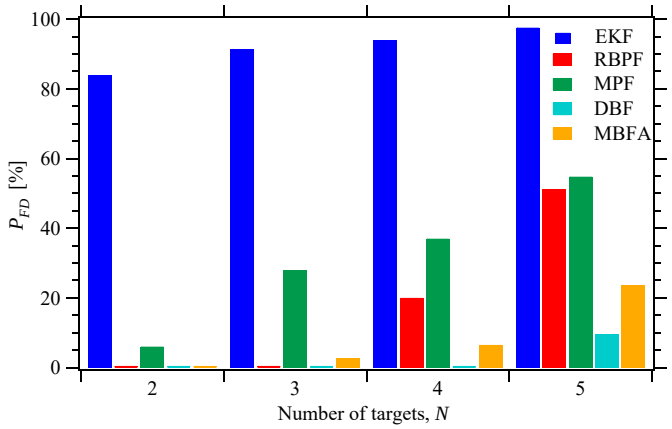


Figure 2.8: Probability of divergence versus N for the RBPF, the MPF and the DBF techniques, and the MBFA.

Bayesian filtering, is based on the factor graph representation of Bayesian filtering. The application of our graphical approach to a network consisting of two Bayesian filters has been illustrated. Moreover, a specific instance of the proposed approach has been analyzed in detail for the case in which the considered SSM is CLG, and the interconnected filters are an extended Kalman filter and a particle filter. Simulation results for two specific SSMs evidence that the devised filtering techniques perform closely to other well known filtering methods, but are appreciably faster or offer a better tracking capability.

Three

Double Bayesian Smoothing as Message Passing

Recently, a novel method for developing filtering algorithms, based on the interconnection of two Bayesian filters and called double Bayesian filtering, has been proposed. In this chapter we show that the same conceptual approach can be exploited to devise a new smoothing method, called double Bayesian smoothing. A double Bayesian smoother combines a double Bayesian filter, employed in its forward pass, with the interconnection of two backward information filters used in its backward pass. As a specific application of our general method, a detailed derivation of double Bayesian smoothing algorithms for conditionally linear Gaussian systems is illustrated. Numerical results for two specific dynamic systems evidence that these algorithms can achieve a better complexity-accuracy tradeoff and tracking capability than other smoothing techniques recently appeared in the literature.

Part of this Chapter has been published in [47].

3.1 Introduction

The problem of *Bayesian smoothing* for a *state space model* (SSM) concerns the development of recursive algorithms able to estimate the *probability density function* (pdf) of the model state on a given observation interval, given a batch of noisy measurements acquired over it [3], [48]; the estimated pdf is known as a *smoothed* or *smoothing* pdf. Two general methods are available in the literature for recursively calculating smoothing densities; they are known as the *forward filtering-backward smoothing recursion* (e.g., see [49] and [50]) and the method based on the *two-filter smoothing formula* (e.g., see [51] and [52]). Both methods are based on the idea that the smoothing densities can be computed by combining the predicted and/or filtered densities generated by a Bayesian filtering method with the statistical information produced in the backward pass by a different filtering method; the latter method is *paired with the first one* and, in the case of the two-filter smoothing formula, is known as *backward information filtering* (BIF). Unluckily, closed form solutions for Bayesian smoothing are available for *linear Gaussian* and *linear Gaussian mixture* models only [3, 48, 53]. This has motivated the development of various methods based on approximating smoothing densities in different ways. For instance, the use of Gaussian approximations for the smoothing densities and of sigma points techniques for solving moment matching integrals has been investigated in [54–56]. Another class of methods (usually known as *particle smoothers*) is based on the exploitation of *sequential Monte Carlo* techniques, i.e. on approximating smoothing densities through a set of weighted particles (e.g., see [49, 51, 57–60] and references therein). Recently, substantial attention has been also paid to the development of smoothing algorithms for the class of *conditionally linear Gaussian* SSMs [7, 23, 24, 61, 62]. In this case, the above mentioned approximate methods can benefit from the so called *Rao-Blackwellization* technique, i.e. from the marginalisation of the linear substructure of any conditionally linear Gaussian model; this can significantly reduce the overall computational complexity of both sigma-point based Gaussian smoothing [24] and particle smoothing [7, 23, 61, 62] (that is usually known as *Rao-Blackwellized particle smoothing*, RBPS, in this case).

In this chapter, we propose a novel general method for the development of *computationally efficient* particle smoothers. Our method exploits the same conceptual approach illustrated in Chapter 2 (see [32]) in the context of Bayesian filtering and dubbed *multiple Bayesian filtering*. That approach is based on the idea of developing new filtering algorithms by: a) interconnecting multiple heterogeneous Bayesian filters; b) representing the processing accomplished by each Bayesian filter and the exchange of statistical information among distinct filters as a message passing over a proper factor graph. In [32] the exploitation of this approach has been investigated in detail for the case in which two Bayesian filters are interconnected, i.e. *dual Bayesian filtering* (DBF) is employed. Moreover, it has been shown that accurate and computationally efficient DBF algorithms can be devised if the considered SSM is conditionally linear Gaussian. In this chapter, we show that, if DBF is employed in the *forward* pass of a smoothing method, a BIF method, paired with DBF and based on the interconnection of two backward information filters can be devised by following some simple rules. Similarly as DBF, our derivation of such a BIF method, called *double backward information filtering* (DBIF), is based on a *graphical model*. Such a graphical model allows us to show that: a) the pdfs computed in DBIF can be represented as messages passed on it; b) all the expressions of the passed messages can be derived by applying the same rule, namely the so called *sum-product algorithm* (SPA) [17], [18], to it; c) iterative algorithms can be developed in a natural fashion once the cycles it contains have been identified and the order according to which messages are passed on them (i.e., the *message scheduling*) has been established; d) the statistical information generated by a DBIF algorithm in the backward pass can be easily merged with those produced by its paired DBF technique in the forward pass in order to evaluate the required smoothed pdfs. To exemplify the usefulness of the resulting smoothing method, based on the combination of DBF and DBIF, and called *double Bayesian smoothing* (DBS), the two DBF algorithms proposed in [32] for the class of conditionally linear Gaussian SSMs are taken into consideration, and the BIF algorithm paired with each of them and a simplified version of it are derived. This leads to the development of four new DBS algorithms, two generating an estimate of the joint smoothing density over the whole observation interval, the other two an estimate of the marginal smoothing densities over the same interval. Our computer simulations for two specific conditionally linear Gaussian SSMs evidence that, in the first case, the derived DBS algorithms perform very closely to the RBPS technique proposed in [7] and to the particle smoothers devised in [23], but at lower computational cost and time. In the second case, instead, two of the devised DBS techniques represent the only technically useful options, thanks to their good tracking capability. In fact, such techniques are able to operate reliably even when their competitors diverge in the forward pass.
It is worth stressing that the technical contribution provided by this chapter represents a significant advancement with respect to the application of factor graph theory to particle smoothing illustrated in [23]. In fact, in that manuscript, we also focus on conditionally linear Gaussian models, but assume that the forward pass is accomplished by *marginalized particle filtering* (MPF; also known as *Rao-Blackwellized particle filtering*); in other words, Bayesian filtering is based on the interconnection of a particle filter with a *bank of Kalman filters*. In this chapter, instead, the general method we propose applies to a couple of arbitrary interconnected Bayesian filters. Moreover, the specific smoothing algorithms we derive assume that the forward pass is carried out by a filtering algorithm based on the interconnection of a particle filter with a *single* extended Kalman filter.

The remaining part of this chapter is organized as follows. In Section 3.2, a general graphical model, on which the processing accomplished in DBF, DBIF, and DBS is based, is illustrated. In Section 3.3, a specific instance of the graphical model illustrated in the previous section is developed under the assumptions that the filters employed in the forward pass are an extended Kalman filter and a particle filter, and that the considered SSM is conditionally linear Gaussian. Then, the scheduling and the computation of the messages passed over this model are analysed in detail and new DBS algorithms are devised. The differences and similarities between these algorithms and other known smoothing techniques are analysed in Section 3.4. A comparison,

in terms of accuracy, computational complexity, and execution time, between the proposed techniques and three smoothers recently appeared in the literature, is provided in Section 3.5 for two conditionally linear Gaussian SSMs. Finally, some conclusions are offered in Section 3.6.

3.2 Graphical Model for a Couple of Interconnected Bayesian Information Filters and Message Passing on it

In this chapter, we consider a discrete-time SSM whose $D-$dimensional *hidden state* in the $k-$th interval is denoted $\mathbf{x}_k \triangleq [x_{0,k}, x_{1,k}, ..., x_{D-1,k}]^T$, and whose *state update* and *measurement models* are expressed by

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k \tag{3.1}$$

and

$$\begin{aligned} \mathbf{y}_k &\triangleq [y_{0,k}, y_{1,k}, ..., y_{P-1,k}]^T \\ &= \mathbf{h}_k(\mathbf{x}_k) + \mathbf{e}_k, \end{aligned} \tag{3.2}$$

respectively, with $k = 1, 2, ..., T$. Here, $\mathbf{f}_k(\mathbf{x}_k)$ ($\mathbf{h}_k(\mathbf{x}_k)$) is a time-varying $D-$dimensional ($P-$dimensional) real function, T is the duration of the observation interval and $\mathbf{w}_k$ ($\mathbf{e}_k$) is the $k-$th element of the process (measurement) noise sequence $\{\mathbf{w}_k\}$ ($\{\mathbf{e}_k\}$); this sequence consists of $D-$dimensional ($P-$dimensional) *independent and identically distributed* (iid) Gaussian noise vectors, each characterized by a zero mean and a covariance matrix $\mathbf{C}_w$ ($\mathbf{C}_e$). Moreover, statistical independence between $\{\mathbf{e}_k\}$ and $\{\mathbf{w}_k\}$ is assumed.

From a statistical viewpoint, a complete statistical description of the considered SSM is provided by the pdf $f(\mathbf{x}_1)$ of its initial state, its *Markov model* $f(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and its *observation model* $f(\mathbf{y}_k|\mathbf{x}_k)$ for any k; the first pdf is assumed to be known, whereas the last two pdfs can be easily derived from Eq. (3.1) and Eq. (3.2), respectively.

In the following, we focus on the problem of developing novel smoothing algorithms and, in particular, algorithms for the estimation of the *joint smoothed pdf* $f(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ (problem **P.1**) and the sequence of *marginal smoothed pdfs* $\{f(\mathbf{x}_k|\mathbf{y}_{1:T}),\ k = 1, 2, ..., T\}$ (problem **P.2**); here, $\mathbf{y}_{1:T} \triangleq \left[\mathbf{y}_1^T, \mathbf{y}_2^T, ..., \mathbf{y}_T^T\right]^T$ is a $P \cdot T-$dimensional vector. Note that, in principle, once problem **P.1** is solved, problem **P.2** can be easily tackled; in fact, if the joint pdf $f(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ is known, all the posterior pdfs $\{f(\mathbf{x}_k|\mathbf{y}_{1:T})\}$ can be evaluated by *marginalization*.

The development of our smoothing algorithms is mainly based on the graphical approach illustrated in our previous manuscripts [23, Sec. III], [32, Sec. II] and [1, Sec. III] for Bayesian filtering and smoothing. This approach consists in the following steps:

1. The state vector $\mathbf{x}_k$ is partitioned in two substates, denoted $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$ and having sizes D_1 and $D_2 = D - D_1$, respectively. Note that, if $\bar{\mathbf{x}}_k^{(i)}$ represents the portion of $\mathbf{x}_k$ not included in $\mathbf{x}_k^{(i)}$ (with $i = 1$ and 2), our assumptions entail that $\bar{\mathbf{x}}_k^{(1)} = \mathbf{x}_k^{(2)}$ and $\bar{\mathbf{x}}_k^{(2)} = \mathbf{x}_k^{(1)}$.

2. A *sub-graph* that allows to represent both Bayesian filtering and BIF for the substate $\mathbf{x}_k^{(i)}$ (with $i = 1$ and 2) as message passing algorithms on it is developed, under the assumption that the complementary substate $\bar{\mathbf{x}}_k^{(i)}$ is statistically known. This means that filtered and predicted densities of $\mathbf{x}_k^{(i)}$ are represented as messages passed on the edges of this sub-graph and the rules for computing them result from the application of the SPA to it.

3. The two sub-graphs devised in the previous step (one referring to $\mathbf{x}_k^{(1)}$, the other one to $\mathbf{x}_k^{(2)}$) are *interconnected*, so that a single graphical model referring to the whole state $\mathbf{x}_k$ is obtained.

4. Algorithms for Bayesian filtering and BIF for the whole state $\mathbf{x}_k$ are derived by applying the SPA to the graphical model obtained in the previous step.

Let us analyse now the steps 2.-4. in more detail. As far as **step 2.** is concerned, the sub-graph devised for the substate $\mathbf{x}_k^{(i)}$ is based on the same principles illustrated in our manuscripts cited above (in particular, ref. [32]) and is illustrated in Fig. 3.1. The $k-$th recursion (with $k = 1, 2, ..., T$) of Bayesian filtering for the sub-state $\mathbf{x}_k^{(i)}$ is represented as a *forward* message

passing on this factor graph, that involves the Markov model $f(\mathbf{x}_{k+1}^{(i)}|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)})$ and the observation model $f(\mathbf{y}_k|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)})$. This allows to compute the messages $\vec{m}_{\rm fe1}(\mathbf{x}_k^{(i)})$, $\vec{m}_{\rm fe2}(\mathbf{x}_k^{(i)})$ and $\vec{m}_{\rm fp}(\mathbf{x}_{k+1}^{(i)})$, that convey the *first filtered* pdf of $\mathbf{x}_k^{(i)}$, the *second filtered* pdf of $\mathbf{x}_k^{(i)}$ and the *predicted* pdf of $\mathbf{x}_{k+1}^{(i)}$, respectively, on the basis of the messages $\vec{m}_{\rm fp}(\mathbf{x}_k^{(i)})$, $m_{\rm ms}(\mathbf{x}_k^{(i)})$ and $m_{\rm pm}(\mathbf{x}_k^{(i)})$; the last three messages represent the predicted pdf of $\mathbf{x}_k^{(i)}$ evaluated in the previous (i.e., in the $(k-1)-$th) recursion of Bayesian filtering, and the messages conveying the *measurement* and the *pseudo-measurement* information, respectively, available in the $k-$recursion. The considered filtering algorithm requires the availability of the messages $m_{\rm pm}(\mathbf{x}_k^{(i)})$, $m_{\rm mg1}(\bar{\mathbf{x}}_k^{(i)})$, $m_{\rm mg2}(\bar{\mathbf{x}}_k^{(i)})$, that are computed on the basis of external statistical information. The presence of the messages $m_{\rm mg1}(\bar{\mathbf{x}}_k^{(i)})$ and $m_{\rm mg2}(\bar{\mathbf{x}}_k^{(i)})$ is due the fact that the substate $\bar{\mathbf{x}}_k^{(i)}$ represents a *nuisance state* for the considered filtering algorithm; in fact, these messages convey filtered (or predicted) pdfs of $\bar{\mathbf{x}}_k^{(i)}$ and are employed to integrate out the dependence of the pdfs $f(\mathbf{y}_k|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)})$ and $f(\mathbf{x}_{k+1}^{(i)}|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)})$, respectively, on $\bar{\mathbf{x}}_k^{(i)}$. Note also that these two messages are not necessarily equal, since more refined information about $\bar{\mathbf{x}}_k^{(i)}$ could become available after that the message $m_{\rm ms}(\mathbf{x}_k^{(i)})$ has been computed. On the other hand, the message $m_{\rm pm}(\mathbf{x}_k^{(i)})$ conveys the statistical information provided by a pseudo-measurement[1] about $\mathbf{x}_k^{(i)}$. In Fig. 3.1, following [32, Sec. II], it is assumed that the pseudo-measurement $\mathbf{z}_k^{(i)}$ is available in the estimation of $\mathbf{x}_k^{(i)}$ and that $m_{\rm pm}(\mathbf{x}_k^{(i)})$ represents the pdf of $\mathbf{z}_k^{(i)}$ conditioned on $\mathbf{x}_k^{(i)}$, that is

$$m_{\rm pm}\big(\mathbf{x}_k^{(i)}\big) \triangleq f\big(\mathbf{z}_k^{(i)}\big|\mathbf{x}_k^{(i)}\big). \tag{3.3}$$

The computation of the messages $\vec{m}_{\rm fe1}(\mathbf{x}_k^{(i)})$, $\vec{m}_{\rm fe2}(\mathbf{x}_k^{(i)})$ and $\vec{m}_{\rm fp}(\mathbf{x}_{k+1}^{(i)})$ on the basis of the messages $\vec{m}_{\rm fp}(\mathbf{x}_k^{(i)})$, $m_{\rm ms}(\mathbf{x}_k^{(i)})$, $m_{\rm pm}(\mathbf{x}_k^{(i)})$, $m_{\rm mg1}(\bar{\mathbf{x}}_k^{(i)})$ and $m_{\rm mg2}(\bar{\mathbf{x}}_k^{(i)})$ is based on the two simple rules illustrated in [1, Figs. 8-a) and 8-b), p. 1535] and can be summarized as follows. The first and second filtered pdfs (i.e., the first and the second forward estimates) of $\mathbf{x}_k^{(i)}$ are evaluated as

$$\vec{m}_{\rm fe1}\big(\mathbf{x}_k^{(i)}\big) = \vec{m}_{\rm fp}\big(\mathbf{x}_k^{(i)}\big) m_{\rm ms}\big(\mathbf{x}_k^{(i)}\big), \tag{3.4}$$

and

$$\vec{m}_{\rm fe2}\big(\mathbf{x}_k^{(i)}\big) = \vec{m}_{\rm fe1}\big(\mathbf{x}_k^{(i)}\big) m_{\rm pm}\big(\mathbf{x}_k^{(i)}\big), \tag{3.5}$$

respectively, where

$$m_{\rm ms}\big(\mathbf{x}_k^{(i)}\big) \triangleq \int f\big(\mathbf{y}_k\big|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)}\big)\, m_{\rm mg1}\big(\bar{\mathbf{x}}_k^{(i)}\big)\, d\bar{\mathbf{x}}_k^{(i)} \tag{3.6}$$

and $m_{\rm pm}(\mathbf{x}_k^{(i)})$ is defined in Eq. (3.3). Equations (3.4)-(3.6) describe the processing accomplished in the *measurement update* of the considered recursion. This is followed by the *time update*, in which the new predicted pdf (i.e., the new forward prediction)

$$\vec{m}_{\rm fp}\big(\mathbf{x}_{k+1}^{(i)}\big) = \int\int f\big(\mathbf{x}_{k+1}^{(i)}\big|\mathbf{x}_k^{(i)}, \bar{\mathbf{x}}_k^{(i)}\big) \vec{m}_{\rm fe2}\big(\mathbf{x}_k^{(i)}\big) m_{\rm mg2}\big(\bar{\mathbf{x}}_k^{(i)}\big) d\mathbf{x}_k\, d\bar{\mathbf{x}}_k^{(i)}, \tag{3.7}$$

is computed. The message passing procedure described above is initialised by setting $\vec{m}_{\rm fp}(\mathbf{x}_1^{(i)}) = f(\mathbf{x}_1^{(i)})$ (where $f(\mathbf{x}_1^{(i)})$ is the pdf resulting from the marginalization of $f(\mathbf{x}_1)$ with respect to $\bar{\mathbf{x}}_1^{(i)}$) in the first recursion and is run for $k = 1, 2, ..., T$. Once this procedure is over, BIF is executed for the substate $\mathbf{x}_k^{(i)}$; its $(T-k)-$th recursion (with $k = T-1, T-2, ..., 1$) can be represented as a *backward* message passing on the factor graph shown in Fig. 3.1. In this case, the messages $\overleftarrow{m}_{\rm bp}(\mathbf{x}_k^{(i)})$, $\overleftarrow{m}_{\rm be1}(\mathbf{x}_k^{(i)})$, $\overleftarrow{m}_{\rm be2}(\mathbf{x}_k^{(i)}) = \overleftarrow{m}_{\rm be}(\mathbf{x}_k^{(i)})$, that convey the *backward predicted* pdf of $\mathbf{x}_k^{(i)}$, the

[1] Generally speaking, a pseudo-measurement is a *fictitious* measurement that is computed on the basis of statistical information provided by a filtering algorithm different from the one benefiting from it.

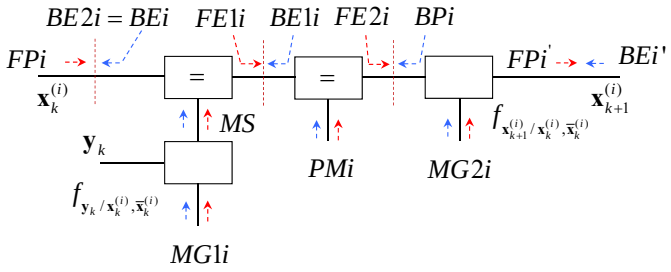


Figure 3.1: Factor graph involved in the $k-$th $((T-k)-$th) recursion of Bayesian filtering (BIF) for the substate $\mathbf{x}_k^{(i)}$ and forward (backward) message passing on it. The flow of messages in the forward (backward) pass are indicated by red (blue) arrows, respectively; the brown vertical lines cutting each graph identify the partitioning associated with formulas (3.10) (left cut), (3.11) (central cut) and (3.12) (right cut). The messages $\vec{m}_{\rm fp}(\mathbf{x}_k^{(i)})$, $\overleftarrow{m}_{\rm bp}(\mathbf{x}_k^{(i)})$, $\vec{m}_{\rm fp}(\mathbf{x}_{k+1}^{(i)})$, $\overleftarrow{m}_{\rm be}(\mathbf{x}_{k+1}^{(i)})$, $m_{\rm ms}(\mathbf{x}_k^{(i)})$, $m_{\rm mgl}(\bar{\mathbf{x}}_k^{(i)})$, $m_{\rm pm}(\mathbf{x}_k^{(i)})$, $\vec{m}_{\rm fel}(\mathbf{x}_k^{(i)})$ and $\overleftarrow{m}_{\rm bel}(\mathbf{x}_k^{(i)})$ are denoted FPi, BPi, $FPi^{'}$, $BEi^{'}$, MSi, $MGli$, PMi, $FEli$ and $BEli$ respectively, to ease reading.

first backward filtered pdf of $\mathbf{x}_k^{(i)}$ and the *second backward filtered* pdf of $\mathbf{x}_k^{(i)}$, respectively, are evaluated on the basis of the messages $\overleftarrow{m}_{\rm be}(\mathbf{x}_{k+1}^{(i)})$, $m_{\rm pm}(\mathbf{x}_k^{(i)})$ and $m_{\rm ms}(\mathbf{x}_k^{(i)})$, respectively; note that $\overleftarrow{m}_{\rm be}(\mathbf{x}_{k+1}^{(i)})$ represents the backward filtered pdf of $\mathbf{x}_k^{(i)}$ computed in the previous (i.e., in the $(T-(k+1))-$th) recursion of BIF. Moreover, the first and second backward filtered pdfs of $\mathbf{x}_k^{(i)}$ are evaluated as (see Fig. 3.1)

$$\overleftarrow{m}_{\rm be1}\big(\mathbf{x}_k^{(i)}\big) = \overleftarrow{m}_{\rm bp}\big(\mathbf{x}_k^{(i)}\big)\, m_{\rm pm}\big(\mathbf{x}_k^{(i)}\big), \tag{3.8}$$

and

$$\overleftarrow{m}_{\rm be2}\big(\mathbf{x}_k^{(i)}\big) = \overleftarrow{m}_{\rm be}\big(\mathbf{x}_k^{(i)}\big) = \overleftarrow{m}_{\rm be1}\big(\mathbf{x}_k^{(i)}\big)\, m_{\rm ms}\big(\mathbf{x}_k^{(i)}\big), \tag{3.9}$$

respectively, where $m_{\rm pm}(\mathbf{x}_k^{(i)})$ and $m_{\rm ms}(\mathbf{x}_k^{(i)})$ are still expressed by Eq. (3.3) and Eq. (3.6), respectively. The BIF message passing is initialised by setting $\overleftarrow{m}_{\rm be}(\mathbf{x}_T^{(i)}) = m_{\rm fe}(\mathbf{x}_T^{(i)})$ in its first recursion and is run for $k = T-1, T-2, ..., 1$. Once the backward pass is over, a solution to problem **P.2** becomes available for the substate $\mathbf{x}_k^{(i)}$, since the marginal smoothed pdf $f(\mathbf{x}_k^{(i)}, \mathbf{y}_{1:T}, \mathbf{z}_{1:T}^{(i)})$ (where $\mathbf{z}_{1:T}^{(i)}$ is the $P \cdot T-$dimensional vector resulting from the ordered concatenation of the all the observed pseudo-measurements $\{\mathbf{z}_k^{(i)}\}$) can be evaluated as[2]

$$f\big(\mathbf{x}_k^{(i)}, \mathbf{y}_{1:T}, \mathbf{z}_{1:T}^{(i)}\big) = \vec{m}_{\rm fp}\big(\mathbf{x}_k^{(i)}\big)\overleftarrow{m}_{\rm be2}\big(\mathbf{x}_k^{(i)}\big) \tag{3.10}$$

$$= \vec{m}_{\rm fe1}\big(\mathbf{x}_k^{(i)}\big)\overleftarrow{m}_{\rm be1}\big(\mathbf{x}_k^{(i)}\big) \tag{3.11}$$

$$= \vec{m}_{\rm fe2}\big(\mathbf{x}_k^{(i)}\big)\overleftarrow{m}_{\rm bp}\big(\mathbf{x}_k^{(i)}\big), \tag{3.12}$$

with $k = 1, 2, ..., T$. Note that, from a graphical viewpoint, formulas (3.10)-(3.12) can be related with the three different partitionings of the graph shown in Fig. 3.1 (where a specific partitioning is identified by a brown dashed vertical line cutting the graph in two parts).

Given the graphical model represented in Fig. 3.1, **step 3.** can be accomplished by adopting the same conceptual approach as [23, Sec. III] and [32, Par. II-B], where the factor graphs on which smoothing and filtering, respectively, are based are obtained by merging two sub-graphs, each referring to a distinct substate. For this reason, in this case, the graphical model for the whole state $\mathbf{x}_k$ is obtained by interconnecting two distinct factor graphs, each structured like the one shown in Fig. 3.1. In [32, Par. II-B], message passing on the resulting graph is described in detail for the case of *Bayesian filtering*. In this chapter, instead, our analysis of message passing concerns BIF *and smoothing only*. The devised graph and the messages passed on it are shown in

[2]Note that, similarly as refs. [23] and [1], a *joint* smoothed pdf is considered here in place of the corresponding *posterior* pdf.

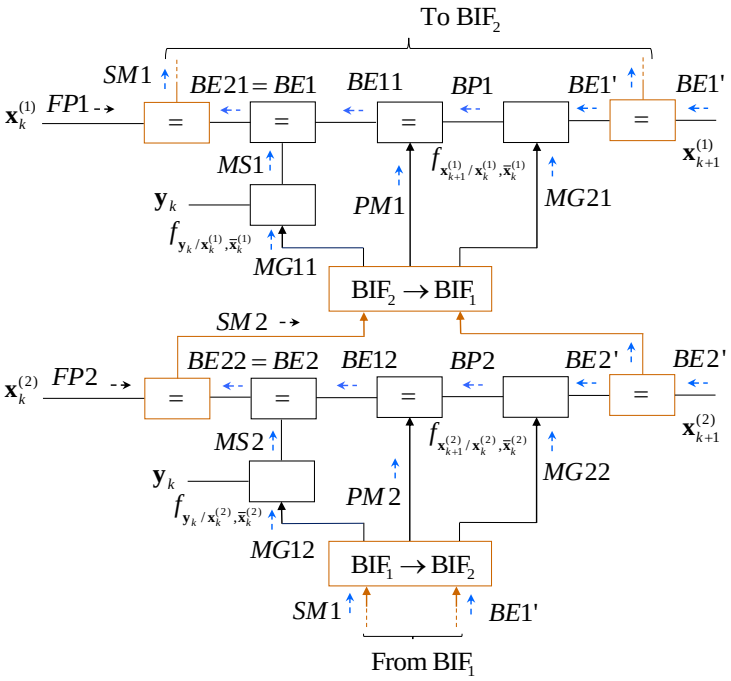


Figure 3.2: Graphical model based on the sub-graph shown in Fig. 3.1 and referring to the interconnection of two backward information filters. The message computed in the backward (forward) pass are identified by blue (black) arrows. The message $m_{\rm sm}(\mathbf{x}_k^{(i)})$ is denoted SMi to ease reading.

Fig. 3.2. Note that, in developing our graphical model, it has been assumed that the *smoothed* pdf referring to $\mathbf{x}_k^{(i)}$ (and conveyed by the message $m_{\rm sm}(\mathbf{x}_k^{(i)})$) is computed on the basis of Eq. (3.10), i.e. by merging the messages $\vec{m}_{\rm fp}(\mathbf{x}_k^{(i)})$ and $\overleftarrow{m}_{\rm be}(\mathbf{x}_k^{(i)}) = \overleftarrow{m}_{\rm be2}(\mathbf{x}_k^{(i)})$. Moreover, the following elements (identified by brown lines) have been added to its $i-$th sub-graph (with $i = 1$ and 2): a) two equality nodes; b) the block BIF$_i \rightarrow$BIF$_j$ for extracting useful information from the messages computed on the $i-$th sub-graph and delivered to the $j-$th one. The former elements allow the $i-$th backward information filter to generate copies of the messages $\overleftarrow{m}_{\rm be}(\mathbf{x}_{k+1}^{(i)})$ and $m_{\rm sm}(\mathbf{x}_k^{(i)})$, that are made available to the other sub-graphs. In the latter element, instead, the messages $m_{\rm pm}(\mathbf{x}_k^{(i)})$ (see Eq. (3.3)) and $m_{{\rm mg}q}(\bar{\mathbf{x}}_k^{(i)})$ (with $q = 1$ and 2; see Eqs. (3.6) and (3.7)) are computed; note that this block is connected to *oriented* edges only, i.e. to edges on which the flow of messages is unidirectional.

Given the graphical model represented in Fig. 3.2, **step 4**. can be easily accomplished. In fact, recursive BIF and smoothing algorithms can be derived by systematically applying the SPA to it after that a proper *scheduling* has been established for message passing. In doing so, we must always keep in mind that:

1) Message passing on the $i-$th subgraph represents BIF/smoothing for the substate $\mathbf{x}_k^{(i)}$; the exchange of messages between the sub-graphs, instead, allows us to represent the interaction of two interconnected BIF/smoothing algorithms in a effective and rigorous way.

2) Different approximations can be used for the predicted/filtered/smoothed pdfs computed in the message passing on each of the two sub-graphs and for the involved Markov/observation models. For this reason, generally speaking, the two interconnected filtering/BIF/smoothing algorithms are not required to be of the same type.

3) The $k-$th recursion of the overall BIF algorithm is fed by the backward estimates $\overleftarrow{m}_{\rm be}(\mathbf{x}_{k+1}^{(1)})$ $(BE1')$ and $\overleftarrow{m}_{\rm be}(\mathbf{x}_{k+1}^{(2)})$ $(BE2')$, and generates the new backward predictions $\overleftarrow{m}_{\rm bp}(\mathbf{x}_k^{(1)})$ $(BP1)$ and $\overleftarrow{m}_{\rm bp}(\mathbf{x}_k^{(2)})$ $(BP2)$, and the two couples of filtered densities $\{(\overleftarrow{m}_{\rm be1}(\mathbf{x}_k^{(i)}), \overleftarrow{m}_{\rm be2}(\mathbf{x}_k^{(i)})), i = 1, 2\}$ $(\{BE1i, BE2i, i = 1, 2\})$. Moreover, merging the predicted densities computed in the forward pass (i.e., the messages $\{FPi\}$) with the second backward filtered densities (i.e., the messages

$\{BE2i = BEi\}$) allows us to generate the smoothed pdfs for each substate according to Eq. (3.10). However, a *joint* filtered/smoothed density for the whole state $\mathbf{x}_k$ is *unavailable*.
4) Specific algorithms are employed to compute the pseudo-measurement and the nuisance substate pdfs in the BIF$_i$→BIF$_j$ blocks appearing in Fig. 3.2. These algorithms depend on the considered SSM and on the selected message scheduling; for this reason, a general description of their structure cannot be provided.
5) The graphical model shown in Fig. 3.2, unlike the one illustrated in Fig. 3.1, is *not cycle free*. The presence of cycles raises the problems of identifying all the messages that can be iteratively refined and establishing the order according to which they are computed. Generally speaking, iterative message passing on the devised graphical model involves both the couple of measurement updates and the backward prediction accomplished in each of the interconnected backward information filters. In fact, this should allow each filter to progressively refine the nuisance substate density employed in its second measurement update and backward prediction, and improve the quality of the pseudo-measurements exploited in its first measurement update. For this reason, if n_i iterations are run, the overall computational complexity of each recursion is multiplied by n_i.
The final important issue about the graphical model devised for both Bayesian filtering and BIF concerns the possible presence of *redundancy*. In all the considerations illustrated above, *disjoint* substates $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$ have been assumed. Actually, in ref. [32], it has been shown that our graphical approach can be also employed if the substates $\mathbf{x}_k^{(1)}$ and $\mathbf{x}_k^{(2)}$ cover $\mathbf{x}_k$, but do not necessarily form a partition of it. In other words, some overlapping between these two substates is admitted. When this occurs, the forward/backward filtering algorithm run over the whole graphical model contains a form of *redundancy*, since $N_d \triangleq D_1 + D_2 - D$ elements of the state vector $\mathbf{x}_k$ are independently estimated by the interconnected forward/backward filters. The parameter N_d can be considered as the *degree of redundancy* characterizing the filtering/smoothing algorithm. Moreover, in ref. [32], it has been shown that the presence of redundancy in a Bayesian filtering algorithm can significantly enhance its tracking capability (i.e., reduce its probability of divergence); however, this result is obtained at the price of an increased complexity with respect to the case in which the interconnected filters are run over disjoint substates.

3.3 Double Backward Information Filtering and Smoothing Algorithms for Conditionally Linear Gaussian State Space Models

In this section we focus on the development of two new DBS algorithms for conditionally linear Gaussian models. We first describe the graphical models on which these algorithms are based; then, we provide a detailed description of the computed messages and their scheduling in a specific case.

3.3.1 Graphical Modelling

In this paragraph, we focus on a specific instance of the graphical model illustrated in Fig. 3.2, since we make the same specific choices as ref. [32] for both the considered SSM and the two Bayesian filters employed in the forward pass. For this reason, we assume that: a) the SSM described by eqs. (3.1)–(3.2) is conditionally linear Gaussian [7], [1], [6], so that its state vector $\mathbf{x}_k$ can be partitioned into its *linear component* $\mathbf{x}_k^{(L)}$ and its *nonlinear component* $\mathbf{x}_k^{(N)}$ (having sizes D_L and D_N, respectively, with $D_N + D_L = D$); b) the dual Bayesian filter employed in the forward pass results from the interconnection of an *extended Kalman filter* with a *particle filter*[3] (these filters are denoted F_1 and F_2, respectively), as described in detail in ref. [32]. As far as the last point is concerned, it is also worth mentioning that, on the one hand, filter F_2 estimates the nonlinear state component only (so that $\mathbf{x}_k^{(2)} = \mathbf{x}_k^{(N)}$ and $\bar{\mathbf{x}}_k^{(2)} = \mathbf{x}_k^{(L)}$) and approximates the

[3] In particular, a *sequential importance resampling* filter is employed [2].

predicted/filtered densities of this component through a set of N_p weighted particles. On the other hand, filter F_1 employs a Gaussian approximation of all its predicted/filtered densities, and works on the *whole system state* or on the *linear state component*. In the first case (denoted **C.1** in the following), we have that $\mathbf{x}_k^{(1)} = \mathbf{x}_k$ and $\bar{\mathbf{x}}_k^{(1)}$ is empty, so that both F_1 and F_2 estimate the nonlinear state component (for this reason, the corresponding degree of redundancy in the developed smoothing algorithm is $N_d = D_N$); in the second case (denoted **C.2** in the following), instead, $\mathbf{x}_k^{(1)} = \mathbf{x}_k^{(L)}$ and $\bar{\mathbf{x}}_k^{(1)} = \mathbf{x}_k^{(N)}$, so that filters F_1 and F_2 estimate *disjoint* substates (consequently, $N_d = 0$).

Our selection of the forward filtering scheme has the following implications on the developed DBIF scheme. The first backward information filter (denoted BIF_1) is the backward filter associated with an extended Kalman filter operating over on the *whole system state* (case **C.1**) or on the *linear state component* (case **C.2**). The second backward filter (denoted BIF_2), instead, is a backward filter associated with a particle filter operating on the nonlinear state component only. In practice, following [7, 23, 62], BIF_2 is employed to update the weights of all the elements of the particle set generated by filter F_2 in the forward pass. Then, based on the graphical model shown in Fig. 3.2, the factor graph illustrated in Fig. 3.3 can be drawn for case **C.1**. It is important to point out that:

1) The first backward information filter (BIF_1) is based on *linearised* (and, consequently, *approximate*) Markov/measurement models, whereas the second one (BIF_2) relies on *exact* models, as explained in more detail below. These models are the same as those employed in ref. [32].

2) Since the nuisance substate $\bar{\mathbf{x}}_k^{(1)}$ is empty, no marginalization is required in BIF_1; for this reason, the messages $\{m_{\mathrm{mg}q}(\bar{\mathbf{x}}_k^{(1)});\, q = 1, 2\}$ (i.e., $MG11$ and $MG21$) visible in Fig. 3.2 do not appear in Fig. 3.3. Moreover, the message $m_{\mathrm{sm}}(\mathbf{x}_k^{(1)}) = m_{\mathrm{sm}}(\mathbf{x}_k)$ is generated on the basis of Eq. (3.11), instead of Eq. (3.10).

3) The backward filtered pdf $\overleftarrow{m}_{\mathrm{be}}(\mathbf{x}_{k+1}^{(2)}) = \overleftarrow{m}_{\mathrm{be}}(\mathbf{x}_{k+1}^{(N)})$ and the smoothed pdf $m_{\mathrm{sm}}(\mathbf{x}_k^{(2)}) = m_{\mathrm{sm}}(\mathbf{x}_k^{(N)})$ (i.e., the messages $BE2'$ and $SM2$, respectively) feed the $\mathrm{BIF}_2 \rightarrow \mathrm{BIF}_1$ block, where they are processed jointly to generate the pseudo-measurement message $m_{\mathrm{pm}}(\mathbf{x}_k^{(1)}) = m_{\mathrm{pm}}(\mathbf{x}_k)$ ($PM1$) feeding filter F_1. Similarly, the backward filtered pdf $\overleftarrow{m}_{\mathrm{be}}(\mathbf{x}_{k+1}^{(1)}) = \overleftarrow{m}_{\mathrm{be}}(\mathbf{x}_{k+1})$ ($BE1'$) and the smoothed pdf $m_{\mathrm{sm}}(\mathbf{x}_k^{(1)}) = m_{\mathrm{sm}}(\mathbf{x}_k)$ ($SM1$) feed the $\mathrm{BIF}_1 \rightarrow \mathrm{BIF}_2$ block, where the pseudo-measurement message $m_{\mathrm{pm}}(\mathbf{x}_k^{(2)}) = m_{\mathrm{pm}}(\mathbf{x}_k^{(N)})$ ($PM2$) and the messages $\{m_{\mathrm{mg}q}(\bar{\mathbf{x}}_k^{(2)}) = m_{\mathrm{mg}q}(\mathbf{x}_k^{(L)});\, q = 1, 2\}$ (i.e., $MG12$ and $MG22$) are evaluated.

In the remaining part of this paragraph, we first provide various details about the backward filters BIF_1 and BIF_2, and the way pseudo-measurements are computed for each of them; then, we comment on how the factor graph shown in Fig. 3.3 should be modified if case **C.2** is considered.

BIF_1 - This backward filter is based on the *linearized* versions of Eqs. (3.1) and (3.2), i.e. on the models (e.g., see [3, pp. 194-195] and [32, Par. III-A])

$$\mathbf{x}_{k+1} = \mathbf{F}_k\, \mathbf{x}_k + \mathbf{u}_k + \mathbf{w}_k \tag{3.13}$$

and

$$\mathbf{y}_k = \mathbf{H}_k^T\, \mathbf{x}_k + \mathbf{v}_k + \mathbf{e}_k, \tag{3.14}$$

respectively; here, $\mathbf{F}_k \triangleq [\partial \mathbf{f}_k(\mathbf{x})/\partial \mathbf{x}]_{\mathbf{x}=\mathbf{x}_{\mathrm{fe},k}}$, $\mathbf{u}_k \triangleq \mathbf{f}_k(\mathbf{x}_{\mathrm{fe},k}) - \mathbf{F}_k\, \mathbf{x}_{\mathrm{fe},k}$, $\mathbf{H}_k^T \triangleq [\partial \mathbf{h}_k(\mathbf{x})/\partial \mathbf{x}]_{\mathbf{x}=\mathbf{x}_{\mathrm{fp},k}}$, $\mathbf{v}_k \triangleq \mathbf{h}_k(\mathbf{x}_{\mathrm{fp},k}) - \mathbf{H}_k^T \mathbf{x}_{\mathrm{fp},k}$ and $\mathbf{x}_{\mathrm{fp},k}$ ($\mathbf{x}_{\mathrm{fe},k}$) is the *forward prediction* (*forward estimate*) of $\mathbf{x}_k$ computed by F_1 in its $(k-1)$−th (k−th) recursion. Consequently, the approximate models

$$\tilde{f}(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{F}_k\, \mathbf{x}_k + \mathbf{u}_k, \mathbf{C}_w) \tag{3.15}$$

and

$$\tilde{f}(\mathbf{y}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{H}_k^T\, \mathbf{x}_k + \mathbf{v}_k, \mathbf{C}_e) \tag{3.16}$$

appear in the graphical model shown in Fig. 3.3.

BIF_2 - In developing this backward filter, the state vector $\mathbf{x}_k$ is represented as the ordered concatenation of its linear component $\mathbf{x}_k^{(L)} \triangleq [x_{0,k}^{(L)}, x_{1,k}^{(L)}, ..., x_{D_L-1,k}^{(L)}]^T$ and its nonlinear component $\mathbf{x}_k^{(N)} \triangleq [x_{0,k}^{(N)}, x_{1,k}^{(N)}, ..., x_{D_N-1,k}^{(N)}]^T$. Based on [1, eq. (3)], the Markov model

$$\mathbf{x}_{k+1}^{(N)} = \mathbf{A}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big)\mathbf{x}_k^{(L)} + \mathbf{f}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big) + \mathbf{w}_k^{(N)} \tag{3.17}$$

is adopted for the nonlinear state component (this model corresponds to the last D_N lines of Eq. (3.1)); here, $\mathbf{f}_k^{(N)}(\mathbf{x}_k^{(N)})$ ($\mathbf{A}_k^{(N)}(\mathbf{x}_k^{(N)})$) is a time-varying D_N-dimensional real function ($D_N \times D_L$ real matrix) and $\mathbf{w}_k^{(N)}$ consists of the last D_N elements of the noise term $\mathbf{w}_k$ appearing in Eq. (3.1) (the covariance matrix of $\mathbf{w}_k^{(N)}$ is denoted $\mathbf{C}_w^{(N)}$). Moreover, it is assumed that the observation model (3.2) can be put in the form (see [32, eq. (31)] or [1, eq. (4)])

$$\mathbf{y}_k = \mathbf{g}_k\big(\mathbf{x}_k^{(N)}\big) + \mathbf{B}_k\big(\mathbf{x}_k^{(N)}\big)\mathbf{x}_k^{(L)} + \mathbf{e}_k, \tag{3.18}$$

where $\mathbf{g}_k(\mathbf{x}_k^{(N)})$ ($\mathbf{B}_k(\mathbf{x}_k^{(N)})$) is a time-varying $P-$dimensional real function ($P \times D_L$ real matrix). Consequently, the considered backward filter is based on the *exact* pdfs

$$f\big(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{x}_k^{(N)}; \mathbf{A}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big)\mathbf{x}_k^{(L)} + \mathbf{f}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big), \mathbf{C}_w^{(N)}\big) \tag{3.19}$$

and

$$f\big(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{x}_k; \mathbf{g}_k\big(\mathbf{x}_k^{(N)}\big) + \mathbf{B}_k\big(\mathbf{x}_k^{(N)}\big)\mathbf{x}_k^{(L)}, \mathbf{C}_e\big), \tag{3.20}$$

both appearing in the graphical model drawn in Fig. 3.3.

Computation of the pseudo-measurements for the first backward filter - Filter BIF_1 is fed by pseudo-measurement information about the *whole state* $\mathbf{x}_k$. The method for computing these information is similar to the one illustrated in ref. [23, Sects. III-IV] and can be summarised as follows. The pseudo-measurements about the nonlinear state component are represented by the N_p particles conveyed by the smoothed pdf $m_{\text{sm}}(\mathbf{x}_k^{(N)})$ ($SM2$). On the other hand, N_p pseudo-measurements about the linear state component are evaluated by means of the same method employed by *marginalized particle filtering* (MPF) for this task. This method is based on the idea that the random vector (see Eq. (3.17))

$$\mathbf{z}_k^{(L)} \triangleq \mathbf{x}_{k+1}^{(N)} - \mathbf{f}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big), \tag{3.21}$$

depending on the *nonlinear state component only*, must equal the sum

$$\mathbf{A}_k^{(N)}\big(\mathbf{x}_k^{(N)}\big)\mathbf{x}_k^{(L)} + \mathbf{w}_k^{(N)}, \tag{3.22}$$

that depends on the *linear state component*. For this reason, N_p realizations of $\mathbf{z}_k^{(L)}$ (3.21) are computed in the $\text{BIF}_2 \rightarrow \text{BIF}_1$ block on the basis of the messages $\overleftarrow{m}_{\text{be}}(\mathbf{x}_{k+1}^{(N)})$ ($BE2'$) and $m_{\text{sm}}(\mathbf{x}_k^{(N)})$, and are treated as measurements about $\mathbf{x}_k^{(L)}$.

Computation of the pseudo-measurements for the second backward filter - The messages $\overleftarrow{m}_{\text{be}}(\mathbf{x}_{k+1})$ ($BE1'$) and $m_{\text{sm}}(\mathbf{x}_k)$ ($SM1$) feeding the $\text{BIF}_1 \rightarrow \text{BIF}_2$ block are employed for: a) generating the messages $\{m_{\text{mg}q}(\mathbf{x}_k^{(L)});\ q = 1, 2\}$ required to integrate out the dependence of the state update and measurement models (i.e., of the densities $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ (3.19) and $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ (3.20), respectively) on the substate $\mathbf{x}_k^{(L)}$; b) generating pseudo-measurement information about $\mathbf{x}_k^{(N)}$. As far as the last point is concerned, the approach we adopt is the same as that developed for *dual marginalized particle filtering* (dual MPF) in ref. [1, Sec. V] and also adopted in particle smoothing [23, Sects. III-IV]. The approach relies on the Markov model

$$\mathbf{x}_{k+1}^{(L)} = \mathbf{A}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)\mathbf{x}_k^{(L)} + \mathbf{f}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big) + \mathbf{w}_k^{(L)}, \tag{3.23}$$

referring to the *linear* state component (see [23, eq. (1)] or [1, eq. (3)]); in the last expression, $\mathbf{f}_k^{(L)}(\mathbf{x}_k^{(N)})$ ($\mathbf{A}_k^{(L)}(\mathbf{x}_k^{(N)})$) is a time-varying D_L-dimensional real function ($D_L \times D_L$ real matrix),

and $\mathbf{w}_k^{(L)}$ consists of the first D_L elements of the noise term $\mathbf{w}_k$ appearing in (3.1) (the covariance matrix of $\mathbf{w}_k^{(L)}$ is denoted $\mathbf{C}_w^{(L)}$, and independence between $\{\mathbf{w}_k^{(L)}\}$ and $\{\mathbf{w}_k^{(N)}\}$ is assumed for simplicity). From Eq. (3.23) it is easily inferred that the random vector

$$\mathbf{z}_k^{(N)} \triangleq \mathbf{x}_{k+1}^{(L)} - \mathbf{A}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)\,\mathbf{x}_k^{(L)}, \tag{3.24}$$

must equal the sum

$$\mathbf{f}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big) + \mathbf{w}_k^{(L)}, \tag{3.25}$$

that depends on $\mathbf{x}_k^{(N)}$ *only*; for this reason, $\mathbf{z}_k^{(N)}$ (3.24) can be interpreted as a pseudo-measurement about $\mathbf{x}_k^{(N)}$. In this case, the pseudo-measurement information is conveyed by the message $m_{\rm pm}(\mathbf{x}_k^{(N)})$ $(PM2)$ that expresses the *correlation* between the pdf of the random vector $\mathbf{z}_k^{(N)}$ (3.24) (computed on the basis of the statistical information about the linear state component made available by BIF_1) and the pdf obtained for $\mathbf{z}_k^{(N)}$ under the assumption that this vector is expressed by Eq. (3.25). The message $m_{\rm pm}(\mathbf{x}_k^{(N)})$ is evaluated for each of the particles representing $\mathbf{x}_k^{(N)}$ in BIF_2; this results in a set of N_p particle weights employed in the first measurement update of BIF_2 and different from those computed on the basis of $\mathbf{y}_k$ (3.18) in its second measurement update.

A graphical model similar to the one shown in Fig. 3.3 can be easily derived from the general model appearing in Fig. 3.2 for case **C.2** too. The relevant differences with respect to case **C.1** can be summarized as follows:

1) The backward filters BIF_1 and BIF_2 estimate $\mathbf{x}_k^{(1)} = \mathbf{x}_k^{(L)}$ and $\mathbf{x}_k^{(2)} = \mathbf{x}_k^{(N)}$, respectively; consequently, their nuisance substates are $\bar{\mathbf{x}}_k^{(1)} = \mathbf{x}_k^{(N)}$ and $\bar{\mathbf{x}}_k^{(2)} = \mathbf{x}_k^{(L)}$, respectively.

2) The $\text{BIF}_2 \rightarrow \text{BIF}_1$ block is fed by the backward predicted/smoothed pdfs computed by BIF_2; such pdfs are employed for: a) generating the messages $m_{\rm mg1}(\mathbf{x}_k^{(N)})$ $(MG11)$ and $m_{\rm mg2}(\mathbf{x}_k^{(N)})$ $(MG21)$ required to integrate out the dependence of the Markov model (see Eq. (3.23))

$$f\big(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{x}_k^{(L)};\mathbf{A}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)\mathbf{x}_k^{(L)} + \mathbf{f}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big),\mathbf{C}_w^{(L)}\big) \tag{3.26}$$

and of the measurement model $f(\mathbf{y}_k|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)})$ (3.20), respectively, on $\mathbf{x}_k^{(N)}$; b) generating pseudo-measurement information about the substate $\mathbf{x}_k^{(L)}$ *only*. As far as point a) is concerned, it is also important to point out that the model $f(\mathbf{y}_k|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)})$ $(f(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)}))$ on which BIF_1 is based can be derived from Eq. (3.20) (Eq. (3.26)) after setting $\mathbf{x}_k^{(N)} = \mathbf{x}_{{\rm fp},k}^{(N)}$ $(\mathbf{x}_k^{(N)} = \mathbf{x}_{{\rm fe},k}^{(N)})$; here, $\mathbf{x}_{{\rm fp},k}^{(N)}$ $(\mathbf{x}_{{\rm fe},k}^{(N)})$ denotes the prediction (the estimate) of $\mathbf{x}_k^{(N)}$ evaluated by the filter F_2 in the forward pass (further details about this can be found in ref. [32, Par. III-A])

The derivation of specific DBS algorithms based on the graphical model illustrated in Fig. 3.3 requires defining the scheduling of the messages passed on it and deriving mathematical expressions for such messages. These issues are investigated in detail in the following paragraph.

3.3.2 Message Scheduling and Computation

In this paragraph, the scheduling of a new *recursive smoothing algorithm*, called *double Bayesian smoothing algorithm* (DBSA) and based on the graphical model illustrated in Fig. 3.3, and a simplified version of it are described. Moreover, the expression of the messages computed by the DBSA are illustrated.

The scheduling adopted in the DBSA mimics the one employed in ref. [23] (which, in turn, has been inspired by [62] and [7]). Moreover, in devising it, the presence of cycles in the underlying graphical model has been accounted for by allowing multiple passes of messages over the edges which such cycles consist of; this explains why an iterative procedure is embedded in each recursion of the DBSA. Our description of the devised scheduling is based on Fig. 3.4, that refers to the $(T-k)-$th recursion of the backward pass of the DBSA (with $k = T-1, T-2, ..., 1$) and to

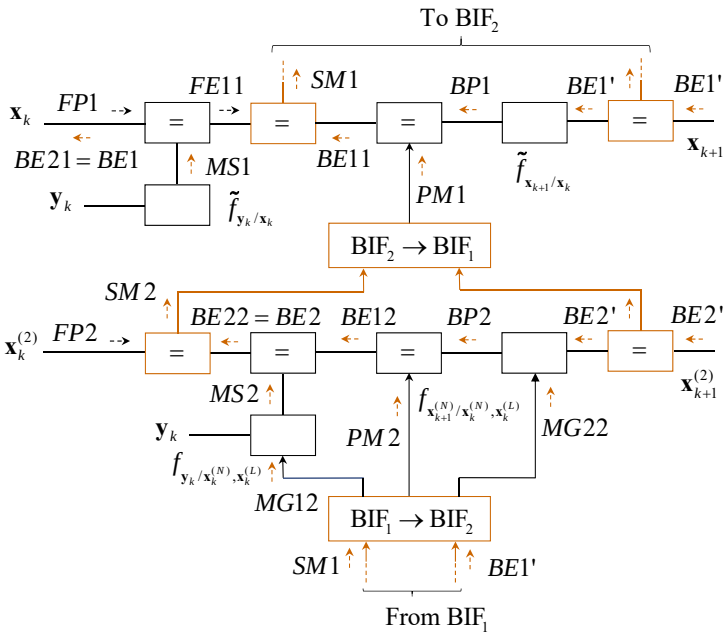


Figure 3.3: Graphical model referring to the interconnection of two backward information filters, one paired with an extended Kalman filter, the other one with a particle filter.

the $n-$th iteration accomplished within this recursion (with $n = 1, 2, ..., n_i$, where n_i represents the overall number of iterations). Note that, in this figure, a simpler notation is adopted for most of the considered messages to ease reading; in particular, the symbols q , $q^{(n)}$, qL, $qL^{(n)}$, qN and $qN^{(n)}$ are employed to represent the messages $m_q(\mathbf{x}_k)$, $m_q^{(n)}(\mathbf{x}_k)$, $m_q(\mathbf{x}_k^{(L)})$, $m_q^{(n)}(\mathbf{x}_k^{(L)})$, $m_q(\mathbf{x}_k^{(N)})$, $m_q^{(n)}(\mathbf{x}_k^{(N)})$, respectively (independently of the presence of an arrow and of its direction in the considered message), and the presence of the superscript (n) in a given message means that such a message is computed in the n-th iteration. Moreover, each of the passed messages conveys a Gaussian pdf or a pdf in particle form. In the first case, the pdf of a state/substate $\mathbf{x}$ is conveyed by the message

$$m_G(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \eta, \mathbf{C}), \tag{3.27}$$

where η and $\mathbf{C}$ denote the mean and the covariance of $\mathbf{x}$, respectively. In the second case, instead, its pdf is conveyed by the message

$$m_P(\mathbf{x}) = \sum_{j=1}^{N_p} m_{P,j}(\mathbf{x}), \tag{3.28}$$

where

$$m_{P,j}(\mathbf{x}) \triangleq w_j \, \delta(\mathbf{x} - \mathbf{x}_j) \tag{3.29}$$

is its $j-$th *component*; this represents the contribution of the $j-$th particle $\mathbf{x}_j$ and its weight w_j to $m_P(\mathbf{x})$ (3.28). The nature of each message can be easily inferred from Fig. 3.4, since Gaussian messages and messages in particle form are identified by blue and red arrows, respectively.

Before analysing the adopted scheduling, we need to define the *input* messages feeding the considered recursion of the DBSA and the outputs that such a recursion produces. In the considered recursion, the DBSA *input* messages originate from:

1) The k-th recursion of the forward pass. These messages have been generated by the DBF technique paired with the considered BIF scheme and, in particular, by the DBF algorithm derived in ref. [32, Par. III-B], and have been stored (so that they are made available to the backward pass).

2) The previous recursion (i.e., the $(T - k - 1)-$th recursion) of the DBSA itself.

As far as the input messages computed in the forward pass are concerned, BIF_1 is fed by the Gaussian messages (see Fig. 3.4)

$$\vec{m}_{\text{fp}}(\mathbf{x}_k) \triangleq \mathcal{N}(\mathbf{x}_k; \eta_{\text{fp},k}, \mathbf{C}_{\text{fp},k}). \tag{3.30}$$

and

$$\vec{m}_{\text{fe1}}(\mathbf{x}_k) \triangleq \mathcal{N}(\mathbf{x}_k; \eta_{\text{fe1},k}, \mathbf{C}_{\text{fe1},k}), \tag{3.31}$$

that convey the predicted pdf and the first filtered pdf, respectively, computed by F_1 (in its $(k-1)-$th and in its $k-$th recursion, respectively). The covariance matrix $\mathbf{C}_{\text{fe1},k}$ and the mean vector $\eta_{\text{fe1},k}$ are evaluated on the basis of the associated precision matrix (see [23, Eqs. (14)-(17)])

$$\mathbf{W}_{\text{fe1},k} = \mathbf{H}_k \, \mathbf{W}_e \, \mathbf{H}_k^T + \mathbf{W}_{\text{fp},k} \tag{3.32}$$

and of the associated transformed mean vector

$$\mathbf{w}_{\text{fe1},k} = \mathbf{H}_k \, \mathbf{W}_e (\mathbf{y}_k - \mathbf{v}_k) + \mathbf{w}_{\text{fp},k}, \tag{3.33}$$

respectively; here, $\mathbf{W}_e \triangleq \mathbf{C}_e^{-1}$, $\mathbf{W}_{\text{fp},k} \triangleq (\mathbf{C}_{\text{fp},k})^{-1}$ and $\mathbf{w}_{\text{fp},k} \triangleq \mathbf{W}_{\text{fp},k} \, \eta_{\text{fp},k}$.

The backward filter BIF_2, instead, is fed by the forward messages $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(N)})$ and $\vec{m}_{\text{fe1}}(\mathbf{x}_k^{(N)})$, both in particle form (see Fig. 3.4); their $j-$th components are represented by

$$\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(N)}) \triangleq w_p \, \delta(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)}), \tag{3.34}$$

and

$$\vec{m}_{\text{fe1},j}(\mathbf{x}_k^{(N)}) \triangleq w_{\text{fe},k,j} \, \delta(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)}), \tag{3.35}$$

respectively, with $j = 1, 2, ..., N_p$; here, $\mathbf{x}_{k,j}^{(N)}$ is the $j-$th particle predicted by F_2 in the $(k-1)$-th recursion of the forward pass (i.e., the $j-$th element of the particle set $S_k \triangleq \{\mathbf{x}_{k,1}^{(N)}, \mathbf{x}_{k,2}^{(N)}, ..., \mathbf{x}_{k,N_p}^{(N)}\}$), whereas $w_p \triangleq 1/N_p$ and $w_{\text{fe},k,j}$ represent the (normalised) weights assigned to this particle in the messages $\vec{m}_{\text{fp}}(\mathbf{x}_k^{(N)})$ and $\vec{m}_{\text{fe1}}(\mathbf{x}_k^{(N)})$, respectively.

On the other hand, the input messages originating from the previous recursion of the backward pass are the backward filtered Gaussian pdf

$$\overleftarrow{m}_{\text{be}}(\mathbf{x}_{k+1}) \triangleq \mathcal{N}(\mathbf{x}_{k+1}; \eta_{\text{be},k+1}, \mathbf{C}_{\text{be},k+1}) \tag{3.36}$$

and the backward pdf

$$\overleftarrow{m}_{\text{be}}(\mathbf{x}_{k+1}^{(N)}) \triangleq \delta(\mathbf{x}_{k+1}^{(N)} - \mathbf{x}_{\text{be},k+1}^{(N)}), \tag{3.37}$$

that represents $\mathbf{x}_{k+1}^{(N)}$ through a single particle having unit weight; these are computed by BIF_1 and BIF_2, respectively. Consequently, in the considered recursion of the backward pass, all the forward/backward input messages described above are processed to compute: 1) the new backward pdfs $\overleftarrow{m}_{\text{be}}(\mathbf{x}_k)$ and $\overleftarrow{m}_{\text{be}}(\mathbf{x}_k^{(N)})$; 2) the smoothed statistical information about $\mathbf{x}_k$ ($\mathbf{x}_k^{(N)}$) by properly merging forward and backward messages generated by F_1 and BIF_1 (F_2 and BIF_2). As far as the last point is concerned, the evaluation of smoothed information is based on the same conceptual approach as [7, 23, 62]. In fact, in our work, the *joint* smoothing pdf $f(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ is estimated by providing multiple (say, M) *realizations* of it. A single realization (i.e., a single *smoothed* state trajectory) is computed in each backward pass; consequently, generating the whole output of the DBSA requires running a single forward pass and M distinct backward passes. Moreover, the evaluation of the smoothed information is based on the factorisation (3.10) or (3.11). In fact, these formulas are exploited to merge the statistical information emerging from the forward pass with that computed in any of the M backward passes.

The message passing on which the DBSA is based can be divided in the three consecutive phases listed below.

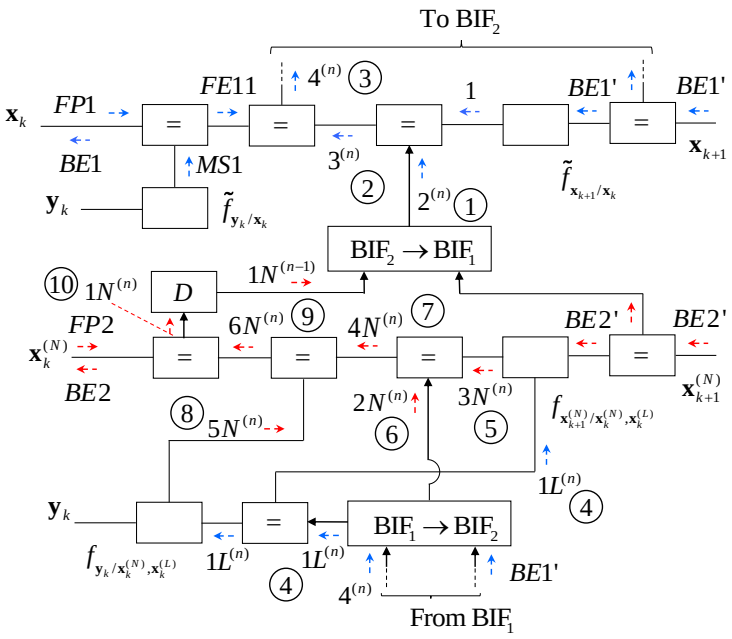


Figure 3.4: Representation of the message scheduling accomplished within the $(T-k)-$th recursion of the backward pass of the DBSA; the circled integers $1-10$ specify the order according to which the passed messages are computed in the $n-$th iteration embedded in the considered recursion. Blue and red arrows are employed to identify Gaussian messages and messages in particle form, respectively.

Phase I - In the first phase, the backward predicted pdf $\overleftarrow{m}_1(\mathbf{x}_k)$ (1) is computed on the basis of the backward filtered pdf $\overleftarrow{m}_{\text{be}}(\mathbf{x}_{k+1})$ ($BE1'$).

Phase II - In this phase, an iterative procedure for computing and progressively refining the first backward filtered and the smoothed pdfs of the whole state (BIF_1), and the second filtered and the smoothed pdfs of the nonlinear state component (BIF_2) is carried out. More specifically, in the n-th iteration of this procedure (with $n = 1, 2, ..., n_i$), the ordered computation of the following messages is accomplished in eight consecutive steps (see Fig. 3.4): 1) $m_2^{(n)}(\mathbf{x}_k)$ ($2^{(n)}$; pdf conveying pseudo-measurement information about $\mathbf{x}_k$); 2) $\overleftarrow{m}_3^{(n)}(\mathbf{x}_k)$ ($3^{(n)}$; *first backward filtered pdf of* $\mathbf{x}_k$); 3) $m_4^{(n)}(\mathbf{x}_k)$ ($4^{(n)}$; *smoothed pdf of* $\mathbf{x}_k$); 4) $m_1^{(n)}(\mathbf{x}_k^{(L)})$ ($1L^{(n)}$; pdf for integrating out the dependence of $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ and $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ on $\mathbf{x}_k^{(L)}$), $\overleftarrow{m}_3^{(n)}(\mathbf{x}_k^{(N)})$ ($3N^{(n)}$; *backward predicted pdf of* $\mathbf{x}_k^{(N)}$); 5) $m_2^{(n)}(\mathbf{x}_k^{(N)})$ ($2N^{(n)}$; pdf conveying pseudo-measurement information about $\mathbf{x}_k^{(N)}$); 6) $m_4^{(n)}(\mathbf{x}_k^{(N)})$ ($4N^{(n)}$; *first backward filtered pdf of* $\mathbf{x}_k^{(N)}$); 7) $m_5^{(n)}(\mathbf{x}_k^{(N)})$ ($5N^{(n)}$; message conveying measurement-based information about $\mathbf{x}_k^{(N)}$); 8) $m_6^{(n)}(\mathbf{x}_k^{(N)})$ ($6N^{(n)}$; *second backward filtered pdf of* $\mathbf{x}_k^{(N)}$), $m_1^{(n)}(\mathbf{x}_k^{(N)})$ ($1N^{(n)}$; *smoothed pdf of* $\mathbf{x}_k^{(N)}$). Note that the message $m_1^{(n)}(\mathbf{x}_k^{(N)})$ computed in the last step of the n-th iteration is stored in a memory cell (identified by the label 'D'), so that it becomes available at the beginning of the next iteration.

Phase III - In the third phase, the final smoothed pdf $m_1^{(n_i)}(\mathbf{x}_k^{(N)})$ is exploited to compute: a) the final backward pdf (i.e., the output message of BIF_2) $\overleftarrow{m}_{\text{be}}(\mathbf{x}_k^{(N)})$; b) the new pseudo-measurement message $m_2^{(n_i+1)}(\mathbf{x}_k)$, the final backward filtered pdf $\overleftarrow{m}_3^{(n_i+1)}(\mathbf{x}_k)$, the final smoothed pdf $m_4^{(n_i+1)}(\mathbf{x}_k)$ of $\mathbf{x}_k$ and, finally, the final backward pdf (i.e., the output message of BIF_1) $m_{\text{be}}(\mathbf{x}_k)$.

3.3.3 Message Computation

The expressions of all the messages evaluated by the DBSA, with the exception of the messages emerging from the $\text{BIF}_1 \rightarrow \text{BIF}_2$ block and the $\text{BIF}_2 \rightarrow \text{BIF}_1$ block, can be easily derived by applying the few mathematical rules listed in Tables A.1–A.3; all such rules result from the application

of the SPA to equality nodes or nodes representing Gaussian functions. The derivation of the algorithms for computing the pseudo-measurement messages $m_2^{(n)}(\mathbf{x}_k)$ $(2^{(n)})$ and $m_2^{(n)}(\mathbf{x}_k^{(N)})$ $(2N^{(n)})$ emerging from the $\text{BIF}_2\rightarrow\text{BIF}_1$ block and the $\text{BIF}_1\rightarrow\text{BIF}_2$ block, respectively, is based on the same approach illustrated in refs. [23, Par. IV-B] and [1, Sects. IV-V]. On the other hand, the message $m_1^{(n)}(\mathbf{x}_k^{(L)})$ $(1L^{(n)})$ originating from the $\text{BIF}_1\rightarrow\text{BIF}_2$ block results from marginalizing $m_4^{(n)}(\mathbf{x}_k)$ $(4^{(n)})$ with respect to $\mathbf{x}_k^{(N)}$.

In the remaining part of this paragraph, the expressions of all the messages computed in each of the three phases described above are provided for the $(T-k)-$th recursion of the backward pass; the derivation of these expressions is sketched in Appendix D and is omitted here for space limitations.

Phase I - The computation of the *backward predicted pdf*

$$\overleftarrow{m}_1\big(\mathbf{x}_k\big) \triangleq \mathcal{N}\big(\mathbf{x}_k; \eta_{1,k}, \mathbf{C}_{1,k}\big) \tag{3.38}$$

of $\mathbf{x}_k$ involves $\overleftarrow{m}_{\text{be}}(\mathbf{x}_{k+1})$ (3.36) and the pdf $\tilde{f}(\mathbf{x}_{k+1}|\mathbf{x}_k)$ (3.15). Its parameters $\eta_{1,k}$ and $\mathbf{C}_{1,k}$ are evaluated on the basis of the associated precision matrix

$$\mathbf{W}_{1,k} \triangleq \big(\mathbf{C}_{1,k}\big)^{-1} = \mathbf{F}_k^T\, \mathbf{P}_{k+1}\, \mathbf{W}_{\text{be},k+1} \mathbf{F}_k \tag{3.39}$$

and of the associated transformed mean vector

$$\mathbf{w}_{1,k} \triangleq \mathbf{W}_{1,k}\eta_{1,k} = \mathbf{F}_k^T[\mathbf{P}_{k+1}\, \mathbf{w}_{\text{be},k+1} - \mathbf{W}_{\text{be},k+1}\, \mathbf{Q}_{k+1}\, \mathbf{W}_w\, \mathbf{u}_k], \tag{3.40}$$

respectively; here, $\mathbf{W}_{\text{be},k+1} \triangleq (\mathbf{C}_{\text{be},k+1})^{-1}$, $\mathbf{P}_{k+1} \triangleq \mathbf{I}_D - \mathbf{W}_{\text{be},k+1}\, \mathbf{Q}_{k+1}$, $\mathbf{Q}_{k+1} \triangleq (\mathbf{W}_w + \mathbf{W}_{\text{be},k+1})^{-1}$, $\mathbf{W}_w \triangleq (\mathbf{C}_w)^{-1}$ and $\mathbf{w}_{\text{be},k+1} \triangleq \mathbf{W}_{\text{be},k+1}\, \eta_{\text{be},k+1}$.

Phase II - In the $n-$th iteration of this phase, the eight consecutive steps listed below are carried out; for each step, all the computed messages are described.

Step 1) - In this step, the message

$$m_1^{(n-1)}\big(\mathbf{x}_k^{(N)}\big) = \sum_{j=1}^{N_p} W_{1,k,j}^{(n-1)}\, \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)}\big), \tag{3.41}$$

computed in the previous iteration and conveying the smoothed pdf of $\mathbf{x}_k^{(N)}$ generated by F_2 and BIF_2 (see step 8)) is processed jointly with $\overleftarrow{m}_{\text{be}}(\mathbf{x}_{k+1}^{(N)})$ (3.37) in the $\text{BIF}_2\rightarrow\text{BIF}_1$ block to generate the message

$$m_2^{(n)}\big(\mathbf{x}_k\big) = \mathcal{N}\big(\mathbf{x}_k; \eta_{2,k}^{(n)}, \mathbf{C}_{2,k}^{(n)}\big), \tag{3.42}$$

that conveys the pseudo-measurement information provided to BIF_1. The mean vector $\eta_{2,k}^{(n)}$ and the covariance matrix $\mathbf{C}_{2,k}^{(n)}$ are evaluated as

$$\eta_{2,k}^{(n)} = \big[\big(\eta_{L,k}^{(n)}\big)^T, \big(\eta_{N,k}^{(n)}\big)^T\big]^T \tag{3.43}$$

and

$$\mathbf{C}_{2,k}^{(n)} = \begin{bmatrix} \mathbf{C}_{LL,k}^{(n)} & \mathbf{C}_{LN,k}^{(n)} \\ \big(\mathbf{C}_{LN,k}^{(n)}\big)^T & \mathbf{C}_{NN,k}^{(n)} \end{bmatrix}, \tag{3.44}$$

respectively, where

$$\eta_{X,k}^{(n)} \triangleq \sum_{j=1}^{N_p} W_{1,k,j}^{(n-1)}\, \eta_{X,k,j} \tag{3.45}$$

is a D_X-dimensional mean vector (with $X = L$ and N),

$$\mathbf{C}_{XY,k}^{(n)} \triangleq \sum_{j=1}^{N_p} W_{1,k,j}^{(n-1)}\, \mathbf{r}_{XY,k,j} - \eta_{X,k}\big(\eta_{Y,k}\big)^T \tag{3.46}$$

is a $D_X \times D_Y$ covariance (or cross-covariance) matrix (with $XY = LL$, NN and LN), $\eta_{L,k,j} = \tilde{\eta}_{k,j}$, $\eta_{N,k,j} = \mathbf{x}_{k,j}^{(N)}$, $\mathbf{r}_{LL,k,j} \triangleq \tilde{\mathbf{C}}_{k,j} + \tilde{\eta}_{k,j}(\tilde{\eta}_{k,j})^T$, $\mathbf{r}_{NN,k,j} \triangleq \mathbf{x}_{k,j}^{(N)}(\mathbf{x}_{k,j}^{(N)})^T$ and $\mathbf{r}_{LN,k,j} \triangleq \tilde{\eta}_{k,j}(\mathbf{x}_{k,j}^{(N)})^T$. The covariance matrix $\tilde{\mathbf{C}}_{k,j}$ and the mean vector $\tilde{\eta}_{k,j}$ are computed on the basis of the associated precision matrix

$$\tilde{\mathbf{W}}_{k,j} \triangleq \big(\tilde{\mathbf{C}}_{k,j}\big)^{-1} = \big(\mathbf{A}_{k,j}^{(N)}\big)^T \mathbf{W}_w^{(N)} \mathbf{A}_{k,j}^{(N)} \tag{3.47}$$

and of the associated transformed mean vector

$$\tilde{\mathbf{w}}_{k,j} \triangleq \tilde{\mathbf{W}}_{k,j}\tilde{\eta}_{k,j} = \big(\mathbf{A}_{k,j}^{(N)}\big)^T \mathbf{W}_w^{(N)} \mathbf{z}_{k,j}^{(L)}, \tag{3.48}$$

respectively; here, $\mathbf{A}_{k,j}^{(N)} \triangleq \mathbf{A}_k^{(N)}(\mathbf{x}_{k,j}^{(N)})$,

$$\mathbf{z}_{k,j}^{(L)} \triangleq \mathbf{x}_{\mathrm{be},k+1}^{(N)} - \mathbf{f}_{k,j}^{(N)} \tag{3.49}$$

is an iteration-independent pseudo-measurement (see Eq. (3.21)) and $\mathbf{f}_{k,j}^{(N)} \triangleq \mathbf{f}_k^{(N)}(\mathbf{x}_{k,j}^{(N)})$. Note that, in the first iteration,

$$W_{1,k,j}^{(n-1)} = W_{1,k,j}^{(0)} = w_{\mathrm{fe},k,j}, \tag{3.50}$$

for any j, i.e. $m_1^{(0)}(\mathbf{x}_k^{(N)}) = \vec{m}_{\mathrm{fe1}}(\mathbf{x}_k^{(N)})$ (see Eq. (3.35)) since the initial information available about the particle set are those originating from the forward pass. For this reason, the particles $\{\mathbf{x}_{k,j}^{(N)}\}$ and their weights $\{w_{\mathrm{fe},k,j}\}$ are stored in the memory cell at the beginning of the first iteration.

Step 2) - In this step, the first backward filtered pdf $\overleftarrow{m}_3^{(n)}(\mathbf{x}_k)$ of $\mathbf{x}_k$ is computed as (see Fig. 3.4)

$$\overleftarrow{m}_3^{(n)}\big(\mathbf{x}_k\big) = \overleftarrow{m}_1(\mathbf{x}_k)\, m_2^{(n)}(\mathbf{x}_k) \tag{3.51}$$

$$= \mathcal{N}\big(\mathbf{x}_k; \eta_{3,k}^{(n)}, \mathbf{C}_{3,k}^{(n)}\big), \tag{3.52}$$

where the messages $\overleftarrow{m}_1(\mathbf{x}_k)$ and $m_2^{(n)}(\mathbf{x}_k)$ are given by Eq. (3.38) and Eq. (3.42), respectively. The covariance matrix $\mathbf{C}_{3,k}^{(n)}$ and the mean vector $\eta_{3,k}^{(n)}$ are computed on the basis of the associated precision matrix

$$\mathbf{W}_{3,k}^{(n)} \triangleq (\mathbf{C}_{3,k}^{(n)})^{-1} = \mathbf{W}_{1,k} + \mathbf{W}_{2,k}^{(n)} \tag{3.53}$$

and the associated transformed mean vector

$$\mathbf{w}_{3,k}^{(n)} \triangleq \mathbf{W}_{3,k}^{(n)}\, \eta_{3,k}^{(n)} = \mathbf{w}_{1,k} + \mathbf{w}_{2,k}^{(n)}, \tag{3.54}$$

respectively; here, $\mathbf{W}_{2,k}^{(n)} \triangleq (\mathbf{C}_{2,k}^{(n)})^{-1}$, $\mathbf{w}_{2,k}^{(n)} \triangleq \mathbf{W}_{2,k}^{(n)}\, \eta_{2,k}^{(n)}$, and $\mathbf{W}_{1,k}$ and $\mathbf{w}_{1,k}$ are given by Eqs. (3.39) and (3.40), respectively. From Eqs. (3.53)-(3.54) the expressions

$$\mathbf{C}_{3,k}^{(n)} = \mathbf{W}_k^{(n)} \mathbf{C}_{2,k}^{(n)} \tag{3.55}$$

and

$$\eta_{3,k}^{(n)} = \mathbf{W}_k^{(n)} \big[\mathbf{C}_{2,k}^{(n)}\, \mathbf{w}_{1,k} + \eta_{2,k}^{(n)}\big] \tag{3.56}$$

can be easily inferred; here, $\mathbf{W}_k^{(n)} \triangleq [\mathbf{C}_{2,k}^{(n)} \mathbf{W}_{1,k} + \mathbf{I}_D]^{-1}$.

Step 3) - In this step, the smoothed pdf $m_4^{(n)}(\mathbf{x}_k)$ of $\mathbf{x}_k$ is evaluated as (see Fig. 3.4)

$$m_4^{(n)}\big(\mathbf{x}_k\big) = \vec{m}_{\mathrm{fe1}}\big(\mathbf{x}_k\big) \overleftarrow{m}_3^{(n)}\big(\mathbf{x}_k\big) \tag{3.57}$$

$$= \mathcal{N}\big(\mathbf{x}_k; \eta_{4,k}^{(n)}, \mathbf{C}_{4,k}^{(n)}\big), \tag{3.58}$$

where the messages $\vec{m}_{\mathrm{fe1}}\big(\mathbf{x}_k\big)$ and $\overleftarrow{m}_3^{(n)}\big(\mathbf{x}_k\big)$ are given by Eqs. (3.31) and (3.52), respectively. The covariance matrix $\mathbf{C}_{4,k}^{(n)}$ and the mean vector $\eta_{4,k}^{(n)}$ are computed on the basis of the associated precision matrix

$$\mathbf{W}_{4,k}^{(n)} = \mathbf{W}_{\mathrm{fe1},k} + \mathbf{W}_{3,k}^{(n)} \tag{3.59}$$

and of the associated transformed mean vector

$$\mathbf{w}_{4,k}^{(n)} = \mathbf{w}_{\text{fe1},k} + \mathbf{w}_{3,k}^{(n)}, \tag{3.60}$$

respectively. Note that Eq. (3.57) represents an instance of Eq. (3.11), since $\vec{m}_{\text{fe1}}\big(\mathbf{x}_k\big)$ and $\overleftarrow{m}_3^{(n)}\big(\mathbf{x}_k\big)$ correspond to $\vec{m}_{\text{fe1}}(\mathbf{x}_k^{(i)})$ and $\overleftarrow{m}_{\text{be1}}(\mathbf{x}_k^{(i)})$, respectively ($\mathbf{x}_k^{(i)} = \mathbf{x}_k^{(N)}$ in this case).

Step 4) - In this step, the message

$$m_1^{(n)}\big(\mathbf{x}_k^{(L)}\big) \triangleq \int m_4^{(n)}\big(\mathbf{x}_k\big)\, d\mathbf{x}_k^{(N)} = \mathcal{N}(\mathbf{x}_k^{(L)}; \tilde{\eta}_{1,k}^{(n)}, \tilde{\mathbf{C}}_{1,k}^{(n)}), \tag{3.61}$$

is computed in the BIF$_1$→BIF$_2$ block. In practice, the mean $\tilde{\eta}_{1,k}^{(n)}$ and the covariance matrix $\tilde{\mathbf{C}}_{1,k}^{(n)}$ are extracted from the mean $\eta_{4,k}^{(n)}$ and the covariance matrix $\mathbf{C}_{4,k}^{(n)}$ of $m_4^{(n)}(\mathbf{x}_k)$ (3.58), respectively, since $\mathbf{x}_k^{(L)}$ consists of the first D_L elements of $\mathbf{x}_k$.

Then, the backward predicted pdf $\overleftarrow{m}_3^{(n)}(\mathbf{x}_k^{(N)})$ is evaluated as (see Fig. 3.4)

$$\overleftarrow{m}_3^{(n)}\big(\mathbf{x}_k^{(N)}\big) = \int\int f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})\, \overleftarrow{m}_{\text{be}}\big(\mathbf{x}_{k+1}^{(N)}\big) m_1^{(n)}\big(\mathbf{x}_k^{(L)}\big)\, d\mathbf{x}_k^{(N)} d\mathbf{x}_{k+1}^{(N)}. \tag{3.62}$$

Actually, what is really needed in our computations is the value taken on by this message (and also by messages $m_2^{(n)}(\mathbf{x}_k^{(N)})$ and $m_5^{(n)}(\mathbf{x}_k^{(N)})$ evaluated in step 5) and in step 7), respectively) for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}$ (see Eqs. (3.75), (3.87) and (3.92)); such a value, denoted $w_{3,k,j}^{(n)}$, is computed as

$$w_{3,k,j}^{(n)} = D_{3,k,j}^{(n)} \exp\big(-\frac{1}{2} Z_{3,k,j}^{(n)}\big), \tag{3.63}$$

where

$$D_{3,k,j}^{(n)} = (2\pi)^{-D_N/2} (\det(\mathbf{C}_{3,k,j}^{(N)}[n]))^{-1/2}, \tag{3.64}$$

$$Z_{3,k,j}^{(n)} \triangleq \big\|\mathbf{x}_{\text{be},k+1}^{(N)} - \eta_{3,k,j}^{(N)}[n]\big\|^2_{\mathbf{W}_{3,k,j}^{(N)}[n]}, \tag{3.65}$$

$\big\|\mathbf{x}\big\|^2_{\mathbf{W}} \triangleq \mathbf{x}^T\mathbf{W}\mathbf{x}$ denotes the square of the norm of the vector $\mathbf{x}$ with respect to the positive definite matrix $\mathbf{W}$,

$$\eta_{3,k,j}^{(N)}[n] \triangleq \mathbf{A}_{k,j}^{(N)}\, \tilde{\eta}_{1,k}^{(n)} + \mathbf{f}_{k,j}^{(N)}, \tag{3.66}$$

$\mathbf{W}_{3,k,j}^{(N)}[n] \triangleq (\mathbf{C}_{3,k,j}^{(N)}[n])^{-1}$ and

$$\mathbf{C}_{3,k,j}^{(N)}[n] \triangleq \mathbf{A}_{k,j}^{(N)}\, \tilde{\mathbf{C}}_{1,k}^{(n)}\, \big(\mathbf{A}_{k,j}^{(N)}\big)^T + \mathbf{C}_w^{(N)}. \tag{3.67}$$

Step 5) - In this step, the message $m_2^{(n)}(\mathbf{x}_k^{(N)})$, conveying pseudo-measurement information about the nonlinear state component, is computed in the BIF$_1$→BIF$_2$ block. The value $w_{2,k,j}^{(n)}$ taken on by this message for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}$ is evaluated as

$$w_{2,k,j}^{(n)} = D_{2,k,j}^{(n)} \exp\big(-\frac{1}{2} Z_{2,k,j}^{(n)}\big) \tag{3.68}$$

for any j; here,

$$Z_{2,k,j}^{(n)} \triangleq \big\|\check{\eta}_{z,k,j}^{(n)}\big\|^2_{\check{\mathbf{W}}_{z,k,j}^{(n)}} - \big\|\mathbf{f}_{k,j}^{(L)}\big\|^2_{\mathbf{W}_w^{(L)}} - \big\|\check{\eta}_{2,k,j}^{(n)}\big\|^2_{\check{\mathbf{W}}_{2,k,j}^{(n)}}, \tag{3.69}$$

$\mathbf{W}_w^{(L)} \triangleq [\mathbf{C}_w^{(L)}]^{-1}$, $\mathbf{f}_{k,j}^{(L)} \triangleq \mathbf{f}_k^{(L)}(\mathbf{x}_{k,j}^{(N)})$, $\check{\mathbf{W}}_{z,k,j}^{(n)} \triangleq (\check{\mathbf{C}}_{z,k,j}^{(n)})^{-1}$, $\check{\mathbf{w}}_{z,k,j}^{(n)} \triangleq \check{\mathbf{W}}_{z,k,j}^{(n)} \check{\eta}_{z,k,j}^{(n)}$,

$$\check{\eta}_{z,k,j}^{(n)} = \tilde{\eta}_{\text{be},k+1} - \mathbf{A}_{k,j}^{(L)}\, \tilde{\eta}_{1,k}^{(n)}, \tag{3.70}$$

$$\check{\mathbf{C}}_{z,k,j}^{(n)} = \tilde{\mathbf{C}}_{\text{be},k+1} - \mathbf{A}_{k,j}^{(L)}\, \tilde{\mathbf{C}}_{1,k}^{(n)}\, \big(\mathbf{A}_{k,j}^{(L)}\big)^T, \tag{3.71}$$

$$\check{\mathbf{W}}_{2,k,j}^{(n)} \triangleq \big(\check{\mathbf{C}}_{2,k,j}^{(n)}\big)^{-1} = \check{\mathbf{W}}_{z,k,j}^{(n)} + \mathbf{W}_{w}^{(L)}, \tag{3.72}$$

$$D_{2,k,j}^{(n)} \triangleq (2\pi)^{-D_L/2}[\det(\check{\mathbf{C}}_{k,j}^{(n)})]^{-1/2} \tag{3.73}$$

$\check{\mathbf{C}}_{k,j}^{(n)} \triangleq \check{\mathbf{C}}_{z,k,j}^{(n)} + \mathbf{C}_{w}^{(L)}$, $\mathbf{A}_{k,j}^{(L)} \triangleq \mathbf{A}_{k}^{(L)}(\mathbf{x}_{k,j}^{(N)})$, $\check{\eta}_{2,k,j}^{(n)}$ is evaluated on the basis of the associated transformed mean vector

$$\check{\mathbf{w}}_{2,k,j}^{(n)} \triangleq \check{\mathbf{W}}_{2,k,j}^{(n)}\,\check{\eta}_{2,k,j}^{(n)} = \check{\mathbf{w}}_{z,k,j}^{(n)} + \mathbf{W}_{w}^{(L)}\,\mathbf{f}_{k,j}^{(L)}, \tag{3.74}$$

and the mean $\tilde{\eta}_{\mathrm{be},k+1}$ and the covariance matrix $\tilde{\mathbf{C}}_{\mathrm{be},k+1}$ are extracted from the mean $\eta_{\mathrm{be},k+1}$ and the covariance matrix $\mathbf{C}_{\mathrm{be},k+1}$ of $\overleftarrow{m}_{\mathrm{be}}(\mathbf{x}_{k+1})$ (3.36), since they refer to $\mathbf{x}_{k}^{(L)}$ only.

Step 6) - In this step, the message $m_4^{(n)}(\mathbf{x}_k^{(N)})$, conveying the first backward filtered pdf of $\mathbf{x}_k^{(N)}$, is computed as (see Fig. 3.4)

$$\overleftarrow{m}_4^{(n)}(\mathbf{x}_k^{(N)}) = \overleftarrow{m}_3^{(n)}(\mathbf{x}_k^{(N)})\, m_2^{(n)}(\mathbf{x}_k^{(N)}). \tag{3.75}$$

The value $w_{4,k,j}^{(n)}$ taken on by this message for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}$ is given by (see Eqs. (3.63) and (3.68))

$$w_{4,k,j}^{(n)} \triangleq w_{2,k,j}^{(n)}\, w_{3,k,j}^{(n)} \tag{3.76}$$

for any j.

Step 7) - In this step, the message conveying measurement-based information about $\mathbf{x}_k^{(N)}$ is computed as (see Fig. 3.4)

$$m_5^{(n)}(\mathbf{x}_k^{(N)}) = \int f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})\, m_1^{(n)}(\mathbf{x}_k^{(L)})\, d\mathbf{x}_k^{(L)} \tag{3.77}$$

$$= \mathcal{N}\big(\mathbf{y}_l; \bar{\eta}_{5,k}^{(n)}\big(\mathbf{x}_k^{(N)}\big), \bar{\mathbf{C}}_{5,k}^{(n)}\big(\mathbf{x}_k^{(N)}\big)\big) \tag{3.78}$$

where

$$\bar{\eta}_{5,k}^{(n)}\big(\mathbf{x}_k^{(N)}\big) \triangleq \mathbf{B}_k\big(\mathbf{x}_k^{(N)}\big)\,\tilde{\eta}_{1,k}^{(n)} + \mathbf{g}_k\big(\mathbf{x}_k^{(N)}\big) \tag{3.79}$$

and

$$\bar{\mathbf{C}}_{5,k}^{(n)}\big(\mathbf{x}_k^{(N)}\big) \triangleq \mathbf{B}_k\big(\mathbf{x}_k^{(N)}\big)\,\tilde{\mathbf{C}}_{1,k}^{(n)}\,\mathbf{B}_k^T\big(\mathbf{x}_k^{(N)}\big) + \mathbf{C}_e. \tag{3.80}$$

Consequently, the value taken on by $m_5^{(n)}(\mathbf{x}_k^{(N)})$ for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}$ is

$$w_{5,k,j}^{(n)} = \mathcal{N}\big(\mathbf{y}_k; \bar{\eta}_{5,k,j}^{(n)}, \bar{\mathbf{C}}_{5,k,j}^{(n)}\big) \tag{3.81}$$

$$= D_{5,k,j}^{(n)} \exp\big(-\frac{1}{2} Z_{5,k,j}^{(n)}\big), \tag{3.82}$$

where

$$\bar{\eta}_{5,k,j}^{(n)}\big(\mathbf{x}_k^{(N)}\big) \triangleq \bar{\eta}_{5,k}^{(n)}\big(\mathbf{x}_{k,j}^{(N)}\big) = \mathbf{B}_{k,j}\,\tilde{\eta}_{1,k}^{(n)} + \mathbf{g}_{k,j}, \tag{3.83}$$

$$\bar{\mathbf{C}}_{5,k,j}^{(n)} \triangleq \bar{\mathbf{C}}_{5,k}^{(n)}\big(\mathbf{x}_{k,j}^{(N)}\big) = \mathbf{B}_{k,j}\,\tilde{\mathbf{C}}_{1,k}^{(n)}\,\mathbf{B}_{k,j}^T + \mathbf{C}_e, \tag{3.84}$$

$\mathbf{B}_{k,j} \triangleq \mathbf{B}_k(\mathbf{x}_{k,j}^{(N)})$, $\mathbf{g}_{k,j} \triangleq \mathbf{g}_l(\mathbf{x}_{k,j}^{(N)})$,

$$D_{5,k,j}^{(n)} \triangleq (2\pi)^{-P/2}[\det(\bar{\mathbf{C}}_{5,k,j}^{(n)})]^{-1/2} \tag{3.85}$$

$$Z_{5,k,j}^{(n)} \triangleq \big\|\mathbf{y}_k - \bar{\eta}_{5,k,j}^{(n)}\big\|_{\bar{\mathbf{W}}_{5,k,j}^{(n)}}^2 \tag{3.86}$$

and $\bar{\mathbf{W}}_{5,k,j}^{(n)} \triangleq (\bar{\mathbf{C}}_{5,k,j}^{(n)})^{-1}$. Then, the message $\overleftarrow{m}_6^{(n)}(\mathbf{x}_k^{(N)})$ is evaluated as (see Fig. 3.4)

$$\overleftarrow{m}_6^{(n)}\big(\mathbf{x}_k^{(N)}\big) = \overleftarrow{m}_4^{(n)}\big(\mathbf{x}_k^{(N)}\big)\, m_5^{(n)}\big(\mathbf{x}_k^{(N)}\big)\,. \tag{3.87}$$

Its value for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}$ is given by (see Eqs. (3.63), (3.68) and (3.82))

$$w_{6,k,j}^{(n)} = w_{4,k,j}^{(n)} w_{5,k,j}^{(n)} = w_{2,k,j}^{(n)}\, w_{3,k,j}^{(n)} w_{5,k,j}^{(n)} \tag{3.88}$$

$$= D_{6,k,j}^{(n)} \exp\big(-\frac{1}{2} Z_{6,k,j}^{(n)}\big) \tag{3.89}$$

where

$$D_{6,k,j}^{(n)} \triangleq D_{2,k,j}^{(n)}\, D_{3,k,j}^{(n)}\, D_{5,k,j}^{(n)} \tag{3.90}$$

and

$$Z_{6,k,j}^{(n)} \triangleq Z_{2,k,j}^{(n)} + Z_{3,k,j}^{(n)} + Z_{5,k,j}^{(n)}. \tag{3.91}$$

Note that the weight $w_{6,k,j}^{(n)}$ conveys the information provided by the backward state transition ($w_{3,k,j}^{(n)}$), the pseudo-measurements ($w_{2,k,j}^{(n)}$) and the measurements ($w_{5,k,j}^{(n)}$).

Step 8) - In this step, the message $m_1^{(n)}(\mathbf{x}_k^{(N)})$, conveying the smoothed pdf of $\mathbf{x}_k^{(N)}$ evaluated in the n-th iteration, is computed as (see Fig. 3.4)

$$m_1^{(n)}\big(\mathbf{x}_k^{(N)}\big) = \vec{m}_{\rm fp}\big(\mathbf{x}_k^{(N)}\big)\, \overleftarrow{m}_6^{(n)}\big(\mathbf{x}_k^{(N)}\big); \tag{3.92}$$

this formula represents an instance of Eq. (3.10), since $\vec{m}_{\rm fp1}(\mathbf{x}_k^{(N)})$ and $\overleftarrow{m}_6^{(n)}(\mathbf{x}_k^{(N)})$ correspond to $\vec{m}_{\rm fp}(\mathbf{x}_k^{(i)})$ and $\overleftarrow{m}_{\rm be2}(\mathbf{x}_k^{(i)})$, respectively ($\mathbf{x}_k^{(i)} = \mathbf{x}_k^{(N)}$ in this case). The $j-$th component of $m_1^{(n)}(\mathbf{x}_k^{(N)})$ is evaluated as (see Eqs. (3.34) and (3.88))

$$m_{1,j}^{(n)}\big(\mathbf{x}_k^{(N)}\big) = \vec{m}_{{\rm fp},j}\big(\mathbf{x}_k^{(N)}\big)\, w_{6,k,j}^{(n)} \tag{3.93}$$

$$= w_{1,k,j}^{(n)}\; \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)}\big), \tag{3.94}$$

where

$$w_{1,k,j}^{(n)} \triangleq w_p\, w_{6,k,j}^{(n)}. \tag{3.95}$$

Then, the weights $\{w_{1,k,j}^{(n)}\}$ are normalized; the j-th normalised weight is computed as

$$W_{1,k,j}^{(n)} \triangleq C_k^{(n)}\, w_{1,k,j}^{(n)}\,, \tag{3.96}$$

with $j = 1, 2, ..., N_p$, where $C_k^{(n)} \triangleq 1/\sum_{j=0}^{N_p-1} w_{1,k,j}^{(n)}$. Moreover, the weights $\{W_{1,k,j}^{(n)}\}$ are stored for the next iteration. This concludes the $n-$th iteration. Then, the index n is increased by one, and a new iteration is started by going back to step 1) if $n < n_i + 1$; otherwise (i.e., if $n = n_i + 1$), we proceed with the next phase.

Phase III - In this phase, $\overleftarrow{m}_{\rm be}(\mathbf{x}_k^{(N)})$ (i.e., the BIF$_2$ output message) is computed first; then, steps 1) and 2) of phase II are accomplished in order to compute all the statistical information required for the evaluation of the backward estimate $\overleftarrow{m}_{\rm be}\big(\mathbf{x}_k\big)$ (i.e., the BIF$_1$ output message). More specifically, we first sample the set S_k once on the basis of the particle weights $\{W_{1,k,j}^{(n_i)}\}$ computed in the last iteration; if the j_k-th particle (i.e., $\mathbf{x}_{k,j_k}^{(N)}$) is selected, we set

$$\mathbf{x}_{{\rm be},k}^{(N)} = \mathbf{x}_{k,j_k}^{(N)}, \tag{3.97}$$

so that the message (see Eq. (3.37))

$$\overleftarrow{m}_{\rm be}\big(\mathbf{x}_k^{(N)}\big) \triangleq \delta\big(\mathbf{x}_k^{(N)} - \mathbf{x}_{{\rm be},k}^{(N)}\big), \tag{3.98}$$

can be made available at the output of BIF$_2$. On the other hand, the evaluation of the message $\overleftarrow{m}_{\rm be}\big(\mathbf{x}_k\big)$ is accomplished as follows. The messages $m_2^{(n_i+1)}(\mathbf{x}_k)$ and $\overleftarrow{m}_3^{(n_i+1)}\big(\mathbf{x}_k\big)$ are computed

first (see Eqs. (3.42)–(3.49) and Eqs. (3.51)–(3.54), respectively). Then, the message $\overleftarrow{m}_{\rm be}(\mathbf{x}_k)$ is computed as (see Fig. 3.4)

$$\overleftarrow{m}_{\rm be}(\mathbf{x}_k) = \overleftarrow{m}_{\rm be2}(\mathbf{x}_k) = \overleftarrow{m}_{\rm be1}^{(n_i+1)}(\mathbf{x}_k)\, m_{\rm ms}(\mathbf{x}_k) \tag{3.99}$$

$$= \mathcal{N}(\mathbf{x}_k; \eta_{\rm be2,k}, \mathbf{C}_{\rm be2,k}), \tag{3.100}$$

where

$$m_{\rm ms}(\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_k; \eta_{\rm ms,k}, \mathbf{C}_{\rm ms,k}) \tag{3.101}$$

is the message conveying measurement information.

Moreover, the covariance matrices $\mathbf{C}_{\rm ms,k}$ and $\mathbf{C}_{\rm be2,k}$, and the mean vectors $\eta_{\rm ms,k}$ and $\eta_{\rm be2,k}$ are evaluated on the basis of the associated precision matrices

$$\mathbf{W}_{\rm ms,k} \triangleq (\mathbf{C}_{\rm ms,k})^{-1} = \mathbf{H}_k \mathbf{W}_e \mathbf{H}_k^T, \tag{3.102}$$

$$\mathbf{W}_{\rm be2,k} \triangleq (\mathbf{C}_{\rm be2,k})^{-1} = \mathbf{W}_{\rm ms,k} + \mathbf{W}_{\rm be1,k}^{(n_i+1)}, \tag{3.103}$$

and of the transformed mean vectors

$$\mathbf{w}_{\rm ms,k} \triangleq \mathbf{W}_{\rm ms,k}\, \eta_{\rm ms,k} = \mathbf{H}_k \mathbf{W}_e (\mathbf{y}_k - \mathbf{v}_k), \tag{3.104}$$

$$\mathbf{w}_{\rm be2,k} \triangleq \mathbf{W}_{\rm be2,k}\, \eta_{\rm be2,k} = \mathbf{w}_{\rm ms,k} + \mathbf{w}_{\rm be1,k}^{(n_i+1)}, \tag{3.105}$$

respectively. The k-th recursion is now over.

In the DBSA, the *first recursion* of the backward pass (corresponding to $k = T-1$) requires the knowledge of the input messages $\overleftarrow{m}_{\rm be}(\mathbf{x}_T)$ and $\overleftarrow{m}_{\rm be}(\mathbf{x}_T^{(N)})$. Similarly as any BIF algorithm, the evaluation of these messages in DBIF is based on the statistical information generated in the last recursion of the forward pass. In particular, the above mentioned messages are still expressed by Eqs. (3.36) and (3.37) (with $k = T-1$ in both formulas), respectively. However, the vector $\mathbf{x}_{{\rm be},T}^{(N)}$ is generated by sampling the particle set S_T on the basis of the forward weights $\{w_{{\rm fe},T,j}\}$, since backward predictions are unavailable at the final instant $k = T$. Therefore, if the j_T-th particle of S_T is selected, we set

$$\mathbf{x}_{{\rm be},T}^{(N)} = \mathbf{x}_{{\rm fe},T,j_T}^{(N)} \tag{3.106}$$

in the message $\overleftarrow{m}_{\rm be}(\mathbf{x}_T^{(N)})$ entering the BIF$_2$ in the first recursion (see Eq. (3.37)). As far as BIF$_1$ is concerned, following [23], we choose

$$\mathbf{W}_{{\rm be},T} = \mathbf{W}_{{\rm fe1},T} \tag{3.107}$$

and

$$\mathbf{w}_{{\rm be},T} = \mathbf{w}_{{\rm fe1},T} \tag{3.108}$$

for the message $\overleftarrow{m}_{\rm be}(\mathbf{x}_T)$.

The DBSA is summarized in Algorithm 1. It generates all the statistical information required to solve problems **P.1** and **P.2**. Let us now discuss how this can be done in detail. As far as problem **P.1** is concerned, it is useful to point out that the DBSA produces a trajectory $\{\mathbf{x}_{{\rm be},k}^{(N)}, k = 1, 2, ..., T\}$ for the *nonlinear* component (see Eq. (3.97)). Another trajectory, representing the time evolution of the *linear* state component only and denoted $\{\mathbf{x}_{{\rm be},k}^{(L)}, k = 1, 2, ..., T\}$, can be evaluated by sampling the message $m_1^{(n_i)}(\mathbf{x}_k^{(L)})$ (see Eq. (3.61)) or by simply setting $\mathbf{x}_{{\rm be},k}^{(L)} = \tilde{\eta}_{1,k}^{(n_i)}$ (this task can be accomplished in phase III, after sampling the particle set S_k; see also the task g- in phase III of Algorithm 1.

Since the DBSA solves problem **P.1**, it also solves problem **P.2**; in fact, once it has been run, an approximation of the marginal smoothed pdf at any instant can be simply obtained by marginalization. Unluckily, the last result is achieved at the price of a significant computational cost, since M backward passes are required. However, if we are interested in solving problem

Algorithm 4: Double Bayesian Smoothing

1 **Forward filtering**: For $k = 1$ to T: Run the DBF, and store $\mathbf{W}_{\text{fe1},k}$ (3.32), $\mathbf{w}_{\text{fe1},k}$ (3.33), $S_k = \{\mathbf{x}_{k,j}^{(N)}\}$ and $\{w_{\text{fe},k,j}\}_{j=1}^{N_p}$.

2 **Initialisation of backward filtering**: compute $\mathbf{x}_{\text{be},T}^{(N)}$ (3.106), $\mathbf{W}_{\text{be},T}$ (3.107) and $\mathbf{w}_{\text{be},T}$ (3.108); then, compute $\mathbf{C}_{\text{be},T} = (\mathbf{W}_{\text{be},T})^{-1}$, $\eta_{\text{be},T} = \mathbf{C}_{\text{be},T}\mathbf{w}_{\text{be},T}$.

3 **Backward filtering and smoothing**:

for $k = T-1$ *to* 1 **do**

a- Phase I:

- *Backward prediction in* BIF_1: compute $\mathbf{W}_{1,k}$ (3.39) and $\mathbf{w}_{1,k}$ (3.40).
- *Computation of iteration-independent information required in task b*:For $j = 1$ to N_p: compute $\mathbf{z}_{k,j}^{(L)}$ (3.49), $\tilde{\mathbf{W}}_{k,j}$ (3.47), $\tilde{\mathbf{w}}_{k,j}$ (3.48), $\tilde{\mathbf{C}}_{k,j} = (\tilde{\mathbf{W}}_{k,j})^{-1}$ and $\tilde{\eta}_{k,j} = \tilde{\mathbf{C}}_{k,j}\tilde{\mathbf{w}}_{k,j}$.
- *Initialisation of particle weights*: Set $W_{1,k,j}^{(0)} = w_{\text{fe},k,j}$.

Phase II:

for $n = 1$ *to* n_i **do**

b- Compute $\eta_{2,k}^{(n)}$ (3.43) and $\mathbf{C}_{2,k}^{(n)}$ (3.44).

c- Compute $\mathbf{C}_{3,k}^{(n)}$ (3.55), $\eta_{3,k}^{(n)}$ (3.56), $\mathbf{W}_{3,k}^{(n)} = (\mathbf{C}_{3,k}^{(n)})^{-1}$, $\mathbf{w}_{3,k}^{(n)} = \mathbf{W}_{3,k}^{(n)}\eta_{3,k}^{(n)}$, $\mathbf{W}_{4,k}^{(n)}$ (3.59), $\mathbf{w}_{4,k}^{(n)}$ (3.60), $\mathbf{C}_{4,k}^{(n)} = (\mathbf{W}_{4,k}^{(n)})^{-1}$ and $\eta_{4,k}^{(n)} = \mathbf{C}_{4,k}^{(n)}\mathbf{w}_{4,k}^{(n)}$. Then, extract $\tilde{\eta}_{1,k}^{(n)}$ ($\tilde{\mathbf{C}}_{1,k}^{(n)}$) from $\eta_{4,k}^{(n)}$ ($\mathbf{C}_{4,k}^{(n)}$).

d- For $j = 1$ to N_p: compute $\eta_{3,k,j}^{(N)}[n]$ (3.66) and $\mathbf{C}_{3,k,j}^{(N)}[n]$ (3.67). Then, compute $D_{3,k,j}^{(n)}$ (3.64) and $Z_{3,k,j}^{(n)}$ (3.65).

e- For $j = 1$ to N_p: compute $\check{\eta}_{z,k,j}^{(n)}$ (3.70), $\check{\mathbf{C}}_{z,k,j}^{(n)}$ (3.71), $\check{\mathbf{W}}_{z,k,j}^{(n)} = (\check{\mathbf{C}}_{z,k,j}^{(n)})^{-1}$, $\check{\mathbf{w}}_{z,k,j}^{(n)} = \check{\mathbf{W}}_{z,k,j}^{(n)}\check{\eta}_{z,k,j}^{(n)}$, $\check{\mathbf{W}}_{2,k,j}^{(n)}$ (3.72), $\check{\mathbf{w}}_{2,k,j}^{(n)}$ (3.74). Then, compute $D_{2,k,j}^{(n)}$ (3.73) and $Z_{2,k,j}^{(n)}$ (3.69).

f- For $j = 1$ to N_p: Compute $\bar{\eta}_{5,k,j}^{(n)}$ (3.83), $\bar{\mathbf{C}}_{5,k,j}^{(n)}$ (3.84), $\bar{\mathbf{W}}_{5,k,j}^{(n)} = (\bar{\mathbf{C}}_{5,k,j}^{(n)})^{-1}$, $D_{5,k,j}^{(n)}$ (3.85) and $Z_{5,k,j}^{(n)}$ (3.86). Then, compute $D_{6,k,j}^{(n)}$ (3.90), $Z_{6,k,j}^{(n)}$ (3.91), $w_{6,k,j}^{(n)}$ (3.89), $w_{1,k,j}^{(n)}$ (3.95) and $W_{1,k,j}^{(n)}$ (3.96). Store the weights $\{W_{1,k,j}^{(n)}\}$ for the next iteration.

end

g- Phase III - BIF_2: Select the j_k-th particle $\mathbf{x}_{k,j_k}^{(N)}$ by sampling the set S_k on the basis of the weights $\{W_{1,k,j}^{(n_i)}\}$, set $\mathbf{x}_{\text{be},k}^{(N)} = \mathbf{x}_{k,j_k}^{(N)}$ and store $\mathbf{x}_{\text{be},k}^{(N)}$ for the next recursion.

h- Phase III - BIF_1: Compute $\eta_{2,k}^{(n_i+1)}$, $\mathbf{C}_{2,k}^{(n_i+1)}$, $\mathbf{W}_{3,k}^{(n_i+1)}$ and $\mathbf{w}_{3,k}^{(n_i+1)}$ (see steps 1) and 2)). Then, compute $\mathbf{W}_{\text{ms},k}$ (3.102), $\mathbf{w}_{\text{ms},k}$ (3.104), $\mathbf{W}_{\text{be2},k}$ (3.103), $\mathbf{w}_{\text{be2},k}$ (3.105), $\mathbf{C}_{\text{be},k} = (\mathbf{W}_{\text{be2},k})^{-1}$ and $\eta_{\text{be},k} = \mathbf{C}_{\text{be},k}\mathbf{w}_{\text{be2},k}$, and store $\mathbf{C}_{\text{be},k}$ and $\eta_{\text{be},k}$ for the next recursion.

end

P.2 only, a simpler particle smoother can be developed following the approach illustrated in ref. [23], so that a single backward pass has to be run. In this pass, the evaluation of the message $\check{m}_{\text{be}}(\mathbf{x}_k^{(N)})$ (i.e., of the particle $\mathbf{x}_{\text{be},k}^{(N)}$) involves the whole particle set S_k and their weights $\{W_{1,k,j}^{(n_i)}\}$ (see Eq. (3.96)) evaluated in the last phase of the $(T-k)-$th recursion. More specifically, a new smoother is obtained by employing a different method for evaluating $\mathbf{x}_{\text{be},k}^{(N)}$ (see phase III-BIF_2); it consists in computing the smoothed estimate

$$\mathbf{x}_{\text{sm},k}^{(N)} = \sum_{j=1}^{N_p} W_{1,k,j}^{(n_i)}\, \mathbf{x}_{k,j}^{(N)} \tag{3.109}$$

of $\mathbf{x}_k^{(N)}$ and, then, setting

$$\mathbf{x}_{\text{be},k}^{(N)} = \mathbf{x}_{\text{sm},k}^{(N)}. \tag{3.110}$$

The resulting smoother is called *simplified* DBSA (SDBSA) in the following.

The computational complexity of the DBSA and the SDBSA can be reduced by reusing the forward weights $\{w_{\text{fe},k,j}\}$ in all the iterations of phase II, so that step 7) can be skipped; this

means that, for any n, we set $w_{5,k,j}^{(n)} = w_{\mathrm{fe},k,j}$ in the evaluation of the $j-$th particle weight $w_{6,k,j}^{(n)}$ according to Eq. (3.88) in step 8) of phase II. Our simulation results have evidenced that, at least for the SSMs considered in Section 3.5, this modification does not affect the estimation accuracy of the derived algorithms; for this reason, it is always employed in our simulations.

The DBSA and the SDBSA refer to case **C.1**, i.e. to the case in which the substates estimated by the interconnected forward/backward filters share the substate $\mathbf{x}_k^{(N)}$. Let us focus now on case **C.2**, i.e. on the case on which the filters are run on disjoint substates. A filtering technique, called *simplified* DBF (SDBF), and based on the interconnection of a particle filter (F_2) with a single Kalman filter (F_1), is developed for this case in ref. [32, Par. III-B]. The BIF algorithm paired with it can be easily derived following the approach illustrated above for the DBSA; the resulting smoothing algorithm is dubbed *disjoint* DBSA (DDBSA) in the following. It is important to mention that, in deriving the DBSA, the following relevant changes are made with respect to the DBSA (see Fig. 3.2):

1) The iterative procedure embedded in the $(T-k)-$th recursion of the backward pass involves both the computation of the backward predicted pdf ($BP1$) and of the message $MS1$ in BIF_1; for this reason, it requires marginalizing the pdfs $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ and $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$, respectively, with respect to $\mathbf{x}_k^{(L)}$. This result is achieved in the first iteration by setting $\mathbf{x}_k^{(N)} = \mathbf{x}_{\mathrm{fe},k}^{(N)}$ in both these pdfs, where $\mathbf{x}_{\mathrm{fe},k}^{(N)}$ denotes the estimate of $\mathbf{x}_k^{(N)}$ computed by F_2 in the forward pass. In the following iterations, we set $\mathbf{x}_k^{(N)} = \mathbf{x}_{\mathrm{sm},k}^{(N)}$, where $\mathbf{x}_{\mathrm{sm},k}^{(N)}$ represents the estimate of $\mathbf{x}_k^{(N)}$ evaluated on the basis of the statistical information provided by BIF_2 (through the message $SM2$).

2) The pseudo-measurement message $PM1$ (corresponding to $m_2^{(n)}(\mathbf{x}_k)$ (3.42) in the DBSA) conveys information about $\mathbf{x}_k^{(L)}$only. Moreover, it is a Gaussian message, and its mean and covariance matrix are given by $\eta_{L,k}^{(n)}$ and $\mathbf{C}_{LL,k}^{(n)}$ (see Eqs. (3.45) and (3.46), respectively).

Finally, it is worth mentioning that a *simplified* version of the DDBSA (called SDDBSA) can be easily developed by making the same modifications as those adopted in deriving the SDBSA from the DBSA.

3.4 Comparison of the Developed Double Smoothing Algorithms with Related Techniques

The DBSA and the DDBSA developed in the previous Section are conceptually related to the Rao-Blackwellised particle smoothers proposed by Fong *et al.* [62] and by Lindsten *et al.* [7] (these algorithms are denoted Alg-F and Alg-L respectively, in the following) and to the RBSS algorithm devised by Vitetta *et al.* in ref. [23]. In fact, all these techniques share with the DBSA and the DDBSA the following important features: 1) all of them estimate the *joint* smoothing density over the whole observation interval by generating multiple *realizations* from it; 2) they accomplish a single forward pass and as many backward passes as the overall number of realizations; 3) they combine Kalman filtering with particle filtering. However, Alg-F, Alg-L and the RBSS algorithm employ, in both their forward and backward passes, as many Kalman filters as the number of particles (N_p) to generate a particle-dependent estimate of the linear state component only. On the contrary, the DBSA (DDBSA) employs a *single* extended Kalman filter (a *single* Kalman filter), that estimates the whole system state (the linear state component only); this substantially reduces the memory requirements of particle smoothing and, consequently, the overall number of memory accesses accomplished on the hardware platform on which smoothing is run. As far as the last point is concerned, the memory requirements of a smoothing algorithm can be roughly assessed by estimating the overall number of real quantities that need to be stored in both its forward pass and its backward pass. It can be shown that overall number of real quantities to be stored by MPF, DBF and SDBF in the forward pass of the considered smoothing algorithms is of

order $\mathcal{O}(M_{\mathrm{MPF}})$, $\mathcal{O}(M_{\mathrm{DBF}})$, and $\mathcal{O}(M_{\mathrm{SDBF}})$, respectively, with[4]

$$M_{\mathrm{MPF}} = N_p T\,(2D_L^2 + 2D_L + D_N + 1), \tag{3.111}$$

$$M_{\mathrm{DBF}} = T\,(2D^2 + 2D + N_p\,D_N + N_p) \tag{3.112}$$

and

$$M_{\mathrm{SDBF}} = T\,(2D_L^2 + 2D_L + N_p\,D_N + N_p\,). \tag{3.113}$$

Moreover, the overall number of real quantities to be stored by Alg-L, RBSS, the DBSA and the DDBSA is approximately of order $\mathcal{O}(M_{\mathrm{Alg-L}})$, $\mathcal{O}(M_{\mathrm{RBSS}})$, $\mathcal{O}(M_{\mathrm{DBSA}})$ and $\mathcal{O}(M_{\mathrm{DDBSA}})$, respectively, with

$$M_{\mathrm{Alg-L}} = M_{\mathrm{MPF}} + D_L^2 + D, \tag{3.114}$$

$$M_{\mathrm{RBSS}} = M_{\mathrm{MPF}} + D_L^2 + D, \tag{3.115}$$

$$M_{\mathrm{DBSA}} = M_{\mathrm{DBF}} + N_p + D^2 + D + D_N \tag{3.116}$$

and

$$M_{\mathrm{DDBSA}} = M_{\mathrm{SDBF}} + N_p + D_L^2 + D. \tag{3.117}$$

The memory requirements of the SDBSA and the SDDBSA (the SPS algorithm) are the same as those of the DBSA and the DDBSA (the RBSS algorithm), respectively. Note also that the quantities M_{DBSA} (3.116) and M_{DDBSA} (3.117) are smaller than $M_{\mathrm{Alg-L}}$ (3.114) and M_{RBSS} (3.115), since M_{MPF} is larger than M_{DBF} and M_{SDBF} because of its dependence on N_p.

The differences in the overall execution time measured for the simulated smoothing algorithms are related not only to their requirements in terms of memory resources, but also to their computational complexity. In our work, the computational cost of the smoothing algorithms derived in the previous section has been carefully assessed in terms of number of *floating point operations* (flops) to be executed over the whole observation interval. The general criteria adopted in estimating the computational cost of an algorithm are the same as those illustrated in [40, App. A, p. 5420] and are not repeated here for space limitations. A detailed analysis of the cost required by each of the tasks accomplished by our smoothing algorithms is provided in Appendix E.5. Our analysis leads to the conclusion that the overall computational cost of the DBSA and of the DDBSA is approximately of order $\mathcal{O}(N_{\mathrm{DBSA}})$ and $\mathcal{O}(N_{\mathrm{DDBSA}})$, respectively, with

$$\begin{aligned} N_{\mathrm{DBSA}} = T\,\{&N_{\mathrm{DBF}} + M\,[38D^3/3 + 20D_N^3/3 + \\ &n_i N_p(2D_L^2 D_N + 2D_L D_N^2 \\ &+ D_N^3/3 + 5D_L^3) + 6n_i D^3]\,\}, \end{aligned} \tag{3.118}$$

and

$$\begin{aligned} N_{\mathrm{DDBSA}} = T\,\{&N_{\mathrm{SDBF}} + M\,[38D_L^3/3 + 20D_N^3/3 \\ &+ n_i N_p(2D_L^2 D_N + 2D_L D_N^2 \\ &+ D_N^3/3 + 5D_L^3) + 6n_i D_L^3]\,\}; \end{aligned} \tag{3.119}$$

here, N_{DBF} and N_{SDBF} represent the computational complexity of a single recursion of the DBF and SDBF, respectively (see [32, Eqs. (97) and (98)]). Each of the expressions (3.118)–(3.119) has been derived as follows. First, the costs of all the tasks identified in Appendix E.5 have been summed; then, the resulting expression has been simplified, keeping only the dominant contributions due to matrix inversions, matrix products and Cholesky decompositions, and discarding all the contributions that originate from the evaluation of the matrices $\mathbf{A}_k^{(Z)}(\mathbf{x}_k^{(N)})$ (with $Z = L$ and N), $\mathbf{F}_k$, $\mathbf{H}_k$ and $\mathbf{B}_k$ and the functions $\mathbf{f}_k^{(Z)}(\mathbf{x}_k^{(N)})$ (with $Z = L$ and N), $\mathbf{f}_k(\mathbf{x}_k)$

[4] Note that the expressions (3.111)–(3.113) also account for the contributions due to measurement-based information (see Eqs. (3.102) and (3.104)).

and $\mathbf{g}_k(\mathbf{x}_k^{(N)})$. Moreover, the sampling of the particle set in each recursion of the backward pass has been ignored.

From Eqs. (3.118)–(3.119) it is easily inferred that the computational complexities of the DBSA and the DDBSA are approximately of order $\mathcal{O}(n_i M\, N_p D_L^3 T)$. A similar approach can be followed for Alg-L and the RBSS algorithm; this leads to the conclusion that their complexities are approximately of order $\mathcal{O}(M\, N_p D_L^3 T)$, i.e. of the same order of the complexities of the DBSA and of the DDBSA if $n_i = 1$ is assumed.

On the other hand, the SDBSA and the SDDBSA are conceptually related to the SPS algorithm devised by Vitetta *et al.* in ref. [23]. In fact, all these algorithms aim at solving problem **P.2** only (consequently, they are unable to generate the joint smoothed pdf $f(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$) and carry out a *single backward pass*. This property makes them much faster than Alg-L, the RBSS algorithm, the DBSA and the DDBSA in the computation of marginal smoothed densities. Finally, note that, similarly as the DBSA and the DDBSA techniques, the use of the SDBSA and the SDDBSA requires a substantially smaller number of memory accesses than the SPS algorithm, since the last algorithm employs MPF in its forward pass. Moreover, the computational cost of the SDBSA and the SDDBSA is approximately of order $\mathcal{O}(n_i N_p D_L^3 T)$, whereas that of the SPS algorithm is approximately of order $\mathcal{O}(N_p D_L^3 T)$; consequently, they are all of the same order if $n_i = 1$ is assumed.

3.5 Numerical Results

In this section we first compare, in terms of accuracy and execution time, the DBSA, the SDBSA, the DDBSA and the SDDBSA with Alg-L, the RBSS algorithm, and the SPS algorithm for a specific conditionally linear Gaussian SSM. The considered SSM is the same as the SSM#2 defined in [23] and describes the bidimensional motion of an agent. Its state vector in the k-th observation interval is defined as $\mathbf{x}_k \triangleq [\mathbf{v}_k^T, \mathbf{p}_k^T]^T$, where $\mathbf{v}_k \triangleq [v_{x,k}, v_{y,k}]^T$ and $\mathbf{p}_k \triangleq [p_{x,k}, p_{y,k}]^T$ (corresponding to $\mathbf{x}_k^{(L)}$ and $\mathbf{x}_k^{(N)}$, respectively) represent the agent velocity and position, respectively (their components are expressed in m/s and in m, respectively). The state update equations are

$$\mathbf{v}_{k+1} = \rho\, \mathbf{v}_k + T_s\, \mathbf{a}_k(\mathbf{p}_k) + \big(1-\rho\big)\, \mathbf{n}_{v,k} \tag{3.120}$$

and

$$\mathbf{p}_{k+1} = \mathbf{p}_k + T_s\, \mathbf{v}_k + (T_s^2/2)\ \ \mathbf{a}_k(\mathbf{p}_k) + \mathbf{n}_{p,k}, \tag{3.121}$$

where ρ is a forgetting factor (with $0 < \rho < 1$), T_s is the sampling interval, $\mathbf{n}_{v,k}$ is an *additive Gaussian noise* (AGN) vector characterized by the covariance matrix $\mathbf{I}_2$,

$$\mathbf{a}_k\big(\mathbf{p}_k\big) = -a_0 \frac{\mathbf{p}_k}{\big\|\mathbf{p}_k\big\|} \frac{1}{1+\big(\big\|\mathbf{p}_k\big\|/d_0\big)^2} \tag{3.122}$$

is the acceleration due to a force applied to the agent (and pointing towards the origin of our reference system), a_0 is a scale factor (expressed in m/s^2), d_0 is a *reference distance* (expressed in m), and $\mathbf{n}_{p,k}$ is an AGN vector characterized by the covariance matrix $\sigma_p^2\, \mathbf{I}_2$ and accounting for model inaccuracy. The measurement vector available in the k-th interval for state estimation is

$$\mathbf{y}_k = \mathbf{x}_k + \mathbf{e}_k, \tag{3.123}$$

where $\mathbf{e}_k \triangleq [\mathbf{e}_{v,k}^T, \mathbf{e}_{p,k}^T]^T$ and $\mathbf{e}_{v,k}$ ($\mathbf{e}_{p,k}$) is an AGN vector characterized by the covariance matrix $\sigma_{ev}^2\, \mathbf{I}_2$ ($\sigma_{ep}^2\, \mathbf{I}_2$).

In our computer simulations, following [23] and [1], the estimation accuracy of the considered smoothing techniques has been assessed by evaluating two *root mean square errors* (RMSEs), one for the linear state component, the other for the nonlinear one, over an observation interval lasting $T = 200\ T_s$; these are denoted RMSE$_L$(alg) and RMSE$_N$(alg), respectively, where 'alg' is the acronym of the algorithm these parameters refer to. Our assessment of the computational

requirements is based, instead, on evaluating the average *computation time* required for processing a single *block* of measurements (this quantity is denoted CTB(alg) in the following). Moreover, the following values have been selected for the parameters of the considered SSM: $\rho = 0.995$, $T_s = 0.01$ s, $\sigma_p = 5 \cdot 10^{-3}$ m, $\sigma_{e,p} = 2 \cdot 10^{-2}$ m, $\sigma_{e,v} = 2 \cdot 10^{-2}$ m/s, $a_0 = 0.5$ m/s^2, $d_0 = 5 \cdot 10^{-3}$ m and $v_0 = 1$ m/s (the initial position $\mathbf{p}_0 \triangleq [p_{x,0}, p_{y,0}]^T$ and the initial velocity $\mathbf{v}_0 \triangleq [v_{x,0}, v_{y,0}]^T$ have been set to $[0.01 \text{ m}, 0.01 \text{ m}]^T$ and $[0.01 \text{ m/s}, 0.01 \text{ m/s}]^T$, respectively).

Some numerical results showing the dependence of RMSE_L and RMSE_N on the number of particles (N_p) for some of the considered smoothing algorithms are illustrated in Figs. 3.5 and 3.6, respectively (simulation results are indicated by markers, whereas continuous lines are drawn to fit them, so facilitating the interpretation of the available data). In this case, $n_i = 1$ has been selected for all the derived particle smoothers, $M = N_p$ has been chosen for all the smoothing algorithms generating multiple trajectories and the range $[10, 150]$ has been considered for N_p (since no real improvement is found for $N_p \gtrsim 150$). Moreover, RMSE_L and RMSE_N results are also provided for MPF and DBF, since these filtering techniques are employed in the forward pass of Alg-L, the RBSS algorithm and the SPS algorithm, and the DBSA and the SDBSA, respectively; this allows us to assess the improvement in estimation accuracy provided by the backward pass with respect to the forward pass for each smoothing algorithm. These results show that:

1) The DBSA, the SDBSA, Alg-L and the RBSS algorithm achieve similar accuracies in the estimation of both the linear and nonlinear state components.

2) The SPS algorithm is slightly outperformed by the other smoothing algorithms in terms of RMSE_N only; for instance, RMSE_N(SPS) is about 1.11 times larger than RMSE_N(SDBSA) for $N_p = 100$.

3) Even if the RBSS algorithm and the DBSA provide by far richer statistical information than their simplified counterparts (i.e., than the SPS algorithm and the SDBSA, respectively), they do not provide a significant improvement in the accuracy of state estimation; for instance, RMSE_N(SPS) (RMSE_N(SDBSA)) is about 1.12 (1.03) time larger than RMSE_N(RBSS) (RMSE_N(DBSA)) for $N_p = 100$.

4) The accuracy improvement in terms of RMSE_L (RMSE_N) provided by all the smoothing algorithms except the SPS (by Alg-L, the RBSS algorithm, the DBSA and the SDBSA) is about 24% (about 23%) with respect to MPF and DBF, for $N_p = 100$. Moreover, the accuracy improvement in terms of RMSE_L (RMSE_N) achieved by the SPS algorithm is about 24% (about 14%) with respect to the MPF for $N_p = 100$.

5) In the considered scenario, DBF is slightly outperformed by (perform similarly as) MPF in the estimation of the linear (nonlinear) state component; a similar result is reported in [32] for a different SSM. Our simulations have also evidenced that the DBSA and the SDBSA perform similarly as the DDBSA and the SDDBSA; for this reason, RSME results referring to the last two algorithms are not shown in Figs. 3.5 and 3.6. This leads to the conclusion that, in the considered scenario, the presence of redundancy in double Bayesian smoothing does not provide any improvement with respect to the case in which the two interconnected filters operate on disjoint substates in the forward and in the backward passes. Note that the same conclusion had been reached in ref. [32, Sec. IV] for DBF only. Despite their similar accuracies, the considered smoothing algorithms require different computational efforts; this is easily inferred from the numerical results appearing in Fig. 3.7 and illustrating the dependence of the CTB on N_p for all the above mentioned filtering and smoothing algorithms. In fact, these results show that CTB(DBSA) is approximately 0.85 (0.48) times smaller than CTB(Alg-L) (CTB(RBSS)); this is in agreement with the mathematical results illustrated in Section 3.4 about the complexity of Alg-L, the RBSS algorithms and the DBSA, i.e. with the fact the complexities of all these smoothers are approximately of order $\mathcal{O}(M\, N_p\, D_L^3 T)$ (provided that $n_i = 1$ is selected for the DBSA). Moreover, we have found that a 5.5% reduction in CTB is obtained if the DDBSA is employed in place of the DBSA (i.e., if double Bayesian smoothing is not redundant). Similar considerations hold for the SDBSA, the SDDBSA and the SPS algorithm. In fact, CTB(SDBSA)

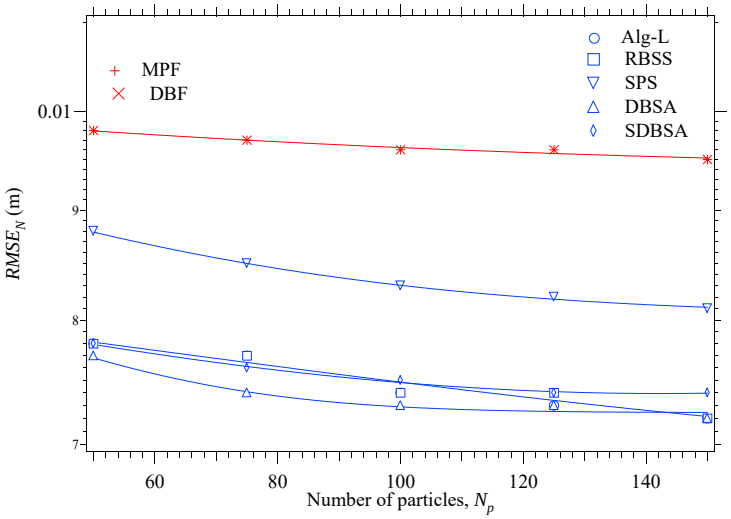


Figure 3.5: RMSE performance versus N_p for the nonlinear component (RMSE_N) of the state of SSM #1; five smoothing algorithms (Alg-L, the DBSA, the SDBSA, the RBSS algorithm and the SPS algorithm) and two filtering techniques (MPF and DBF) are considered.

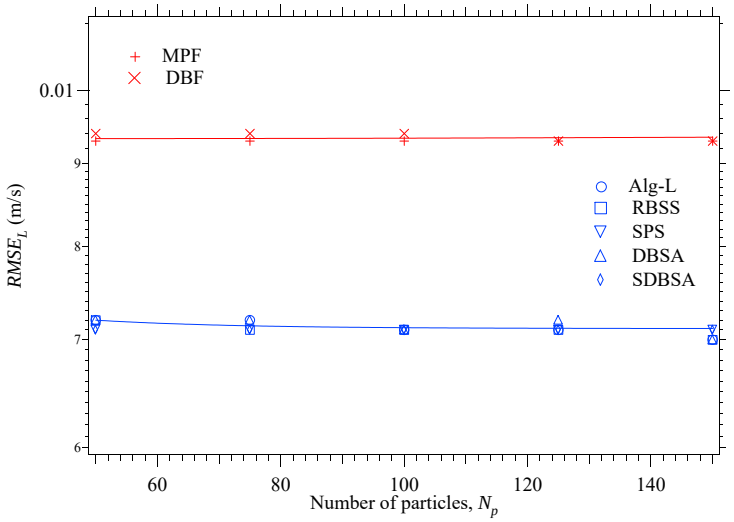


Figure 3.6: RMSE performance versus N_p for the linear component (RMSE_L) of the state of SSM #1; five smoothing algorithms (Alg-L, the DBSA, the SDBSA, the RBSS algorithm and the SPS algorithm) and two filtering techniques (MPF and DBF) are considered.

is approximately 0.57 times smaller than CTB(SPS); moreover, the CTB is reduced by 6.8% if the SDDBSA is employed in place of the SDBSA. It is also interesting to note that CTB(DBF) is approximately 0.55 times smaller than CTB(MPF) for the same value of N_p; once again, this result is in agreement with the results shown in [32] for a different SSM. All the numerical results illustrated above lead to the conclusion that, in the considered scenario, the DDBSA and the SDDBSA achieve the best accuracy-complexity tradeoff in their categories of smoothing techniques. The second SSM we considered is the same as the second SSM illustrated in [32, Sec. IV] and refers to a sensor network employing P sensors placed on the vertices of a square grid (partitioning a square area having side equal to l m); these sensors receive the reference signals radiated, at the same power level and at the same frequency, by N independent targets moving on a plane. Each target evolves according to the motion model described by Eqs. (3.120)–(3.121) with $\mathbf{a}_k(\mathbf{p}_k) = \mathbf{0}$ for any k. In this case, the considered SSM (denoted SSM#2 in the following) refers to the whole set of targets and its state vector $\mathbf{x}_k$ results from the ordered concatenation of the vectors $\{\mathbf{x}_k^{(i)};\ i = 1,\ 2,\ ...,\ N\}$, where $\mathbf{x}_k^{(i)} \triangleq [(\mathbf{v}_k^{(i)})^T, (\mathbf{p}_k^{(i)})^T]^T$, and $\mathbf{v}_k^{(i)}$ and $\mathbf{p}_k^{(i)}$ represent the

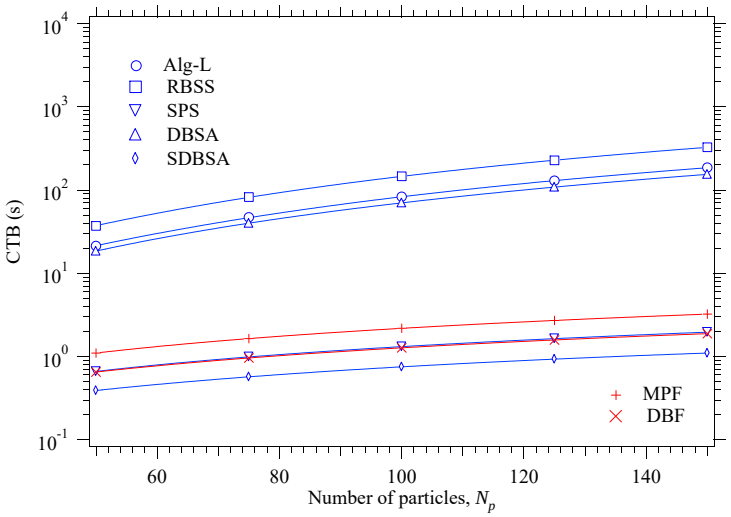


Figure 3.7: CTB versus N_p for five smoothing algorithms (Alg-L, DBSA, SDBSA, the RBSS algorithm and the SPS algorithm) and two filtering techniques (MPF and DBF); SSM #1 is considered.

$i-$th target *velocity* and the *position*, respectively. Moreover, the following additional assumptions have been made about this SSM: 1) the process noises $\mathbf{n}_{p,k}^{(i)}$ and $\mathbf{n}_{v,k}^{(i)}$, affecting the $i-$th target position and velocity, respectively, are given by $\mathbf{n}_{p,k}^{(i)} = (T_s^2/2)\,\mathbf{n}_{a,k}^{(i)}$ and $\mathbf{n}_{v,k}^{(i)} = T_s\,\mathbf{n}_{a,k}^{(i)}$, where $\{\mathbf{n}_{a,k}^{(i)}\}$ is two-dimensional AWGN, representing a random acceleration and having covariance matrix $\sigma_a^2\,\mathbf{I}_2$ (with $i = 1, 2, ..., N$); 2) the measurement acquired by the $q-$th sensor (with $q = 1, 2, ..., P$) in the k-th observation interval is given by

$$y_{q,k} = 10\log_{10}\left(\Psi\sum_{i=1}^{N}\frac{d_0^2}{\left|\left|\mathbf{s}_q - \mathbf{p}_k^{(i)}\right|\right|^2}\right) + e_k, \tag{3.124}$$

where the measurement noise $\{e_k\}$ is AWGN with variance σ_e^2, Ψ denotes the normalised power received by each sensor from any target at a distance d_0 from the sensor itself and $\mathbf{s}_q$ is the position of the considered sensor; 3) the overall measurement vector $\mathbf{y}_k$ results from the ordered concatenation of the measurements $\{y_{q,k};\ q = 1, 2, ..., P\}$ and, consequently, provides information about the position only; 4) the initial position $\mathbf{p}_0^{(i)} \triangleq [p_{x,0}^{(i)}, p_{y,0}^{(i)}]^T$ and the initial velocity $\mathbf{v}_0^{(i)} \triangleq [v_{x,0}^{(i)}, v_{y,0}^{(i)}]^T$ of the $i-$th target are randomly selected (with $i = 1, 2, ..., N$). As far as the last point is concerned, it is important to mention that, in our computer simulations, distinct targets are placed in different squares of the partitioned area in a random fashion; moreover, the initial velocity of each target is randomly selected within the interval $(v_{\min}, v_{\max})$ in order to ensure that the trajectories of distinct targets do not cross each other in the observation interval. The following values have been selected for the parameters of SSM#2: $P = 25$, $l = 10^3$ m, $T_s = 1$ s, $\rho = 1$, $\sigma_a^2 = 0.1$ m/s^2, $\sigma_e^2 = -35$ dB, $\Psi = 1$, $d_0 = 1$ m, $v_{\min} = 0$ m/s and $v_{\min} = 0.1$ m/s. Moreover, $N = 3$ targets have been observed over a time interval lasting $T = 60\ T_s$ s. Our computer simulations have aimed at evaluating the accuracy achieved by the considered smoothing algorithms in tracking the position of all the targets. In practice, such an accuracy has been assessed by estimating the average RMSE referring to the estimates of the whole set $\{\mathbf{p}_k^{(i)};\ i = 1, 2, 3\}$; note that, if the $i-$th target is considered, its position $\mathbf{p}_k^{(i)}$ represents the *nonlinear* component of the associated substate $\mathbf{x}_k^{(i)}$, because of the nonlinear dependence of $\mathbf{y}_k$ on it (see Eq. (3.124)). Our computer simulations have evidenced that, in the considered scenario, the MPF and the SDBF techniques diverge frequently in the observation interval (some numerical results about the probability of divergence area available in [32, Sec. IV]); unluckily, when this occurs, all the smoothing algorithms that employ these techniques in their forward pass (namely, Alg-L, the RBSS algorithm, the SPS algorithm, the DDBSA and the SDDBSA) are unable to

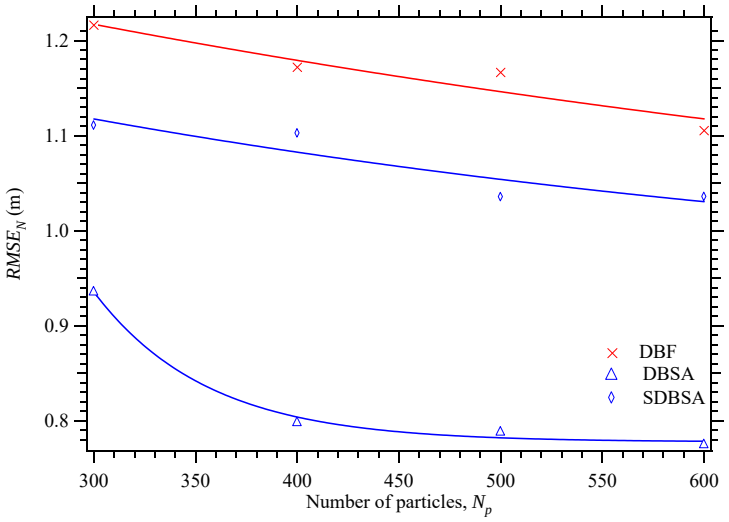


Figure 3.8: RMSE performance versus N_p for the nonlinear component (RMSE_N) of the state of SSM #2; two smoothing algorithms (the DBSA and the SDBSA) and one filtering technique (DBF) are considered.

recover from this event and, consequently, are useless. The DBF technique, instead, thanks to its inner redundancy, is still able to track all the targets. Moreover, the two smoothing algorithms employing this technique in their forward pass (namely, the DBSA and the SDBSA), are able to improve the accuracy of position estimates in their backward pass; this is evidenced by Fig. 3.8, that shows the dependence of RMSE_N on the overall number of particles (N_p) for the DBF technique, the DBSA and the SDBSA (the range $[300, 600]$ is considered for N_p). Note that the SDBSA is outperformed by the DBSA in terms of RMSE_N; for instance, RMSE_N(SDBSA) is about 1.31 times larger than RMSE_N(DBSA) for $N_p = 500$. However, this result is achieved at the price of a significantly higher complexity; in fact, CTB(SDBSA) is approximately equal to $2 \cdot 10^{-3}$·CTB(DBSA).

3.6 Conclusions

In this chapter, factor graph methods have been exploited to develop new smoothing algorithms based on the interconnection of two Bayesian filters in the forward pass and of two backward information filters in the backward pass. This has allowed us to develop a new approximate method for Bayesian smoothing, called *double Bayesian smoothing*. Four double Bayesian smoothers have been derived for the class of conditionally linear Gaussian systems and have been compared, in terms of both accuracy and execution time, with other smoothing algorithms for two specific dynamic models. Our simulation results lead to the conclusion that the devised algorithms can achieve a better complexity-accuracy tradeoff and a better tracking capability than other smoothing techniques recently appeared in the literature.

A

Factor Graphs and the Sum-Product Algorithm

A *factor graph* is a graphical model representing the factorization of any function $f(\cdot)$ expressible as a product of factors $\{f_i(\cdot)\}$, each depending on a set of variables $\{x_l\}$. In this thesis, Forney-style factor graphs are considered [17]. This means that the factor graph associated with the function $f(\cdot)$ consists of *nodes*, *edges* (connecting distinct nodes) and *half-edges* (connected to a single node only). Moreover, the following rules are employed for its construction: a) every *factor* is represented by a single node (a *rectangle* in our pictures); b) every *variable* is represented by a unique *edge* or *half edge*; c) the node representing a factor $f_i(\cdot)$ is connected with the edge (or half-edge) representing the variable x_l if and only if such a factor depends on x_l; d) an *equality constraint node* (represented by a rectangle labelled by "=") is used as a branching point when more than two factors are required to share the same variable[1]. For instance, the factorization of the function

$$f(x_1, x_2, x_3, x_4) = f_1(x_1)\ f_2(x_1, x_2)\ f_3(x_1, x_3)\ f_4(x_3, x_4) \tag{A.1}$$

can be represented through the factor graph shown in Fig. A.1.

Throughout this dissertation, factorizable functions represent joint pdfs. It is well known that the *marginalization* of $f(\cdot)$ with respect to one or more of its variables can be usually split into a sequence of simpler marginalizations if the factor graph representing $f(\cdot)$ is *cycle free*, that is it does not contain closed paths (known as *cycles*). Our interest in the graph representing $f(\cdot)$ is motivated by the fact that the function resulting from each of these marginalizations can be represented as a *message* (conveying a joint pdf of the variables it depends on) passed along an edge of the graph itself. For this reason, the overall procedure for marginalizing $f(\cdot)$ with respect to each of its variables can be seen as a step-by-step message passing over a graph. For instance, let us assume that we are interested in the marginal $f(x_3)$ originating from the joint pdf $f(x_1, x_2, x_3, x_4)$ (A.1). In this case, it in not difficult to prove that the required marginal can be evaluated as

$$f(x_3) = \vec{m}_4(x_3)\ \overleftarrow{m}_5(x_3), \tag{A.2}$$

where $\vec{m}_4(x_3)$ and $\overleftarrow{m}_5(x_3)$ are messages referring to the same edge, but coming from opposite directions; moreover, the evaluation of the last two messages is based on the ordered computation of the messages

$$\vec{m}_2(x_1) = \int f_2(x_1, x_2)\ \vec{m}_0(x_2)\ dx_2, \tag{A.3}$$

$$\vec{m}_3(x_1) = \vec{m}_1(x_1)\ \vec{m}_2(x_1), \tag{A.4}$$

$$\vec{m}_4(x_3) = \int f_3(x_1, x_3)\ \vec{m}_3(x_1)\ dx_1 \tag{A.5}$$

and

$$\overleftarrow{m}_5(x_3) = \int f_4(x_3, x_4)\ \overleftarrow{m}_6(x_4)\ dx_4, \tag{A.6}$$

[1] In the following, degree−3 equality constraint nodes will be always employed in our graphical models for the sake of clarity. Note that, generally speaking, an equality node of degree d can be represented by a tree with $(d-2)$ equality nodes of degree 3.

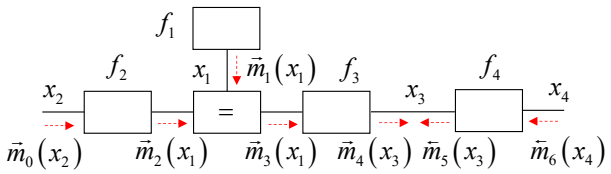


Figure A.1: Factor graph representing the structure of the function $f(x_1, x_2, x_3, x_4)$ (A.1) and message passing on it for the evaluation of the marginal $f(x_3)$.

where $\vec{m}_1(x_1) \triangleq f_1(x_1)$, $\vec{m}_0(x_2) \triangleq 1$ and $\overleftarrow{m}_6(x_4) \triangleq 1$. Eqs. (A.3)–(A.6) represent specific instances of a general rule, known as SPA and that can be formulated as follows (e.g., see [17, Sec. IV]): the message emerging from a node, representing a factor $f_i(\cdot)$, along the edge associated with a variable x_l is expressed by the product of $f_i(\cdot)$ and the messages along all the incoming edges (except that associated with x_l), integrated over all the involved variables except x_l. Note that, in the following, a straightforward generalization of this rule will be employed any time the joint pdf of a vector $\mathbf{x}_1$ is evaluated by marginalizing a factorizable pdf that refers to the disjoint random vectors $\mathbf{x}_1$ and $\mathbf{x}_2$.

Two simple applications of the SPA are illustrated in Fig. A.2-a) and in Fig. A.2-b), that refer to an *equality constraint node* and to a *function node*, respectively (note that, generally speaking, these nodes are connected to edges representing *vectors of variables* instead of single variables). On the one hand, the message $\vec{m}_{out}(\mathbf{x})$ emerging from the *equality node* shown in Fig. A.2-a) is evaluated as[2]

$$\vec{m}_{out}(\mathbf{x}) = \vec{m}_{in,1}(\mathbf{x})\ \vec{m}_{in,2}(\mathbf{x}), \tag{A.7}$$

where $\vec{m}_{in,1}(\mathbf{x})$ and $\vec{m}_{in,2}(\mathbf{x})$ are the two messages entering the node itself[3] and $\mathbf{x}$ is the vector of variables all these message refer to. On the other hand, the message $\vec{m}_{out}(\mathbf{x}_2)$ emerging from the *function node* shown Fig. A.2-b), that refers to the function $f(\mathbf{x}_1, \mathbf{x}_2)$ depending on the vectors of variables $\mathbf{x}_1$ and $\mathbf{x}_2$, is given by

$$\vec{m}_{out}(\mathbf{x}_2) = \int \vec{m}_{in}(\mathbf{x}_1)\ f(\mathbf{x}_1, \mathbf{x}_2)\ d\mathbf{x}_1, \tag{A.8}$$

where $\vec{m}_{in}(\mathbf{x}_1)$ denotes the message entering it. Note that Eq. (A.2), and Eqs. (A.3), (A.5) and (A.6) result from the application of Eq. (A.7) and Eq. (A.8), respectively. Moreover, it is worth mentioning that the following two rules have been also exploited in applying the SPA to the factor graph shown in Fig. A.1: a) the marginal pdf $f(x_l)$, referring to the variable x_l only, is expressed by the product of two messages associated with the edge x_l, but coming from opposite directions (see Eq. (A.2)); b) the half-edge associated with a variable x_l may be thought as carrying a constant message of unit value as incoming message (e.g., see the messages $\vec{m}_0(x_2)$ and $\overleftarrow{m}_6(x_4)$).

The result expressed by Eq. (A.2) is *exact* since the graph representing the joint pdf $f(x_1, x_2, x_3, x_4)$ (A.1) is *cycle free*. In principle, since the SPA is a 'local' rule, it can be also applied to factor graphs with cycles (e.g., see [17, Par. III.A] and [18, Sec. V]), but usually produces approximate results. Moreover, in this case, the messages passed on a cycle are usually recomputed[4] according to some schedule until a certain stopping criterion is satisfied; this unavoidably leads to the development of *iterative* message passing algorithms. Despite this, it is widely accepted that the most important applications of the SPA refer to graphs with cycles [18].

The last important issue related to the application of the SPA is the availability of closed-form expressions for the passed messages when, like in the filtering problem investigated in this thesis, the involved variables are continuous. In the following, the pdfs of all the considered random

[2] Note that, if the messages $\vec{m}_{in,1}(\mathbf{x})$ and $\vec{m}_{in,2}(\mathbf{x})$ represent pdfs, their product does not usually satisfy the normalization condition, so that resulting pdf is known up to a scale factor.

[3] If a single message $\vec{m}(\mathbf{x})$ enters an equality node, it is conventionally assumed that the two messages emerging from are copies of it.

[4] Note that the messages passed over a cycle need to be properly rescaled in order to ensure convergence (this prevents them from tending to zero or infinity).

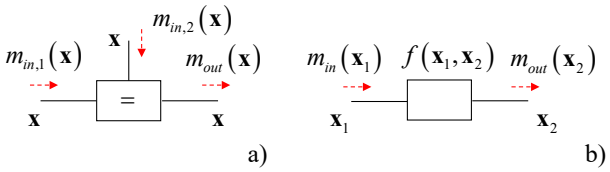


Figure A.2: Representation of the graphical models which Eqs. (A.7) (diagram a)) and (A.8) (diagram b)) refer to.

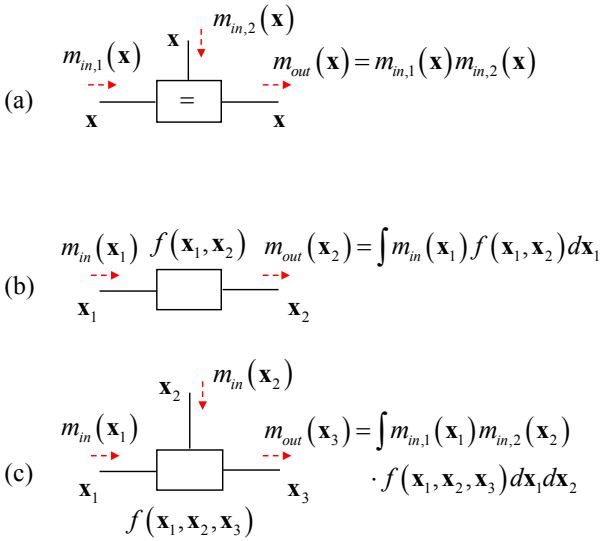


Figure A.3: Ordered representation of the graphs which the message passing formulas listed in Table A.1, Table A.2 and Table A.3, respectively, refer to.

vectors are Gaussian or are approximated through a set of N_p weighted particles. In the first case, the pdf of a random vector $\mathbf{x}$ is conveyed by the message

$$\vec{m}_G(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \eta, \mathbf{C}), \tag{A.9}$$

where η and $\mathbf{C}$ denote the mean vector and the covariance matrix of $\mathbf{x}$, respectively. In the second case, instead, its pdf is conveyed by the message

$$\vec{m}_P(\mathbf{x}) = \sum_{j=1}^{N_p} \vec{m}_{P,j}(\mathbf{x}), \tag{A.10}$$

where

$$\vec{m}_{P,j}(\mathbf{x}) \triangleq w_j \, \delta(\mathbf{x} - \mathbf{x}_j) \tag{A.11}$$

represents the j-th *component* of the message $\vec{m}_P(\mathbf{x})$ (A.10), that is the contribution of the j-th particle $\mathbf{x}_j$ and its weight w_j to such a message. Luckily, various closed-form results are available for these two types of messages; the few mathematical rules required in the computation of all the messages appearing in the algorithms described in Chapter 1, 2 and 3 are summarised in Tables A.1, A.2 and A.3, which refer to the FGs illustrated in Fig. A.3-(a), Fig. A.3-(b) and Fig. A.3-(c), respectively (note that in these Tables $\mathbf{a}$ denotes a constant vector); these results are provided by [17, Table 2, p. 1303] or based on standard mathematical results about Gaussian random variables (e.g., see [63, Par. 2.3.3]).

Table A.1: Mathematical rules for the evaluation of the message $m_{out}(\mathbf{x})$, emerging from the *equality node* shown in Fig. A.3-(a), on the basis of the given input messages $m_{in,1}(\mathbf{x})$ and $m_{in,2}(\mathbf{x})$.

Formula no.	$m_{in,1}(\mathbf{x})$	$m_{in,2}(\mathbf{x})$	$m_{out}(\mathbf{x})$
1	$\delta(\mathbf{x}-\mathbf{a})$	$f(\mathbf{x})$	$f(\mathbf{a})\,\delta(\mathbf{x}-\mathbf{a})$
2	$\mathcal{N}(\mathbf{x};\eta_1,\mathbf{C}_1)$	$\mathcal{N}(\mathbf{x};\eta_2,\mathbf{C}_2)$	$\mathcal{N}(\mathbf{x};\eta,\mathbf{C}), \mathbf{w}=\mathbf{w}_1+\mathbf{w}_2,$ $\mathbf{W}=\mathbf{W}_1+\mathbf{W}_2$

Table A.2: Mathematical rules for the evaluation of the message $m_{out}(\mathbf{x}_2)$, emerging from the *function node* shown in Fig. A.3-(b), on the basis of the given input messages $m_{in,1}(\mathbf{x}_1)$ and the function $f(\mathbf{x}_1,\mathbf{x}_2)$.

Formula no.	$m_{in}(\mathbf{x}_1)$	$f(\mathbf{x}_1,\mathbf{x}_2)$	$m_{out}(\mathbf{x}_2)$
1	$\mathcal{N}(\mathbf{x}_1;\eta_1,\mathbf{C}_1)$	$\mathcal{N}(\mathbf{x}_2;\mathbf{A}\mathbf{x}_1+\mathbf{g},\mathbf{C}_2)$	$\mathcal{N}(\mathbf{x}_2;\mathbf{A}\eta_1+\mathbf{g},\mathbf{A}\mathbf{C}_1\mathbf{A}_l^T+\mathbf{C}_2)$
2	$\delta(\mathbf{x}_1-\mathbf{a})$	$\mathcal{N}(\mathbf{x}_2;\mathbf{A}\mathbf{x}_1+\mathbf{g},\mathbf{C}_2)$	$\mathcal{N}(\mathbf{x}_2;\mathbf{A}\mathbf{a}+\mathbf{g},\mathbf{C}_2)$
3	$\delta(\mathbf{x}_1-\mathbf{a})$	$\mathcal{N}(\mathbf{x}_1;\mathbf{A}\mathbf{x}_2,\mathbf{C}_2)$	$\mathcal{N}(\mathbf{a};\mathbf{A}\mathbf{x}_2,\mathbf{C}_2)$
4	$\mathcal{N}(\mathbf{x}_1;\eta_1,\mathbf{C}_1)$	$\mathcal{N}(\mathbf{x}_1;\eta_2,\mathbf{C}_2)$	$\exp\{\frac{1}{2}[\eta^T\mathbf{W}\eta-\eta_1^T\mathbf{W}_1\eta_1-\eta_2^T\mathbf{W}_2\eta_2]\}$ $\mathbf{W}=\mathbf{W}_1+\mathbf{W}_2, \mathbf{w}=\mathbf{W}_1\eta_1+\mathbf{W}_2\eta_2$

Table A.3: Mathematical rule for the evaluation of the message $m_{out}(\mathbf{x}_3)$, emerging from the *function node* shown in Fig. A.3-(c), on the basis of the input messages $m_{in,1}(\mathbf{x}_1)$ and $m_{in,2}(\mathbf{x}_2)$, and of the function $f(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$.

Formula no.	$m_{in,1}(\mathbf{x}_1)$	$m_{in,2}(\mathbf{x}_2)$	$f(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$	$m_{out}(\mathbf{x}_3)$
1	$\delta(\mathbf{x}_1-\mathbf{a})$	$\mathcal{N}(\mathbf{x}_2;\eta_2,\mathbf{C}_2)$	$\mathcal{N}(\mathbf{x}_3;\mathbf{g}(\mathbf{x}_1)+\mathbf{A}(\mathbf{x}_1)\,\mathbf{x}_2,\mathbf{C}_3)$	$\mathcal{N}(\mathbf{x}_3;\mathbf{g}(\mathbf{a})+\mathbf{A}(\mathbf{a})\eta_2,$ $\mathbf{C}_3+\mathbf{A}(\mathbf{a})\mathbf{C}_2(\mathbf{A}(\mathbf{a}))^T)$

B

Message computation for Marginalized Particle Filtering and Dual Marginalized Particle Filtering

In this Appendix the derivation of the expressions of various messages evaluated by the MPF and the DMPF algorithms is sketched.

B.1 Marginalized Particle Filtering

Step 1) - The message $\vec{m}_{1,j}(\mathbf{x}_k^{(N)})$ emerges from the function node representing $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)}) = \mathcal{N}(\mathbf{y}_k; \mathbf{B}_k(\mathbf{x}_k^{(N)})\,\mathbf{x}_k^{(L)} + \mathbf{h}_k(\mathbf{x}_k^{(N)}), \mathbf{C}_e)$ and fed by $\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(L)})$ (1.12) (see Fig. 1.3); for this reason, the derivation of Eq. (1.13) is based on formula no. 1 of Table A.2. Eq. (1.15), instead, is obtained by first substituting Eqs. (1.11) and (1.13) in the *right-hand side* (RHS) of (1.14) and, then, applying formula no. 1 of Table A.1.
Step 2) - The message $\vec{m}_{1,j}(\mathbf{x}_k^{(L)})$ emerges from the function node representing $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ and fed by $\vec{m}_{\text{fp},j}(\mathbf{x}_k^{(N)})$ (1.11) (see Fig. 1.3); for this reason, based on Eq. (1.11) and formula no. 2 of Table A.2, it is easy to show that

$$\vec{m}_{1,j}\big(\mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{y}_k; \mathbf{B}_{k,j}\,\mathbf{x}_k^{(L)} + \mathbf{h}_{k,j}, \mathbf{C}_e\big); \tag{B.1}$$

putting this message in an equivalent Gaussian form referring to $\mathbf{x}_k^{(L)}$ results in Eq. (1.21). Eq. (1.23), instead, results from substituting Eqs. (1.12) and (1.21) in the RHS of Eq. (1.22) and, then, applying formula no. 2 of Table A.1.
Step 3) - The message $\vec{m}_{3,j}(\mathbf{x}_{k+1}^{(N)})$ emerges from the function node representing $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)}) = \mathcal{N}(\mathbf{x}_{k+1}^{(N)}; \mathbf{f}_k^{(N)}(\mathbf{x}_k^{(N)}) + \mathbf{A}_k^{(N)}(\mathbf{x}_k^{(N)})\,\mathbf{x}_k^{(L)}, \mathbf{C}_w^{(N)})$, and fed by the messages $\vec{m}_{2,j}(\mathbf{x}_k^{(N)})$ and $\vec{m}_{2,j}(\mathbf{x}_k^{(L)})$ (see Fig. 1.3). Therefore, substituting Eqs. (1.20) and (1.23) in the RHS of Eq. (1.28) and applying formula no. 1 of Table A.3 produces Eq. (1.29).
Step 4) - The message $\vec{m}_j(\mathbf{z}_k^{(L)})$ is given by (see Fig. 1.3)

$$\vec{m}_j\big(\mathbf{z}_k^{(L)}\big) = \int\int f\big(\mathbf{z}_k^{(L)}\big|\mathbf{x}_k^{(N)}, \mathbf{x}_{k+1}^{(N)}\big) \cdot f\big(\mathbf{x}_{k+1}^{(N)}\big|\mathbf{x}_k^{(N)}\big)\, f\big(\mathbf{x}_k^{(N)}\big)\, d\mathbf{x}_k^{(N)} d\mathbf{x}_{k+1}^{(N)}. \tag{B.2}$$

Since, in this case, $f(\mathbf{z}_k^{(L)}|\mathbf{x}_k^{(L)}, \mathbf{x}_k^{(N)}) = \mathcal{N}(\mathbf{z}_k^{(L)};\ \mathbf{A}_k^{(N)}(\mathbf{x}_k^{(N)})\ \mathbf{x}_k^{(L)}, \mathbf{C}_w^{(N)})$, $f(\mathbf{x}_k^{(N)}) = \delta(\mathbf{x}_k^{(N)} - \mathbf{x}_{\text{fe},k,j}^{(N)})$, $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_{\text{fp},k,j}^{(N)}) = \delta(\mathbf{x}_k^{(N+1)} - \mathbf{x}_{\text{fp},k+1,j}^{(N)})$ (see Eq. (1.20) and Eq. (1.32), respectively), Eq. (B.2) leads easily to Eq. (1.33). The message $\vec{m}_{3,j}(\mathbf{x}_k^{(L)})$ flows out of the node representing the function $f(\mathbf{z}_k^{(L)}|\mathbf{x}_k^{(L)}, \mathbf{x}_k^{(N)}) = \mathcal{N}(\mathbf{z}_k^{(L)};\mathbf{A}_k^{(N)}(\mathbf{x}_k^{(N)})\,\mathbf{x}_k^{(L)}, \mathbf{C}_w^{(N)})$ and fed by the messages $\vec{m}_j(\mathbf{z}_k^{(L)})$ and $\vec{m}_{2,j}(\mathbf{x}_k^{(N)})$ (see Fig. 1.3); consequently, based on Eq. (1.33), formulas no. 2 and 3 of Table A.2, it is easy to prove that

$$\vec{m}_{3,j}\big(\mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{z}_{k,j}^{(L)}; \mathbf{A}_{k,j}^{(N)}\,\mathbf{x}_k^{(L)}, \mathbf{C}_w^{(N)}\big); \tag{B.3}$$

this message can be easily put in the equivalent Gaussian form expressed by Eq. (1.34). Eq. (1.36), instead, is obtained by first substituting Eqs. (1.23) and (1.34) in the RHS of Eq. (1.35) and, then,

by applying formula no. 2 of Table A.1. Finally, the output message $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_{k+1}^{(L)})$ emerges from the node representing the function $f(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(L)},\mathbf{x}_k^{(N)}) = \mathcal{N}(\mathbf{x}_{k+1}^{(N)};\, \mathbf{f}_k^{(L)}(\mathbf{x}_k^{(N)}) + \mathbf{A}_k^{(L)}(\mathbf{x}_k^{(N)})\, \mathbf{x}_k^{(L)}, \mathbf{C}_w^{(L)})$ and fed by the messages $\vec{m}_{2,j}(\mathbf{x}_k^{(N)})$ and $\vec{m}_{4,j}(\mathbf{x}_k^{(L)})$. Therefore, substituting Eqs. (1.20) and (1.36) in the RHS of Eq. (1.39) and applying formula no. 1 of Table A.3 yields Eq. (1.40).

B.2 Dual Marginalized Particle Filtering

Step 3) - The message $\vec{m}_{3,j}(\mathbf{x}_{k+1}^{(L)})$ emerges from the function node representing $f(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(N)},\, \mathbf{x}_k^{(L)}) = \mathcal{N}(\mathbf{x}_{k+1}^{(L)};\, \mathbf{A}_k^{(L)}(\mathbf{x}_k^{(N)})\, \mathbf{x}_k^{(L)} + \mathbf{f}_k^{(L)}(\mathbf{x}_k^{(N)}), \mathbf{C}_w^{(L)})$ and fed by $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(L)})$ (1.12) and $\vec{m}_{2,j}(\mathbf{x}_k^{(L)})$ (1.23) (see Fig. 1.4); for this reason, the derivation of Eq. (1.45) is based on the application of formula no. 1 of Table A.3 to Eq. (1.44).
Step 4) The message $m_j(\mathbf{z}_k^{(N)})$ can be put in a *particle-dependent* Gaussian form since $\mathbf{x}_{k+1}^{(L)}$ and $\mathbf{x}_k^{(L)}$, conditioned on $\mathbf{x}_k^{(N)} = \mathbf{x}_{\mathrm{fp},k,j}^{(N)}$, are modelled as jointly Gaussian random vectors. In fact, if $\mathbf{x}_k^{(N)} = \mathbf{x}_{\mathrm{fp},k,j}^{(N)}$, the random vector $\mathbf{z}_k^{(N)}$ (1.10) should equal $\mathbf{x}_{k+1}^{(N)} - \mathbf{f}_k^{(N)}(\mathbf{x}_{\mathrm{fp},k,j}^{(N)})$; consequently, adopting the particle-dependent Gaussian model (1.60) for $\mathbf{x}_{k+1}^{(N)}$ results in a particle-dependent Gaussian model for $\mathbf{z}_k^{(N)}$ too. Moreover, since, as shown in Fig. 1.4, the statistical representations of $\mathbf{x}_k^{(L)}$ and $\mathbf{x}_{k+1}^{(L)}$ employed in the computation of $m_j(\mathbf{z}_k^{(N)})$ are conveyed by messages $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(L)})$ (1.12) and $\vec{m}_{3,j}(\mathbf{x}_{k+1}^{(L)})$ (1.45), respectively, the mean and covariance of $\mathbf{z}_k^{(N)}$ (1.10) can be evaluated as

$$\eta_{\mathbf{z},k,j}^{(N)} = \eta_{3,k,j}^{(L)} - \bar{\mathbf{A}}_{k,j}^{(L)}\, \eta_{\mathrm{fp},k,j}^{(L)} \tag{B.4}$$

and

$$\mathbf{C}_{\mathbf{z},k,j}^{(N)} = \mathbf{C}_{3,k,j}^{(L)} + \bar{\mathbf{A}}_{k,j}^{(L)}\, \mathbf{C}_{\mathrm{fp},k,j}^{(L)} \big(\bar{\mathbf{A}}_{k,j}^{(L)}\big)^T - \bar{\mathbf{A}}_{k,j}^{(L)}\, \mathbf{C}_{\mathbf{x},k,j}^{(L)} - \big(\mathbf{C}_{\mathbf{x},k,j}^{(L)}\big)^T \big(\bar{\mathbf{A}}_{k,j}^{(L)}\big)^T, \tag{B.5}$$

respectively; here, $\mathbf{C}_{\mathbf{x},k,j}^{(L)}$ denotes the cross covariance matrix for the vectors $\mathbf{x}_k^{(L)}$ and $\mathbf{x}_{k+1}^{(L)}$ (conditioned on $\mathbf{x}_k^{(N)} = \mathbf{x}_{\mathrm{fp},k,j}^{(N)}$) and $\bar{\mathbf{A}}_{k,j}^{(L)} \triangleq \mathbf{A}_k^{(L)}(\mathbf{x}_{\mathrm{fp},k,j}^{(N)})$. Given $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(L)})$ (1.12) and the conditional pdf $f(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(L)},\mathbf{x}_{\mathrm{fp},k,j}^{(N)}) = \mathcal{N}(\mathbf{x}_{k+1}^{(L)};\, \bar{\mathbf{f}}_{k,j}^{(L)} + \bar{\mathbf{A}}_{k,j}^{(L)}\, \mathbf{x}_k^{(L)}, \mathbf{C}_w^{(L)})$ (where $\bar{\mathbf{f}}_{k,j}^{(N)} \triangleq \mathbf{f}_k^{(N)}(\mathbf{x}_{\mathrm{fp},k,j}^{(N)})$), it is easy to show that $\mathbf{C}_{\mathbf{x},k,j}^{(L)} = \mathbf{C}_{2,k,j}^{(L)}(\bar{\mathbf{A}}_{k,j}^{(L)})^T$ (e.g., see [63, Par. 2.3.3, eq. (2.104)]); consequently, Eq. (B.5) can be rewritten as

$$\mathbf{C}_{\mathbf{z},k,j}^{(N)} = \mathbf{C}_{3,k,j}^{(L)} - \bar{\mathbf{A}}_{k,j}^{(L)}\, \mathbf{C}_{\mathrm{fp},k,j}^{(L)} \big(\bar{\mathbf{A}}_{k,j}^{(L)}\big)^T. \tag{B.6}$$

Then, substituting Eqs. (1.46) and (1.47) in Eqs. (B.4) and (B.6), respectively, produces Eqs. (1.49) and (1.50), respectively. The message $\vec{m}_{3,j}(\mathbf{x}_k^{(N)})$ emerges from the function node referring to the pdf $f(\mathbf{z}_k^{(N)}|\mathbf{x}_k^{(N)}) = \mathcal{N}(\mathbf{z}_k^{(N)}; \mathbf{f}_k^{(L)}(\mathbf{x}_k^{(N)}), \mathbf{C}_w^{(L)})$ and fed by the messages $\vec{m}_j(\mathbf{z}_k^{(N)})$ and $\vec{m}_{\mathrm{fp},j}(\mathbf{x}_k^{(N)})$ (see Fig. 1.4); therefore, substituting Eqs. (1.11) and (1.48) in the RHS of Eq. (1.51) and applying formulas no. 2 and 4 of Table A.2 produces Eq. (1.52).
Step 5) The message $\vec{m}_{5,j}(\mathbf{x}_{k+1}^{(N)})$ emerges from the function node representing $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)},\mathbf{x}_k^{(L)}) = \mathcal{N}(\mathbf{x}_{k+1}^{(N)};\, \mathbf{f}_k^{(N)}(\mathbf{x}_k^{(N)}) + \mathbf{A}_k^{(N)}(\mathbf{x}_l^{(N)})\, \mathbf{x}_k^{(L)},\, \mathbf{C}_w^{(N)})$ and fed by the messages $\vec{m}_{2,j}(\mathbf{x}_k^{(L)})$ and $\vec{m}_{4,j}(\mathbf{x}_k^{(N)})$; therefore, substituting Eqs. (1.23) and (1.58) in the RHS of Eq. (1.59) and applying formula no. 1 of Table A.3 yields Eq. (1.60).

C

Message computation for Double Bayesian Filtering

In this Appendix, the derivation of the expressions of various messages evaluated in each of the three phases which the DBF technique consists of is sketched. Our derivations are mainly based on various formulas listed in the tables of refs. [17] and Appendix A, and resulting from the application of the SPA to specific function nodes/messages.

Phase I - Message $\vec{m}_1(\mathbf{x}_k)$ (2.31) conveys the pdf $\tilde{f}(\mathbf{y}_k|\mathbf{x}_k)$ (2.17); therefore, it can be expressed as $\vec{m}_1(\mathbf{x}_k) = \mathcal{N}\left(\mathbf{y}_k; \mathbf{H}_k^T \mathbf{x}_k + \mathbf{v}_k, \mathbf{C}_e\right)$. The last formula can be easily put in the equivalent form (2.31) (see [17, Table 3, p. 1304, Eqs. (III.5) and (III.6)]). Then, substituting eqs. (2.28) and (2.31) in the RHS of Eq. (2.34) and applying formula no. 2 of Table A.1 yields Eqs. (2.35)–(2.37).

Phase II - Step 1) The derivation of the formulas (2.39), (2.42) and (2.43) referring to the message $\vec{m}_3^{(n)}(\mathbf{x}_k)$ can be considered as a straightforward application of formula no. 2 of Table A.1, since Eq. (2.38) has exactly the same structure as Eq. (A.7), and both $\vec{m}_2(\mathbf{x}_k)$ and $\vec{m}_4^{(n-1)}(\mathbf{x}_k)$ are Gaussian messages.

Step 2) - The expression (2.48) of the weight $w_{1,k,j}^{(n)}$ can be derived as follows. We first substitute Eq. (2.21) (conditioned on $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}[n]$) and Eq. (2.44) in the RHS of Eq. (2.47); then, the resulting integral is solved by applying formula no. 1 of Table A.2.

Step 3) - The derivation of the expression (2.52) for the weight $w_{3,k,j}^{(n)}$ is similar to that illustrated for the particle weights originating from the pseudo-measurements in *dual* RBPF and can be summarized as follows (additional mathematical details can be found in [1, Sec. V, pp. 1528–1529]). Two different Gaussian densities are derived for the random vector $\mathbf{z}_k^{(N)}$ (2.25), *conditioned on* $\mathbf{x}_k^{(N)}$. The expression of the first density originates from the definition (2.25) and from the knowledge of the *joint* pdf of $\mathbf{x}_k^{(L)}$ and $\mathbf{x}_{k+1}^{(L)}$; this joint density is obtained from: a) the statistical information provided by the messages $\vec{m}_2(\mathbf{x}_k^{(L)}) = \mathcal{N}(\mathbf{x}_k^{(L)}; \tilde{\eta}_{2,k}, \tilde{\mathbf{C}}_{2,k})$ and $\vec{m}_3^{(n)}(\mathbf{x}_k^{(L)}) = \mathcal{N}(\mathbf{x}_k^{(L)}; \tilde{\eta}_{3,k}^{(n)}, \tilde{\mathbf{C}}_{3,k}^{(n)})$, resulting from the marginalization of $\vec{m}_2(\mathbf{x}_k)$ (2.35) and $\vec{m}_3^{(n)}(\mathbf{x}_k)$ (2.39), respectively, with respect to $\mathbf{x}_k^{(N)}$; b) the Markov model $f(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ (2.27). This leads to the pdf

$$f_1^{(n)}\left(\mathbf{z}_k^{(N)}|\mathbf{x}_k^{(N)}\right) = \mathcal{N}\left(\mathbf{z}_k^{(N)}; \check{\eta}_{z,k}^{(n)}\left(\mathbf{x}_k^{(N)}\right), \check{\mathbf{C}}_{z,k}^{(n)}\left(\mathbf{x}_k^{(N)}\right)\right), \tag{C.1}$$

where

$$\check{\eta}_{z,k}^{(n)}\left(\mathbf{x}_k^{(N)}\right) = \mathbf{A}_k^{(L)}\left(\mathbf{x}_k^{(N)}\right)\left[\tilde{\eta}_{3,k}^{(n)} - \tilde{\eta}_{2,k}\right] + \mathbf{f}_k^{(L)}\left(\mathbf{x}_k^{(N)}\right), \tag{C.2}$$

and

$$\check{\mathbf{C}}_{z,k}^{(n)}\left(\mathbf{x}_k^{(N)}\right) = \mathbf{C}_w^{(L)} + \mathbf{A}_k^{(L)}\left(\mathbf{x}_k^{(N)}\right)\left[\tilde{\mathbf{C}}_{3,k}^{(n)} - \tilde{\mathbf{C}}_{2,k}\right] \cdot \left(\mathbf{A}_k^{(L)}\left(\mathbf{x}_k^{(N)}\right)\right)^T. \tag{C.3}$$

The second pdf of $\mathbf{z}_k^{(N)}$, instead, results from the fact that this vector $\mathbf{z}_k^{(N)}$ must equal the sum (2.26); consequently, it is given by

$$f_2\left(\mathbf{z}_k^{(N)}|\mathbf{x}_k^{(N)}\right) = \mathcal{N}\left(\mathbf{z}_k^{(N)}; \mathbf{f}_k^{(L)}\left(\mathbf{x}_k^{(N)}\right), \mathbf{C}_w^{(N)}\right). \tag{C.4}$$

Given the pdfs (C.1) and (C.4), the message $\vec{m}_3^{(n)}(\mathbf{x}_k^{(N)})$ is expressed by their *correlation*, since it is computed as

$$\vec{m}_3^{(n)}(\mathbf{x}_k^{(N)}) = \int f_1^{(n)}\big(\mathbf{z}_k^{(N)}|\mathbf{x}_k^{(N)}\big) \cdot f_2\big(\mathbf{z}_k^{(N)}|\mathbf{x}_k^{(N)}\big) d\mathbf{z}_k^{(N)}. \tag{C.5}$$

Substituting (C.1) and (C.4) in the RHS of the last expression, setting $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}[n]$ and applying formula no. 4 of Table A.2 to the evaluation of the resulting integral yields Eq. (2.52); note that $\check{\eta}_{z,k,j}^{(n)}$ (2.56) and $\check{\mathbf{C}}_{z,k,j}^{(n)}$ (2.55) represent the values taken on by $\check{\eta}_{z,k}^{(n)}(\mathbf{x}_k^{(N)})$ (C.2) and $\check{\mathbf{C}}_{z,k}^{(n)}(\mathbf{x}_k^{(N)})$ (C.3), respectively, for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}[n]$.

Step 4) - Formula (2.59), that refers to the message $\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(N)})$, is obtained by substituting $\vec{m}_{2,j}(\mathbf{x}_k^{(N)})$ (2.51) in the RHS of Eq. (2.58) and observing that $w_{3,k,j}^{(n)}$ (2.52) represents the value taken on by the message $\vec{m}_3^{(n)}(\mathbf{x}_k^{(N)})$ for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}[n]$.

Step 5) Eq. (2.63) is results from substituting Eqs. (2.44) and (C.5) in Eq. (2.62) and, then, applying formula no. 1 of Table A.3 to evaluate the resulting integral.

Step 6) - The message $\vec{m}_4^{(n)}(\mathbf{x}_k)$ (2.67) results from merging, in the $F_2 \rightarrow F_1$ block, the statistical information about the *nonlinear* state component conveyed by the message $\vec{m}_4^{(n)}(\mathbf{x}_k^{(N)})$ (and, consequently, by its components $\{\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(N)})\}$; see Eq. (2.61)) with those provided by the pseudo-measurement $\mathbf{z}_k^{(L)}$ (2.22) about the *linear* state component. The method employed for processing this pseudo-measurement is the same as that developed for RBPF and can be summarized as follows (additional mathematical details can be found in [1, Sec. IV, p. 1527]):

a) The particles $\mathbf{x}_{k,j}^{(N)}[n+1]$ and $\bar{\mathbf{x}}_{k+1,j}^{(N)}[n+1]$, conveyed by the messages $\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(N)})$ (2.61) and $\vec{m}_{\mathrm{fp},j}^{(n)}(\mathbf{x}_{k+1}^{(N)})$ (2.66), respectively, are employed to compute the j-th realization $\mathbf{z}_{k,j}^{(L)}[n+1]$ (2.77) of the vector $\mathbf{z}_k^{(L)}$ (2.22) according to Eq. (2.77).

b) The pseudo-measurement $\mathbf{z}_{k,j}^{(L)}[n+1]$ (2.77) is exploited to generate the (particle-dependent) message

$$\vec{m}_{4,j}^{(n)}\big(\mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{x}_k^{(L)}; \tilde{\eta}_{4,k,j}^{(n)}, \tilde{\mathbf{C}}_{4,k,j}^{(n)}\big), \tag{C.6}$$

that conveys pseudo-measurement information about $\mathbf{x}_k^{(L)}$; the covariance matrix $\tilde{\mathbf{C}}_{4,k,j}^{(n)}$ and the mean vector $\tilde{\eta}_{4,k,j}^{(n)}$ of this message are computed on the basis of the precision matrix $\tilde{\mathbf{W}}_{4,k,j}^{(n)}$ (2.75) and the transformed mean vector $\tilde{\mathbf{w}}_{4,k,j}^{(n)}$ (2.76), respectively. Finally, the message $\vec{m}_4^{(n)}(\mathbf{x}_k)$ (2.67) results from merging the message $\vec{m}_4^{(n)}(\mathbf{x}_k^{(N)})$ (its j-th component is expressed by Eq. (2.61)) with the pdfs $\{\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(L)})\}$ (see Eq. (C.6)); the adopted approach is based on the fact that: a) as it can be easily inferred from our previous derivations, the Gaussian message $\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(L)})$ (C.6) is evaluated under the condition that $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}[n+1]$; b) the messages $\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(N)})$ and $\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(L)})$ provide *complementary* information, because they refer to the two different components of the overall state $\mathbf{x}_k$. Consequently, the statistical information conveyed by the sets $\{\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(N)})\}$ and $\{\vec{m}_{4,j}^{(n)}(\mathbf{x}_k^{(L)})\}$ can be merged in the joint pdf

$$f^{(k)}\big(\mathbf{x}_k^{(L)}, \mathbf{x}_k^{(N)}\big) \triangleq w_p \sum_{j=1}^{N_p} \vec{m}_{4,j}^{(n)}\big(\mathbf{x}_k^{(N)}\big)\, \vec{m}_{4,k}^{(n)}\big(\mathbf{x}_k^{(L)}\big). \tag{C.7}$$

referring to $\mathbf{x}_k$. Then, the message $\vec{m}_4^{(n)}(\mathbf{x}_k)$ (2.67) is evaluated by *projecting* the pdf $f^{(k)}(\mathbf{x}_k^{(L)}, \mathbf{x}_k^{(N)})$ (C.7) onto a single Gaussian pdf having the same *mean vector* and *covariance matrix*. It is worth noting that, unlike all the other messages considered above and below, the computation of the last message is not based on the SPA.

Phase III - The message $\vec{m}_{\text{fp}}(\mathbf{x}_{k+1})$ (2.81) is computed as follows. Substituting the expressions (2.16) of $\tilde{f}(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and (2.78) of $\vec{m}_3^{(n_i+1)}(\mathbf{x}_k)$ in the RHS of Eq. (2.80) and applying formula no. 1 of Table A.2 to the evaluation of the resulting integral produces Eqs. (2.81)–(2.83).

D

Message computation for Double Bayesian Smoothing Algorithm

In this Appendix, the derivation of the expressions of various messages evaluated in each of the three phases the DBSA consists of is sketched.

Phase I - Formulas (3.39) and (3.40), referring to the message $\bar{m}_1(\mathbf{x}_k)$ (3.38), can be easily computed by applying eqs. (IV.6)-(IV.8) of ref. [17, Table 4, p.1304] in their backward form (with $A \rightarrow \mathbf{I}_D$, $X \rightarrow \mathbf{F}_k\mathbf{x}_k$, $Z \rightarrow \mathbf{x}_{k+1}$ and $Y \rightarrow \mathbf{u}_k + \mathbf{w}_k$) and, then, eqs. (III.5)–(III.6) of [17, Table 3, p.1304] (with $A \rightarrow \mathbf{F}_k$, $X \rightarrow \mathbf{x}_k$ and $Y \rightarrow \mathbf{F}_k\mathbf{x}_k$).

Phase II -Step 1) The message $m_2^{(n)}(\mathbf{x}_k)$ (3.42) results from merging, in the $\text{BIF}_2 \rightarrow \text{BIF}_1$ block, the statistical information about the *nonlinear* state component conveyed by the message $m_1^{(n-1)}(\mathbf{x}_k^{(N)})$ (3.41) (and, consequently, by its N_p components $\{m_{1,j}^{(n-1)}(\mathbf{x}_k^{(N)}) = W_{1,k,j}^{(n-1)}\,\delta(\mathbf{x}_k^{(N)} - \mathbf{x}_{k,j}^{(N)})\}$) with those provided by the pseudo-measurement $\mathbf{z}_k^{(L)}$ (3.21) about the *linear* state component. The method employed for processing this pseudo-measurement is the same as that developed for MPF and can be summarised as follows (additional mathematical details can be found in [1, Sec. IV, p. 1527]):

1) The particles $\mathbf{x}_{k,j}^{(N)}$ and $\mathbf{x}_{\text{be},k+1}^{(N)}$, conveyed by the messages $m_1^{(n-1)}(\mathbf{x}_k^{(N)})$ (3.41) and $\bar{m}_{\text{be}}(\mathbf{x}_{k+1}^{(N)})$ (3.37), respectively, are employed to compute the $j-$th realization $\mathbf{z}_{k,j}^{(L)}$ (3.49) of $\mathbf{z}_k^{(L)}$ for $j = 1, 2, ..., N_p$.

2) The pseudo-measurement $\mathbf{z}_{k,j}^{(L)}$ (3.49) is exploited to generate the (particle-dependent) pdf

$$f_j^{(n)}\big(\mathbf{x}_k^{(L)}\big) = \mathcal{N}\big(\mathbf{x}_k^{(L)}; \tilde{\eta}_{k,j}, \tilde{\mathbf{C}}_{k,j}\big), \tag{D.1}$$

that conveys pseudo-measurement information about $\mathbf{x}_k^{(L)}$ for any j; the covariance matrix $\tilde{\mathbf{C}}_{k,j}$ and the mean vector $\tilde{\eta}_{k,j}$ of this message are computed on the basis of the precision matrix $\tilde{\mathbf{W}}_{k,j}$ (3.47) and the transformed mean vector $\tilde{\mathbf{w}}_{k,j}$ (3.48), respectively.

3) The messages $\{m_{1,j}^{(n-1)}(\mathbf{x}_k^{(N)})\}$ are merged with the pdfs $\{f_j^{(n)}(\mathbf{x}_k^{(L)})\}$ to generate the message $m_2^{(n)}(\mathbf{x}_k)$ (3.42). The approach we adopt to achieve this result is based on the fact that the message $m_{1,j}^{(n-1)}(\mathbf{x}_k^{(N)})$ and the pdf $f_j^{(n)}(\mathbf{x}_k^{(L)})$ refer to the same particle (i.e., to the $j-$th particle $\mathbf{x}_{k,j}^{(N)}$, but provide *complementary* information (since they refer to the two different components of the overall state $\mathbf{x}_k$). This allows us to condense the statistical information conveyed by the sets $\{m_{1,j}^{(n-1)}(\mathbf{x}_k^{(N)})\}$ and $\{f_j^{(n)}(\mathbf{x}_k^{(L)})\}$ in the joint pdf

$$f^{(n)}(\mathbf{x}_k^{(L)}, \mathbf{x}_k^{(N)}) \triangleq w_p \sum_{j=1}^{N_p} m_{1,j}^{(n-1)}\big(\mathbf{x}_k^{(N)}\big)\, f_j^{(n)}\big(\mathbf{x}_k^{(L)}\big). \tag{D.2}$$

referring to the whole state $\mathbf{x}_k$. Then, the message $m_2^{(n)}(\mathbf{x}_k)$ (3.42) is computed by projecting the pdf $f^{(k)}(\mathbf{x}_k^{(L)}, \mathbf{x}_k^{(N)})$ (D.2) onto a single Gaussian pdf having the same *mean* and *covariance*.

Steps 2 and 3) The expression (3.52) of $\overleftarrow{m}_3^{(n)}(\mathbf{x}_k)$ represents a straightforward application of formula no. 2 of Table A.1 (with $\mathbf{W}_1 \rightarrow \mathbf{W}_{1,k}$, $\mathbf{W}_2 \rightarrow \mathbf{W}_{2,k}^{(n)}$, $\mathbf{w}_1 \rightarrow \mathbf{w}_{1,k}$ and $\mathbf{w}_2 \rightarrow \mathbf{w}_{2,k}^{(n)}$). The same considerations apply to the derivation of the expression (3.58) of $m_4^{(n)}(\mathbf{x}_k)$.

Step 4) The expression (3.63) of the weight $w_{3,k,j}^{(n)}$ is derived as follows. First, we substitute the expression (3.19) of $f(\mathbf{x}_{k+1}^{(N)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$, and the expressions of the messages $\overleftarrow{m}_{be}(\mathbf{x}_{k+1}^{(N)})$ (3.37) and $m_1^{(n)}(\mathbf{x}_k^{(L)})$ (3.61) in the *right-hand side* (RHS) of Eq. (3.62). Then, the resulting integral is solved by applying formula no. 1 of Table A.2 in the integration with respect to $\mathbf{x}_k^{(L)}$ and the sifting property of the Dirac delta function in the integration with respect to $\mathbf{x}_{k+1}^{(N)}$.

Step 5) - The derivation of the expression (3.68) of the weight $w_{2,k,j}^{(n)}$ is similar to that illustrated for the particle weights originating from the pseudo-measurements in dual MPF and can be summarised as follows (additional mathematical details can be found in ref. [1, Sec. V, pp. 1528-1529]). Two different Gaussian densities are derived for the random vector $\mathbf{z}_k^{(N)}$ (3.24), *conditioned on* $\mathbf{x}_k^{(N)}$. The expression of the first density originates from the definition (3.24) and from the knowledge of the *joint* pdf of $\mathbf{x}_k^{(L)}$ and $\mathbf{x}_{k+1}^{(L)}$; this joint density is obtained from: a) the statistical information provided by the message $m_1^{(n)}(\mathbf{x}_k^{(L)})$ (3.61) and the pdf $\mathcal{N}(\mathbf{x}_{k+1}^{(L)}, \tilde{\eta}_{\mathrm{be},k+1}\tilde{\mathbf{C}}_{\mathrm{be},k+1})$ (resulting from integrating out the dependence of $\overleftarrow{m}_{\mathrm{be}}(\mathbf{x}_{k+1})$ (3.36) on $\mathbf{x}_k^{(N)}$); b) the Markov model $f(\mathbf{x}_{k+1}^{(L)}|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$ (3.26). This leads to the pdf

$$f_1^{(n)}\big(\mathbf{z}_k^{(N)}\big|\mathbf{x}_k^{(N)}\big) = \mathcal{N}\big(\mathbf{z}_k^{(N)}; \check{\eta}_{z,k}^{(n)}\big(\mathbf{x}_k^{(N)}\big), \check{\mathbf{C}}_{z,k}^{(n)}\big(\mathbf{x}_k^{(N)}\big)\big), \tag{D.3}$$

where

$$\check{\eta}_{z,k}^{(n)}\big(\mathbf{x}_k^{(N)}\big) = \tilde{\eta}_{\mathrm{be},k+1} - \mathbf{A}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)\,\tilde{\eta}_{1,k}^{(n)} \tag{D.4}$$

and

$$\check{\mathbf{C}}_{z,k}^{(n)}\big(\mathbf{x}_k^{(N)}\big) = \tilde{\mathbf{C}}_{\mathrm{be},k+1} - \mathbf{A}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)\tilde{\mathbf{C}}_{1,k}^{(n)}\left(\mathbf{A}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big)\right)^T. \tag{D.5}$$

The second pdf of $\mathbf{z}_k^{(N)}$, instead, results from the fact that this vector $\mathbf{z}_k^{(N)}$ must equal the sum (3.25); consequently, it is given by

$$f_2\big(\mathbf{z}_k^{(N)}\big|\mathbf{x}_k^{(N)}\big) = \mathcal{N}\big(\mathbf{z}_k^{(N)}; \mathbf{f}_k^{(L)}\big(\mathbf{x}_k^{(N)}\big), \mathbf{C}_w^{(N)}\big). \tag{D.6}$$

Given the pdfs (D.3) and (D.6), the message $\vec{m}_3^{(n)}(\mathbf{x}_k^{(N)})$ is expressed by their *correlation*, i.e. it is computed as

$$\vec{m}_3^{(n)}(\mathbf{x}_k^{(N)}) = \int f_1^{(n)}\big(\mathbf{z}_k^{(N)}\big|\mathbf{x}_k^{(N)}\big) \cdot f_2\big(\mathbf{z}_k^{(N)}\big|\mathbf{x}_k^{(N)}\big) d\mathbf{z}_k^{(N)}. \tag{D.7}$$

Substituting Eqs. (D.3) and (D.6) in the RHS of the last expression, setting $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}$ and applying formula no. 4 of Table A.2 to the evaluation of the resulting integral yields Eq. (3.68); note that $\check{\eta}_{z,k,j}^{(n)}$ (3.70) and $\check{\mathbf{C}}_{z,k,j}^{(n)}$ (3.71) represent the values taken on by $\check{\eta}_{z,k}^{(n)}(\mathbf{x}_k^{(N)})$ (D.4) and $\check{\mathbf{C}}_{z,k}^{(n)}(\mathbf{x}_k^{(N)})$ (D.5), respectively, for $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}$.

Step 7) The expression (3.82) of the weight $w_{5,k,j}^{(n)}$ is derived as follows. First, we substitute the expressions (3.61) and (3.20) of $m_1^{(n)}(\mathbf{x}_k^{(L)})$ and $f(\mathbf{y}_k|\mathbf{x}_k^{(N)}, \mathbf{x}_k^{(L)})$, respectively, in the RHS of Eq. (3.77). Then, solving the resulting integral (see formula no. 1 of Table A.2) produces Eq. (3.78). Finally, setting $\mathbf{x}_k^{(N)} = \mathbf{x}_{k,j}^{(N)}$ in Eq. (3.78) yields Eq. (3.82).

Phase III - The expression (3.100) of the message $\overleftarrow{m}_{\mathrm{be}}(\mathbf{x}_k)$ results from the application of formula no. 2 of Table A.1 to Eq. (3.99).

E

Computational complexity of various techniques

In this chapter, the computational complexity of various techniques employed in this work is evaluated in terms of flops.

E.1 Computational complexity of the DBF and SDBF techniques

In this appendix, the computational complexity of the tasks accomplished in a single recursion of the DBF technique is assessed in terms of flops. Moreover, we comment on how the illustrated results can be also exploited to assess the computational complexity of a single recursion of the SDBF technique. In the following, $\mathcal{C}_{\mathbf{H}}$, $\mathcal{C}_{\mathbf{B}}$, $\mathcal{C}_{\mathbf{F}}$, $\mathcal{C}_{\mathbf{A}^{(L)}}$ and $\mathcal{C}_{\mathbf{A}^{(N)}}$, and $\mathcal{C}_{\mathbf{g}}$, $\mathcal{C}_{\mathbf{f}^{(L)}}$, $\mathcal{C}_{\mathbf{f}^{(N)}}$ and $\mathcal{C}_{\mathbf{f}_k}$denote the cost due to the evaluation of the matrices $\mathbf{H}_k$, $\mathbf{B}_k$, $\mathbf{F}_k$, $\mathbf{A}_k^{(L)}(\mathbf{x}_k^{(N)})$ and $\mathbf{A}_k^{(N)}(\mathbf{x}_k^{(N)})$, and of the functions $\mathbf{g}_k(\mathbf{x}_k^{(N)})$, $\mathbf{f}_k^{(L)}(\mathbf{x}_k^{(N)})$, $\mathbf{f}_k^{(N)}(\mathbf{x}_k^{(N)})$ and $\mathbf{f}_k(\mathbf{x}_k)$, respectively. Moreover, similarly as [40], it is assumed that the computation of the inverse of any covariance matrix involves a Cholesky decomposition of the matrix itself and the inversion of a lower or upper triangular matrix.

1. Filter F_1, first measurement update - The overall computational cost of this task is (see Eqs. (2.32)–(2.33) and (2.36)–(2.37))

$$\mathcal{C}_{MU1}^{(1)} = \mathcal{C}_{\mathbf{W}_{2,k}} + \mathcal{C}_{\mathbf{w}_{2,k}} + \mathcal{C}_{\mathbf{C}_{2,k}} + \mathcal{C}_{\eta_{2,k}} \tag{E.1}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\mathbf{W}_{2,k}}$ is equal to $\mathcal{C}_{\mathbf{H}} + 2PD^2 + 2P^2D - PD$ flops; 2) the cost $\mathcal{C}_{\mathbf{w}_{2,k}}$ is equal to $\mathcal{C}_{\mathbf{B}} + \mathcal{C}_{\mathbf{g}} + 2P^2D + 5PD_L + 3PD_N - P$ flops ($\mathbf{H}_k$ has been already computed at point 1); 3) the cost $\mathcal{C}_{\mathbf{C}_{2,k}}$ is equal to $2D^3/3 + 3D^2/2 + 5D/6$ flops; 4) the cost $\mathcal{C}_{\eta_{2,k}}$ is equal to $D(2D-1)$ flops. The expressions listed at points 1)-4) can be exploited for the SDBF too; in the last case, however, $D_N = 0$ and $D = D_L$ must be assumed.

2. Filter F_1, second measurement update - The overall computational cost of this task is (see Eqs. (2.42)–(2.43))

$$\mathcal{C}_{MU2}^{(1)} = n_i \left(\mathcal{C}_{\mathbf{C}_{3,k}^{(n)}} + \mathcal{C}_{\eta_{3,k}^{(n)}}\right), \tag{E.2}$$

where the costs $\mathcal{C}_{\mathbf{C}_{3,k}^{(n)}}$ and $\mathcal{C}_{\eta_{3,k}^{(n)}}$ are equal to $D^2(2D-1)$ flops and $4D^2 - D$ flops, respectively; if the SDBF is considered, we have that $D = D_L$ in the last two expressions.

3. Filter F_2, first measurement update - The overall computational cost of this task is (see Eqs. (2.48)–(2.50))

$$\mathcal{C}_{MU1}^{(2)} = n_i\, N_p \left(\mathcal{C}_{\tilde{\eta}_{1,k,j}^{(n)}} + \mathcal{C}_{\tilde{\mathbf{C}}_{1,k,j}^{(n)}} + \mathcal{C}_{w_{1,k,j}^{(n)}}\right). \tag{E.3}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\tilde{\eta}_{1,k,j}^{(n)}}$ is equal to $\mathcal{C}_{\mathbf{B}} + \mathcal{C}_{\mathbf{g}} + 2PD_L$ flops; 2) the cost $\mathcal{C}_{\tilde{\mathbf{C}}_{1,k,j}^{(n)}}$ is equal to $2PD_L^2 + 2P^2D_L - PD_L$ flops (the cost for computing $\mathcal{C}_{\mathbf{B}}$ has been already accounted for at point 1)); 3) the cost $\mathcal{C}_{w_{1,k,j}^{(n)}}$ is equal to $(4P^3 + 21P^2 + 17P + 6)/6$ flops.

4. Filter F_2, second measurement update - The overall computational cost of this task is (see Eqs. (2.57) and (2.60))

$$\mathcal{C}_{MU2}^{(2)} = \mathcal{C}_{w_{4,k,j}^{(n)}} + \mathcal{C}_{W_{4,k,j}^{(n)}} + n_i\, \mathcal{C}_R(N_p), \tag{E.4}$$

where the costs $\mathcal{C}_{w_{4,k,j}^{(n)}}$ and $\mathcal{C}_{W_{4,k,j}^{(n)}}$ are equal to $n_i N_p$ flops and $2N_p - 1$ flops, respectively, and $\mathcal{C}_R(N_p)$ denotes the total cost of the resampling step (that involves a particle set of size N_p).

5. Computation of the pseudo-measurements for filter F_2 - The overall computational cost of this task is (see Eqs. (2.52)–(2.56))

$$\begin{aligned}\mathcal{C}_{1\to 2} = n_i\, N_p \big(&\mathcal{C}_{\check{\eta}_{z,k}^{(n)}} + \mathcal{C}_{\check{\mathbf{C}}_{z,k}^{(n)}} + \mathcal{C}_{\check{\mathbf{W}}_{z,k}^{(n)}} + \mathcal{C}_{\check{\mathbf{w}}_{z,k}^{(n)}} + \\ &\mathcal{C}_{\check{\mathbf{W}}_{3,k,j}^{(n)}} + \mathcal{C}_{\check{\mathbf{w}}_{3,k,j}^{(n)}} + \mathcal{C}_{\check{\mathbf{C}}_{3,k,j}^{(n)}} + \mathcal{C}_{\eta_{3,k,j}^{(n)}} + \mathcal{C}_{w_{3,k,j}^{(n)}}\big).\end{aligned} \tag{E.5}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\check{\eta}_{z,k}^{(n)}}$ is equal to $\mathcal{C}_{\mathbf{A}^{(L)}} + \mathcal{C}_{\mathbf{f}^{(L)}} + 2D_L^2 + D_L$ flops; 2) the cost $\mathcal{C}_{\check{\mathbf{C}}_{z,k}^{(n)}}$ is equal $4D_L^3$ flops (since the cost for computing $\mathcal{C}_{\mathbf{A}^{(L)}}$ has been already accounted for at point 1); 3) the cost $\mathcal{C}_{\check{\mathbf{W}}_{z,k}^{(n)}}$ is equal to $2D_L^3/3 + 3D_L^2/2 + 5D_L/6$ flops; 4) the cost $\mathcal{C}_{\check{\mathbf{w}}_{z,k}^{(n)}}$ is equal to $D_L(2D_L - 1)$ flops; 5) the cost $\mathcal{C}_{\check{\mathbf{W}}_{3,k,j}^{(n)}}$ is equal to D_L^2 flops; 6) the cost $\mathcal{C}_{\check{\mathbf{w}}_{3,k,j}^{(n)}}$ is equal to $2D_L^2$ flops (the cost for computing $\mathcal{C}_{\mathbf{f}^{(L)}}$ has been already accounted for at point 1); 7) the cost $\mathcal{C}_{\check{\mathbf{C}}_{3,k,j}^{(n)}}$ is equal to $2D_L^3/3 + 3D_L^2/2 + 5D_L/6$ flops; 8) the cost $\mathcal{C}_{\eta_{3,k,j}^{(n)}}$ is equal to $D_L(2D_L - 1)$ flops; 9) the cost $\mathcal{C}_{w_{3,k,j}^{(n)}}$ is equal to $6D_L^2 + 3D_L + 1$ flops (the cost for computing $\mathcal{C}_{\mathbf{f}^{(L)}}$ has been already accounted for at point 1)).

6. Computation of the pseudo-measurements for filter F_1 - The overall computational cost of this task is (see Eqs. (2.69)–(2.77))

$$\begin{aligned}\mathcal{C}_{2\to 1} = n_i\, N_p \big(&\mathcal{C}_{\mathbf{z}_{k,j}^{(L)}} + \mathcal{C}_{\tilde{\mathbf{W}}_{4,k,j}^{(n)}} + \mathcal{C}_{\tilde{\mathbf{w}}_{4,k,j}^{(n)}} + \mathcal{C}_{\tilde{\mathbf{C}}_{4,k,j}^{(n)}} + \mathcal{C}_{\tilde{\eta}_{4,k,j}^{(n)}}\big) \\ &+ \mathcal{C}_{\mathbf{C}_{4,k}^{(n)}} + \mathcal{C}_{\eta_{4,k}^{(n)}} + \mathcal{C}_{\mathbf{W}_k^{(n)}}.\end{aligned} \tag{E.6}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\mathbf{z}_{k,j}^{(L)}}$ is equal to D_N flops (the cost for computing $\mathcal{C}_{\mathbf{f}^{(N)}}$ has been already accounted for in the time update of filter F_2); 2) the cost $\mathcal{C}_{\tilde{\mathbf{W}}_{4,k,j}^{(n)}}$ is equal to $DD_L(2D_N - 1)$ flops (the cost for computing $\mathcal{C}_{\mathbf{A}^{(N)}}$ has been already accounted for in the time update of filter F_2); 3) the cost $\mathcal{C}_{\tilde{\mathbf{w}}_{4,k,j}^{(n)}}$ is equal to $D_L(2D_N^2 + D_N - 1)$ flops (the cost for computing $\mathcal{C}_{\mathbf{A}^{(N)}}$ has been already accounted for in the time update of filter F_2); 4) the cost $\mathcal{C}_{\tilde{\mathbf{C}}_{4,k,j}^{(n)}}$ is equal to $2D_L^3/3 + 3D_L^2/2 + 5D_L/6$ flops; 5) the cost $\mathcal{C}_{\tilde{\eta}_{4,k,j}^{(n)}}$ is equal to $D_L(2D_L - 1)$ flops; 6) the cost $\mathcal{C}_{\mathbf{C}_{4,k}^{(n)}}$ is equal to $n_i(2N_pD_L^2 + N_pD_N^2 + 2D_L^2 + 2D_N^2 + N_pD_LD_N + 2D_LD_N + 3N_p)$ flops; 7) the cost $\mathcal{C}_{\eta_{4,k}^{(n)}}$ is equal to $n_i(D(N_p - 1) + 1)$ flops; 8) the cost $\mathcal{C}_{\mathbf{W}_k^{(n)}}$ is $n_i(16D^3 + 9D^2 + 5D)/6$ flops.

If the SDBF is considered, the total costs 1)-5) remain unchanged, whereas $D_N = 0$ and $D = D_L$ in the costs $\mathcal{C}_{\eta_{4,k}^{(n)}}$ and $\mathcal{C}_{\mathbf{W}_k^{(n)}}$ (see points 7) and 8)); moreover, the cost $\mathcal{C}_{\mathbf{C}_{4,k}^{(n)}}$ becomes $n_i(2N_pD_L^2 + 2D_L^2 + N_p)$ flops (see point 6)).

7. Filter F_1, time update - The overall computational cost of this task is (see Eqs. (2.82)–(2.83))

$$\mathcal{C}_{TU}^{(1)} = \mathcal{C}_{\eta_{\mathrm{fp},k+1}} + \mathcal{C}_{\mathbf{C}_{\mathrm{fp},k+1}} + \mathcal{C}_{\mathbf{W}_{\mathrm{fp},k+1}} + \mathcal{C}_{\mathbf{w}_{\mathrm{fp},k+1}}, \tag{E.7}$$

since $\mathcal{C}_{\mathbf{C}_{3,k}^{(n_i+1)}}$ and $\mathcal{C}_{\eta_{3,k}^{(n_i+1)}}$ have been already computed in the previous time update of filter F_2. Moreover, we have that:

1) $\mathcal{C}_{\eta_{\mathrm{fp},k+1}}$ is equal to $\mathcal{C}_{\mathbf{f}_k}$ flops; 2) $\mathcal{C}_{\mathbf{C}_{\mathrm{fp},k+1}}$ is equal to $\mathcal{C}_{\mathbf{F}} + D^2(4D - 1)$ flops; 3) $\mathcal{C}_{\mathbf{W}_{\mathrm{fp},k+1}}$ is equal to $2D^3/3 + 3D^2/2 + 5D/6$ flops; 4) $\mathcal{C}_{\mathbf{w}_{\mathrm{fp},k+1}}$ is equal to $D(2D - 1)$ flops. If the SDBF is considered, $D = D_L$ is set in the expressions of the costs listed at points 1)-4).

8. Filter F_2, time update - The overall computational cost of this task is (see Eqs. (2.64)–(2.66))

$$\mathcal{C}_{TU}^{(2)} = n_i\, N_p \big(\mathcal{C}_{\eta_{3,k,j}^{(N)}} + \mathcal{C}_{\mathbf{C}_{3,k,j}^{(N)}} + \mathcal{C}_{\mathbf{x}_{k+1,j}^{(N)}}\big). \tag{E.8}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\eta^{(N)}_{3,k,j}}$ is equal to $\mathcal{C}_{\mathbf{A}^{(N)}} + \mathcal{C}_{\mathbf{f}^{(N)}} + 2D_L D_N$ flops; 2) the cost $\mathcal{C}_{\mathbf{C}^{(N)}_{3,k,j}}$ is equal to $D_L D_N(2D-1)$ flops ($\mathcal{C}_{\mathbf{A}^{(N)}}$ has been already accounted for at point 1)); 3) the cost $\mathcal{C}_{\mathbf{x}^{(N)}_{k+1,j}}$ is equal to $D_N^3/3 + 3D_N^2 + 5D_N/3$ flops.

Based on the results illustrated above, it can be proved that the overall computational cost of the DBF/SDBF techniques is approximately of order $\mathcal{O}(N_{\text{alg}})$, where the parameter alg is equal to DBF or SDBF (see Eqs. (2.85) and (2.86), respectively). Moreover, in both cases, N_{alg} can be put in the form

$$N_{\text{alg}} = N^{(1)}_{\text{alg}} + n_i\, N_p\, (N^{(2)}_{\text{alg}} + N^{(1\to 2)}_{\text{alg}} + N^{(2\to 1)}_{\text{alg}}), \tag{E.9}$$

where $N^{(1)}_{\text{alg}}$, $N^{(2)}_{\text{alg}}$, $N^{(1\to 2)}_{\text{alg}}$ and $N^{(2\to 1)}_{\text{alg}}$ denote the contributions to N_{alg} originating from filter F_1, from filter F_2, from the $\text{F}_1 \to \text{F}_2$ block and from the $\text{F}_2 \to \text{F}_1$ block, respectively. The expressions of these contributions are

$$N^{(1)}_{\text{DBF}} = 2\,(n_i\, D^3 + PD^2 + 2P^2 D + 8D^3/3), \tag{E.10}$$

$$N^{(2)}_{\text{DBF}} = 2\,(PD_L^2 + P^2 D_L + P^3/3 + DD_L D_N + D_N^3/6), \tag{E.11}$$

$$N^{(1\to 2)}_{\text{DBF}} = 16\, D_L^3/3 \tag{E.12}$$

and

$$N^{(2\to 1)}_{\text{DBF}} = 2\,(DD_L D_N + D_L D_N^2 + D_L^3/3 + 4D^3/(3\, N_p)) \tag{E.13}$$

for the DBF technique. On the other hand, if the SDBF technique is considered, we have that $N^{(2)}_{\text{SDBF}} = N^{(2)}_{\text{DBF}}$ and $N^{(1\to 2)}_{\text{SDBF}} = N^{(1\to 2)}_{\text{DBF}}$ (see Eqs. (E.11) and (E.12), respectively); moreover, $N^{(2\to 1)}_{\text{SDBF}} = 2\,(D\, D_L\, D_N + D_L\, D_N^2 + D_L^3/3 + 4D_L^3/(3\, N_p))$, whereas the expression of $N^{(1)}_{\text{SDBF}}$ is easily obtained setting $D = D_L$ in the RHS of Eq. (E.10).

E.2 Computational complexity of the EKF technique

In this appendix analysis of EKF complexity is illustrated; the notation is the same as [3, pp. 194-195]. In the following, $\mathcal{C}_{\mathbf{H}}$ and $\mathcal{C}_{\mathbf{F}}$, $\mathcal{C}_{\mathbf{h}_k}$ and $\mathcal{C}_{\mathbf{f}_k}$ denote the cost due to the evaluation of the matrices $\mathbf{H}_k$ and $\mathbf{F}_k$, and of the functions $\mathbf{h}_k(\mathbf{x}_k)$ and $\mathbf{f}_k(\mathbf{x}_k)$, respectively. Moreover, similarly as [40], it is assumed that the computation of the inverse of any covariance matrix involves a Cholesky decomposition of the matrix itself and the inversion of a lower or upper triangular matrix.

1. **Measurement update**

The overall computational cost of this task is

$$\mathcal{C}_{MU} = \mathcal{C}_{\mathbf{\Omega}_k} + \mathcal{C}_{\mathbf{L}_k} + \mathcal{C}_{\eta_{k|k}} + \mathcal{C}_{\mathbf{C}_{k|k}}. \tag{E.14}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\mathbf{\Omega}_k}$ is equal to $\mathcal{C}_{\mathbf{H}} + 2P^2 D + 2PD^2 - PD$ flops; 2) $\mathcal{C}_{\mathbf{L}_k}$ is equal to $2P^3/3 + 3P^2/2 + 5P/6 + 2PD^2 + 2P^2 D - 2PD$ flops; 3) $\mathcal{C}_{\eta_{k|k}}$ is equal to $\mathcal{C}_{\mathbf{h}_k} + 2PD + P$ flops; 4) $\mathcal{C}_{\mathbf{C}_{k|k}}$ is equal $2D^3 + 2PD^2 - D^2$ flops.

2. **Time update**

The overall computational cost of this task is

$$\mathcal{C}_{TU} = \mathcal{C}_{\eta_{k+1|k}} + \mathcal{C}_{\mathbf{C}_{k+1|k}}, \tag{E.15}$$

where the costs $\mathcal{C}_{\eta_{k+1|k}}$ and $\mathcal{C}_{\mathbf{C}_{k+1|k}}$ are equal to $\mathcal{C}_{\mathbf{f}_k}$ flops and $\mathcal{C}_{\mathbf{F}} + 4D^3 - D^2$ flops, respectively.

E.3 Computational complexity of the RBPF technique

In this appendix a detailed analysis of the RBPF complexity is provided; the adopted notation is the same as [1]. In the following, $\mathcal{C}_{\mathbf{B}}$, $\mathcal{C}_{\mathbf{A}^{(L)}}$ and $\mathcal{C}_{\mathbf{A}^{(N)}}$, and $\mathcal{C}_{\mathbf{g}}$, $\mathcal{C}_{\mathbf{f}^{(L)}}$ and $\mathcal{C}_{\mathbf{f}^{(N)}}$ denote the cost due to the evaluation of the matrices $\mathbf{B}_k$, $\mathbf{A}_k^{(L)}(\mathbf{x}_k^{(N)})$ and $\mathbf{A}_k^{(N)}(\mathbf{x}_k^{(N)})$, and of the functions $\mathbf{g}_k(\mathbf{x}_k^{(N)})$, $\mathbf{f}_k^{(L)}(\mathbf{x}_k^{(N)})$ and $\mathbf{f}_k^{(N)}(\mathbf{x}_k^{(N)})$, respectively. Moreover, similarly as [40], it is assumed that the computation of the inverse of any covariance matrix involves a Cholesky decomposition of the matrix itself and the inversion of a lower or upper triangular matrix.

1. **Measurement update nonlinear part**

The overall computational cost of this task is

$$\mathcal{C}_{MU}^{(N)} = N_p \left(\mathcal{C}_{\eta_{1,k,j}^{(N)}} + \mathcal{C}_{\mathbf{C}_{1,k,j}^{(N)}} + \mathcal{C}_{w_{fe,k,j}} \right) + \mathcal{C}_{W_{fe,k,j}} + \mathcal{C}_R(N_p). \tag{E.16}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\eta_{1,k,j}^{(N)}}$ is equal to $\mathcal{C}_{\mathbf{B}} + \mathcal{C}_{\mathbf{g}} + 2PD_L$ flops; 2) $\mathcal{C}_{C_{1,k,j}^{(N)}}$ is equal to $2PD_L^2 + 2P^2D_L - PD_L$ flops ($\mathcal{C}_{\mathbf{B}}$ has been already accounted for at point 1)); 3) $\mathcal{C}_{w_{fe,k,j}}$ is equal to $(4P^3 + 21P^2 + 17P + 6)/6$ flops; 4) $\mathcal{C}_{W_{fe,k,j}}$ is equal to $2N_p - 1$ flops; 5) $\mathcal{C}_R(N_p)$ denotes the total cost of the resampling step (that involves a particle set of size N_p).

2. **First measurement update linear part**

The overall computational cost of this task is

$$\mathcal{C}_{MU1}^{(L)} = N_p \left(\mathcal{C}_{\mathbf{w}_{1,k,j}^{(L)}} + \mathcal{C}_{\mathbf{W}_{1,k,j}^{(L)}} + \mathcal{C}_{\mathbf{C}_{2,k,j}^{(L)}} + \mathcal{C}_{\eta_{2,k,j}^{(L)}} \right). \tag{E.17}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\mathbf{w}_{1,k,j}^{(L)}}$ is equal to $\mathcal{C}_{\mathbf{B}}+\mathcal{C}_{\mathbf{g}}+2P^2D_L+2PD_L-PD_L-D_L+P$ flops; 2) $\mathcal{C}_{\mathbf{W}_{1,k,j}^{(L)}}$ is equal to $2PD_L^2+2P^2D_L-D_L^2-PD_L$ flops; 3) $\mathcal{C}_{\mathbf{C}_{2,k,j}^{(L)}}$ is equal to $4D_L^3/3+4D_L^2+5D_L/3$ flops; 4) $\mathcal{C}_{\eta_{2,k,j}^{(L)}}$ is equal to $D_L(4D_L - 1)$ flops.

3. **Second measurement update linear part**

The overall computational cost of this task is

$$\mathcal{C}_{MU2}^{(L)} = N_p \left(\mathcal{C}_{\mathbf{z}_{k,j}^{(L)}} + \mathcal{C}_{\mathbf{C}_{4,k,j}^{(L)}} + \mathcal{C}_{\eta_{4,k,j}^{(L)}} \right). \tag{E.18}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\mathbf{z}_{k,j}^{(L)}}$ is equal to $\mathcal{C}_{\mathbf{f}^{(N)}} + D_N$ flops; 2) $\mathcal{C}_{\mathbf{C}_{4,k,j}^{(L)}}$ is equal to $\mathcal{C}_{\mathbf{A}^{(N)}} + 2D_L^3/3 + 2D_L^2D_N + 2D_LD_N^2 + 3D_L^2/2 - D_LD_N + 5D_L/6$ flops; 3) $\mathcal{C}_{\eta_{4,k,j}^{(L)}}$ is equal to $2D_LD_N^2 + 2D_L^2 + D_LD_N - 2D_L + D_N$ flops.

4. **Time update nonlinear part**

The overall computational cost of this task is

$$\mathcal{C}_{TU}^{(N)} = N_p \left(\mathcal{C}_{\eta_{3,k,j}^{(N)}} + \mathcal{C}_{\mathbf{C}_{3,k,j}^{(N)}} + \mathcal{C}_{\mathbf{x}_{\mathrm{fp},k+1,j}^{(N)}} \right). \tag{E.19}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\eta_{3,k,j}^{(N)}}$ is equal to $\mathcal{C}_{\mathbf{A}^{(N)}} + \mathcal{C}_{\mathbf{f}^{(N)}} + 2D_LD_N$ flops; 2) $\mathcal{C}_{\mathbf{C}_{3,k,j}^{(N)}}$ is equal to $D_LD_N(2D - 1)$ flops ($\mathcal{C}_{\mathbf{A}^{(N)}}$ has been already accounted for at point 1)); 3) $\mathcal{C}_{\mathbf{x}_{\mathrm{fp},k+1,j}^{(N)}}$ is equal to $D_N^3/3 + 3D_N^2 + 5D_N/3$ flops.

5. **Time update linear part**

The overall computational cost of this task is

$$\mathcal{C}_{TU}^{(L)} = N_p \left(\mathcal{C}_{\eta_{\mathrm{fp},k+1,j}^{(L)}} + \mathcal{C}_{\mathbf{C}_{\mathrm{fp},k+1,j}^{(L)}} \right). \tag{E.20}$$

where the costs $\mathcal{C}_{\eta_{\mathrm{fp},k+1}}$ and $\mathcal{C}_{\mathbf{C}_{\mathrm{fp},k+1,j}^{(L)}}$ are equal to $\mathcal{C}_{\mathbf{A}^{(L)}} + \mathcal{C}_{\mathbf{f}^{(L)}} + 2D_L^2$ flops and $D_L^2(4D_L - 1)$ flops, respectively.

E.4 Computational complexity of the MPF technique developed in ref. [37]

In this appendix a detailed analysis of the MPF complexity is illustrated. The notation is the same as [37]. In the following, $\mathcal{C}_{f_x}$ and $\mathcal{C}_{f_y}$ denote the cost due to the evaluation of the functions $f_x(x_{t-1}, u_t)$ and $f_y(x_t, v_t)$, respectively. Moreover, similarly as [40], it is assumed that the computation of the inverse of any covariance matrix involves a Cholesky decomposition of the matrix itself and the inversion of a lower or upper triangular matrix.

1. **Measurement update**

The overall computational cost of this task is

$$\mathcal{C}_{MU} = n\left(\mathcal{C}_{x_{i,t-1}^{(m)}} + \mathcal{C}_{w_{i,t}^{(m)}} + \mathcal{C}_{x_{i,t}^{(m)}} + \mathcal{C}_R(M)\right). \tag{E.21}$$

Moreover, we have that: 1) $\mathcal{C}_{x_{i,t-1}^{(m)}}$ is equal $ML(N_f - 1)$ flops; 2) $\mathcal{C}_{w_{i,t}^{(m)}}$ is equal to $ML(6\mathcal{C}_{f_y} + 4d_y^3 + 21d_y^2 + 17d_y)/6 + 2ML + 2M - 1$ flops; 3) $\mathcal{C}_{x_{i,t}^{(m)}}$ is equal to $d_{x,i}(2M-1)$ flops; 4) $\mathcal{C}_R(M)$ denotes the total cost of the resampling step (that involves a particle set of size M).

2. **Time update**

The overall computational cost of this task is

$$\mathcal{C}_{TU} = n\mathcal{C}_{x_{i,t}^{(m)}}, \tag{E.22}$$

where the cost $\mathcal{C}_{x_{i,t}^{(m)}}$ is equal to $M(3\mathcal{C}_{f_x} + d_{x,i}^3 + 9d_{x,i}^2 + 5d_{x,i})/3$ flops.

E.5 Computational complexity of the DBSA and DDBSA

In this appendix, the computational complexity of the tasks accomplished *in a single recursion* of backward filtering and smoothing of the DBSA is assessed in terms of flops. Moreover, we comment on how the illustrated results can be also exploited to assess the computational complexity of a single recursion of the DDBSA. In the following, $\mathcal{C}_{\mathbf{H}}$, $\mathcal{C}_{\mathbf{B}}$, $\mathcal{C}_{\mathbf{F}}$, $\mathcal{C}_{\mathbf{A}^{(L)}}$ and $\mathcal{C}_{\mathbf{A}^{(N)}}$, and $\mathcal{C}_{\mathbf{g}}$, $\mathcal{C}_{\mathbf{f}^{(L)}}$, $\mathcal{C}_{\mathbf{f}^{(N)}}$ and $\mathcal{C}_{\mathbf{f}_k}$ denote the cost due to the evaluation of the matrices $\mathbf{H}_k$, $\mathbf{B}_k$, $\mathbf{F}_k$, $\mathbf{A}_k^{(L)}(\mathbf{x}_k^{(N)})$ and $\mathbf{A}_k^{(N)}(\mathbf{x}_k^{(N)})$, and of the functions $\mathbf{g}_k(\mathbf{x}_k^{(N)})$, $\mathbf{f}_k^{(L)}(\mathbf{x}_k^{(N)})$, $\mathbf{f}_k^{(N)}(\mathbf{x}_k^{(N)})$ and $\mathbf{f}_k(\mathbf{x}_k)$, respectively. Moreover, similarly as [40], it is assumed that the computation of the inverse of any covariance matrix involves a Cholesky decomposition of the matrix itself and the inversion of a lower or upper triangular matrix. Finally, it is assumed that the computation of the determinant of any matrix involves a Cholesky decomposition of the matrix itself and the product of the diagonal entries of a triangular matrix.

Phase I - The overall computational cost of this task is evaluated as (see Eqs. (3.39)–(3.40) and (3.47)–(3.49))

$$\begin{aligned}\mathcal{C}_1 = \mathcal{C}_{\mathbf{W}_{1,k}} + \mathcal{C}_{\mathbf{w}_{1,k}} + N_p\big(&\mathcal{C}_{\mathbf{z}_{k,j}^{(L)}} + \mathcal{C}_{\tilde{\mathbf{W}}_{k,j}} \\ &+\mathcal{C}_{\tilde{\mathbf{w}}_{k,j}} + \mathcal{C}_{\tilde{\mathbf{C}}_{k,j}} + \mathcal{C}_{\tilde{\eta}_{k,j}}\big) \triangleq \mathcal{C}_{\text{bp}}^{(1)}.\end{aligned} \tag{E.23}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\mathbf{W}_{1,k}}$ is equal to $\mathcal{C}_{\mathbf{F}} + 26D^3/3 - D^2/2 + 5D/6$ flops; 2) the cost $\mathcal{C}_{\mathbf{w}_{1,k}}$ is equal to $4D^3 + 4D^2 - 2D$ flops (the cost for computing $\mathcal{C}_{\mathbf{F}}$ has been already accounted for at point 1)); 3) the cost $\mathcal{C}_{\mathbf{z}_{k,j}^{(L)}}$ is equal to $\mathcal{C}_{\mathbf{f}^{(N)}} + D_N$ flops; 4) the cost $\mathcal{C}_{\tilde{\mathbf{W}}_{k,j}}$ is equal to $\mathcal{C}_{\mathbf{A}^{(N)}} + 4D_N^3 - 2D_N^2$ flops; 5) the cost $\mathcal{C}_{\tilde{\mathbf{w}}_{k,j}}$ is equal to $2D_N^3 + D_N^2 - D_N$ flops (the cost for computing $\mathcal{C}_{\mathbf{A}^{(N)}}$ has been already accounted for at point 4)); 6) the cost $\mathcal{C}_{\tilde{\mathbf{C}}_{k,j}}$ is equal to $2D_N^3/3 + 3D_N^2/2 + 5D_N/6$ flops; 7) the cost $\mathcal{C}_{\tilde{\eta}_{k,j}}$ is equal to $2D_N^2 - D_N$ flops. The expressions listed at points 1)-2) can be exploited for the DDBSA too; in the last case, however, $D_N = 0$ and $D = D_L$ must be assumed.

Phase II - The overall computational cost of this task is evaluated as

$$\mathcal{C}_2 = n_i\big(\mathcal{C}_{\mathrm{pm}^{(1)}} + \mathcal{C}_{\mathrm{be1}^{(1)}} + \mathcal{C}_{\mathrm{sm}^{(1)}} + \mathcal{C}_{\mathrm{bp}^{(2)}} + \mathcal{C}_{\mathrm{pm}^{(2)}} + \mathcal{C}_{\mathrm{ms}^{(2)}} + \mathcal{C}_{\mathrm{be2}^{(2)}} + \mathcal{C}_{\mathrm{sm}^{(2)}}\big). \tag{E.24}$$

The terms appearing in the RHS of the last equation can be computed as follows. First of all, we have that

$$\mathcal{C}_{\mathrm{pm}^{(1)}} = \mathcal{C}_{\eta_{2,k}^{(n)}} + \mathcal{C}_{\mathbf{C}_{2,k}^{(n)}}, \tag{E.25}$$

where (see Eqs. (3.43)-(3.44)): 1) the cost $\mathcal{C}_{\eta_{2,k}^{(n)}}$ is equal to $2N_pD - D$ flops; 2) the cost $\mathcal{C}_{\mathbf{C}_{2,k}^{(n)}}$ is equal to $5N_pD_L^2 + 4N_pD_N^2 + 4N_pD_LD_N + D_L^2 + D_N^2 + D_LD_N$ flops. The expressions listed at points 1)-2) can be exploited for the DDBSA too; in the last case, however, $D_N = 0$ and $D = D_L$ must be assumed.

The second term appearing in the RHS of Eq. (E.24) is evaluated as

$$\mathcal{C}_{\mathrm{be1}^{(1)}} = \mathcal{C}_{\mathbf{C}_{3,k}^{(n)}} + \mathcal{C}_{\eta_{3,k}^{(n)}} + \mathcal{C}_{\mathbf{W}_{3,k}^{(n)}} + \mathcal{C}_{\mathbf{w}_{3,k}^{(n)}}, \tag{E.26}$$

where (see Eqs. (3.55)-(3.56)): 1) the cost $\mathcal{C}_{\mathbf{C}_{3,k}^{(n)}}$ is equal to $14D^3/3 + D^2/2 + 5D/6$ flops; 2) the cost $\mathcal{C}_{\eta_{3,k}^{(n)}}$ is equal to $4D^2 - D$ flops (the cost for computing $\mathcal{C}_{\mathbf{W}_k^{(n)}}$ has been already accounted for at point 1)); 3) the cost $\mathcal{C}_{\mathbf{W}_{3,k}^{(n)}}$ is equal to $2D^3/3 + 3D^2/2 + 5D/6$ flops; 4) the cost $\mathcal{C}_{\mathbf{w}_{3,k}^{(n)}}$ is equal to $2D^2 - D$ flops. The expressions listed at points 1)-4) can be exploited for the DDBSA too; in the last case, however, $D_N = 0$ and $D = D_L$ must be assumed.

The third term appearing in the RHS of Eq. (E.24) is computed as

$$\mathcal{C}_{\mathrm{sm}^{(1)}} = \mathcal{C}_{\mathbf{W}_{4,k}^{(n)}} + \mathcal{C}_{\mathbf{w}_{4,k}^{(n)}} + \mathcal{C}_{\mathbf{C}_{4,k}^{(n)}} + \mathcal{C}_{\eta_{4,k}^{(n)}}, \tag{E.27}$$

where (see Eqs. (3.59)-(3.60)): 1) the cost $\mathcal{C}_{\mathbf{W}_{4,k}^{(n)}}$ is equal to D^2 flops; 2) the cost $\mathcal{C}_{\mathbf{w}_{4,k}^{(n)}}$ is equal to D flops; 3) the cost $\mathcal{C}_{\mathbf{C}_{4,k}^{(n)}}$ is equal to $2D^3/3 + 3D^2/2 + 5D/6$ flops; 4) the cost $\mathcal{C}_{\eta_{4,k}^{(n)}}$ is equal to $2D^2 - D$ flops. The expressions listed at points 1)-4) can be exploited for the DDBSA too; in the last case, however, $D_N = 0$ and $D = D_L$ must be assumed.

The fourth term appearing in the RHS of Eq. (E.24) is given by

$$\mathcal{C}_{\mathrm{bp}^{(2)}} = N_p\big(\mathcal{C}_{\eta_{3,k,j}^{(N)}} + \mathcal{C}_{\mathbf{C}_{3,k,j}^{(N)}} + \mathcal{C}_{D_{3,k,j}^{(n)}} + \mathcal{C}_{Z_{3,k,j}^{(n)}}\big), \tag{E.28}$$

where (see Eqs. (3.66)-(3.67) and (3.64)-(3.65)): 1) the cost $\mathcal{C}_{\eta_{3,k,j}^{(N)}}$ is equal to $\mathcal{C}_{\mathbf{A}^{(N)}} + \mathcal{C}_{\mathbf{f}^{(N)}} + 2D_LD_N$ flops; 2) the cost $\mathcal{C}_{\mathbf{C}_{3,k,j}^{(N)}}$ is equal to $2D_L^2D_N + 2D_LD_N^2 - D_LD_N$ flops (the cost for computing $\mathcal{C}_{\mathbf{A}^{(N)}}$ and $\mathcal{C}_{\mathbf{f}^{(N)}}$ has been already accounted for at point 1)); 3) the cost $\mathcal{C}_{D_{3,k,j}^{(n)}}$ is equal to $D_N^3/3 + D_N^2 + 5D_N/3 + 2$ flops; 4) the cost $\mathcal{C}_{Z_{3,k,j}^{(n)}}$ is equal to $2D_N^2 + 2D_N - 1$ flops.

The fifth term appearing in the RHS of Eq. (E.24) is evaluated as

$$\mathcal{C}_{\mathrm{pm}^{(2)}} = N_p\big(\mathcal{C}_{\check{\eta}_{z,k,j}^{(n)}} + \mathcal{C}_{\check{\mathbf{C}}_{z,k,j}^{(n)}} + \mathcal{C}_{\check{\mathbf{W}}_{2,k,j}^{(n)}} + \mathcal{C}_{\check{\mathbf{w}}_{2,k,j}^{(n)}} + \mathcal{C}_{D_{2,k,j}^{(n)}} + \mathcal{C}_{Z_{2,k,j}^{(n)}}\big), \tag{E.29}$$

where (see Eqs. (3.69)-(3.74)): 1) the cost $\mathcal{C}_{\check{\eta}_{z,k,j}^{(n)}}$ is equal to $\mathcal{C}_{\mathbf{A}^{(L)}} + 2D_L^2$ flops; 2) the cost $\mathcal{C}_{\check{\mathbf{C}}_{z,k,j}^{(n)}}$ is equal to $4D_L^3 - D_L^2$ flops (the cost for computing $\mathcal{C}_{\mathbf{A}^{(L)}}$ has been already accounted for at point 1)); 3) the cost $\mathcal{C}_{\check{\mathbf{W}}_{2,k,j}^{(n)}}$ is equal to $2D_L^3/3 + 5D_L^2/2 + 5D_L/6$ flops; 4) the cost $\mathcal{C}_{\check{\mathbf{w}}_{2,k,j}^{(n)}}$ is equal to $\mathcal{C}_{\mathbf{f}^{(L)}} + 4D_L^2 - D_L$ flops; 5) the cost $\mathcal{C}_{D_{2,k,j}^{(n)}}$ is equal to $D_L^3/3 + 2D_L^2 + 5D_L/3 + 2$ flops; 6) the cost $\mathcal{C}_{Z_{2,k,j}^{(n)}}$ is equal to $6D_L^2 + 3D_L - 1$ flops.

The sixth term appearing in the RHS of Eq. (E.24) is computed as

$$\mathcal{C}_{\mathrm{ms}^{(2)}} = N_p\big(\mathcal{C}_{\bar{\eta}_{5,k,j}^{(n)}} + \mathcal{C}_{\bar{\mathbf{C}}_{5,k,j}^{(n)}} + \mathcal{C}_{D_{5,k,j}^{(n)}} + \mathcal{C}_{Z_{5,k,j}^{(n)}}\big), \tag{E.30}$$

where (see Eqs. (3.83)-(3.86)): 1) the cost $\mathcal{C}_{\bar{\eta}_{5,k,j}^{(n)}}$ is equal to $\mathcal{C}_{\mathbf{B}} + \mathcal{C}_{\mathbf{g}} + 2PD_L$ flops; 2) the cost $\mathcal{C}_{\bar{\mathbf{C}}_{5,k,j}^{(n)}}$ is equal to $2PD_L^2 + 2P^2D_L - PD_L$ flops (the cost for computing $\mathcal{C}_{\mathbf{B}}$ has been already accounted for at point 1)); 3) the cost $\mathcal{C}_{D_{5,k,j}^{(n)}}$ is equal to $D_L^3/3 + D_L^2 + 5D_L/3 + 2$ flops; 4) the cost $\mathcal{C}_{Z_{5,k,j}^{(n)}}$ is equal to $2P^3/3 + 7P^2/2 + 17P/6 - 1$ flops. It is important to note that, if the forward weights $\{w_{\mathrm{fe},k,j}\}$ are reused, the cost $\mathcal{C}_{\mathrm{ms}^{(2)}}$ appearing in Eq. (E.30) is equal to zero.

The seventh term appearing in the RHS of Eq. (E.24) is given by

$$\mathcal{C}_{\mathrm{be2}^{(2)}} = N_p\big(\mathcal{C}_{D_{6,k,j}^{(n)}} + \mathcal{C}_{Z_{6,k,j}^{(n)}} + \mathcal{C}_{w_{6,k,j}^{(n)}}\big), \tag{E.31}$$

where the costs $\mathcal{C}_{D_{6,k,j}^{(n)}}$ and $\mathcal{C}_{Z_{6,k,j}^{(n)}}$ are equal to 2 flops, and the cost $\mathcal{C}_{w_{6,k,j}^{(n)}}$ is equal to 3 flops (see Eqs. (3.89)-(3.91)). If the forward weights $\{w_{\mathrm{fe},k,j}\}$ are reused, the costs $\mathcal{C}_{D_{6,k,j}^{(n)}}$ and $\mathcal{C}_{Z_{6,k,j}^{(n)}}$ are equal to 1 flops, whereas the cost $\mathcal{C}_{w_{6,k,j}^{(n)}}$ remains unchanged.

The last term appearing in the RHS of Eq. (E.24) is evaluated as

$$\mathcal{C}_{\mathrm{sm}^{(2)}} = \mathcal{C}_{w_{1,k,j}^{(n)}} + \mathcal{C}_{W_{1,k,j}^{(n)}}, \tag{E.32}$$

where the costs $\mathcal{C}_{w_{1,k,j}^{(n)}}$ and $\mathcal{C}_{W_{1,k,j}^{(n)}}$ are equal to N_p and $2N_p - 1$ flops, respectively (see Eqs. (3.95)-(3.96)).

Phase III - The overall computational cost of this task is evaluated as

$$\mathcal{C}_3 = \mathcal{C}_{\mathrm{be}^{(2)}} + \mathcal{C}_{\mathrm{pm}^{(1)}} + \mathcal{C}_{\mathrm{be1}^{(1)}} + \mathcal{C}_{\mathrm{be}^{(1)}}. \tag{E.33}$$

Here, the cost $\mathcal{C}_{\mathrm{be}^{(2)}}$ is equal to $\mathcal{C}_S(N_p)$, that represents the total cost of a sampling step that involves a particle set of size N_p; moreover, the costs $\mathcal{C}_{\mathrm{pm}^{(1)}}$ and $\mathcal{C}_{\mathrm{be1}^{(1)}}$ are the same as those appearing in the RHS of Eq. (E.24), and $\mathcal{C}_{\mathrm{be}^{(1)}}$ is computed as (see Eqs. (3.102)–(3.105))

$$\mathcal{C}_{\mathrm{be}^{(1)}} = \mathcal{C}_{\mathbf{W}_{\mathrm{ms},k}} + \mathcal{C}_{\mathbf{w}_{\mathrm{ms},k}} + \mathcal{C}_{\mathbf{W}_{\mathrm{be2},k}} + \mathcal{C}_{\mathbf{w}_{\mathrm{be2},k}} + \mathcal{C}_{\mathbf{C}_{\mathrm{be}}} + \mathcal{C}_{\eta_{\mathrm{be}}}. \tag{E.34}$$

Moreover, we have that: 1) the cost $\mathcal{C}_{\mathbf{W}_{\mathrm{ms},k}}$ is equal to $\mathcal{C}_{\mathbf{H}} + 2P^2D + 2PD^2 - D^2 - PD$ flops; 2) the cost $\mathcal{C}_{\mathbf{w}_{\mathrm{ms},k}}$ is equal to $\mathcal{C}_{\mathbf{B}} + \mathcal{C}_{\mathbf{g}} + 2P^2D + 3PD + 2PD_L - P - D$ flops (the cost for computing $\mathcal{C}_{\mathbf{H}}$ has been already accounted for at point 1)); 3) the cost $\mathcal{C}_{\mathbf{W}_{\mathrm{be2},k}}$ is equal to D^2 flops; 4) the cost $\mathcal{C}_{\mathbf{w}_{\mathrm{be2},k}}$ is equal to D flops; 5) the cost $\mathcal{C}_{\mathbf{C}_{\mathrm{be}}}$ is equal to $2D^3/3 + 3D^2/2 + 5D/6$ flops; 6) the cost $\mathcal{C}_{\eta_{\mathrm{be}}}$ is equal to $2D^2 - D$ flops. The expressions listed at points 1)-6) can be exploited for the DDBSA too; in the last case, however, $D_N = 0$ and $D = D_L$ must be assumed. Note that the costs $\mathcal{C}_{\mathbf{W}_{\mathrm{ms},k}}$ and $\mathcal{C}_{\mathbf{w}_{\mathrm{ms},k}}$ (see points 1) and 2)) are ignored if the precision matrix $\mathbf{W}_{\mathrm{ms},k}$ and the transformed mean vector $\mathbf{w}_{\mathrm{ms},k}$ are stored in the forward pass (so that they do not need to be recomputed in the backward pass). Moreover, if the SDBSA or the SDDBSA is used, the cost $\mathcal{C}_{\mathrm{be}^{(2)}}$ in the RHS of Eq. (E.33) becomes $D_N(2N_p - 1)$ flops.

Finally, it is worth stressing that, if the DBSA or the DDBSA (the SDBSA or the SDDBSA) is employed, the overall computational complexity is obtained by multiplying the computational cost assessed for a single recursion by MT (by T), where M and T denote the overall number of accomplished backward passes and the duration of the observation interval, respectively.

Bibliography

[1] G. M. Vitetta, E. Sirignano, P. Di Viesti, F. Montorsi, and M. Sola, "Marginalized Particle Filtering and Related Filtering Techniques as Message Passing," *IEEE Trans. Signal Process.*, vol. 67, no. 6, pp. 1522–1536, Mar. 2019.

[2] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb./2002.

[3] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, ser. Prentice-Hall Information and System Sciences Series. Englewood Cliffs, N.J: Prentice-Hall, 1979.

[4] A. Doucet, N. Freitas, and N. Gordon, "An Introduction to Sequential Monte Carlo Methods," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. Freitas, and N. Gordon, Eds. New York, NY: Springer New York, 2001, pp. 3–14.

[5] B. Paige, F. Wood, A. Doucet, and Y. W. Teh, "Asynchronous Anytime Sequential Monte Carlo," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3410–3418.

[6] T. Schon, F. Gustafsson, and P. Nordlund, "Marginalized Particle Filters for Mixed Linear/Nonlinear State-Space Models," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[7] F. Lindsten, P. Bunch, S. Särkkä, T. B. Schön, and S. J. Godsill, "Rao-Blackwellized Particle Smoothers for Conditionally Linear Gaussian Models," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 2, pp. 353–365, Mar. 2016.

[8] R. Chen and J. S. Liu, "Mixture Kalman Filters," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 62, no. 3, pp. 493–508, 2000.

[9] C. Andrieu and A. Doucet, "Particle Filtering for Partially Observed Gaussian State Space Models," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 64, no. 4, pp. 827–836, 2002.

[10] V. Smidl and A. Quinn, "Variational Bayesian Filtering," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 5020–5030, Oct. 2008.

[11] T. Lu, M. F. Bugallo, and P. M. Djuric, "Simplified Marginalized Particle Filtering for Tracking Multimodal Posteriors," in *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, Aug. 2007, pp. 269–273.

[12] C.-J. Kim, "Dynamic Linear Models with Markov-Switching," *Journal of Econometrics*, vol. 60, no. 1, pp. 1–22, Jan. 1994.

[13] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund, "Particle Filters for Positioning, Navigation, and Tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002.

[14] S. Särkkä, A. Solin, A. Nummenmaa, A. Vehtari, T. Auranen, S. Vanni, and F.-H. Lin, "Dynamic Retrospective Filtering of Physiological Noise in BOLD fMRI: DRIFTER," *NeuroImage*, vol. 60, no. 2, pp. 1517–1527, Apr. 2012.

[15] X. Xu and B. Li, "Learning Motion Correlation for Tracking Articulated Human Body with a Rao-Blackwellised Particle Filter," in *2007 IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8.

[16] C. Andrieu and S. J. Godsill, "A Particle Filter for Model Based Audio Source Separation," p. 7, 2000.

[17] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The Factor Graph Approach to Model-Based Signal Processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, Jun. 2007.

[18] F. R. Kschischang, B. J. Frey, and H. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[19] H.-A. Loeliger, "Least Squares and Kalman Filtering on Forney Graphs," in *Codes, Graphs, and Systems: A Celebration of the Life and Career of G. David Forney, Jr. on the Occasion of His Sixtieth Birthday*, ser. The Kluwer International Series in Engineering and Computer Science, R. E. Blahut and R. Koetter, Eds. Boston, MA: Springer US, 2002, pp. 113–135.

[20] M. Briers, A. Doucet, S. S. Singh, and K. Weekes, "Particle Filters for Graphical Models," in *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, Sep. 2006, pp. 59–64.

[21] J. Dauwels, S. Korl, and H. Loeliger, "Particle Methods as Message Passing," in *2006 IEEE International Symposium on Information Theory*, Jul. 2006, pp. 2052–2056.

[22] G. M. Vitetta, E. Sirignano, and F. Montorsi, "A Novel Message Passing Algorithm for Online Bayesian Filtering: Turbo Filtering," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 645–651.

[23] ——, "Particle Smoothing for Conditionally Linear Gaussian Models as Message Passing Over Factor Graphs," *IEEE Trans. Signal Process.*, vol. 66, no. 14, pp. 3633–3648, Jul. 2018.

[24] R. Hostettler and S. Särkkä, "Rao-Blackwellized Gaussian Smoothing," *IEEE Trans. Autom. Control*, vol. 64, no. 1, pp. 305–312, Jan. 2019.

[25] G. M. Vitetta, P. Di Viesti, E. Sirignano, and F. Montorsi, "Parallel Concatenation of Bayesian Filters: Turbo Filtering," *ArXiv180604632 Stat*, Jun. 2018.

[26] F. R. Kschischang and B. J. Frey, "Iterative Decoding of Compound Codes by Probability Propagation in Graphical Models," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 219–230, Feb. 1998.

[27] A. P. Worthen and W. E. Stark, "Unified Design of Iterative Receivers Using Factor Graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 843–849, Feb. 2001.

[28] T. Li, M. Bolic, and P. M. Djuric, "Resampling Methods for Particle Filtering: Classification, Implementation, and Strategies," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015.

[29] R. Karlsson, T. Schon, and F. Gustafsson, "Complexity Analysis of the Marginalized Particle Filter," *IEEE Trans. Signal Process.*, vol. 53, no. 11, pp. 4408–4411, Nov. 2005.

[30] F. Mustiere, M. Bolic, and M. Bouchard, "A Modified Rao-Blackwellised Particle Filter," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 3, May 2006, pp. III–III.

[31] A. R. Runnalls, "Kullback-Leibler Approach to Gaussian Mixture Reduction," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 989–999, Jul. 2007.

[32] G. M. Vitetta, P. Di Viesti, E. Sirignano, and F. Montorsi, "Multiple Bayesian Filtering as Message Passing," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1002–1020, 2020.

[33] S. Mazuelas, Y. Shen, and M. Z. Win, "Belief Condensation Filtering," *IEEE Trans. Signal Process.*, vol. 61, no. 18, pp. 4403–4415, Sep. 2013.

[34] J. Prieto, S. Mazuelas, and M. Z. Win, "Context-Aided Inertial Navigation via Belief Condensation," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3250–3261, June 2016.

[35] F. Daum and J. Huang, "Curse of Dimensionality and Particle Filters," in *2003 IEEE Aerospace Conference Proceedings (Cat. No.03TH8652)*, vol. 4, Mar. 2003, pp. 4–1979–4–1993.

[36] T. Bengtsson, P. Bickel, and B. Li, "Curse-of-Dimensionality Revisited: Collapse of the Particle Filter in Very Large Scale Systems," in *Institute of Mathematical Statistics Collections*. Beachwood, Ohio, USA: Institute of Mathematical Statistics, 2008, pp. 316–334.

[37] P. M. Djurić and M. F. Bugallo, "Particle Filtering for High-Dimensional Systems," in *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2013, pp. 352–355.

[38] B. Krach and P. Roberston, "Cascaded Estimation Architecture for Integration of Foot-Mounted Inertial Sensors," in *2008 IEEE/ION Position, Location and Navigation Symposium*, May 2008, pp. 112–119.

[39] P. Closas and M. F. Bugallo, "Improving Accuracy by Iterated Multiple Particle Filtering," *IEEE Signal Process. Lett.*, vol. 19, no. 8, pp. 531–534, Aug. 2012.

[40] B. Ait-El-Fquih and I. Hoteit, "A Variational Bayesian Multiple Particle Filtering Scheme for Large-Dimensional Systems," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5409–5422, Oct. 2016.

[41] P. M. Djuric, T. Lu, and M. F. Bugallo, "Multiple Particle Filtering," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 3, Apr. 2007, pp. III–1181–III–1184.

[42] M. F. Bugallo and P. M. Djuric, "Gaussian Particle Filtering in High-Dimensional Systems," in *2014 IEEE Workshop on Statistical Signal Processing (SSP)*. Gold Coast, Australia: IEEE, Jun. 2014, pp. 129–132.

[43] I. Urteaga, M. F. Bugallo, and P. M. Djurić, "Multiple Particle Filtering for Inference in the Presence of State Correlation of Unknown Mixing Parameters," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 3849–3853.

[44] P. Chavali and A. Nehorai, "Multiple Rao-Blackwellized Particle Filtering for Target Tracking in Urban Environments," in *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2011, pp. 409–412.

[45] ——, "Scheduling and Power Allocation in a Cognitive Radar Network for Multiple-Target Tracking," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 715–729, Feb. 2012.

[46] F. Montorsi, "Localization and Tracking for Indoor Environments," Ph.D. dissertation, 2013.

[47] P. Di Viesti, G. M. Vitetta, and E. Sirignano, "Double bayesian smoothing as message passing," *IEEE Transactions on Signal Processing*, vol. 67, no. 21, p. 5495–5510, Nov 2019.

[48] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge, United Kingdom: Cambridge University Press, 2013.

[49] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, Jul. 2000.

[50] G. Kitagawa, "Non-Gaussian State-Space Modeling of Nonstationary Time Series," *J. Am. Stat. Assoc.*, vol. 82, no. 400, pp. 1032–1041, 1987.

[51] ——, "The Two-Filter Formula for Smoothing and an Implementation of the Gaussian-Sum Smoother," *Ann Inst Stat Math*, vol. 46, no. 4, pp. 605–623, Dec. 1994.

[52] Y. Bresler, "Two-Filter Formulae for Discrete-Time Non-Linear Bayesian Smoothing," *Int. J. Control*, vol. 43, no. 2, pp. 629–641, Feb. 1986.

[53] B. Vo, B. Vo, and R. P. S. Mahler, "Closed-Form Solutions to Forward-Backward Smoothing," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 2–17, Jan. 2012.

[54] S. Sarkka and J. Hartikainen, "On Gaussian Optimal Smoothing of Non-Linear State Space Models," *IEEE Trans. Autom. Control*, vol. 55, no. 8, pp. 1938–1941, Aug. 2010.

[55] J. Kokkala, A. Solin, and S. Särkkä, "Sigma-Point Filtering and Smoothing Based Parameter Estimation in Nonlinear Dynamic Systems," *J. Adv. Inf. Fusion*, vol. 11, no. 1, pp. 15–30, 2016.

[56] Á. F. García-Fernández, L. Svensson, and S. Särkkä, "Iterated Posterior Linearization Smoother," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 2056–2063, Apr. 2017.

[57] R. Douc, A. Garivier, E. Moulines, and J. Olsson, "Sequential Monte Carlo Smoothing for General State Space Hidden Markov Models," *Ann. Appl. Probab.*, vol. 21, no. 6, pp. 2109–2145, Dec. 2011.

[58] G. Kitagawa, "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, Mar. 1996.

[59] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo Smoothing for Nonlinear Time Series," *J. Am. Stat. Assoc.*, vol. 99, no. 465, pp. 156–168, Mar. 2004.

[60] F. Lindsten, "Backward Simulation Methods for Monte Carlo Statistical Inference," *Found. Trends. Mach. Learn.*, vol. 6, no. 1, pp. 1–143, 2013.

[61] M. Briers, A. Doucet, and S. Maskell, "Smoothing Algorithms for State-Space Models," *Ann Inst Stat Math*, vol. 62, no. 1, p. 61, Jun. 2009.

[62] W. Fong, S. J. Godsill, A. Doucet, and M. West, "Monte Carlo Smoothing with Application to Audio Signal Enhancement," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 438–449, Feb. 2002.

[63] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. New York, NY: Springer, 2006.

Part II

Radar Imaging

Four

Novel Methods for Approximate Maximum Likelihood Estimation of Multiple Superimposed Undamped Tones

In this chapter, novel methods for the detection of multiple superimposed tones in noise and the estimation of their parameters are derived. These methods are based on a maximum likelihood approach and combine an innovative single tone estimator with a serial cancellation procedure. Our numerical results lead to the conclusion that these methods can achieve a substantially better accuracy-complexity tradeoff than various related techniques in the presence of multiple closely spaced tones.

Part of this Chapter has been published in [1] and is patent pending.

4.1 Introduction

The problem of estimating the amplitude, phase and frequency of multiple (say, L) tones in *additive white Gaussian noise* (AWGN) has received significant attention for a number of years because of its relevance in various fields, including radar systems [2] and wireless communications; in particular, in the last field, it plays a fundamental role in frequency synchronization [3], and in the estimation of mobile channels [4–6] and of the *direction of arrival* (DOA) in massive *multiple-input multiple-output* (MIMO) systems employing uniform linear arrays (ULAs) in their base stations [7].

It is well known that the *maximum-likelihood* (ML) approach to this problem leads to a complicated nonlinear optimization problem. Substantial simplifications can be made when the L tone frequencies are sufficiently well separated and the number N of available signal samples is large enough [8–10]. In fact, under these assumptions, each tone has a limited influence on the estimation of the others, so that approximate ML estimation can be achieved through a conceptually simple sequential procedure, that consists in iteratively executing two steps [9]. In the first step of this procedure, the parameters of the dominant tone (i.e., of the tone associated with the largest peak in the periodogram of the observed signal) are estimated in a ML fashion. In its second step, instead, the estimated tone is subtracted from the available signal samples and a new periodogram is computed for the resulting residual. These steps are repeated until all the detectable tones are estimated. The technical relevance of this procedure is motivated by the following relevant advantages [11]: 1. It turns a complicated multidimensional problem (whose dimensionality is usually unknown a priori) into a sequence of lower dimensional subproblems. Consequently, its overall complexity is proportional to that required to solve each of such subproblems and is usually much lower than that of *parametric estimation methods* (e.g., the MUSIC [12] and the ESPRIT [13]) and *non parametric spectral estimators* (e.g., the Capon method [14], the APES [15] and the IAA-APES [16]). 2. It performs better than independently estimating the tones associated with the largest peaks of the original periodogram. In fact, it allows to identify peaks that are initially masked by the leakage due to nearby stronger tones. 3. It is able to estimate an unknown L in a simple fashion. In fact, this result can be achieved setting the initial value of this parameter to zero and applying a suitable test to establish whether, at

each repetition of its first step, the largest peak detected in the periodogram of the last residual is significant [8] or whether, at each repetition of its second step, the energy of the new residual is large enough [17]. If one of these conditions is satisfied, the estimate of L is incremented by one and the next step is carried out; otherwise, the estimation process is terminated. It is worth stressing that various estimation methods (e.g., the MUSIC and the ESPRIT) require prior knowledge of L and that, in these cases, the use of some methods, like the generalized *Akaike information criterion* [18] or the *minimum description length* [19] is commonly proposed for the estimation of this parameter; however, the computational effort they require is not negligible.

The two-step procedure described above, despite its advantages, suffers from the following two shortcomings: 1. Any inaccuracy in the estimation of each single tone accomplished in its first step results in an imperfect cancellation of the tone itself and, consequently, in *error accumulation*; the intensity of this phenomenon increases with iterations, so affecting the estimation accuracy of the weakest tones. 2. The estimate of each tone is potentially *biased*, because of the presence of other tones [10]. Biases are influenced by the relative phase, frequency and amplitude of the superimposed tones and are expected to be more relevant in the first estimated frequencies, since these suffer from stronger interference from other tones. For this reason, the overall accuracy of this procedure depends on that of the employed single tone estimator and can be improved by adopting specific methods for mitigating the estimation bias. As far as the first issue is concerned, it is important to point out that optimal (i.e., ML) estimation of a single tone in AWGN is a computationally hard task. This is mainly due to the fact that the ML metric is an highly nonlinear function, that does not lend itself to easy maximisation (e.g., see [20]). In practice, the most accurate ML-based single tone estimators available in the technical literature achieve approximate maximisation of this metric through a two-step procedure; the first step consists in a *coarse search* of tone frequency, whereas the second one in a *fine estimation* generating an estimate of the so called *frequency residual* (i.e., of the difference between the real frequency and its coarse estimate). Coarse estimation is always based on the maximization of the periodogram of the observed signal, whereas fine estimation can be accomplished in an open loop fashion or through an iterative procedure. On the one hand, all the open loop estimators exploit spectral *interpolation* to infer the frequency residual from the analysis of the *fast Fourier transform* (FFT) coefficients at the maxima of the associated periodogram and at frequencies adjacent to it [11, 21–30]. Unfortunately, unlike iterative estimators, the accuracy they achieve is frequency dependent and gets smaller when the signal frequency approaches the center of one of the FFT bins. On the other hand, the iterative estimation techniques available in the technical literature are based on various methods, namely on: a) standard numerical methods for locating the global maximum of a function (e.g., the secan method [31] or the Newton's method [32]); b) an iterative method for binary search, known as the *dichotomous search of the periodogram peak* [33, 34]; c) interpolation methods amenable to iterative implementation [35–40]; d) the combination of the above mentioned dichotomous search with various interpolation techniques [41]; e) the computation of the first derivative of the spectrum [42].

The use of some of these algorithms in multiple tone estimators based on the above mentioned serial cancellation approach has been investigated in refs. [11, 17, 18] and [43–46]. More specifically, on the one hand, the periodogram-based (coarse) estimation method has been employed in the CLEAN algorithm [47–49], in the *more* CLEAN (MCLEAN) [17] and in the RELAX algorithm [18]. Note that, since a fine estimation step is missing in all these algorithms, achieving high accuracy requires the use of zero padding and of a large FFT order. On the other hand, the exploitation of more refined single tone estimators has been investigated in refs. [11] and [43–45]. In particular, the use of open-loop interpolation methods exploiting three or five adjacent spectral coefficients (including the one associated with the coarse frequency estimate) has been studied in ref. [11], whereas that of the iterative methods developed in refs. [35] and [50] has been analysed in refs. [43], [44] and [45], respectively.

As far as the second technical issue (i.e., estimation bias) is concerned, it is worth mentioning that the most straightforward methods for bias mitigation rely on the use of a) zero-padding for

enhancing periodogram spectral resolution and b) window functions [10, 20, 51–54]; the price to be paid for these choices is an increase in the overall computational cost and in the variance of computed estimates, respectively. More refined methods are represented by interpolators with intrinsic leakage rejection [11] and nonlinear optimization methods. The last class of methods includes the *expectation maximization* (EM) algorithm [55], the *space-alternating generalized* EM (SAGE) algorithm [4, 56], the Newton's method [32, 57] and different optimization algorithms that employ cyclic cancellation procedures [17, 18, 45, 57]. In the last case, tone re-estimation is accomplished after removing the interference of both stronger and weaker tones as the iterations of the serial cancellation procedure evolve [17, 18] or after detecting and estimating the parameters of all tones [45]; the most refined version of the first method is described in ref. [18], where tone re-estimation is iterated after the estimation of each new tone, in order to generate excellent initial estimates for the next step (i.e., for the estimation of the next tone). Tone re-estimation reduces error accumulation and leads to convergence to the ML solution in the absence of noise if the frequency spacing of the detected tones is large enough; however, this result is achieved at the price of an increase of the overall computational cost and latency [17].

This chapter aims at providing various new results about the estimation of multiple superimposed tones. Its contribution is threefold. First, a novel ML-based iterative estimator of a single complex tone is developed. The derivation of this estimator is based on: a) expressing the dependence of the ML metric on the frequency 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 [58, 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 technical literature. Secondly, it is shown how serial cancellation in the frequency domain can be combined with our iterative estimator to detect multiple tones and estimate their parameters. Thirdly, the accuracy of our single and multiple estimators is assessed by extensive computer simulations. Our results lead to the conclusion that our estimators outperform all the other related estimators in terms of probability of convergence and accuracy in the presence of arbitrary frequency residuals.

The remaining part of this chapter is organised as follows. In Section 4.2, the employed signal model is defined and its relevance in a number of applications is briefly discussed. Section 4.3 is devoted to the derivation of our single tone and multiple tone estimation algorithms, to the assessment of their computational complexity, and to the analysis of their similarities and differences with related estimators available in the technical literature. In Section 4.4, the performance of our estimation algorithms is assessed and compared with that achieved by other estimators. Finally, some conclusions are offered in Section 4.5.

4.2 Signal Model

In this chapter, we focus on the problem of estimating all the parameters of the complex sequence

$$x_{c,n} = \sum_{l=0}^{L-1} A_l \exp\left(j2\pi n F_l\right) + w_{c,n}, \tag{4.1}$$

and its real counterpart

$$\begin{aligned} x_{r,n} &= \sum_{l=0}^{L-1} \Re\left\{A_l \exp\left(j2\pi n F_l\right)\right\} + w_{r,n} \\ &= \sum_{l=0}^{L-1} a_l \cos\left(2\pi n F_l + \psi_l\right) + w_{r,n}, \end{aligned} \tag{4.2}$$

with $n = 0, 1, ..., N-1$; here, A_l and $F_l \in [0, 1)$ denote the *complex amplitude* and the *normalised frequency*, respectively, of the l-th complex tone appearing in the *right-hand side* (RHS) of eq.

(4.1), $a_l \triangleq |A_l|$, $\psi_l \triangleq \angle(A_l)$, $w_{c,n}$ is the n-th sample of an *additive white Gaussian noise* (AWGN) sequence (whose elements have zero mean and variance $2\sigma^2$), $w_{r,n} \triangleq \Re\{w_{c,n}\}$, N is the overall number of samples, and $\Re\{x\}$ and $\angle(x)$ denote the real part and the phase, respectively, of the complex quantity x. It is useful to point out that the signal model (4.2) can be rewritten as

$$x_{r,n} = \sum_{l=0}^{L-1} \left[C_l \exp\left(j2\pi n F_l\right) + C_l^* \exp\left(-j2\pi n F_l\right)\right] + w_{r,n}, \tag{4.3}$$

where

$$C_l \triangleq \frac{1}{2} a_l \exp\left(j\psi_l\right) = \frac{1}{2} A_l \tag{4.4}$$

represents the *complex amplitude* of the real tone appearing in the RHS of eq. (4.2).

The signal model (4.1) appears in a number of problems concerning biomedical applications, wireless communications and radar systems. Our interest in the considered estimation problem has been motivated by its relevance in the last field and, in particular, in colocated MIMO FMCW radar systems operating at mmWave [1, 2]. In this work, we always refer to the radar system model illustrated in [59, Fig. 2] and assume that: a) this system is equipped with N_T *transmit* (TX) and N_R *receive* (RX) antenna elements placed at different positions of the same planar shield; b) it operates in *time division multiplexing* (TDM) mode. Then, if the frequency modulated waveform radiated by the radar transmitter is reflected by L *static point targets*, and its receiver extracts both the in-phase and quadrature components of the received signal, the n-th received signal sample acquired through the v-th *virtual antenna element* can be expressed as (e.g., see [60, Par. 4.6, eq. (4.27)] or [59, Par. II-C])

$$x_n^{(v)} = \sum_{l=0}^{L-1} A_l^{(v)} \exp\left(j2\pi n\, F_l^{(v)}\right) + w_n^{(v)} \tag{4.5}$$

$$= \sum_{l=0}^{L-1} a_l \exp\left(j\left(2\pi n\, F_l^{(v)} + \psi_l^{(v)}\right)\right) + w_n^{(v)}, \tag{4.6}$$

with $n = 0, 1, ..., N-1$ and $v = 0, 1, ..., N_{VR}-1$. Here,

$$A_l^{(v)} \triangleq a_l \exp\left(j\psi_l^{(v)}\right) \tag{4.7}$$

for any v and l, a_l is the amplitude[1] of the l-th useful component of the received signal,

$$F_l^{(v)} \triangleq f_l^{(v)} T_s \tag{4.8}$$

is the normalised version of the frequency

$$f_l^{(v)} \triangleq \mu\, \tau_l^{(v)}, \tag{4.9}$$

characterizing the l-th target detected on the v-th virtual RX antenna,

$$\tau_l^{(v)} \triangleq \frac{2}{c} \left[R_l + x_v \cos\left(\phi_l\right) \sin\left(\theta_l\right) + y_v \sin\left(\phi_l\right)\right] \tag{4.10}$$

is the delay of the echo generated by the l-th target and observed on the v-th virtual channel, x_v and y_v are the abscissa and the ordinate[2], respectively, of the v-th *virtual antenna element*, R_l, θ_l and ϕ_l denote the range, the azimuth and the elevation, respectively, of the l-th target,

$$\psi_l^{(v)} \cong 2\pi f_0 \tau_l^{(v)} \tag{4.11}$$

[1]This amplitude quantifies the *radar cross section* (RCS) of the l-th target. It depends on both the range and the reflectivity of this target and is assumed to be independent of the virtual antenna index (i.e., of v), for simplicity.

[2]A reference system lying on the physical antenna array is adopted in the evaluation of these coordinates.

is the phase of the complex gain $A_l^{(v)}$ (4.7), $w_n^{(v)}$ is a complex Gaussian random variable having zero mean and variance $2\sigma^2$ (assumed to be independent of v), N is the overall number of samples acquired over a chirp period and $N_{VR} \triangleq N_T \cdot N_R$ represents the overall number of available *virtual antennas* associated with the given physical array. Moreover, T_s is the sampling period adopted in the analog-to-digital conversion accomplished at the receive side,

$$\mu \triangleq B/T \tag{4.12}$$

is the *chirp rate* (i.e., the steepness of the generated frequency chirp), T and B are the *chirp interval* and the *bandwidth*, respectively, of the transmitted signal and c is the speed of light.

As it can be easily inferred from (4.5)–(4.9), in the considered radar system, the problem of target detection and range estimation on the v-th virtual channel is equivalent to the problem of detecting multiple overlapped complex tones in the presence of complex AWGN and estimating their frequencies. In fact, if the l-th tone is found at the frequency $\hat{f}_l^{(v)}$, the presence of a target at the range (see (4.9) and (4.10))

$$\hat{R}_l^{(v)} = \frac{c}{2\,\mu}\,\hat{f}_l^{(v)} \tag{4.13}$$

is detected. Information about the angular coordinates (namely, the azimuth and the elevation) of this target, instead, can be acquired through the estimation of the set of N_{VR} phases $\{\psi_l^{(v)}; v = 0, 1, ..., N_{VR} - 1\}$ observed over the available virtual antennas. In fact, since (see (4.10) and (4.11))

$$\psi_l^{(v)} \cong \frac{4\pi}{\lambda}\left[R_l + x_v \cos(\phi_l)\sin(\theta_l) + y_v \sin(\phi_l)\right], \tag{4.14}$$

where

$$\lambda \triangleq c/f_0 \tag{4.15}$$

is the wavelength associated with the *start frequency* f_0 of the generated frequency chirp, the sequence $\{\psi_l^{(v)}; v = 0, 1, ..., N_{VR}-1\}$ exhibits a periodic behavior characterized by the *normalised horizontal spatial frequency*

$$F_{\mathrm{H},l} \triangleq 2\frac{d_{\mathrm{H}}}{\lambda}\cos(\phi_l)\sin(\theta_l), \tag{4.16}$$

if the considered virtual elements form an horizontal *uniform linear array* (ULA), whose adjacent elements are spaced d_{H} m apart. Dually, if a virtual vertical ULA is assumed, the periodic variations observed in the same sequence of phases are characterized by the *normalised vertical spatial frequency*

$$F_{\mathrm{V},l} \triangleq 2\frac{d_{\mathrm{V}}}{\lambda}\sin(\phi_l), \tag{4.17}$$

where d_{V} denotes the distance between adjacent elements of the virtual array itself.

From the considerations illustrated above, the following conclusions can be easily inferred: a) on the one hand, achieving precise estimation of the range of a given target requires the availability of an accurate estimate of the normalised *frequency* of the complex tone associated with the target itself over at least one RX antenna; b) on the other hand, the quality of the estimate of the *direction of arrival* (DOA) characterizing the radar echo generated by a given target depends on the accuracy of the *phase* estimated over multiple virtual channels. The last consideration motivates the importance of accurately estimating this parameter over multiple antennas. It is worth stressing that limited attention is often paid to this technical issue in most of the technical literature dealing with the estimation of the parameters of a single tone or multiple superimposed tones; in fact, a lot of emphasis is usually put on the accuracy of frequency estimation, but the quality of phase estimates is neglected (e.g., see [11, 17, 35, 43–45]). Finally, in analysing the suitability of multiple tone estimators to colocated MIMO radar systems operating at millimetre waves, the following additional technical issues need to be taken carefully into account:

1) These radar systems often operate at short ranges and in the presence of extended targets. Each of resulting radar images is a cloud of point targets whose mutual spacing can be very

small. For this reason, the accuracy of these images depends on the frequency resolution achieved on each virtual antenna. In fact, this makes the radar receiver able to separate point targets characterized by similar ranges.

2) Information about the RCS of each point targets can be exploited in the classification of extended targets; for this reason, the availability of an accurate estimate of the amplitude of each radar echo can be very useful in a number of applications (e.g., in SAR imaging [15, 18]).

3) Distinct radar echoes can be characterized by substantially different *signal-to-noise ratios* (SNRs).

4) The number N of samples acquired over each virtual channel usually ranges from few hundreds to few thousands.

The last two issues explain why significant attention must be paid to the accuracy achieved by the adopted estimation algorithms at low SNRs and/or for relatively small values of N, since this can appreciably influence the quality of the generated radar image.

4.3 Approximate Maximum Likelihood Estimation of Single and Multiple Tones

In this section, we first derive a new method for estimating the parameters of a complex tone and we show how the same approach can be exploited to derive a method for estimating the parameters of a real tone. Then, we illustrate how these methods can be exploited to detect multiple superimposed tones and estimate their parameters through a deterministic procedure based on successive cancellation. Finally, we analyse the computational complexity of the developed estimation methods and compare them with some related techniques available in the technical literature.

4.3.1 Estimation of a single complex tone

Let us focus on the problem of estimating the parameters (namely, the frequency and complex amplitude) of a single tone contained in the *complex* sequence $\{x_{c,n};\, n = 0,\, 1,\, ...,\, N-1\}$, whose n-th sample is expressed by eq. (4.1) with $L = 1$, i.e. as

$$x_{c,n} = A \exp\left(j2\pi nF\right) + w_{c,n}, \tag{4.18}$$

with $n = 0,\, 1,\, ...,\, N-1$ where A and F are the complex amplitude and the normalised frequency, respectively, of the tone itself. It is well known that the ML estimates F_{ML} and A_{ML} of the parameters F and A, respectively, represent the solution of the *least square problem* (e.g., see [31, eq. (22)])

$$(F_{\mathrm{ML}}, A_{\mathrm{ML}}) \triangleq \arg\min_{\tilde{F},\tilde{A}} \varepsilon(\tilde{F}, \tilde{A}), \tag{4.19}$$

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

$$\varepsilon(\tilde{F}, \tilde{A}) \triangleq \frac{1}{N} \sum_{n=0}^{N-1} \varepsilon_n(\tilde{F}, \tilde{A}) \tag{4.20}$$

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

$$\varepsilon_n(\tilde{F}, \tilde{A}) \triangleq \left| x_{c,n} - s_n(\tilde{F}, \tilde{A}) \right|^2 \tag{4.21}$$

is the *square error* between the noisy sample $x_{c,n}$ (4.18) and its useful component

$$s_n(\tilde{F}, \tilde{A}) \triangleq \tilde{A} \exp\left(j2\pi n\tilde{F}\right), \tag{4.22}$$

evaluated under the assumption that $F = \tilde{F}$ and $A = \tilde{A}$. Substituting the RHS of the last equation in that of eq. (4.21) yields

$$\varepsilon_n(\tilde{F}, \tilde{A}) = \tilde{A}_R^2 + \tilde{A}_I^2 - 2(x_{c,n}^{(R)}\tilde{A}_R + x_{c,n}^{(I)}\tilde{A}_I)\cos(\tilde{\phi}_n) - 2(x_{c,n}^{(I)}\tilde{A}_R - x_{c,n}^{(R)}\tilde{A}_I)\sin(\tilde{\phi}_n) + |x_{c,n}|^2, \tag{4.23}$$

where $\tilde{A}_R \triangleq \Re\{\tilde{A}\}$, $\tilde{A}_I \triangleq \Im\{\tilde{A}\}$, $x_{c,n}^{(R)} \triangleq \Re\{x_{c,n}\}$, $x_{c,n}^{(I)} \triangleq \Im\{x_{c,n}\}$ and

$$\tilde{\phi}_n \triangleq 2\pi n\tilde{F}. \tag{4.24}$$

Then, substituting the RHS of eq. (4.23) in that of eq. (4.20) yields, after some manipulation,

$$\varepsilon(\tilde{F},\tilde{A}) = \varepsilon_x + \tilde{A}_R^2 + \tilde{A}_I^2 - 2\tilde{A}_R\,\bar{X}_R(\tilde{F}) - 2\tilde{A}_I\,\bar{X}_I(\tilde{F}), \tag{4.25}$$

where

$$\varepsilon_x \triangleq \frac{1}{N}\sum_{n=0}^{N-1}|x_{c,n}|^2, \tag{4.26}$$

$\Re\{x\}$ ($\Im\{x\}$) denotes the real (imaginary) part of the complex variable x and

$$\bar{X}(\tilde{F}) \triangleq \frac{1}{N}\sum_{n=0}^{N-1}x_{c,n}\exp\left(-j2\pi n\tilde{F}\right), \tag{4.27}$$

is, up to the scale factor $1/N$, the *Fourier transform* of the sequence $\{x_{c,n}\}$. Based on eq. (4.25), it is not difficult to show that the optimization problem expressed by eq. (4.19) does not admit a closed form solution because of the nonlinear 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) (e.g. see [58]). This allows us to transform the *two-dimensional* (2D) optimization problem expressed by eq. (4.19) into a couple of interconnected *one-dimensional* (1D) 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 couple $(F_c\,,\tilde{\delta})$ such that

$$\tilde{F} = F_c + \tilde{\delta}\,F_{\mathrm{DFT}}, \tag{4.28}$$

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

$$F_{\mathrm{DFT}} = 1/N_0 \tag{4.29}$$

is the normalized *fundamental frequency* associated with the N_0-th order *discrete Fourier transform* (DFT)

$$\mathbf{X}_0 = [X_{0,0}, X_{0,1}, ..., X_{0,N_0-1}]^T \tag{4.30}$$

of the zero padded version

$$\mathbf{x}_{0,\mathrm{ZP}} = \left[\mathbf{x}_0^T\;\mathbf{0}_{(M-1)N}^T\right]^T, \tag{4.31}$$

of the vector

$$\mathbf{x}_0 \triangleq [x_{c,0}, x_{c,1}, ..., x_{c,N-1}]^T \tag{4.32}$$

collecting all the elements of the sequence $\{x_{c,n}\}$, M is a positive integer (dubbed *oversampling factor*), $\mathbf{0}_D$ is a $D-$dimensional (column) null vector and $N_0 \triangleq M \cdot N$.

c) Expressing the dependence of the function $\varepsilon(\tilde{F},\tilde{A})$ (4.25) on the variable $\tilde{\delta}$ through its powers $\{\tilde{\delta}^l;\, 0 \le l \le 3\}$; 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.

Let us show now 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 $\varepsilon(\tilde{F},\tilde{A})$ (4.25) 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 variable $\tilde{A}$. In fact, the function $\varepsilon(\hat{F},\tilde{A})$ (4.25) is minimised with respect to $\tilde{A}$ selecting[3]

$$\tilde{A} = \hat{A} = \bar{X}(\hat{F}), \tag{4.33}$$

[3]This is a well known result (e.g., see [31, Sec. IV]).

where $\bar{X}(\hat{F})$ can be computed exactly through its expression (4.27) or, in an approximate fashion, through a computationally efficient procedure based on the fact that the vector

$$\bar{\mathbf{X}}_s \triangleq M\,\mathbf{X}_0 \tag{4.34}$$

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

$$\bar{X}_{0,k} \triangleq \frac{1}{N_0}\sum_{n=0}^{N-1} x_{c,n}\exp\left(-j\frac{2\pi nk}{N_0}\right), \tag{4.35}$$

of the vector $\mathbf{X}_0$ (4.30) can be expressed as

$$\bar{X}_{0,k} = \frac{N}{N_0}\bar{X}\left(F_k\right) = \frac{1}{M}\,\bar{X}\left(F_k\right), \tag{4.36}$$

where $F_k \triangleq k\,F_{\mathrm{DFT}}$, with $k = 0,\,1,\,...,\,N_0-1$. For this reason, an approximate evaluation of the quantity $\bar{X}(\hat{F})$ at a normalised frequency $\hat{F}$ different from any multiple of F_{DFT} (4.29) can be accomplished by interpolating the elements of the vector $\bar{\mathbf{X}}_s$ (4.34); the last vector, in turn, can be easily computed after evaluating the N_0-th *fast Fourier transform* (FFT) of $\mathbf{x}_{0,\mathrm{ZP}}$ (4.31), i.e. the vector $\mathbf{X}_0$ (4.30).

Let us take into consideration now sub-problem **P2**. Unluckily, this sub-problem, unlike the previous one, does not admit a closed form solution. For this reason, an approximate solution is developed below. Such a solution is based on representing the normalized frequency F in the same form as $\tilde{F}$ (see eq. (4.28)), i.e. as $F = F_c + \delta\,F_{\mathrm{DFT}}$ and on a novel method for estimating the real residual δ, i.e. for accomplishing the *fine estimation* of F. This method is derived as follows. Representing the trial normalized frequency $\tilde{F}$ according to eq. (4.28) allows us to express the variable $\tilde{\phi}_n$ (4.24) as

$$\tilde{\phi}_n = \hat{\theta}_n + n\tilde{\Delta}, \tag{4.37}$$

where

$$\tilde{\Delta} \triangleq 2\pi\tilde{\delta}\,F_{\mathrm{DFT}} \tag{4.38}$$

is a new variable and $\hat{\theta}_n \triangleq 2\pi n\,F_c$. Then, substituting the RHS of eq. (4.37) in that of eq. (4.23) (with $\tilde{A} = \hat{A}$) yields

$$\begin{aligned}\varepsilon_n(\tilde{F},\tilde{A}) = |x_{c,n}|^2 + \tilde{A}_R^2 + \tilde{A}_I^2 - 2(x_{c,n}^{(R)}\tilde{A}_R + x_{c,n}^{(I)}\tilde{A}_I)\cdot\left[\cos(\hat{\theta}_n)\cos(n\tilde{\Delta}) - \sin(\hat{\theta}_n)\sin(n\tilde{\Delta})\right]\\ - 2(x_{c,n}^{(I)}\tilde{A}_R - x_{c,n}^{(R)}\tilde{A}_I)\left[\sin(\hat{\theta}_n)\cos(n\tilde{\Delta}) + \cos(\hat{\theta}_n)\sin(n\tilde{\Delta})\right].\end{aligned} \tag{4.39}$$

If the normalized frequency F_{DFT} (4.29) is small enough (i.e., if the FFT order N_0 is large enough), the trigonometric functions $\cos(n\tilde{\Delta})$ and $\sin(n\tilde{\Delta})$ appearing in the RHS of the last equation can be approximated as (F.3) and (F.4), respectively. Then, substituting the RHS of these approximations in that of eq. (4.39) produces, after some manipulation, the approximate expression

$$\begin{aligned}\varepsilon_{\mathrm{CSFE}}\left(\tilde{\Delta},\hat{A}\right) \triangleq \varepsilon_x + \hat{A}_R^2 + \hat{A}_I^2 + \frac{1}{3}\tilde{\Delta}^3\left(\hat{A}_R\,\bar{X}_{3,\rho}^{(I)} - \hat{A}_I\,\bar{X}_{3,\rho}^{(R)}\right) + \tilde{\Delta}^2\left(\hat{A}_R\,\bar{X}_{2,\rho}^{(R)} + \hat{A}_I\,\bar{X}_{2,\rho}^{(I)}\right)\\ + 2\tilde{\Delta}\left(\hat{A}_I\,\bar{X}_{1,\rho}^{(R)} - \hat{A}_R\,\bar{X}_{1,\rho}^{(I)}\right) - 2\left(\hat{A}_R\,\bar{X}_{0,\rho}^{(R)} + \hat{A}_I\,\bar{X}_{0,\rho}^{(I)}\right)\end{aligned} \tag{4.40}$$

for the function $\varepsilon(\tilde{F},\hat{A})$ (4.25) (see Appendix F); here,

$$\rho \triangleq F_c/F_{\mathrm{DFT}}, \tag{4.41}$$

$X_{k,\rho}^{(R)} \triangleq \Re\{X_{k,\rho}\}$, $X_{k,\rho}^{(I)} \triangleq \Im\{X_{k,\rho}\}$ for any k,

$$\bar{X}_{k,\rho} \triangleq \frac{1}{N_0}\sum_{n=0}^{N-1} x_{k,n}\exp\left(-j\frac{2\pi n\rho}{N_0}\right), \tag{4.42}$$

for any ρ and $k = 1, 2, 3$ and $x_{k,n} \triangleq n^k \cdot x_{c,n}$, with $n = 0, 1, ..., N-1$.

It is important to point out that: a) if ρ is an integer, the quantity $\bar{X}_{k,\rho}$ (4.42) (with $k = 1, 2$ and 3) represents the ρ-th element of the vector

$$\mathbf{X}_k = [X_{k,0}, X_{k,1}, ..., X_{k,N_0-1}]^T \tag{4.43}$$

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

$$\mathbf{x}_{k,\mathrm{ZP}} = \left[\mathbf{x}_k^T \, \mathbf{0}_{(M-1)N}^T\right]^T, \tag{4.44}$$

of the vector

$$\mathbf{x}_k \triangleq [x_{k,0}, x_{k,1}, ..., x_{k,N-1}]^T; \tag{4.45}$$

b) if ρ is not an integer, the quantity $\bar{X}_{k,\rho}$ can be evaluated exactly on the basis of eq. (4.42) or, in an approximate fashion, by interpolating I adjacent elements of the N_0-dimensional vectors $\mathbf{X}_k$ (4.43), where I denotes the selected interpolation order; c) the evaluation of the vectors $\{\mathbf{X}_k; k = 1, 2, 3\}$ requires three *additional* FFTs.

Since the function $\varepsilon_{\mathrm{CSFE}}(\tilde{\Delta}, \hat{A})$ (4.40) is a polynomial of degree three in the variable $\tilde{\Delta}$, an estimate $\hat{\Delta}$ of Δ and, consequently, an estimate (see eq. (4.38))

$$\hat{\delta} = \hat{\Delta}/(2\pi F_{\mathrm{DFT}}) \tag{4.46}$$

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

$$a(\rho)\ \tilde{\Delta}^2 + b(\rho)\ \tilde{\Delta} + c(\rho) = 0, \tag{4.47}$$

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

$$a(\rho) \triangleq \Im\left\{\hat{A}^* \bar{X}_{3,\rho}\right\}/2, \tag{4.48}$$

$$b(\rho) \triangleq \Re\{\hat{A}^* \bar{X}_{2,\rho}\}, \tag{4.49}$$

and

$$c(\rho) \triangleq -\Im\{\hat{A}^* \bar{X}_{1,\rho}\}. \tag{4.50}$$

Note that only one of the two solutions of eq. (4.47), namely

$$\hat{\Delta} = -\frac{b(\rho) + \sqrt{(b(\rho))^2 - 4a(\rho)\ c(\rho)}}{2a(\rho)} \tag{4.51}$$

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. (4.47), i.e. setting $a(\rho) = 0$. This leads to a first-degree equation, whose solution is

$$\hat{\Delta} = -c(\rho)/b(\rho). \tag{4.52}$$

Given an estimate $\hat{\Delta}$ of Δ (and, consequently, and estimate $\hat{\delta}$ of δ; see eq. (4.46)), the *fine estimate*

$$\hat{F} = F_c + \hat{\delta}\, F_{\mathrm{DFT}} = F_c + \hat{\Delta}/(2\pi) \tag{4.53}$$

of F can be evaluated on the basis of eq. (4.28).

The mathematical results derived above allow us to derive a novel estimation algorithm, called *complex single frequency estimator* (CSFE), for iteratively estimating the normalised frequency F and the complex amplitude A. This algorithm is initialised by: 1) Evaluating: a) the vector $\mathbf{X}_0$ (4.30); b) the initial coarse estimate $\hat{F}_c^{(0)}$ of F as

$$\hat{F}_c^{(0)} = \hat{\alpha}\, F_{\mathrm{DFT}}, \tag{4.54}$$

where the integer $\hat{\alpha}$ is computed by means of the well known *periodogram method* (e.g., see [31, Sec. IV] or [35, Sec. I]), i.e. as

$$\hat{\alpha} = \arg \max_{\tilde{\alpha} \in \{0,1,...,N_0-1\}} \left| \bar{X}_{0,\tilde{\alpha}} \right| ; \tag{4.55}$$

c) the quantity (see eq. (4.41))

$$\hat{\rho}^{(0)} \triangleq \hat{F}_c^{(0)} / F_{\mathrm{DFT}} = \hat{\alpha}; \tag{4.56}$$

d) the initial estimate $\hat{A}^{(0)}$ of A on the basis of eq. (4.33) with $\hat{F} = \hat{F}_c^{(0)}$; e) the spectral coefficients $\bar{X}_{1,\hat{\alpha}}$, $\bar{X}_{2,\hat{\alpha}}$ and $\bar{X}_{3,\hat{\alpha}}$ on the basis of eq. (4.42); f) the coefficients $\{a(\hat{\alpha}), b(\hat{\alpha}), c(\hat{\alpha})\}$ ($\{b(\hat{\alpha}), c(\hat{\alpha})\}$ according to eqs. (4.48)–(4.50) and the first estimate $\hat{\Delta}^{(0)}$ of Δ on the basis of eq. (4.51) (eq. (4.52)); g) the first fine estimate $\hat{F}^{(0)}$ of F on the basis of eqs. (4.28) and (4.46), i.e. as

$$\hat{F}^{(0)} = \hat{F}_c^{(0)} + \hat{\Delta}^{(0)} / (2\pi) \tag{4.57}$$

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 $i = 1, 2, ..., N_{\mathrm{CSFE}}$, where N_{CSFE} 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 CSFE-Sp).

CSFE-S1 - The new estimate $\hat{\Delta}^{(i)}$ of Δ is computed through eq. (4.51) (eq. (4.52)); 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

$$\rho = \hat{\rho}^{(i-1)} \triangleq \hat{F}^{(i-1)} / F_{\mathrm{DFT}} \tag{4.58}$$

are assumed. Then,

$$\hat{F}^{(i)} = \hat{F}^{(i-1)} + \hat{\Delta}^{(i)} / (2\pi) \tag{4.59}$$

is evaluated.

CSFE-S2 - The new estimate $\hat{A}^{(i)}$ of $\hat{A}$ is evaluated through eq. (4.33); $\hat{F} = \hat{F}^{(i)}$ 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_{CSFE}-th) iteration, the fine estimates $\hat{F} = \hat{F}^{(N_{\mathrm{CSFE}})}$ and $\hat{A} = \hat{A}^{(N_{\mathrm{CSFE}})}$ of F and A, respectively, become available.

The CSFE is summarized in Algorithm 5. It is important to point out that:

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

b) The estimate $\hat{\Delta}^{(i)}$ evaluated according to eq. (4.52) is expected to be less accurate than that computed on the basis of eq. (4.51). However, our numerical results have evidenced that both solutions achieve similar accuracy. Despite this, eq. (4.51) is adopted in Algorithm 5 for generality.

c) The CSFE can be employed even if the single tone appearing in the RHS of eq. (4.18) is replaced by the superposition of L distinct tones (see eq. (4.1)). In this case, the strongest (i.e., the dominant) tone is usually detected through the periodogram method (see eq. (4.55)) and the parameters of this tone are estimated in the presence of both Gaussian noise and the interference due to the remaining tones. Therefore, the estimation accuracy of the CSFE is affected by both the amplitudes and the frequencies of the other $(L-1)$ tones.

d) A stopping criterion, based on the trend of the sequence $\{\hat{\Delta}^{(i)}; i = 1, 2, ...\}$, can be easily formulated for the CSFE. For instance, the execution of its two steps can be stopped if, at the end of the i-th iteration, the condition $|\hat{\Delta}^{(i)}| < \varepsilon_{\Delta}$ is satisfied; here, ε_{Δ} represents a proper threshold.

e) The estimates generated by the CSFE algorithm are *unbiased*, provided the overall number of iterations it accomplishes is large enough; a proof of this statement is provided in Appendix H.

Algorithm 5: Complex single frequency estimation

1 **Initialisation:**

a- Evaluate $\mathbf{X}_0$ (4.30) and $\hat{\alpha}$ on the basis of eq. (4.55); then, compute the initial estimate $\hat{A}^{(0)}$ of A according to eq. (4.33) and set $\hat{\rho}^{(0)} = \hat{\alpha}$ (see eq. (4.56)).

b- Evaluate $\bar{X}_{1,\hat{\alpha}}$, $\bar{X}_{2,\hat{\alpha}}$ and $\bar{X}_{3,\hat{\alpha}}$ according to eq. (4.42); then compute $\{a(\hat{\alpha}),\, b(\hat{\alpha}),\, c(\hat{\alpha})\}$ according to eqs. (4.48)–(4.50). Finally compute $\hat{\Delta}^{(0)}$ and $\hat{F}^{(0)}$ according to eqs. (4.51) and (4.53), respectively.

2 **Refinement: for** $i = 1$ *to* N_{CSFE} **do**

c- *Estimation of* A:
Set $\hat{F} = \hat{F}^{(i-1)}$; then, evaluate $\bar{X}(\hat{F})$ according to eq. (4.27) or by interpolating a few adjacent elements of $\bar{\mathbf{X}}_s$ (4.34). Finally, compute $\hat{A}^{(i)}$ and $\hat{\rho}^{(i-1)}$ according to eqs. (4.33) and (4.58), respectively.

d- *Estimation of* F:
Set $\hat{A} = \hat{A}^{(i)}$; then compute $\bar{X}_{k,\hat{\rho}^{(i-1)}}$ according to eq. (4.42) or or by interpolating a few adjacent elements of $\bar{\mathbf{X}}_k$ (4.43). Finally, compute $\{a(\hat{\rho}^{(i-1)}),\, b(\hat{\rho}^{(i-1)}),\, c(\hat{\rho}^{(i-1)})\}$ according to eqs. (4.48)–(4.50), and $\hat{\Delta}^{(i)}$ and $\hat{F}^{(i)}$ according to eqs. (4.51) and (4.59), respectively.

end

4.3.2 Estimation of a single real tone

All the results developed in the previous paragraph refer to the complex sequence $\{x_{c,n}\}$, whose n-th element is expressed by eq. (4.1). However, a similar estimation method (dubbed *single frequency estimator*, SFE) can be developed for the real counterpart, i.e. for a real sequence $\{x_{r,n};\ n = 0,\, 1,\, ...,\, N-1\}$, whose n-th sample is expressed by eq. (4.2) with $L = 1$, i.e. as

$$x_{r,n} = a\cos\left(2\pi nF + \psi\right) + w_{r,n} \tag{4.60}$$

or, equivalently, as (see eq. (4.3))

$$x_{r,n} = C\exp\left(j2\pi nF\right) + C^*\exp\left(-j2\pi nF\right) + w_{r,n}, \tag{4.61}$$

where (see eq. (4.4))

$$C \triangleq \frac{A}{2} = \frac{a}{2}\exp\left(j\psi\right) \tag{4.62}$$

In this case, ML estimation of the parameters F and C can be formulated in a similar way as eq. (4.19), the only difference being represented by the fact that: a) the parameter A is replaced by C; b) the term $\varepsilon_n(\tilde{F}, \tilde{A})$ appearing in the RHS of eq. (4.20) is replaced by

$$\varepsilon_n(\tilde{F}, \tilde{C}) \triangleq \left[x_{r,n} - s_n(\tilde{F}, \tilde{C})\right]^2. \tag{4.63}$$

where

$$s_n(\tilde{F}, \tilde{C}) \triangleq \tilde{C}\exp\left(j2\pi n\tilde{F}\right) + \tilde{C}^*\exp\left(-j2\pi n\tilde{F}\right), \tag{4.64}$$

represents the useful component of $\{x_{r,n}\}$ (4.60) evaluated under the assumption that $F = \tilde{F}$ and $C = \tilde{C}$. Substituting the RHS of eq. (4.64) in that of eq. (4.63) produces, after some manipulation,

$$\begin{aligned}\varepsilon_n(\tilde{F}, \tilde{C}) = x_{r,n}^2 + 2\left[\tilde{C}_R^2 + \tilde{C}_I^2\right] - 4x_{r,n}\left[\tilde{C}_R\cos(\tilde{\phi}_n) - \tilde{C}_I\sin(\tilde{\phi}_n)\right] \\ + 2\left[\left(\tilde{C}_R^2 - \tilde{C}_I^2\right)\cos(2\tilde{\phi}_n) - 2\tilde{C}_R\,\tilde{C}_I\sin(2\tilde{\phi}_n)\right].\end{aligned} \tag{4.65}$$

Then, substituting the RHS of eq. (4.65) in that of eq. (4.20) yields

$$\begin{aligned}\varepsilon(\tilde{F}, \tilde{C}) = \varepsilon_x + 2\left[\left(\tilde{C}_R^2 - \tilde{C}_I^2\right)\, g_R(\tilde{F}) + 2\tilde{C}_R\,\tilde{C}_I\, g_I(\tilde{F})\right] \\ +2\left[\tilde{C}_R^2 + \tilde{C}_I^2\right] - 4\left[\tilde{C}_R\ \bar{X}_R(\tilde{F}) + \tilde{C}_I\,\bar{X}_I(\tilde{F})\right],\end{aligned} \tag{4.66}$$

where

$$\varepsilon_x \triangleq \frac{1}{N}\sum_{n=0}^{N-1} x_{r,n}^2, \tag{4.67}$$

$g_R(\tilde{F}) \triangleq \Re\{g(\tilde{F})\}$, $g_I(\tilde{F}) \triangleq \Im\{g(\tilde{F})\}$ and

$$g(\tilde{F}) \triangleq \frac{1}{N}\sum_{n=0}^{N-1} \exp\left(-j2\tilde{\phi}_n\right) = \frac{1}{N}\sum_{n=0}^{N-1} \exp\left(-j4\pi n\tilde{F}\right). \tag{4.68}$$

The procedure adopted to minimise the function (4.66) with respect to $\tilde{F}$ and $\tilde{C}$ is similar to that illustrated in the previous paragraph for the CSFE. For this reason, in the following we limit to show essential mathematical results only. First of all, given $\tilde{F} = \hat{F}$, the function $\varepsilon(\tilde{F}, \tilde{C})$ (4.66) is minimised with respect to $\tilde{C}$ by solving the equations

$$\tilde{C}_R - \bar{X}_R(\hat{F}) + \tilde{C}_R\, g_R(\hat{F}) + \tilde{C}_I\, g_I(\hat{F}) = 0 \tag{4.69}$$

and

$$\tilde{C}_I - \bar{X}_I(\hat{F}) - \tilde{C}_I\, g_R(\hat{F}) + \tilde{C}_R\, g_I(\hat{F}) = 0, \tag{4.70}$$

that result from computing the partial derivative of the RHS of eq. (4.66) with respect to $\tilde{C}_R$ and $\tilde{C}_I$, respectively. Solving the linear system of equations (4.69)–(4.70) in the unknowns $\tilde{C}_R$ and $\tilde{C}_I$ produces the optimal values

$$\hat{C}_R = \frac{\bar{X}_R(\hat{F})\left[1 - g_R(\hat{F})\right] - \bar{X}_I(\hat{F})\, g_I(\hat{F})}{1 - \left|g(\hat{F})\right|^2} \tag{4.71}$$

and

$$\hat{C}_I = \frac{-\bar{X}_R(\hat{F}) g_I(\hat{F}) + \bar{X}_I(\hat{F})\left[1 + g_R(\hat{F})\right]}{1 - \left|g(\hat{F})\right|^2}. \tag{4.72}$$

of $\tilde{C}_R$ and $\tilde{C}_I$, respectively. Putting together the last two formulas yields

$$\hat{C} = \hat{C}_R + j\hat{C}_I = \frac{\bar{X}(\hat{F}) - \bar{X}^*(\hat{F}) g(\hat{F})}{1 - \left|g(\hat{F})\right|^2} = \bar{C}(\hat{F}). \tag{4.73}$$

Therefore, given $\tilde{F} = \hat{F}$, the optimal value $\hat{C}$ of the variable $\tilde{C}$ can be computed exactly through the last equation; this requires the evaluation of $\bar{X}(\hat{F})$ and $g(\hat{F})$ (see eqs. (4.27) and (4.68), respectively). Note that, on the one hand, $g(\hat{F})$ can be easily evaluated through its exact expression

$$g(\hat{F}) = \frac{1}{N}\frac{\exp(-j4\pi N\hat{F}) - 1}{\exp(-j4\pi\hat{F}) - 1}, \tag{4.74}$$

which is easily derived from its definition (4.68). On the other hand, $\bar{X}(\hat{F})$ can be computed exactly through its expression (4.27) or, in an approximate fashion, by interpolating the elements of the vector $\bar{\mathbf{X}}_s$ (4.34) following the same mathematical approach as that illustrated in the previous paragraph for the CSFE; note that, in this case, the complex sequence $\{x_{c,n}\}$ appearing in the RHS of (4.35) is substituted with the real sequence $\{x_{r,n}\}$ (see eq. (4.60)). The result given by eq. (4.73) is *exact*. On the contrary, given $\tilde{C} = \hat{C}$, a closed form solution for the value of $\tilde{F}$ minimising the function $\varepsilon(\tilde{F}, \tilde{C})$ cannot be derived because of the *nonlinear* dependence of this function on $\tilde{F}$. However, following the same mathematical approach as that illustrated for the

CSFE, the approximate expression

$$\begin{aligned}\varepsilon_{\mathrm{SFE}}(\tilde{\Delta},\hat{C}) \triangleq \varepsilon_x + 2\left[\hat{C}_R^2+\hat{C}_I^2\right] - 2\hat{C}_R\left[2X_{0,\rho}^{(R)}+2\tilde{\Delta}\,X_{1,\rho}^{(I)}-\tilde{\Delta}^2\,X_{2,\rho}^{(R)}\right]\\ -2\hat{C}_I\left[2X_{0,\rho}^{(I)}-2\tilde{\Delta}\,X_{1,\rho}^{(R)}-\tilde{\Delta}^2\,X_{2,\rho}^{(I)}\right]+2\left(\hat{C}_R^2-\hat{C}_I^2\right)\cdot\\ \left[K_0^{(R)}(2\rho)+2\tilde{\Delta}\,K_1^{(I)}(2\rho)-2\tilde{\Delta}^2\,K_2^{(R)}(2\rho)+\right.\\ \left.-\frac{4}{3}\tilde{\Delta}^3K_3^{(I)}(2\rho)\right]-4\hat{C}_R\,\hat{C}_I\left[-K_0^{(I)}(2\rho)+2\tilde{\Delta}\,K_1^{(R)}(2\rho)+\right.\\ \left.+2\tilde{\Delta}^2K_2^{(I)}(2\rho)-\frac{4}{3}\tilde{\Delta}^3K_3^{(R)}(2\rho)\right]\end{aligned} \tag{4.75}$$

can be obtained for the function $\varepsilon(\tilde{F},\hat{C})$ (4.66) (see Appendix F); here, $X_{k,\rho}$ is still defined by eq. (4.42), $K_p^{(R)}(x) \triangleq \Re\{K_p(x)\}$, $K_p^{(I)}(x) \triangleq \Im\{K_p(x)\}$ for any k, p and $x \geq 0$,

$$K_p(x) \triangleq \frac{1}{N}\sum_{n=0}^{N-1} g_p[n]\exp\left(-j\frac{2\pi nx}{N_0}\right), \tag{4.76}$$

$g_p[n] \triangleq n^p$ for $p = 1$, 2 and 3, $g_0[n] \triangleq 1$ and

$$x_{k,n} \triangleq n^k \cdot x_{r,n} \tag{4.77}$$

with $n = 0$, 1, ..., $N-1$. It is worth stressing that:

a) The considerations expressed about the evaluation of the quantities $\{X_{k,\rho}\}$ in our derivation of the CSFE apply to the SFE too. However, in this case, the additional coefficients $\{K_m; m = 0, 1, 2\}$, need to be computed.

b) The initial estimate $\hat{\rho}^{(0)} = \hat{\alpha}$ of ρ is evaluated in a similar way as the CSFE (see eq. (4.55)), i.e., as

$$\hat{\alpha} = \arg\max_{\tilde{\alpha}\in\{0,1,\ldots,N_0/2\}} \left|\bar{X}_{0,\tilde{\alpha}}\right|. \tag{4.78}$$

The minimization of the function $\varepsilon_{\mathrm{SFE}}(\tilde{\Delta},\hat{C})$ (4.75) with respect to $\tilde{\Delta}$ is achieved by taking its partial derivative with respect to this variable and setting it to zero; this results again in the quadratic equation (4.47), whose coefficients are

$$a(\rho) \triangleq 2\Im\left\{\left(\hat{C}^*\right)^2 K_3(2\rho)\right\}, \tag{4.79}$$

$$b(\rho) \triangleq -\Re\left\{\hat{C}^*\bar{X}_{2,\rho}\right\} + 2\Re\left\{\left(\hat{C}^*\right)^2 K_2(2\rho)\right\} \tag{4.80}$$

and

$$c(\rho) \triangleq \Im\left\{\hat{C}^*\bar{X}_{1,\rho}\right\} - \Im\left\{\left(\hat{C}^*\right)^2 K_1(2\rho)\right\}. \tag{4.81}$$

Following the approach adopted in the development of the CSFE and exploiting the mathematical results expressed by eq. (4.73) and by eqs. (4.79)–(4.81) allow us to easily derive the SFE; this algorithm is summarized in Algorithm 6. Finally, it is important to point out that the coefficients $\{K_p(2\rho);\ p = 1,\ 2,\ 3\}$ can be computed exactly on the basis of eq. (4.76). However, since the definition (4.76) can be put in the equivalent form

$$K_p(x) \triangleq \frac{1}{N}\sum_{n=0}^{N-1} n^p\,(q(x))^n, \tag{4.82}$$

where

$$q(x) \triangleq \exp\left(-j\frac{2\pi x}{N_0}\right), \tag{4.83}$$

the identities

$$(q-1)^2 \sum_{n=0}^{N-1} n\, q^n = (N-1)q^{N+1} - Nq^N + q, \tag{4.84}$$

$$(q-1)^3 \sum_{n=0}^{N-1} n^2 q^n = (N-1)^2 q^{N+2} + \left(-2N^2 + 2N + 1\right) q^{N+1} + N^2 q^N - q^2 - q \tag{4.85}$$

and

$$\begin{aligned}(q-1)^4 \sum_{n=0}^{N-1} n^3 q^n = {} & q + 4q^2 + q^3 - N^3 q^N \\ & + \left(3N^3 - 3N^2 - 3N - 1\right) q^{N+1} \\ & + \left(-3N^3 + 6N^2 - 4\right) q^{N+2} + (N-1)^3 q^{N+3}\end{aligned} \tag{4.86}$$

holding for any $q \in \mathbb{C}$, can be exploited for the efficient computation of $K_p(2\rho)$ for any p and ρ.

Algorithm 6: Single frequency estimation

1 **Initialisation:**

a- Evaluate $\mathbf{X}_0$ (4.30) and $\hat{\alpha}$ on the basis of eq. (4.78); then, compute the initial estimate $\hat{C}^{(0)}$ of C on the basis of eqs. (4.73) and set $\hat{\rho}^{(0)} = \hat{\alpha}$ (see eq. (4.56)).

b- Evaluate the quantities $\{K_p(2\hat{\alpha});\ p = 1,\ 2,\ 3\}$ according to eq. (4.76), and $\bar{X}_{1,\hat{\alpha}}$ and $\bar{X}_{2,\hat{\alpha}}$ according to eq. (4.42); then, compute $\{a(\hat{\alpha}),\ b(\hat{\alpha}),\ c(\hat{\alpha})\}$ according to eqs. (4.79)–(4.81). Finally compute $\hat{\Delta}^{(0)}$ and $\hat{F}^{(0)}$ according to eqs. (4.51) and eq. (4.53), respectively.

2 **Refinement: for** $i = 1$ *to* N_{SFE} **do**

c- *Estimation of* C:
Set $\hat{F} = \hat{F}^{(i-1)}$; then, evaluate $g(\hat{F})$ on the basis of eq. (4.74), and $\bar{X}(\hat{F})$ according to eq. (4.27) or by interpolating a few adjacent elements of $\bar{\mathbf{X}}_s$ (4.34). Finally, compute $\hat{\rho}^{(i-1)}$ and $\hat{C}^{(i)}$ according to eqs. (4.58) and (4.73), respectively.

d- *Estimation of* F:
Set $\hat{C} = \hat{C}^{(i)}$; then, compute $K_p(2\hat{\rho}^{(i-1)})$ by means of eqs. (4.84)–(4.86), and $\bar{X}_{k,\hat{\rho}^{(i-1)}}$ according to eq. (4.42) or or by interpolating a few adjacent elements of $\bar{\mathbf{X}}_k$ (4.43); then, compute $\{a(\hat{\rho}^{(i-1)}),\ b(\hat{\rho}^{(i-1)}),\ c(\hat{\rho}^{(i-1)})\}$ on the basis of eqs. (4.79)–(4.81). Finally, compute $\hat{\Delta}^{(i)}$ and $\hat{F}^{(i)}$ according to eqs. (4.51) and (4.59), respectively.

end

4.3.3 Estimation of multiple tones

Let us analyse now in detail how the techniques derived in the previous paragraphs can be exploited to estimate the multiple tones that form the useful component of the complex (real) sequence $\{x_{c,n}\}$ ($\{x_{r,n}\}$), when its n-th sample is expressed by eq. (4.1) (eq. (4.2)) with $L > 1$. The recursive method we develop to achieve this target is based on the following basic principles:

1) Tones are *sequentially* detected and estimated.

2) The detection of a new tone and the estimation of its parameters are based on the procedure developed for the CSFE (SFE) in the previous paragraph; in addition, a cancellation algorithm is incorporated in this method to remove the contribution of previously detected tones from all the spectral information (namely, the spectrum $\bar{X}(F)$ (4.27), the vector $\mathbf{X}_0$ (4.30) and the coefficients $\{\bar{X}_{k,\rho}\}$ (4.42)), that are processed to detect and estimate the new tone.

3) After detecting a new tone and estimating its parameters, a *re-estimation technique* is executed to improve the accuracy of both this tone and the previously estimated tones; the proposed technique is inspired by the related methods described in refs. [11], [17] and [18].

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 frequency estimation and cancellation* (CSFEC) or *single frequency estimation and cancellation* (SFEC), depending on the fact that the input sequence is complex or real, respectively. In the following, we focus only on the CSFEC method only; however, readers should keep in mind that the differences between this method and the SFEC are similar to those illustrated in our description of the single tone estimation algorithms on which they are based. The CSFEC algorithm is initialised by:

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

2) Setting the recursion index r to 1.

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

$$\hat{\mathbf{F}}[r-1] = \left[\hat{F}_0[r-1], \hat{F}_1[r-1], ..., \hat{F}_{r-1}[r-1]\right]^T, \tag{4.87}$$

and

$$\hat{\mathbf{A}}[r-1] = \left[\hat{A}_0[r-1], \hat{A}_1[r-1], ..., \hat{A}_{r-1}[r-1]\right]^T, \tag{4.88}$$

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

$$\hat{\mathbf{F}}[r] = \left[\hat{F}_0[r], \hat{F}_1[r], ..., \hat{F}_r[r]\right]^T \tag{4.89}$$

and

$$\hat{\mathbf{A}}[r] = \left[\hat{A}_0[r], \hat{A}_1[r], ..., \hat{A}_r[r]\right]^T \tag{4.90}$$

after: a) *estimating* the frequency $\hat{F}_r[r]$ and the associated complex amplitude $\hat{A}_r[r]$ of a new (i.e., of the r-th) tone (if any); b) *refining* the estimates of the r tones 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 CSFEC-Sp).

CSFEC-S1 (CSFE *initialisation with cancellation*) - In this step, the following quantities are evaluated (see the *initialisation* part of Algorithm 5):

a) The *residual spectrum*

$$\mathbf{X}_0[r] = [X_{0,0}[r], X_{0,1}[r], ..., X_{0,N_0-1}[r]]^T \triangleq \mathbf{X}_0 - \mathbf{C}_0\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], r\right), \tag{4.91}$$

where $\mathbf{X}_0$ is the N_0-th order DFT of the zero padded version $\mathbf{x}_{0,\text{ZP}}$ of the vector $\mathbf{x}_0$ collecting all the elements of the sequence $\{x_{c,n}\}$ (see eqs. (4.31)–(4.32)) and the N_0-dimensional vector

$$\mathbf{C}_0\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], r\right) \triangleq \sum_{l=0}^{r-1} \bar{\mathbf{C}}_0\left(\hat{A}_l[r-1], \hat{F}_l[r-1]\right), \tag{4.92}$$

represents the contribution given by all the estimated tones to $\mathbf{X}_0$ (in particular, $\bar{\mathbf{C}}_0(\hat{A}_l[r-1], \hat{F}_l[r-1])$ is the contribution provided by l-th tone to the vector $\mathbf{X}_0$ (the expression of this vector can be found in Appendix G). If the overall energy

$$\mathcal{E}_0[r] \triangleq |\mathbf{X}_0[r]|^2 \tag{4.93}$$

satisfies the inequality $\mathcal{E}_0[r] < \mathcal{T}_{\text{CSFEC}}$, where $\mathcal{T}_{CSFEC}$ is a proper threshold, the algorithm stops and the estimate $\hat{L} = r$ of L is generated.

b) The integer (see eq. (4.55))

$$\hat{\alpha}[r] = \arg \max_{\tilde{\alpha} \in \{0,1,...,N_0-1\}} |X_{0,\tilde{\alpha}}[r]|, \tag{4.94}$$

that represents the index of the element of $\mathbf{X}_0[r]$ (4.91) having the largest absolute value.

c) The preliminary estimate (see eq. (4.33))

$$\bar{A}_r[r] = \bar{X}(\hat{F}_{c,r}[r]) - \bar{X}_{\mathrm{lk},0}\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], \hat{F}_{c,r}[r]\right) \tag{4.95}$$

of the complex amplitude of the new tone; here, $\hat{F}_{c,r}[r] = \hat{\alpha}[r]\, F_{\mathrm{DFT}}$ and

$$\bar{X}_{\mathrm{lk},0}\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], \hat{F}_{c,r}[r]\right) \triangleq \sum_{l=0}^{r-1} \bar{X}_0\left(\hat{A}_l[r-1], \hat{F}_l[r-1], \hat{F}_{c,r}[r]\right) \tag{4.96}$$

represent the *coarse estimate* of the frequency of the new tone (see eq. (4.54)) and the contribution given to $\bar{X}(F)$ by all the estimated tones (i.e., the *leakage*) at the frequency $F = \hat{F}_{c,r}[r]$ (in particular, $\bar{X}_0(\hat{A}_l[r-1], \hat{F}_l[r-1], \hat{F}_{c,r}[r])$ is the leakage due to the l-th tone; the expression of the function $\bar{X}_{\mathrm{lk},k}(\hat{A}, \hat{F}, \bar{F})$ is provided in Appendix G).

d) The spectral coefficient

$$\bar{X}_{k,\rho[r]}[r] = \bar{X}_{k,\rho[r]} - \hat{X}_{\mathrm{lk},k}\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], \hat{F}_{c,r}[r]\right) \tag{4.97}$$

for $k = 1, 2$ and 3; here, we have that (see eq. (4.56))

$$\rho[r] = \hat{F}_{c,r}[r]/F_{\mathrm{DFT}} = \hat{\alpha}[r] \tag{4.98}$$

and

$$\hat{X}_{\mathrm{lk},k}\left(\hat{\mathbf{A}}[r-1], \hat{\mathbf{F}}[r-1], \hat{F}_{c,r}[r]\right) \triangleq \sum_{l=0}^{r-1} \bar{X}_{\mathrm{lk},k}\left(\hat{A}_l[r-1], \hat{F}_l[r-1], \hat{F}_{c,r}[r]\right) \tag{4.99}$$

is the contribution given to $\bar{X}_{k,\rho[r]}[r]$ by all the estimated tones (i.e., the *leakage*) at the frequency $\hat{F}_{c,r}[r]$ (in particular, $\bar{X}_{\mathrm{lk},k}(\hat{A}_l[r-1], \hat{F}_l[r-1], \hat{F}_{c,r}[r])$ represents the leakage due to the l-th estimated tone).

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 frequency

$$\hat{F}_r^{(0)} = \hat{F}_{c,r}[r] + \hat{\Delta}^{(0)}[r]/(2\pi) \tag{4.100}$$

on the basis of eqs. (4.48)–(4.50), eq. (4.51) (or eq. (4.52)), and eqs. (4.54) and (4.57). Note that $\hat{F}_r^{(0)}$ represents the *initial fine estimate* of the normalized frequency of the new tone.

The evaluation of the frequency $\hat{F}_r^{(0)}$ (4.100) concludes the initialization of the modified CSFE executed for the detection and the estimation of the new tone.

CSFEC-S2 (CSFE *refinement with cancellation*) - After carrying out the first step, N_{CSFE} iterations[4] are executed to refine the estimate of the parameters of the new tone. The processing accomplished in this step follows closely that described in the *refinement* part of Algorithm 5. For this reason, in each iteration, a new estimate of the complex amplitude and of the residual of frequency of the r-th tone are computed. This requires re-using eqs. (4.95)–(4.96) and (4.97), 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, of Algorithm 5). At the end of the last iteration, the frequency $\hat{F}_{\mathrm{CSFE},r}[r]$ and the associated complex amplitude $\hat{A}_{\mathrm{CSFE},r}[r]$ of the new tone are available; these represent $\hat{F}_r[r]$ and $\hat{A}_r[r]$, respectively, if the following re-estimation step is not accomplished.

CSFEC-S3 (*re-estimation*) - This step is fed by the $(r+1)$ normalized frequencies $\{\hat{F}_0[r-1], \hat{F}_1[r-1], ..., \hat{F}_{r-1}[r-1], \hat{F}_{\mathrm{CSFE},r}[r]\}$ and their complex amplitudes $\{\hat{A}_0[r-1], \hat{A}_1[r-1], ..., \hat{A}_{r-1}[r-1], \hat{A}_{\mathrm{CSFE},r}[r]\}$. It consists in repeating the previous step for each of the detected tones, starting from the first tone and ending with $(r+1)$-th one. This means that, when re-estimating the l-th tone, the leakage due to the tones whose index belong to set $\{0, 1, .., l-1, l+1, ..., r\}$

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

has to be removed by exploiting equations similar to (4.95) and (4.97)–(4.99), with $l = 0, 1, ..., r$. This allows us to progressively refine the amplitude and the frequency of each tone, so generating the final frequencies $\{\hat{F}_0[r], \hat{F}_1[r], ..., \hat{F}_r[r]\}$ 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.

Our simulation results have evidenced that, unluckily, the estimates generated by the CSFEC algorithm are biased if the values selected for the parameters N_{CSFE} and N_{RES} are not large enough. In principle, this bias can be arbitrarily reduced by increasing the values of these parameters. However, we found out that a computationally efficient alternative to this approach is represented by running an additional (i.e., a fourth) step once that the CSFEC algorithm has been executed. In this final step, the estimation algorithm developed by *Ye* and *Aboutanios* in ref. [43, 44] 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 $\{\hat{A}_0[\hat{L}], \hat{A}_1[\hat{L}], ..., \hat{A}_{\hat{L}-1}[\hat{L}]\}$ generated by the CSFEC. The hybrid technique resulting from interconnecting the CSFEC algorithm with the above mentioned algorithm is dubbed *hybrid* CSFEC (HCSFEC). Finally, 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 CSFE need to be carefully adjusted in order to achieve a good accuracy in the estimation of the parameters of each new tone.

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

c) The threshold $\mathcal{T}_{\mathrm{CSFEC}}$ 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 $\mathcal{T}_{\mathrm{CSFEC}}$ may lead to miss weaker tones; on the other hand, a small value of this parameter may lead to the identification of nonexistent tones.

4.3.4 Comparison with other estimation methods

The CSFEC technique developed in the Paragraphs 4.3.1–4.3.3 is conceptually related to the *multiple tone estimators* developed by *Gough* [17], *Li* and *Stoica* [18], *Macleod* [11], *Ye* and *Aboutanios* [43, 44], *Serbes* and *Qaraqe* [45], and *Serbes* [46] (these algorithms are denoted CLEAN, RELAX, Alg-M, Alg-YA, CFH and Alg-S, respectively, in the following). In fact, all these algorithms are recursive and rely on a serial cancellation procedure since, within each recursion, they detect a single tone, estimate its parameters and subtract its contribution from the residual signal emerging from the previous iteration. Despite their similar structure, they exhibit various differences, that concern the three specific issues listed below.

Single frequency estimator - The main difference is represented by the algorithm they employ in the estimation of a single tone. In fact, on the one hand, the CLEAN and RELAX algorithms rely on the coarse estimate generated by the periodogram method for each detected tone and, eventually, exploit *zero-padding* to improve spectral resolution. On the other hand, the Alg-M, the Alg-YA, the Alg-S and the CFH algorithm compute the frequency residual by means of open-loop interpolation or iterative methods; the last methods refine the estimate of the frequency residual through multiple iterations. All the single tone estimators employed in these algorithms differ from the one used in the CSFEC and its hybrid version.

Use of a re-estimation procedure - In the CLEAN and RELAX algorithms and in Alg-M, once a new tone has been estimated, all the previously computed tones are re-estimated by subtracting the contribution of all the other tones; tone cancellation is accomplished in the time domain in the CLEAN and RELAX algorithms, whereas is carried out in the frequency domain in the Alg-M. The last approach is also adopted in our estimation algorithms. Finally, the CFH algorithm, the Alg-S and the Alg-YA accomplish re-estimation after computing a coarse estimate of all the parameters of the detected tones. However, the CFH algorithm executes this only once, whereas the Alg-S and the Alg-YA repeat it a given number of times.

Use of oversampling - The use of oversampling is proposed for the CLEAN, the RELAX, the Alg-S, and the CSFEC and the HCSFEC algorithms only.

4.3.5 Computational complexity

The complexity of the estimation algorithms developed in Paragraphs 4.3.1–4.3.2 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 overall computational cost of the CSFE and the SFE algorithms are summarised in Appendix I, where a detailed analysis of the contributions due to the different tasks accomplished by each of them is also provided. Our analysis leads to the conclusion that these costs are approximately of order $\mathcal{O}(M_{\mathrm{CSFE}})$ and $\mathcal{O}(M_{\mathrm{SFE}})$, respectively, with

$$M_{\mathrm{CSFE}} = N_0 \log_2 N_0 + K_{\mathrm{CSFE}}\, N_{\mathrm{CSFE}}\, I^2; \tag{4.101}$$

$$M_{\mathrm{SFE}} = N_0 \log_2 N_0 + K_{\mathrm{SFE}}\, N_{\mathrm{SFE}}\, I^2; \tag{4.102}$$

here, N_{CSFE} (N_{SFE}) represents the overall number iterations accomplished by the CSFE (SFE), $K_{\mathrm{CSFE}} = 1/2$ and $K_{\mathrm{SFE}} = 2/3$. Our computer simulations have evidenced that, in the scenarios we considered, a small value of I is required if the so called *barycentric interpolation* is employed (see ref. [61]). For this reason, the contribution of the second term appearing in the RHS of eq. (4.101) can be neglected, so that the order of the overall computational cost is well approximated by the first term, i.e. from the term that originates from FFT processing. Moreover, based on the last result, it is not difficult to show that the computational cost of the CSFEC is approximately of order $\mathcal{O}(M_{\mathrm{CSFEC}})$ and $\mathcal{O}(M_{\mathrm{SFEC}})$, with

$$M_{\mathrm{CSFEC}} = N_0 \log_2(N_0) + K_{\mathrm{CSFE}}\, L\, N_{\mathrm{CSFE}}\, I^2, \tag{4.103}$$

$$M_{\mathrm{SFEC}} = N_0 \log_2(N_0) + K_{\mathrm{SFE}}\, L\, N_{\mathrm{SFE}}\, I^2, \tag{4.104}$$

if no re-estimation is accomplished (see CSFEC-S3 in the description of the CSFEC algorithm) and the algorithm stop after detecting the last tone. Note that the first term appearing in the RHS of eq. (4.103) accounts for the initialization (and, in particular, for the computation of the vectors $\mathbf{X}_0$ (4.30) and $\{\mathbf{X}_k\}$ (4.43)), whereas the second term for the fact that, in the CSFEC (SFEC), the CSFE (SFE) is executed L times. It is also worth noting that the computational costs due to the evaluation of the estimated tones detected after the first one and to their frequency 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 (4.103) is replaced by L^2, since this task involves all the estimated tones. Despite this, the increase in the overall computational cost of the CSFEC with respect to the CSFE is limited since, as evidenced by our simulation results, the use of re-estimation allows these algorithms to achieve convergence with a smaller value of the parameter N_{CSFE}.

Finally, it is important to compare the computational cost of the CSFEC algorithm with that of the CLEAN, RELAX, Alg-M, Alg-YA, CFH and Alg-S techniques considered in the previous paragraph. Their order of complexity is listed in Table 4.1, from which the following considerations can be easily inferred:

1) The CLEAN and the RELAX algorithm are characterized by the same order, expressed by the complexity of a zero-padded FFT multiplied by L^2; the last factor is due to the fact that tone re-estimation is employed in both algorithms.

2) The Alg-M is characterized by the lowest computational cost; in fact, since it performs the cancellation of the detected tones directly in the frequency domain, tone estimation does not require the computation of additional FFTs. Moreover, since tone re-estimation is employed, the cost for the estimation of a single tone (i.e., the parameter K_M) is multiplied by L^2.

3) The order of complexity of the Alg-YA and the Alg-S, and that of the CFH algorithm depend on the fact that a FFT is computed for each tone; morover, an additional term equal to

Table 4.1: Order of the computational complexity of various estimation algorithms.

Algorithm	CLEAN / RELAX	Alg-M	Alg-YA / Alg-S	CFH
$\mathcal{O}(\cdot)$	$L^2 N_0 \log_2(N_0)$	$N_0 \log_2(N_0) + L^2 K_M$	$LN \log_2(N) + L\,Q\,N$	$LN \log_2(N)$

$L\,Q\,N$ is included in the order of the Alg-YA and the Alg-S, since these estimate all the tones Q times.

4) The order of the cost of the CSFEC algorithm is similar to that of the Alg-M; however, in this case, four FFTs are computed in the initialization phase and the cost for the estimation of each tone is multiplied by N_{CSFE} since tone refinement is performed within each recursion.

4.4 Numerical Results

In this section, the accuracy and robustness of the proposed algorithms is assessed on the basis of both synthetically generated data and experimental data acquired through an FMCW colocated MIMO radar.

4.4.1 Numerical results based on synthetically generated measurements

In this section, we compare, in terms of accuracy, convergence rate and failure probability, our single frequency estimator (CSFE) with the A&M [35], the QSE and HAQSE algorithms [50], and our multiple tone estimators (i.e., the CSFEC and HCSFEC algorithms) with the CFH algorithm [45], the Alg-YA [43, 44], the Alg-S [46] and the Alg-DP [62]. As far as the A&M algorithm is concerned, two versions of it are considered; such versions are denoted A&M#1 and A&M#2 in the following and correspond to the Alg-1 and Alg-2, respectively, described in [35].

Five different scenarios have been considered in our computer simulations. In all of them, $N = 512$ and $f_s = N$ Hz have been selected for the overall number of samples of the sequence $\{x_n\}$ and the sampling frequency, respectively; moreover, for any L, the phases of the L tones have been randomly selected over the interval $[0,\, 2\pi]$, each independently of all the other ones. The specific features of the simulated scenarios can be summarised as follows:

Scenario #1 (**S1**) - This is characterized by $L = 1$, i.e. by a single tone, having amplitude equal to one and whose normalised frequency is uniformly distributed over the interval $[8/N,\, 28/N]$.

Scenario #2 (**S2**) - This is characterized by $L = 2$, i.e. by a couple of tones, both having amplitude equal to one. Moreover, the normalised frequency F_0 of the first tone is uniformly distributed over the interval $[8/N,\, 28/N]$, whereas that of the second one is $F_1 = F_0 + 1.1/N$.

Scenario #3 (**S3**) - This is characterized by $L = 2$, i.e. by a couple of tones, both having amplitude equal to one. Moreover, the normalised frequency F_0 of the first tone is uniformly distributed over the interval $[8/N,\, 28/N]$, whereas that of the second one is $F_1 = F_0 + \Delta F_d$; here, $\Delta F_d \triangleq (1 + 0.05\,d)/N$ represents the normalised frequency spacing between the two tones and is controlled through the non negative parameter d ($d = 0,\, 1,\, ...,\, 10$ is assumed in the following).

Scenario #4 (**S4**) - This is characterized by $L \in \{2,\, 3,...,\, 10\}$, i.e. by a varying number of tones. For any L, the amplitude and the frequency of the k-th tone are given by

$$a_k \triangleq 10^{-k\,\Delta_a/10} \tag{4.105}$$

and $F_k \triangleq F_0 + 1.8\,k/N$, respectively, with $k = 0,\, 1,\, ...,\, L-1$; here, $\Delta_a = 2.5/3$ and F_0 is uniformly distributed over the interval $[8/N,\, 28/N]$.

Scenario #5 (**S5**) - This is characterized by $L = 10$ tones, whose amplitudes follow the law expressed by (4.105). Moreover, the normalised frequency of the k-th tone is $F_{k,m} \triangleq F_0 + k\,\Delta F_m$, with $F_0 = 8.3/N$ and $k = 0,\, 1,\, ...,\, 9$; here, $\Delta F_m = (1.5 + 0.2\,m)/N$ represents the normalised frequency spacing between adjacent tones and is controlled through the non negative parameter m ($m = 0,\, 1,\, ...,\, 5$ is assumed in the following).

It important to point out that: a) the interest in **S1** has been uniquely motivated by the need of comparing the performance of the CSFE with that achieved by the other single frequency estimators; b) the study of **S2** and **S3** has allowed us to assess how the considered multiple tone estimators perform in the presence of two close tones whose spacing is fixed and variable, respectively, whereas that of **S4** how their performance changes when L increases; c) the fifth scenario refers to the case in which the observed signal contains many closely spaced tones having different strengths, so that multiple tone estimators may fail detecting all of them and/or the quality of the the estimates of their parameters may turn out to be quite poor.

In our computer simulations, the following performance indices have been assessed for each of the analysed algorithms:

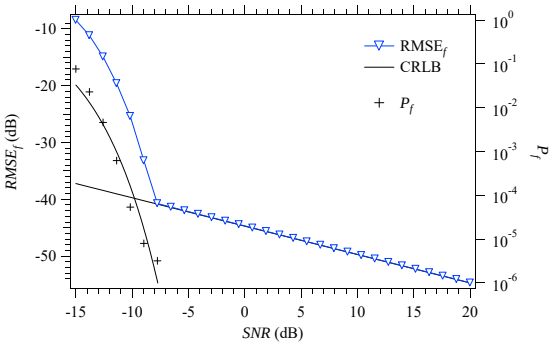
1. The *probability of failure* (P_f), i.e. the probability that the considered algorithm does not converge. In our simulation, a *failure event* is detected whenever the absolute value of the normalised frequency error characterizing the final frequency estimate exceeds the threshold $\Delta\varepsilon_F \triangleq 1/(2N_0)$, i.e. it is greater than half the size of the frequency bin characterizing FFT processing.

2. The *root mean square error* (RMSE) for the estimates of the normalised frequency, of the amplitude and of the phase of one or multiple tones; these are denoted RMSE_f, RMSE_A and RMSE_p, respectively.

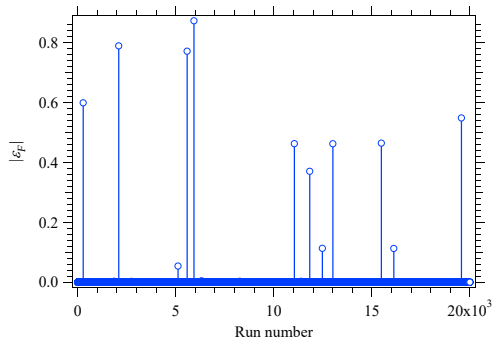
It is important to stress that the first parameter has been never assessed in the technical literature and that its value depends on the intensity of both the additive noise and the interference experienced by each newly detected tone (and due to uncancelled tones). Our interest in it can be motivated as follows. Each of the considered frequency estimation algorithms is highly nonlinear; for this reason, its behavior is characterized by a *threshold*, whose value depends on the specific scenario in which it is employed. In practice, if a frequency estimation algorithm operates above its threshold, failures are very rare events; consequently, the assessed root mean square errors are negligibly influenced by them, i.e., they account for the intensity of the errors observed after the convergence of the algorithm itself. On the contrary, if the algorithm operates below its threshold, a portion of its estimation errors (but not all of them) refers to situations in which it has not converged; when this happens, large frequency estimation errors (i.e., *outliers*) may be observed. In the last case, root mean square errors are not so meaningful since they account for two heterogeneous contributions. The significance of these considerations and our interest in the probability of failure can be fully understood by analyzing the simulation results shown in Figs. 4.1a and 4.1b, that refer to the A&M#1 algorithm operating in **S1**. In particular, the dependence of RMSE_f and P_f on the SNR for this algorithm is illustrated in Fig. 4.1a (where the *Cramer-Rao lower bound*, CRLB, is also shown for comparison), whereas a sample of the absolute value of the normalised frequency errors observed over 20000 consecutive runs at an $\mathrm{SNR} = -10$ dB is represented in Fig. 4.1b; in this case, the overall number of iterations $N_{\mathrm{A\&M}} = 2$ has been adopted for the considered algorithm and 10^7 simulation runs have been executed to generate the numerical results appearing in the first figure. As it can be easily inferred from Fig. 4.1a, the A&M#1 attains the CLRB above its threshold, which, in **S1**, is found at an $\mathrm{SNR} \cong -8$ dB and corresponds to a P_f approximately equal to 10^{-6}; below this SNR (that, in the case of a single tone, uniquely identifies the threshold of the algorithm), the estimated RMSE_f is significantly influenced by the presence of outliers, some of which are clearly visible in Fig. 4.1b. Our simulations have also evidenced that, if the all the failure events are ignored in the evaluation of the root mean square errors, the (negligible) gap between the RMSE_f of the A&M#1 and the CLRB does not change if the SNR drops below its threshold. Based on these considerations we have decided to:

a) Assess the probability of failure in all the considered scenarios.

b) Ignore the failure events in the evaluation of all the RMSEs.



(a) Root mean square error performance versus SNR; the CRLB is also shown for comparison.



(b) Normalised frequency error versus run number.

Figure 4.1: Some numerical results referring to the A&M#1 algorithm (first scenario).

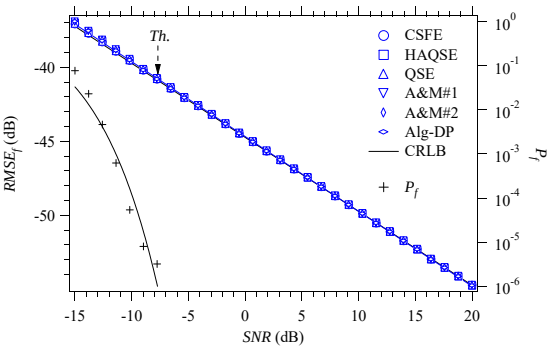
On the one hand, the first choice has allowed us to assess, in all the above mentioned scenarios, if each of the considered estimation algorithms is really operating above its threshold or below it; in doing so, we have always assumed that the threshold is conventionally identified by $P_f = 10^{-6}$, i.e. by one failure over 10^6 simulation runs on the average. On the other hand, the second choice has been made to verify if its accuracy is close to the CLRB when its failures are ignored. In the following, various simulation results are illustrated for the five scenarios described above; in all cases, each value of the considered performance indices have been evaluated by executing 10^6 runs.

The performance of six single tone estimators, namely the Alg-DP, the CSFE, and the A&M#1, A&M#2, QSE and HAQSE algorithms, has been assessed in **S1**. The following parameters have been selected for them: a) overall number of iterations $N_{\text{CSFE}} = 25$ and interpolation order[5] $I = 7$ for the CSFE; b) overall number of iterations $N_{\text{A\&M}} = 2$ for the two A&M algorithms; c) $N_{\text{QSE}} = 3$ for the QSE algorithm; d) $N_{\text{HAQSE}} = 2$ for the HAQSE algorithm; e) frequency displacement $f_d = 1/(10N)$ for the Alg-DP. In addition, the parameter q of the QSE and HAQSE algorithms has been evaluated on the basis of [50, eq. (39)] and the oversampling factor $M = 1$ has been selected for all the considered algorithms. Some numerical results referring to **S1** are illustrated in Figs. 4.2a and 4.2b (in these figures and all the following ones, numerical results are represented by markers, whereas lines are drawn to ease reading; moreover, the abbreviation 'Th.' is employed for the *threshold*, whose position is always indicated by a black arrow). More specifically, the dependence of RMSE$_f$, and RMSE$_A$ and RMSE$_p$ on the SNR is represented in Figs. 4.2a and 4.2b, respectively, for all the considered algorithms. In both figures, the dependence of the probability of failure on the SNR (which is approximately the same for all the considered algorithms) is also shown and the SNR range $[-15, 20]$ dB is considered. From these results it is easily inferred that: 1) all the considered algorithms exhibit a similar dependence of RMSE$_f$, RMSE$_A$ and RMSE$_p$ on the

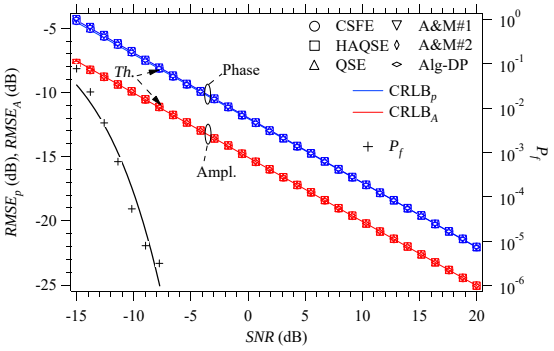
$$\text{SNR} \triangleq 1/\sigma^2 \tag{4.106}$$

and their accuracy approaches the CLRB [63] in frequency, amplitude and phase estimation for SNR$\in[-8, 20]$ dB; 2) they are characterized by a similar probability of failure and, therefore, by a similar SNR threshold; 3) they attain the CRLB in the estimation of frequency, amplitude and phase even below their threshold if failure events are ignored. Our simulation results have also evidenced that the CSFE is characterized by a lower convergence rate than the other algorithms. As far as the last point is concerned, it is worth mentioning that: 1) the A&M#1, A&M#2, QSE and HAQSE algorithms usually require 2-4 iterations to achieve convergence, whereas the overall number of iterations required by the CSFE is 3-4 times larger; 2) the Alg-DP, A&M#1, A&M#2, QSE and HAQSE algorithms are characterised by similar computation times, whereas the CSFE is about 10 times slower.

[5] In all our simulations, the *barycentric interpolation* described in [61] has been always used.



(a) Root mean square error performance achieved in frequency estimation.



(b) Root mean square error performance achieved in phase and amplitude estimation.

Figure 4.2: Root mean square error performance versus SNR (first scenario). The Alg-DP and the CSFE, HAQSE, QSE, A&M#1 and A&M#2 algorithms are considered. The CRLB is also shown for comparison.

Some numerical results referring to **S2** are illustrated in Figs. 4.3, 4.4a and 4.4b. More specifically, the dependence of P_f on the SNR is illustrated in Fig. 4.3, whereas the dependence of RMSE_f, and of RMSE_A and RMSE_p on the SNR[6] is represented in Figs. 4.4a and 4.4b, respectively, for all the considered algorithms; once again, the SNR range $[-15, 20]$ dB is considered. The accuracy of six multiple tone estimators, namely the Alg-YA, the Alg-S, the Alg-DP, and the CSFEC, HCSFEC and CFH algorithms has been assessed in this case; moreover, the following parameters have been selected for these algorithms in **S2** and in the remaining three scenarios: a) overall number of iterations $N_{\mathrm{CSFE}} = 15$, number of re-estimations $N_{\mathrm{RES}} = 5$ and interpolation order $I = 7$ for the CSFEC; b) $N_{\mathrm{CSFE}} = 5$, $N_{\mathrm{RES}} = 1$, $I = 7$ and number of Alg-YA iterations $N_{\mathrm{YA}} = 5$ for the HCSFEC algorithm; c) the same parameters as the HAQSE and Alg-DP in **S1** for the CFH algorithm and the Alg-DP, respectively; d) overall number of iterations $Q = 2$ for the Alg-YA. Moreover, the number of re-estimations carried out by the Alg-S has been evaluated on the basis of [46, eq. (33)]. Our results show that: 1) the CSFEC and HCSFEC techniques are more robust than all the other estimators, since they achieve a substantially lower probability of failure for an SNR > -10 dB; 2) the thresholds of the CSFEC and HCSFEC algorithms are about -3 dB and are substantially lower than that of Alg-YA, which is found at about 3 dB; 3) the CFH, the Alg-S and the Alg-DP exhibit a $P_f > 10^{-2}$ for all the values of SNR (therefore, their RMSE performance is ignored in the following); 4) the Alg-YA is always outperformed by the CSFEC and HCSFEC algorithms both in frequency, amplitude and phase estimation; 5) the trend in the accuracy of the CSFEC and HCSFEC algorithms diverges from that of the CRLB for an SNR > -6 dB because of the small bias introduced by the serial cancellation procedure on which they are based; 6) the CSFEC algorithm is slightly outperformed by the HCSFEC algorithm for an SNR > -6 dB; 7) the trend in the accuracy of the Alg-YA below its threshold diverges from that of the CRLB in frequency, amplitude and phase estimation for all the SNR values, whereas that that of the CSFEC and HCSFEC algorithms is only 1 dB far from the corresponding CRLB. It is also worth mentioning that the CFH algorithm, the Alg-YA and the Alg-DP require similar computation times, whereas the HCSFEC algorithm and the Alg-S (the CSFEC algorithm) are about 2 (16) times slower.

Let us focus now on **S3**. In this scenario, all the performance indices have been evaluated for different values of the normalised tone spacing ($\Delta F \cdot N$). Some numerical results referring to this scenario are illustrated in Figs. 4.5, 4.6a and 4.6b. More specifically, the dependence of P_f on the tone spacing is illustrated in Fig. 4.5, whereas the dependence of RMSE_f, and RMSE_A and RMSE_p on the tone spacing is represented in Figs. 4.6a and 4.6b, respectively, for all the considered algorithms; an SNR $= 10$ dB is assumed. From our results it can be inferred that: 1) the CSFEC and HCSFEC algorithms are substantially more robust than all the other estimators,

[6]Since the amplitudes of both tones are equal to one, the SNR is still computed on the basis of (4.106).

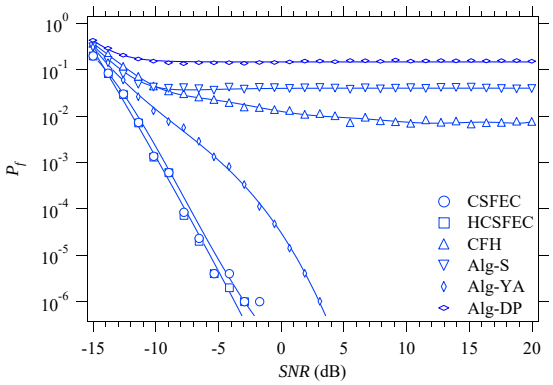
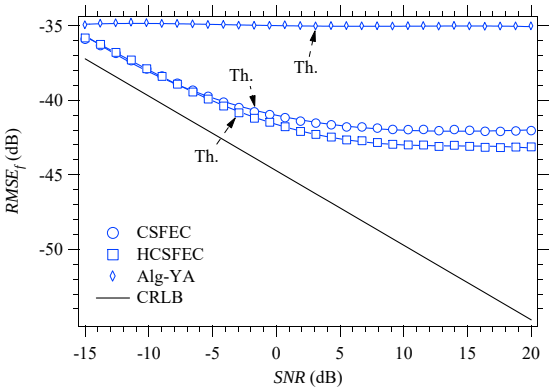
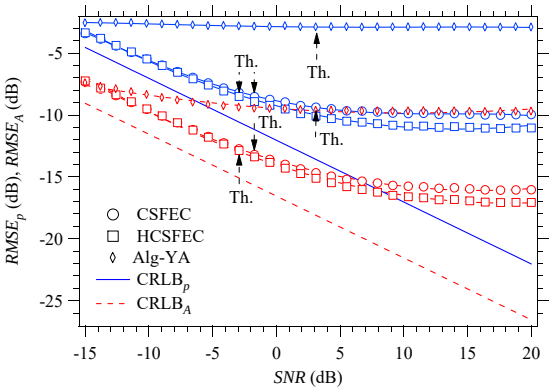


Figure 4.3: Probability of failure versus SNR (second scenario). The Alg-S, the Alg-YA, the Alg-DP, and the CSFEC, HCSFEC and CFH algorithms are considered.



(a) Root mean square error performance achieved in frequency estimation.



(b) Root mean square error performance achieved in phase and amplitude estimation.

Figure 4.4: Root mean square error performance versus SNR (second scenario). The Alg-YA, and the CSFEC and HCSFEC algorithms are considered. The CRLB is also shown for comparison.

since no failure event has been detected for the considered tone spacings over all the simulation runs; 2) the CFH algorithm exhibits a $P_f < 10^{-6}$ for a tone spacing greater than 1.2, whereas the Alg-YA a $P_f > 10^{-6}$ for values of tone spacing between 1.2 and 1.4; 3) the Alg-S and the Alg-DP are characterized by a $P_f > 10^{-6}$ for any value of tone spacing (for this reason, they are ignored in the following); 5) the Alg-YA and the CFH algorithm are outperformed by the CSFEC and HCSFEC algorithms, for all the considered tone spacings, in frequency, amplitude and phase estimation; 6) the CSFEC algorithm is slightly outperformed by the HCSFEC algorithm in frequency, amplitude and phase estimation; 7) the CSFEC and HCSFEC algorithms attain the CRLB for frequency and phase (amplitude) estimation for a tone spacing equal to 1.5 (greater than 1.3).

As already mentioned above, our simulations for **S4** have allowed us to assess how the performance of the considered algorithms is influenced by the overall number of tones. Some results referring to this scenario are shown in Fig. 4.7, which shows the dependence of P_f on L; $M = 1$ and an SNR = 10 dB for the strongest tone have been assumed. From this figure it is easily inferred that: 1) the Alg-DP (Alg-S) exhibits a $P_f > 10^{-2}$ for $L \geq 3$ ($L \geq 4$); 2) the CFH algorithm (Alg-YA) exhibits a $P_f > 10^{-4}$ ($P_f > 10^{-3}$) for $L \geq 5$; 3) the CSFEC and HCSFEC algorithms are substantially more robust than all the other algorithms since are characterized by a $P_f < 10^{-6}$ for $L \leq 8$; 4) the HCSFEC is slightly outperformed by the CSFEC for $L = 10$. Once again, the price to be paid for a lower probability of failure is represented by a larger computational effort. For instance, if $L = 6$, the computation time required by the CSFEC (HCSFEC) algorithm is about 50 (4) times larger than that characterizing the CFH algorithm.

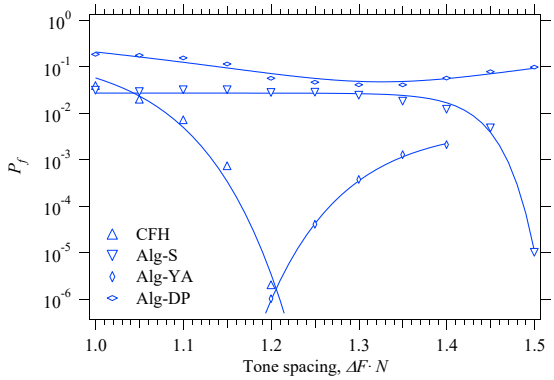
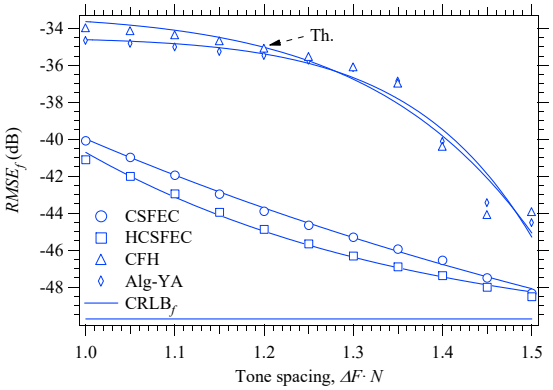
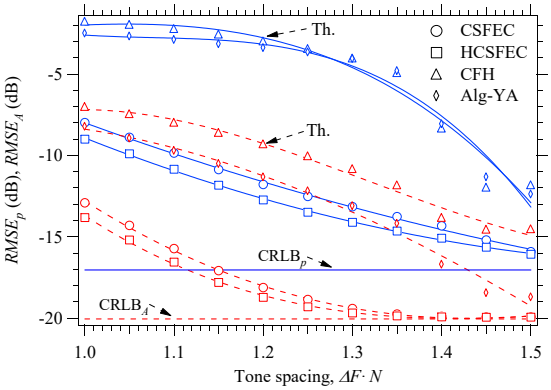


Figure 4.5: Probability of failure versus normalised tone spacing (third scenario). The Alg-S, the Alg-YA, the Alg-DP, and the CSFEC, HCSFEC and CFH algorithms are considered.



(a) Root mean square error performance achieved in frequency estimation.



(b) Root mean square error performance achieved in phase and amplitude estimation

Figure 4.6: Root mean square error performance versus tone separation (third scenario). The Alg-YA, and the CSFEC, HCSFEC and CFH algorithms are considered. The CRLB is also shown for comparison.

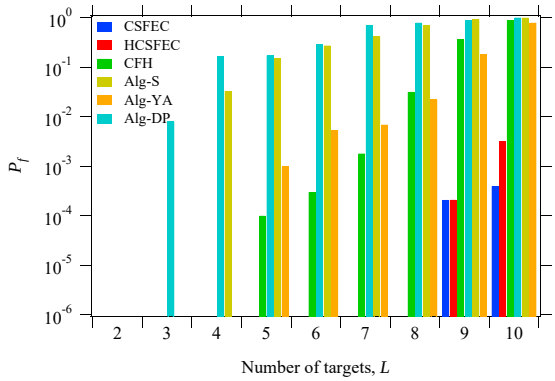


Figure 4.7: Probability of failure versus overall number of tones (fourth scenario). The Alg-S, the Alg-YA, the Alg-DP, and the CSFEC, HCSFEC and CFH algorithms are considered.

Table 4.2: Signal-to-noise ratio characterizing each tone in the fifth scenario and corresponding CRLB evaluated for the estimation of its frequency.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
SNR (dB)	10	8.3	6.7	5	3.3	1.7	0	-1.7	-3.3	-5
CRLB (dB)	-49.7	-48.9	-48	-47.2	-46.4	-45.6	-44.7	-43.9	-43	-42.2

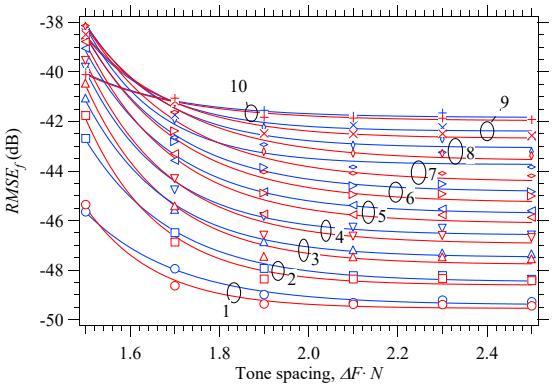


Figure 4.8: Root mean square error performance achieved in frequency estimation versus tone separation (fifth scenario). The CSFEC (blue curves) and HCSFEC (red curves) algorithms are considered. Different tones are identified by distinct markers and numbers.

Finally, let us focus on **S5**. In this case, the performance index RMSE$_f$ has been evaluated for the normalised frequencies of all the tones; moreover, six different values of the tone spacing ($\Delta F \cdot N$) have been considered. Our numerical results refer to the CSFEC and the HCSFEC algorithms only since we found that these are the only algorithms operating above their thresholds for spacings greater than $2/N$. The CFH algorithm, the Alg-S, the Alg-YA and the Alg-DP, instead, are characterized by an high probability of failure (more specifically, $P_f > 10^{-2}$ in all the conditions) and, as a matter of fact, *are unable to detect all the tones and/or to accurately estimate their parameters*. The dependence of RMSE$_f$, obtained under the assumption that SNR = 10 dB for the strongest tone (i.e., for the tone having the smallest frequency; see (4.105)), on ($\Delta F \cdot N$) is illustrated in Fig. 4.8, whereas the values of the SNR and the CRLB referring to each of the ten tones are listed in Table 4.2. Our results lead easily to the following conclusions: 1) the RMSE$_f$ characterizing the HCSFEC algorithm is lower than that achieved by the CSFEC algorithm for each tone and for all the values of tone spacing, since the last algorithm suffers from a larger (even if really limited) bias; 2) weaker tones are characterized by a larger RMSE$_f$ (but also by a larger value of the CRLB, since their SNR is lower) for any ΔF; 3) the RMSE$_f$ characterizing each tone reaches a floor for $\Delta F > 2/N$; 4) the floor appearing in the frequency estimation accuracy of each tone is very close to the CLRB evaluated for that tone.

4.4.2 Numerical results based on experimental measurements

The accuracy of the proposed CSFEC and HCSFEC algorithms has been also assessed on the basis of a real dataset acquired through a commercial MIMO radar. In particular, a measurement campaign has been accomplished in the building of our institution to acquire a data set through a colocated FMCW MIMO radar operating in the E-band. The employed device is the TIDEP-01012 cascade mmWave radar; it is manufactured by *Texas Instrument Inc.* [64] and classified as a *long range radar* (LLR). Its main parameters are: a) chirp rate $\mu = 6.5 \cdot 10^{13}$ Hz/s; b) bandwidth $B = 4.1$ GHz; c) central frequency $f_0 = 77$ GHz; d) sampling frequency $f_s = 8$ MHz; e) number of samples per chirp $N = 512$. Moreover, it is endowed with a planar array made of $N_T = 12$ TX antennas and $N_R = 16$ RX antennas; each of its antennas consists of an array of four patch antennas. In principle, $12 \cdot 16 = 192$ virtual antennas are available in this case;

Figure 4.9: Experimental set-up developed for our acquisitions. The employed radar device is mounted on a wooden bar; the targets are small coins placed on a polystyrene plate.

however, only 86 horizontal-aligned and equally-spaced virtual antennas, forming a virtual ULA with $d_{\mathrm{H}} = \lambda/4$, have been exploited in our work. This choice allows us to achieve the range resolution $\Delta R = c/(2B) = 3.6$ cm and the azimuthal resolution $\Delta\theta = 1.35°$.

All the measurements have been acquired in a large empty room (whose width, depth and height are 10 m, 8 m and 2.5 m, respectively). The radar device has been mounted on an horizontal wooden bar and has been lifted by a tripod at an height of roughly 1.60 m from ground (see Fig. 4.9).

In our experimental set-up, a pico-flexx camera manufactured by PMD Technologies Inc. [65] has been employed as a *reference sensor*; this device is based on a near-infrared *vertical cavity surface emitting laser* (VCSEL), and is able to provide a depth map or, equivalently, a three-dimensional point-cloud of a small region of the observed environment (its maximum depth is equal to 4 m, whereas its *field of view* is $62° \times 45°$).

As far as the acquired measurements are concerned, it is important to point out that: a) all the target ranges have been estimated with respect to the central virtual channel of the ULA; b) the exact target positions have been acquired with respect to the centre of the pico-flexx camera. Therefore, in comparing these positions with their estimates computed on the basis of the radar measurements, the distance $\Delta_{\mathrm{FP}} = 33$ cm between the FMCW radar and the camera was always kept into account; c) all our measurements have been processed in the MATLAB environment (running on a desktop computer equipped with an i7 processor).

The numerical results illustrated in this paragraph refer to two static scenarios. The first scenario is denoted **S6** and is characterized by the presence of an overall number of targets ranging from 1 to 9 (so that $1 \leq L \leq 9$). As shown in Fig. 4.9, the targets, placed on a polystyrene plate, are represented by small coins, each having a diameter equal to 2 cm (note that such coins are grouped in three different clusters, called C1, C2 and C3); their exact positions are listed in Table 4.3 (the data referring to the i-th target are collected in the column identified by T_i, with $i = 1$, 2, ..., 9). In the second scenario (denoted **S7**), instead, five distinct experiments, characterized by a different number of targets, have been conducted. The results obtained in this scenario have allowed us to assess how the performance of our estimation algorithms is influenced by the overall number of targets (i.e., by L).

In processing all the acquired measurements, prior knowledge of L has been always assumed and an oversampling factor M equal to 1 has been adopted for all the algorithms (namely, the CSFEC, HCSFEC and CFH algorithms and the Alg-YA, the Alg-S and the Alg-DP); moreover, $M = 4$ has been also considered for the CSFEC and HCSFEC algorithms. The values of all the other parameters characterizing the considered estimation algorithms have been selected in the same way as the scenarios **S2**, **S3**, **S4** and **S5** described in the previous paragraphs.

In analysing the data acquired in **S6** and **S7**, the accuracy of the *range* estimates evaluated for multiple targets has been assessed by evaluating the RMSE

$$\bar{\varepsilon}_R \triangleq \sqrt{\frac{1}{N_m} \sum_{l=0}^{N_m-1} \left[R_l - \hat{R}_l \right]^2} \tag{4.107}$$

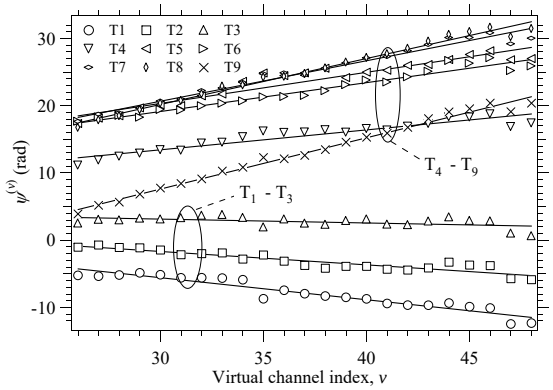


Figure 4.10: Unwrapped phase of the complex gain associated with a given target versus the index of the virtual channel of the employed ULA; scenario **S6** is considered.

and the peak error

$$\hat{\varepsilon}_R \triangleq \max_l \left| R_l - \hat{R}_l \right|, \tag{4.108}$$

where N_m represents the overall number of available measurements. As far as the estimation of the complex amplitude characterizing the echo of a given target on a specific virtual channel is concerned, it is important to point out that the RCS of the targets detected in our experiment was unknown. For this reason, our analysis of the complex gains estimated over the 86 channels of the available virtual ULA and associated with each target has concerned only their (unwrapped) phase. The phases estimated by the CSFEC algorithm over the above mentioned ULA and associated with the nine targets of **S6** are shown in Fig. 4.10. Since the distance d_{H} between adjacent virtual channels is constant, the estimated phases exhibit a linear dependence on the index of the virtual channel (see (4.10) and (4.11)). Moreover, if a linear fitting is drawn for these data, it should be expected that the slope of the resulting lines is proportional to $\sin(\theta)$, where θ is the azimuth of the considered target (see (4.16)); this is confirmed by the results shown in Fig. 4.10, where the three (six) lines associated with the targets T_1–T_3 (T_4–T_9) have a negative (positive) slope[7], as should be expected on the basis of Table 4.3. To assess the quality of the estimated phases, their RMSE $\bar{\varepsilon}_\psi$ has been evaluated on the basis of a formula similar to (4.107) (in this case, N_m represents the overall number of virtual channels for which the estimates of the phases associated with a given target have been computed); in doing so, the linear fitting of the phases estimated over the whole ULA has been taken as a reference with respect to which phase errors have been computed.

The estimates of the target range generated by the all the considered estimation algorithms for each of the targets of **S6** are listed in Table 4.3; in the same table, the value of the phase RMSE $\bar{\varepsilon}_\psi$ computed for each target is also provided. The numerical results collected in this table have been also processed to compute: a) the errors $\bar{\varepsilon}_R$ (4.107) and $\hat{\varepsilon}_R$ (4.108); b) the average errors $\bar{\varepsilon}_{m,R}$ and $\hat{\varepsilon}_{m,R}$ (these represent the average of $\bar{\varepsilon}_R$ and $\hat{\varepsilon}_R$, respectively, over the whole virtual ULA[8]); c) the average of $\bar{\varepsilon}_\psi$, denoted $\bar{\varepsilon}_{m,\psi}$ (this represents the average of the N_m values available for $\bar{\varepsilon}_\psi$); d) the *computation time* (CT). The values of all these performance indices are summarized in Table 4.4 for the six considered estimation algorithms.

From the last results and those listed in Table 4.3 the following conclusions can be drawn:

1. Both the CSFEC and HCSFEC algorithms are able to generate accurate estimates of the range and the amplitude of all the given targets for both the considered values of the oversampling factor M.

[7]The phase trajectories shown in Fig. 4.10 refer to a small portion of the available virtual ULA for better readability.

[8]In this case, the range estimate $\hat{R}_l$ appearing in (4.107) and (4.108) is computed for each channel of the given virtual ULA, while the true value R_l is kept constant along this array.

Table 4.3: Exact position (range and azimuth) of each of the nine targets characterizing the sixth scenario. The range estimates computed by the Alg-YA, the Alg-S, the Alg-DP, and the CFSEC, HCSFEC and CFH algorithms are listed. Moreover, the phase RMSE computed for each target is provided.

Method		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Exact	R (m)	1.755	1.780	1.816	1.950	2.010	2.057	2.195	2.254	2.280
	θ (°)	-5.5	-4.1	-2.7	3.1	4.6	6.0	9.1	10.3	11.5
CSFEC ($M=4$)	$\hat{R}$ (m)	1.801	1.834	1.853	2.013	2.064	2.094	2.193	2.248	2.299
	$\bar{\varepsilon}_\psi$ (rad)	1.228	1.310	2.161	3.066	1.836	3.019	3.863	2.026	4.329
HCSFEC ($M=4$)	R (m)	1.801	1.833	1.853	2.012	2.062	2.094	2.193	2.250	2.290
	$\bar{\varepsilon}_\psi$ (rad)	1.162	1.336	2.064	3.027	1.836	3.021	3.873	2.053	3.262
CSFEC ($M=1$)	$\hat{R}$ (m)	1.809	1.840	1.861	2.007	2.053	2.094	2.196	2.249	2.298
	$\bar{\varepsilon}_\psi$ (rad)	1.879	2.649	2.856	2.456	2.131	4.362	3.444	3.472	3.496
HCSFEC ($M=1$)	R (m)	1.808	1.839	1.860	2.006	2.058	2.093	2.193	2.249	2.301
	$\bar{\varepsilon}_\psi$ (rad)	2.960	2.710	1.795	1.864	3.070	4.299	4.430	1.976	4.234
CFH	$\hat{R}$ (m)	1.795	1.825	1.854	1.875	2.008	2.055	2.092	2.250	2.301
	$\bar{\varepsilon}_\psi$ (rad)	1.413	1.293	2.126	2.849	2.555	2.015	3.991	3.321	2.119
Alg-YA	$\hat{R}$ (m)	1.787	1.854	1.875	1.875	2.012	2.067	2.090	2.190	2.289
	$\bar{\varepsilon}_\psi$ (rad)	1.294	4.020	2.891	2.364	2.123	4.243	3.930	2.152	2.485
Alg-S	$\hat{R}$ (m)	1.809	1.809	1.809	1.809	1.809	1.809	1.822	1.837	2.204
	$\bar{\varepsilon}_\psi$ (rad)	0.965	1.311	2.097	2.966	2.428	3.088	4.037	4.193	3.200
Alg-DP	$\hat{R}$ (m)	1.843	1.854	1.855	1.856	1.856	1.857	1.865	1.867	2.204
	$\bar{\varepsilon}_\psi$ (rad)	3.161	2.635	3.360	3.260	4.691	3.243	5.572	3.207	6.316

Table 4.4: Range RMSE $\bar{\varepsilon}_R$ (and its average $\bar{\varepsilon}_{m,R}$), phase RMSE $\bar{\varepsilon}_{m,\psi}$, range peak error $\hat{\varepsilon}_R$ (and its average $\hat{\varepsilon}_{m,R}$) and CT evaluated for all the considered estimation algorithms; the sixth scenario is considered.

Method	$\bar{\varepsilon}_R$ (m)	$\bar{\varepsilon}_{m,R}$ (m)	$\hat{\varepsilon}_R$ (m)	$\hat{\varepsilon}_{m,R}$ (m)	$\bar{\varepsilon}_{m,\psi}$ (rad)	CT (sec)
CSFEC ($M=4$)	0.04	0.05	0.06	0.11	2.80	0.030
HCSFEC ($M=4$)	0.04	0.05	0.06	0.11	2.80	0.030
CSFEC ($M=1$)	0.04	0.05	0.06	0.12	2.90	0.020
HCSFEC ($M=1$)	0.04	0.05	0.06	0.12	3.0	0.020
CFH	0.05	0.07	0.10	0.12	2.60	0.010
Alg-YA	0.06	0.07	0.10	0.14	2.60	0.040
Alg-S	0.30	0.11	0.40	0.19	3.50	0.050
Alg-DP	0.20	0.15	0.40	0.61	2.90	0.005

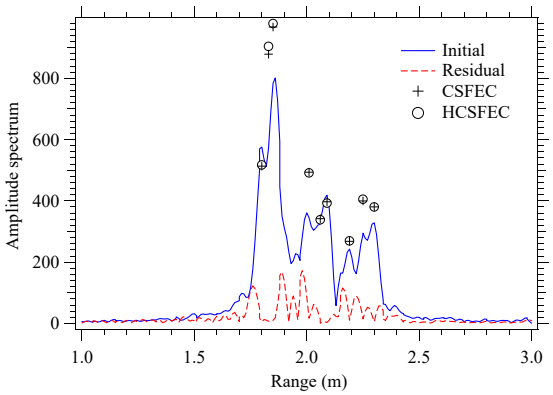


Figure 4.11: Representation of the initial amplitude spectrum of the signal observed on the central virtual channel (blue line) and of the final *residual* amplitude spectrum generated by the CFSEC algorithm (red line). The range and the amplitude characterizing the nine targets of the sixth scenario and estimated by the CSFEC (HCSFEC) algorithm are indicated by black crosses (circles); $M = 4$ is assumed.

2. The values of range RMSE $\bar{\varepsilon}_R$ and peak error $\hat{\varepsilon}_R$ characterizing the CSFEC and HCSFEC algorithms are lower than those provided by all the other algorithms. In particular, even if the values of the range RMSE obtained by the CFH and Alg-YA are quite close to those of the CSFEC and HCSFEC algorithms, the peak errors of the first two algorithms are higher than the minimum range resolution allowed by the employed radar device and are two times larger than those characterizing the last two algorithms. Moreover, the range RMSE and peak errors of the CSFEC and HCSFEC algorithms are much lower those of the Alg-S and the Alg-DP. This result is due to the fact that the ranges estimated by the Alg-S and the Alg-DP are far from their true values, as it can be easily inferred from Table 4.3.

3. The values of the range average errors $\bar{\varepsilon}_{m,R}$ and $\hat{\varepsilon}_{m,R}$ exhibit the same trend as $\bar{\varepsilon}_R$ and $\hat{\varepsilon}_R$, respectively, but are higher than the range RMSE and range peak error, respectively, obtained for the central channel of our virtual ULA. Note also that the large values of $\hat{\varepsilon}_{m,R}$ and $\hat{\varepsilon}_{m,R}$ found for Alg-S and Alg-DP are due to the fact these algorithms does not converge in the considered scenario.

4. The values of the phase average RMSE $\bar{\varepsilon}_{m,\psi}$ evaluated for the CSFEC and HCSFEC algorithms are comparable with (much lower that) those obtained for the CFH and the Alg-YA (the Alg-S and the Alg-DP).

5. The CTs are in the order of few milliseconds for all the algorithms; the best trade-off between accuracy and CT is achieved by the CSFEC and HCSFEC algorithms with an oversampling factor equal to 1. Note also that the fastest algorithm is represented by the Alg-DP, but its estimation accuracy is significantly worse than that provided by the CSFEC and HCSFEC algorithms.

It is also important to point out that the robustness of the CSFEC and HCSFEC algorithms is related to the accuracy of the estimation and cancellation procedure they accomplish. This is exemplified by Fig. 4.11, where the *initial amplitude spectrum* of the signal received on the central virtual channel in the sixth scenario and *its residual*, resulting from the cancellation of the spectral contributions due to the nine targets, are shown. Moreover, the range and the amplitude estimated by the CSFEC and HCSFEC algorithms for each target are shown ($M = 4$ is assumed).

Let us focus on **S7** now. The exact positions of the targets characterizing our five experiments are listed in Table 4.5. In this case, L ranges from five to ten; note also that T_l denotes the l-th target (with $l = 1, 2, ..., 10$).

The values of the range RMSE $\bar{\varepsilon}_R$ and peak error $\hat{\varepsilon}_R$ obtained for all the considered estimation algorithms are listed in table 4.6 and table 4.7, respectively. The values of the range average

Table 4.5: Exact positions of the targets characterizing the seventh scenario; five different experiments are considered.

Exp.	Exact #1 (m)	Exact #2 (m)	Exact #3 (m)	Exact #4 (m)	Exact #5 (m)
T_1	1.775	1.832	1.798	1.755	1.755
T_2	1.876	1.920	1.832	1.780	1.780
T_3	2.003	1.930	1.920	1.816	1.816
T_4	2.145	1.981	1.930	1.950	1.950
T_5	2.228	2.055	1.981	2.010	2.010
T_6		2.100	2.055	2.057	2.057
T_7		2.220	2.100	2.195	2.195
T_8			2.220	2.254	2.254
T_9				2.280	2.280
T_{10}					2.310

errors $\bar{\varepsilon}_{m,R}$ and $\hat{\varepsilon}_{m,R}$ evaluated by averaging the RMSE $\bar{\varepsilon}_R$ and the peak error $\hat{\varepsilon}_R$ obtained for the central channel in all our experiments, and the average CT are listed in Table 4.8. From these results it is easily inferred that:

1. The ability of the CSFEC and HCSFEC algorithms in estimating the range of multiple targets becomes evident when the overall number of targets L increases. In fact, as evidenced by the numerical results collected in Tables 4.6 and 4.7, the CSFEC and HCSFEC algorithms achieve the lowest RMSE and peak errors in all the experiments for both values of M.

2. All the considered algorithms achieve comparable accuracies in the first experiment, i.e. when $L = 5$. In the experiments characterized by $L > 5$, the CSFEC and HCSFEC algorithms achieve the lowest peak errors for both values of M. Moreover, the errors significantly increase for Alg-S and Alg-DP when $L > 8$ (i.e. in the experiments 4 and 5).

3. The average CT achieved by the CSFEC and HCSFEC algorithms is comparable with those of the other algorithms (for example, the average CT of the CSFEC and HCSFEC algorithms for $M = 1$ is equal to that of the CFH algorithm); however, the estimation accuracy they achieve is higher.

4.5 Conclusions

In this chapter, a novel algorithm for detecting and estimating a single tone has been derived; moreover, its has been shown how it can be exploited to estimate multiple tones through a serial cancellation procedure. The accuracy and robustness of the devised single tone and multiple tone estimators have been assessed by means of extensive computer simulations involving both synthetically generated data and the measurements acquired through a commercial colocated MIMO FMCW radar. Our results have evidenced that our multiple tone estimators outperform all the other related estimators available in the technical literature in terms of probability of convergence and accuracy when they operate in the presence of multiple closely-spaced tones. For this reason, if they are employed in FMCW radar systems, they allow to achieve excellent range resolution and to acquire DOA information from the phase estimates computed on an antenna array. Future work concerns the application of the developed algorithms to various fields.

Table 4.6: Range RMSE $\bar{\varepsilon}_R$ evaluated for all our experiments; the seventh scenario is considered.

Method	Exp. 1 (m)	Exp. 2 (m)	Exp. 3 (m)	Exp. 4 (m)	Exp. 5 (m)
CSFEC ($M = 4$)	0.06	0.03	0.04	0.04	0.06
HCSFEC ($M = 4$)	0.06	0.03	0.04	0.04	0.06
CSFEC ($M = 1$)	0.06	0.05	0.04	0.04	0.06
HCSFEC ($M = 1$)	0.06	0.05	0.04	0.04	0.06
CFH	0.06	0.05	0.04	0.05	0.07
Alg-YA	0.07	0.05	0.04	0.06	0.08
Alg-S	0.07	0.07	0.07	0.30	0.14
Alg-DP	0.07	0.12	0.07	0.20	0.22

Table 4.7: Range peak error $\hat{\varepsilon}_R$ evaluated for all our experiments; the seventh scenario is considered.

Method	Exp. 1 (m)	Exp. 2 (m)	Exp. 3 (m)	Exp. 4 (m)	Exp. 5 (m)
CSFEC ($M = 4$)	0.12	0.06	0.06	0.06	0.13
HCSFEC ($M = 4$)	0.12	0.06	0.06	0.06	0.13
CSFEC ($M = 1$)	0.12	0.10	0.05	0.06	0.13
HCSFEC ($M = 1$)	0.12	0.10	0.05	0.06	0.13
CFH	0.12	0.13	0.08	0.10	0.15
Alg-YA	0.12	0.15	0.05	0.10	0.21
Alg-S	0.12	0.13	0.14	0.40	0.24
Alg-DP	0.12	0.24	0.10	0.40	0.37

Table 4.8: Average RMSE $\bar{\varepsilon}_{m,R}$, peak error $\hat{\varepsilon}_{m,R}$, and CT evaluated in the seventh scenario.

Method	$\bar{\varepsilon}_{m,R}$ (m)	$\hat{\varepsilon}_{m,R}$ (m)	CT (sec)
CSFEC ($M = 4$)	0.04	0.08	0.020
HCSFEC ($M = 4$)	0.04	0.08	0.020
CSFEC ($M = 1$)	0.05	0.09	0.010
HCSFEC ($M = 1$)	0.05	0.09	0.010
CFH	0.06	0.11	0.010
Alg-YA	0.06	0.13	0.060
Alg-S	0.13	0.21	0.060
Alg-DP	0.14	0.24	0.005

Five

Novel Deterministic Detection and Estimation Algorithms for Colocated Multiple-Input Multiple-Output Radars

In this chapter, the problem of detecting multiple targets and estimating their spatial coordinates (namely, their *range* and the *direction of arrival* of their electromagnetic echoes) in a *colocated multiple-input multiple-output* radar system operating in a static or slowly changing two-dimensional or three-dimensional propagation scenario is investigated. Various solutions, collectively called *range & angle serial cancellation algorithms*, are developed for both frequency modulated continuous wave radars and stepped frequency continuous wave radars. Moreover, specific technical problems experienced in their implementation are discussed. Finally, the accuracy achieved by these algorithms in the presence of multiple targets is assessed on the basis of both synthetically generated data and of the measurements acquired through three different multiple-input multiple-output radars and is compared with that provided by other methods based on multidimensional Fourier analysis and multiple signal classification.

Part of this Chapter has been published in [66] and is patent pending.

5.1 Introduction

In the last years, the advances in millimeter-wave semiconductor technology and the development of novel signal processing techniques have lead the way to the exploitation of *multiple-input multiple-output* (MIMO) radar systems in a number of fields. These systems can be divided in *statistical* MIMO radars [67, 68] and *colocated* MIMO radars [69] on the basis of the distance between their transmit array and their receive array; in the first case, transmit and receive antennas are widely separated, whereas, in the second one, they are closely spaced and, in particular, they are usually placed on the same shield. In this chapter, we focus on colocated MIMO radars only; such systems play an important role in a number of applications, because of their limited cost, their small size and their ability to detect the presence of multiple targets. The performance achieved by any colocated MIMO radar system depends not only on some important characteristics of its hardware (e.g., the operating frequency, the number of transmit and receive antennas, the configuration of the transmit and receive arrays, etc.), but also on the techniques employed in the generation of its radiated waveforms and in the processing of the measurements acquired through its receive array. As far as the last issue is concerned, it is worth stressing that *optimal* (i.e., *maximum likelihood*, ML) 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 [2]. 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 ref. [70] for *frequency modulated continuous wave* (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-elevation domain or a range-azimuth domain in *three-dimensional* (3D) and *two-dimensional* (2D) imaging, respectively. Despite the practical importance of this technique, the computational effort it requires is still significant, since it involves multidimensional spectral analysis of the acquired signals. Moreover, it suffers from the following relevant drawback: it can miss 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 [71].

Alternative sub-optimal techniques available in the technical 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 *one-dimensional* (1D) or 2D parameter space. Examples of this approach are offered by [72] and [73], and by [74], where range-azimuth estimators for FMCW MIMO radars and for *stepped frequency continuous wave* (SFCW) MIMO radars, respectively, are derived. More specifically, on the one hand, target delays are first estimated by applying the *multiple signal classification* (MUSIC) algorithm to a temporal auto-correlation matrix or by identifying the beat frequencies in the downconverted received signal through spectral analysis (in particular, through the FFT algorithm) in [72] and in [73], respectively; then, the acquired information are exploited to estimate the *direction of arrival* (DOA) of the echoes originating from detected targets. On the other hand, a different approach is proposed in [74], where various *iterative deterministic methods* applicable to a 2D 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, four new detection and estimation algorithms for 2D and 3D radar imaging are developed. They represent different versions of the same algorithm, called *range & angle serial cancellation algorithm* (RASCA), and can be interpreted as embodiments of a general approach to target detection and estimation. In addition, they share some important features with the detection and estimation techniques developed in [73] and [74]. In fact, similarly as the techniques illustrated in [74], they are deterministic, process a single snapshot, operate in an iterative fashion and are computationally efficient; the last feature can be related to the fact they require the evaluation of 1D FFTs only and the search for the global maximum of proper cost functions over 1D (frequency, azimuth or elevation) domains. Moreover, similarly as [73], 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 the iterative estimation techniques developed in [1] and based on a serial cancellation approach for evaluating the parameters of multiple overlapped sinusoids or multiple overlapped complex exponentials in the presence of additive noise. The devised algorithms are able to accurately detect and estimate multiple close targets, and to solve the problem of *merged measurements* or *unresolved measurements* [75–78], in the sense that targets generating merged measurements in the range domain are resolved in the estimation of their angular coordinates.

The remaining part of this chapter is organized as follows. In Section 5.2, the architecture of colocated FMCW MIMO radars and the models adopted in our work for representing their received signals are analysed. The general approach to target detection and estimation on which our algorithms are based is illustrated in Section 5.3, whereas the algorithms themselves are described in Sections 5.4. Various important details about these algorithms are provided in

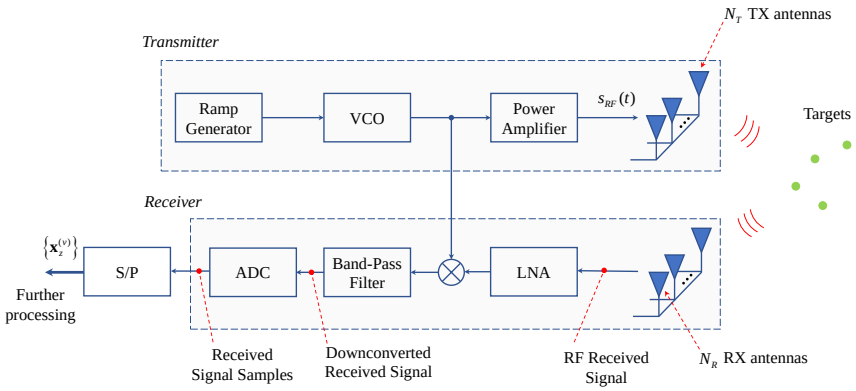


Figure 5.1: Transmitter (upper part) and receiver (lower part) of a colocated MIMO FMCW radar system.

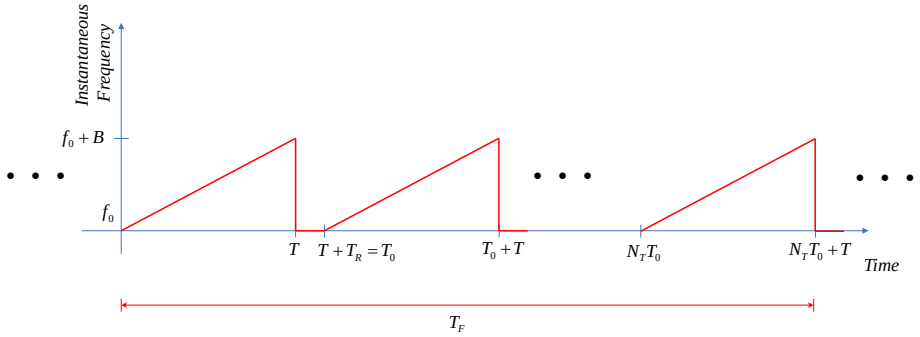


Figure 5.2: Representation of the instantaneous frequency of the RF signal generated by the VCO in a FMCW radar system.

Section 5.5, whereas some technical limitations encountered in their implementation in real world radar systems are discussed in Section 5.6. A description of other FFT-based and MUSIC-based detection and estimation algorithms with which our algorithms are compared is provided in Section 5.7. The computational cost of all the considered algorithms is illustrated in Section 5.8, whereas their performance is analysed in Section 5.9, where various numerical results, based on both synthetically generated data and experimental measurements, are illustrated. Finally, some conclusions are offered in Section 5.10.

5.2 Signal and System Models

This section is devoted to: a) describing the architecture of colocated and bistatic MIMO radar systems operating in *time division multiplexing* (TDM) [2] and at millimeter waves; b) analysing the model of their received signals in the case of FMCW transmission. The considered radar system have the following important characteristics:

1. It is equipped with a *transmit* (TX) and a *receive* (RX) antenna array, consisting of N_T and N_R elements, respectively.

2. These arrays allow to radiate *orthogonal waveforms* from different antennas and to receive distinct replicas of the electromagnetic echoes generated by multiple targets; moreover, the *orthogonality* of the transmitted waveforms is achieved by radiating them through distinct TX antennas over disjoint time intervals.

3. Their TX and RX arrays are made of distinct antenna elements, placed at different positions. However, TX antennas are close to the RX ones and, in particular, are usually placed on the same shield.

The architecture of the FMCW radar system which we always refer to in our work is illustrated in Fig. 5.1. In this case, the radar transmitter consists of a waveform generator feeding a

voltage controlled oscillator (VCO), whose output signal is radiated by the TX array after power amplification. The radiated signal is reflected by multiple targets, whose echoes contribute to the signals acquired through the RX array; each received signal feeds a *low noise amplifier* (LNA), whose output undergoes downconversion, filtering and analog-to-digital conversion. Finally, the resulting stream of signal samples (i.e., of raw data) is processed for target detection and estimation.

Our derivation of the received signal model for the radar system shown in Fig. 5.1 relies on the following assumptions:

a) The radar operates in a static or slowly time varying propagation environment.

b) The signal radiated by the radar transmitter is reflected by L *static point targets*, so that the useful part of the received signal consists of the superposition of L components, each originating from a distinct target.

c) All the TX and RX antennas belong to the same planar shield, so that a two-dimensional reference system lying on the physical antenna array can be defined.

d) Any couple of *physical* TX and RX antennas of the considered bistatic radar is replaced by a single *virtual antenna* of an equivalent *monostatic radar*. The abscissa x_v and the ordinate y_v of the v-th *virtual antenna* (VA) element associated with the p-th TX antenna and the q-th RX antenna (briefly, the (p, q) VA) are computed as[1]

$$x_v = \frac{x_p + x_q}{2} \tag{5.1}$$

and

$$y_v = \frac{y_p + y_q}{2}, \tag{5.2}$$

respectively, with $p = 0, 1, ..., N_T - 1$, $q = 0, 1, ..., N_R - 1$ and $v = 0, 1, ..., N_{\mathrm{VR}} - 1$; here, (x_p, y_p) $((x_q, y_q))$ are the coordinates of the p-th TX (q-th RX) antenna and $N_{\mathrm{VR}} \triangleq N_T \cdot N_R$ represents the overall number of available VAs.

5.2.1 MIMO FMCW radar system

Let us focus now on the FMCW radar shown in Fig. 5.1. Its waveform generator produces a periodic sawtooth signal, so that the instantaneous frequency of the chirp frequency modulated signal available at the output of its VCO evolves periodically, as illustrated in Fig. 5.2. In this figure, the parameters T, T_R and T_0 represent the *chirp interval*, the *reset time* and the *pulse period* (or *pulse repetition interval*), respectively [2], whereas the parameters f_0 and B are the *start frequency* and the *bandwidth*, respectively, of the transmitted signal. Note that, if all the available TX diversity is exploited and a time slot of T_0 s is assigned to each TX antenna, a single *transmission frame*, over which the transmission from the whole TX array is accomplished, lasts $T_F \triangleq N_T T_0$ s; in this interval, a single *snapshot* is acquired at the receive side. Let us focus now on a single chirp interval and, in particular, on the time interval $(0, T)$, and assume that, in that interval, the p-th TX antenna is employed by the considered radar system (with $p \in \{0, 1, \ldots, N_T - 1\}$); the signal radiated by that antenna can be expressed as

$$s_{RF}(t) = A_{RF}\, \Re\{s(t)\}, \tag{5.3}$$

where A_{RF} is its amplitude,

$$s(t) \triangleq \exp(j\theta(t)), \tag{5.4}$$

[1] Note that this is not the only rule adopted in the technical literature to compute the coordinates of the (p, q) VA. For instance, in [79, 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 (5.1) and (5.2), respectively. Keep in mind, however, that if the last rule is adopted, all the following formulas involving such coordinates must be changed accordingly.

$$\theta(t) \triangleq 2\pi\left(f_0 t + \frac{\mu}{2}t^2\right) \tag{5.5}$$

and $\mu \triangleq B/T$ is the *chirp rate* (i.e., the steepness of the generated frequency chirp). It can be proved that, under the assumptions listed above, the n-th received signal sample acquired through the v-th VA element (with $v = 0, 1, ..., N_{\mathrm{VR}} - 1$) in the considered chirp interval is given by (e.g., see [60, Par. 4.6, eq. (4.27)])

$$\begin{aligned} x_{r,n}^{(v)} &= \sum_{l=0}^{L-1} a_l \cos\left(2\pi n\, F_l^{(v)} + \psi_l^{(v)}\right) + w_{r,n}^{(v)} \\ &= \sum_{l=0}^{L-1} \left[C_l^{(v)} \exp\left(j2\pi n\, F_l^{(v)}\right) + (C_l^{(v)})^* \exp\left(-j2\pi n\, F_l^{(v)}\right)\right] + w_{r,n}^{(v)}, \end{aligned} \tag{5.6}$$

with $n = 0, 1, ..., N-1$; here, N is the overall number of samples acquired over each chirp period, a_l is the amplitude[2] of the l-th component of the useful signal,

$$C_l^{(v)} \triangleq \frac{1}{2} a_l \exp\left(j\psi_l^{(v)}\right) \tag{5.7}$$

represents the *complex amplitude* of the real tone appearing in the *right hand side* (RHS) of (5.6),

$$F_l^{(v)} \triangleq f_l^{(v)} T_s = f_l^{(v)}/f_s \tag{5.8}$$

is the normalised version of the frequency

$$f_l^{(v)} \triangleq \mu\, \tau_l^{(v)}, \tag{5.9}$$

characterizing the l-th target detected on the v-th virtual RX antenna, T_s (f_s) denotes the sampling period (frequency) of the employed *analog-to-digital converters* (ADCs),

$$\tau_l^{(v)} \triangleq \frac{2}{c}\left[R_l + x_v \cos(\phi_l)\sin(\theta_l) + y_v \sin(\phi_l)\right] \tag{5.10}$$

is the delay of the echo generated by the l-th target and observed on the v-th virtual channel, R_l, θ_l and ϕ_l denote the range of the l-th target, its azimuth and its elevation (all measured with respect to the center of the receive array), respectively,

$$\psi_l^{(v)} \cong 2\pi f_0 \tau_l^{(v)} \tag{5.11}$$

and $w_{r,n}^{(v)}$ is the n-th sample of the AWGN sequence affecting the received signal (this sample is modelled as a real Gaussian random variable having zero mean and variance σ^2 for any v). It is important to point out that: a) the real signal model (5.6) can be adopted in all the FMCW radar systems acquiring only the *in-phase* component of the signal captured by each RX antenna; b) some commercial MIMO radar devices provide both the *in-phase* and *quadrature* components of the received RF signals (e.g., see [69, Par 2.1 eq. (2.2)]). In the last case, the complex model

$$x_{c,n}^{(v)} = \sum_{l=0}^{L-1} A_l^{(v)} \exp\left(j2\pi n\, F_l^{(v)}\right) + w_{c,n}^{(v)}, \tag{5.12}$$

must be adopted in place of its real counterpart (5.6) for any n; here,

$$A_l^{(v)} \triangleq a_l \exp\left(j\psi_l^{(v)}\right) \tag{5.13}$$

for any v and l, and $w_{c,n}^{(v)}$ is the n-th sample of the AWGN sequence affecting the received signal (this sample is modelled as a complex Gaussian random variable having zero mean and variance σ^2 for any v).

[2]This amplitude quantifies the *radar cross section* (RCS) of the l-th target. It depends on both the range of this target and its reflectivity. Moreover, in this work, it is assumed to be independent of the virtual antenna index (i.e., of v), for simplicity.

5.2.2 Target detection and estimation

In the considered FMCW radar system, the useful component of the received signal observed on each virtual channel can be represented as a superposition of L real or complex oscillations; moreover, the value of the parameter L has to be considered *unknown*. In the following derivations, the real samples $\{x_{r,n}^{(v)};\ n = 0,\ 1,\ ...,\ N-1\}$ or their complex counterpart $\{x_{c,n}^{(v)};\ n = 0,\ 1,\ ...,\ N-1\}$ acquired on the v-th virtual channel are collected in the N-dimensional vector

$$\mathbf{x}_z^{(v)} \triangleq \left[x_{z,0}^{(v)}, x_{z,1}^{(v)}, ..., x_{z,N-1}^{(v)}\right]^T, \tag{5.14}$$

with[3] $z = r$ or c. This vector is processed by the next stages of the radar receiver for target detection and estimation. As it can be easily inferred from (5.6)–(5.9) ((5.12)–(5.13)), in the considered radar system, the problem of *target detection* and *range estimation* on the v-th virtual channel is equivalent to the classic problem of detecting multiple overlapped sinusoids (multiple overlapped complex exponentials) in the presence of AWGN and estimating their *frequencies* [80]. In fact, if, in a FMCW radar system, the l-th tone is found at the frequency $\hat{f}_l^{(v)}$, the presence of a target at the range (see (5.9) and (5.10))

$$\hat{R}_{v,l} = \frac{1}{2}\frac{c}{\mu}\hat{f}_l^{(v)} \tag{5.15}$$

is detected. Information about the angular coordinates (namely, the azimuth and the elevation) of this target, instead, can be acquired through the estimation of the set of N_{VR} phases $\{\psi_l^{(v)};\ v = 0,\ 1,\ ...,\ N_{\mathrm{VR}}-1\}$ observed over the available virtual antennas. In fact, since (see (5.10) and (5.11))

$$\psi_l^{(v)} \cong \frac{4\pi}{\lambda}\left[R_l + x_v\cos(\phi_l)\sin(\theta_l) + y_v\sin(\phi_l)\right], \tag{5.16}$$

where

$$\lambda \triangleq \frac{c}{f_0} \tag{5.17}$$

is the wavelength associated with the frequency f_0, the sequence $\{\psi_l^{(v)};\ v = 0,\ 1,\ ...,\ N_{\mathrm{VR}}-1\}$ exhibits a periodic behavior characterized by the *normalised horizontal spatial frequency*

$$F_{\mathrm{H},l} \triangleq 2\frac{d_{\mathrm{H}}}{\lambda}\cos(\phi_l)\sin(\theta_l), \tag{5.18}$$

if the considered virtual elements form an horizontal *uniform linear array* (ULA), whose adjacent elements are spaced d_{H} m apart. Dually, if a virtual vertical ULA is assumed, the periodic variations observed in the same sequence of phases are characterized by the *normalised vertical spatial frequency*

$$F_{\mathrm{V},l} \triangleq 2\frac{d_{\mathrm{V}}}{\lambda}\sin(\phi_l), \tag{5.19}$$

where d_{V} denotes the distance between adjacent elements of the virtual array itself. Consequently, angle finding can be easily accomplished by *digital beamforming*, i.e. by performing a FFT on the estimated phases taken across multiple elements of the virtual array in a single frame interval [81, 82].

Finally, it is important to note that, in the development of detection and estimation algorithms for colocated MIMO radar systems operating at millimeter waves, the following technical issues need to be taken carefully into account:

1. These radar systems often operate at short ranges and in the presence of extended targets. Each radar image is a *cloud of point targets* whose mutual spacing can be very small [71].

[3] In the following, when the letter z will be used in a subscript, it will be implicitly assumed, unless differently stated, that it can be equal to r or c.

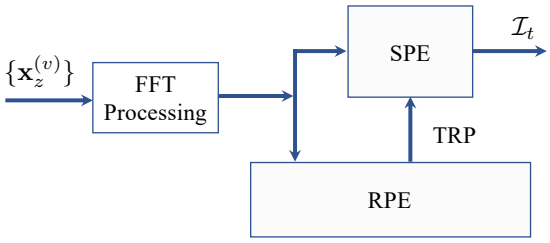


Figure 5.3: Block diagram describing the general approach to target detection and estimation adopted in our work.

For this reason, the accuracy of these images depends, first of all, on the frequency resolution achieved by the detection and estimation algorithm employed on each virtual antenna in a FMCW radar system. In fact, this makes the radar receiver able to separate point targets characterized by similar ranges.

2. Distinct radar echoes can be characterized by substantially different *signal-to-noise ratios* (SNRs), because of relevant differences among the amplitudes of the L overlapped oscillations forming the useful component of the received signal (see (5.6) and (5.12)).

3. The number N of samples acquired over each virtual channel usually ranges from few hundreds to few thousands.

The last two issues explain why significant attention must be paid to the accuracy achieved by the adopted detection and estimation algorithms at low SNRs and/or for relatively small values of N, since this can appreciably influence the quality of the generated radar image.

5.3 Description of the proposed approach to the detection and estimation of multiple targets

All the algorithms developed in the following section can be considered as specific instances of a general approach to target detection and estimation; this approach is described by the block diagram shown in Fig. 5.3. The processing accomplished by the blocks which this diagram consists of can be summarized as follows. Each vector of the set $\{\mathbf{x}_z^{(v)}\}$, collecting N_{VR} vectors (see (5.14)), undergoes FFT processing, so that, in a FMCW radar system, the analysis of the acquired measurements is moved from the time-domain to the frequency-domain. The output of the FFT block is processed by the *range profile estimator* (RPE), that generates the so called *target range profile* (TRP), i.e. a collection of: a) the ranges at which the relevant echoes are detected; b) the associated energies. Note that the last quantities allow us to rank each range on the basis of its perceptual importance. The output of the FFT processing block 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

$$\mathcal{I}_t \triangleq \left\{ \left(\hat{R}_l, \hat{\theta}_l, \hat{\phi}_l, \left| \hat{C}_l \right| \right) ; l = 0, 1, \ldots, \hat{L} - 1 \right\} \tag{5.20}$$

or the set

$$\mathcal{I}_t \triangleq \left\{ \left(\hat{R}_l, \hat{\theta}_l, \left| \hat{C}_l \right| \right) ; l = 0, 1, \ldots, \hat{L} - 1 \right\} \tag{5.21}$$

in the case of 3D and 2D imaging, respectively; here, $\hat{L}$ represents an estimate of the parameter L (i.e., of the overall number of point targets), whereas $\hat{R}_l$, $\hat{\theta}_l$, $\hat{\phi}_l$ and $|\hat{C}_l|$ represent an estimate of the range R_l, azimuth θ_l, elevation ϕ_l and amplitude $|C_l|$, respectively, of the l-th target (with $l = 0, 1, ..., \hat{L} - 1$).

It is important to point out that:

1. If this approach 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 couple 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 to reduce 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 embodiments of the proposed approach.

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 FFT processing block; such outputs are generated on the basis of N_A of the N_{VR} VAs. Note that, on the one hand, a larger N_A allows to compute a 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 represent a *multidimensional problem* since they aim at detecting multiple targets and estimating their ranges. In our method, this multidimensional problem is turned into a sequence of 2D estimation problems by adopting a *serial interference cancellation* (SIC) approach (e.g., see [74]). This means that the noisy signal observed on each VA is processed in an iterative fashion. In 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.

d) *Alternating maximization* – The estimation of the normalised frequency and the complex amplitude of a detected target requires searching for the maximum (or the minimum) of a proper cost function over a 2D domain. In our method, the *alternating maximization* (AM) technique is exploited to develop iterative algorithms that alternate the estimation of the normalised frequency of a given target with that of its complex amplitude; for this reason, a 2D optimization problem is turned into a couple of interacting 1D optimization problems (e.g., see [58, Par. IV-A]).

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

a) *Alternating maximization* – The AM technique is exploited to develop iterative algorithms that alternate the estimation of the elevation of a given target with that of its azimuth. This allows us to decouple the estimation of target elevation from that of its azimuth.

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 the SPE aims at resolving all the targets associated with a given range and estimating their angular coordinates. In the technical literature about radar systems, the detection of an unknown number of targets characterized by the *same range* (or by ranges whose mutual differences are below the range resolution of the employed radar system) and the estimation of their angular parameters is known to be a difficult multidimensional problem (e.g., see [72, Par. III-C]). In our method, a SIC approach is exploited to turn this multidimensional problem into a sequence of 2D (1D) estimation problems in 3D (2D) imaging (see [74] and references therein). 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 FFT processing 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 their 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 a couple of 1D optimization problems). Note also that, once again, the use of a serial cancellation approach 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 all the parallel procedures need to be fused when they 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 one time.

Based on the general approach outlined above and on the techniques listed for the RPE and the SPE, four specific algorithms, called *range & angle serial cancellation algorithms* (RASCAs) are developed in the following. The algorithms are called RASCA-FR2 (RASCA-FC2) and RASCA-FR3 (RASCA-FC3), since they generate 2D and 3D radar images, respectively, on the basis of real (complex) measurements. In the description of these algorithms we assume, without loosing generality, that the available measurements are acquired through the $N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ virtual *uniform rectangular array* (URA) represented in Fig. 5.4 in the case of 3D imaging and through an *horizontal* ULA (HULA), 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 d_{VH} (d_{VV}), whereas, in the second one, is denoted d_{VH}. Moreover, in our considerations, we assume that a *reference* VA, identified by $(p, q) = (p_R, q_R)$ ($p = p_R$) in the 3D (2D) case, is selected in the virtual array, as exemplified by Fig. 5.4.

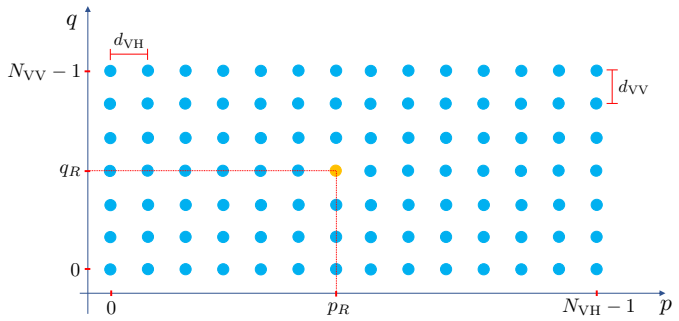


Figure 5.4: Virtual antenna array considered in the description of our detection and estimation algorithms. The reference VA selected in this case is identified by a yellow circle.

In the following sections, all the RASCAs are described. In Section 5.4 we first focus on the RASCA-FR3 and RASCA-FC3, i.e. on the algorithms to be employed in a MIMO FMCW radar system equipped with a 2D antenna array (in particular, with the URA shown in Fig. 5.4). Then, we show how to adapt these algorithms to the case in which this radar system is equipped with a 1D antenna array (in particular, with an ULA); this leads to the RASCA-FR2 and the RASCA-FC2.

5.4 Range & angle serial cancellation algorithms for a frequency modulated continuous wave radar system

In this paragraph, we provide a short description of the architecture of the RASCAs for FMCW radar systems and comment on the method we developed for target detection and cancellation in the angular domain. Then, we illustrate RASCA-FR3 and RASCA-FC3 in detail. Finally, we show how to derive the RASCA-FR2 and RASCA-FC2 from them.

5.4.1 Architecture of the range & angle serial cancellation algorithms for a frequency modulated continuous wave radar system

The inner structure of the RASCAs for an FMCW radar system is described by the block diagram shown in Fig. 5.5, whereas the meaning of the most relevant parameters, sets, vectors and matrices appearing in the description of this algorithm is summarised in Table 5.1. . The processing accomplished inside the blocks appearing in that figure can be summarized as follows. The FFT *processing* block turns the *time domain* information provided by the set of N_{VR} vectors $\{\mathbf{x}_z^{(v)}\}$ into the *frequency domain* information feeding both the RPE and the SPE blocks. This transformation requires the evaluation of $3N_{\mathrm{VR}}$ FFTs, all of order N_0. In fact, it consists in the evaluation of the triad $(\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)})$, collecting three N_0-dimensional vectors, on the basis of $\mathbf{x}_z^{(v)}$, for $v = 0, 1, ..., N_{\mathrm{VR}} - 1$. For this reason, the output of the considered block is represented by the set

$$\mathcal{S}_{\mathrm{FFT}} \triangleq \left\{ \left(\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)} \right); v = 0, 1, ..., N_{\mathrm{VR}} - 1 \right\}, \tag{5.22}$$

consisting of $3 \cdot N_{\mathrm{VR}}$ 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

$$\bar{\mathcal{S}}_{\mathrm{FFT}} \triangleq \left\{ \left(\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)} \right); v \in \mathcal{S}_A \right\}, \tag{5.23}$$

of $\mathcal{S}_{\mathrm{FFT}}$ (5.22); here,

$$\mathcal{S}_A \triangleq \{v_0, v_1, ..., v_{N_A-1}\}, \tag{5.24}$$

represents the set of the values of the VA index v identifying the elements of $\mathcal{S}_{\mathrm{FFT}}$ that belong to $\bar{\mathcal{S}}_{\mathrm{FFT}}$. Each of the triads of $\bar{\mathcal{S}}_{\mathrm{FFT}}$ is processed, independently of all the other ones, by a novel

Table 5.1: Most relevant parameters, sets, vectors and matrices appearing in our description of RASCA-FC3.

Parameter	Description
$\mathbf{x}_{0,\mathrm{ZP}}^{(v)}$	Zero-padded version of the vector $\{\mathbf{x}_z^{(v)}\}$.
$\mathbf{x}_{m,\mathrm{ZP}}^{(v)}$	Zero-padded version of the vector $\{\mathbf{x}_m^{(v)}\}$ (with $m = 1$ and 2).
$\mathbf{X}_m^{(v)}$	DFT of the vector $\mathbf{x}_{m,\mathrm{ZP}}^{(v)}$ (with $m = 0$, 1 and 2).
$\mathcal{S}_{\mathrm{FFT}}$	Set collecting the N_{VR} triads $(\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)})$ such that $v = 0, 1, \ldots, N_{\mathrm{VR}} - 1$.
N_A	Number of virtual antennas exploited by the RPE.
$\mathcal{S}_A$	Set collecting the N_A indices of the virtual antennas exploited by the RPE.
$\bar{\mathcal{S}}_{\mathrm{FFT}}$	Set collecting the N_A triads $(\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)})$ such that $v \in \mathcal{S}_A$.
T_{STDREC}	Threshold set in the STDREC algorithm to limit its overall number of iterations.
$\mathrm{X}_m^{(v_k)}[i]$	Residual spectrum available for the v_k-th antenna at the end of the $(i-1)$-th iteration of the STDREC algorithm.
$E_i^{(v_k)}$	Energy associated with the residual spectrum $\mathrm{X}_m^{(v_k)}[i]$ and computed by the STDREC algorithm for the v_k-th antenna at the end of its $(i-1)$-th iteration.
$\hat{C}_i^{(v_k)}$	Estimate of the complex amplitude $C_i^{(v_k)}$ computed for the v_k-th antenna in the i-th iteration of the STDREC algorithm.
$\hat{F}_i^{(v_k)}$	Estimate of the normalised frequency $F_i^{(v_k)}$ computed for the v_k-th antenna in the i-th iteration of the STDREC algorithm.
$\mathbf{C}_{X_m}^{(v_k)}[i]$	Estimate of the contribution given by the i-th target detected on the v_k-th antenna to the vector $\mathrm{X}_m^{(v_k)}[i]$ (with $m = 0$, 1 and 2).
$\mathcal{S}_{\mathrm{RPE}}$	Set collecting the information generated by the RPE and feeding to the SPE.
T_{STDAEC}	Threshold set in the STDAEC algorithm to limit its overall number of recursions.
$\mathbf{X}^{(i)}[l]$	Matrix representing the spectral contribution given by the l-th frequency bin and available at the beginning of the i-th iteration of the STDAEC algorithm.
$E^{(i)}[l]$	Energy associated with the residual spectrum $\mathbf{X}^{(i)}[l]$ made available by the STDAEC algorithm for the l-th frequency bin at the end of its $(i-1)$-th iteration.
$\mathbf{s}_{\mathrm{VULA},k}^{(i)}[l]$	k - th vector collecting the spectral information computed for the reference VULA and the l-th frequency bin in the i-th iteration of the STDAEC algorithm (with $k = 0, 1$ and 2).
$\hat{C}_{\mathrm{V},i}[l]$	Estimate of the complex amplitude $C_{\mathrm{V},i}[l]$ computed for the target detected in the l-th frequency bin within the i-th iteration of the STDAEC algorithm.
$\hat{F}_{\mathrm{V},i}[l]$	Estimate of the normalised vertical spatial frequency $F_{\mathrm{V},i}[l]$ computed for the target detected in the l-th frequency bin within the i-th iteration of the STDAEC algorithm.
$R_i^{(\mathrm{VF})}[l,q]$	Phase rotation factor computed for the q-th VULA and the l-th frequency bin within the i-th iteration of the STDAEC algorithm (vertical folding).
$\mathbf{X}_i^{(\mathrm{VF})}[l]$	Vertically folded spectrum computed for the l-th frequency bin within the i-th iteration of the STDAEC algorithm.
$\hat{C}_{\mathrm{H},i}[l]$	Estimate of the complex amplitude $C_{\mathrm{H},i}[l]$ computed on the basis of a vertically folded spectrum within the i-th iteration of the STDAEC algorithm for the l-th frequency bin.
$\hat{F}_{\mathrm{H},i}[l]$	Estimate of the normalised horizontal spatial frequency $F_{\mathrm{H},i}[l]$ computed on the basis of a vertically folded spectrum within the i-th iteration of the STDAEC algorithm.
$R_i^{(\mathrm{HF})}[l,p]$	Phase rotation factor computed for the p-th HULA and the l-th frequency bin within the i-th iteration of the STDAEC algorithm (horizontal folding).
$R_i^{(\mathrm{HV})}[l,p,q]$	Phase rotation factor computed for the (p,q) VA and the l-th frequency bin within the i-th iteration of the STDAEC algorithm (overall folding).
$\{\mathbf{X}_{m,\mathrm{OF}}^{(i)}[l]\}$	Overall folded spectrum computed for the l-th frequency bin within the i-th iteration of the STDAEC algorithm.
$\hat{C}_i[l]$	Estimate of the complex amplitude $C_i[l]$ computed on the basis of a overall folded spectrum within the i-th iteration of the STDAEC algorithm.
$\hat{F}_i[l]$	Estimate of the normalised frequency $F_i[l]$ computed on the basis of a overall folded spectrum within the i-th iteration of the STDAEC algorithm.
$\mathbf{C}_{X_0}^{(i)}[l]$	Estimate of the contribution, given by the target detected within i-th iteration of the STDAEC algorithm, to the vector $\mathbf{X}^{(i)}[l]$.
$\hat{R}_i[l]$, $\hat{\theta}_i[l]$, $\hat{\phi}_i[l]$	Estimates of the range, azimuth and elevation made available by the SPE for the i-th target detected in the l-th frequency bin.

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 frequencies associated with these ranges; see (5.9) and (5.15)) and their complex amplitudes (see (5.11) and (5.13)). The name of this algorithm originates from the fact that, in each of its iterations, it *detects* a single target (namely, the strongest target), *estimates* its parameters (and, in particular, the frequency characterizing it, i.e. its range) and *cancels* the target contribution 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[4]

$$\mathcal{S}_{v_k} \triangleq \left\{ \left(\hat{\alpha}_i^{(v_k)}, \hat{C}_i^{(v_k)} \right), i = 0, 1, \ldots, L_k - 1 \right\}, \tag{5.25}$$

with $k = 0$, 1, ..., $N_A - 1$; here, L_k is the overall number of targets detected on the considered VA, whereas $\hat{C}_i^{(v_k)}$ and $\hat{\alpha}_i^{(v_k)}$ represent the estimate of the complex amplitude of the i-th target and the index of the *frequency bin*[5] in which this target has been detected. Finally, the information provided by the N_A sets $\{\mathcal{S}_{v_k}\}$ are merged to generate the single set

$$\mathcal{S}_{\text{RPE}} \triangleq \{ (\hat{\alpha}_l, E_{b,l}), l = 0, 1, \ldots, L_b - 1 \}, \tag{5.26}$$

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

a) 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.

b) The energies $\{E_{b,l}\}$ represent the perceptual importance of the identified frequency bins, in the sense that a larger energy is associated with a more important frequency bin.

Both the sets $\mathcal{S}_{\text{FFT}}$ (5.22) and $\mathcal{S}_{\text{RPE}}$ (5.26) feed the SPE. The aim of this block is to analyse the spectral information associated with the ranges (i.e., with the frequency bins) identified by the RPE in order to: a) estimate the angular coordinates (i.e., azimuth and elevation) of the targets contributing to each frequency bin; b) detect additional targets associated with adjacent frequency 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 $\mathcal{S}_{\text{FFT}}$ (5.22) 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 frequency 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 frequency bins (i.e., ranges) appearing in the TRP (see Fig. 5.5). 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 frequency bin only (with $l = 0$, 1, ..., $L_b - 1$), detects $L[l]$ targets contributing to it 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

[4]Note that the complex amplitude $\hat{C}_i^{(v_k)}$ appearing in the following equations is replaced by $\hat{A}_i^{(v_k)}$ if the received sequence is complex (see eqs. (5.7) and (5.13)). This consideration holds for various equations appearing in the remaining part of this chapter.

[5]Generally speaking, the evaluation of an FFT of order N_0 leads to partitioning the normalised frequency interval [0, 1/2) in N_0 *frequency bins*.

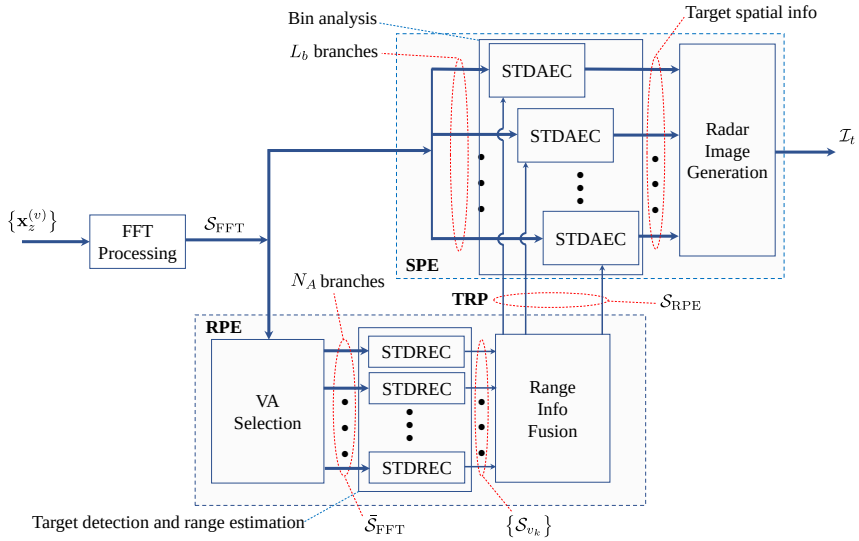


Figure 5.5: Block diagram describing the inner structure of the RASCAs for an FMCW radar system.

(do not forget that the preliminary estimate of this range is provided by the bin index $\hat{\alpha}_l$). If $D[l]$ iterations are accomplished by the l-th STDAEC algorithm, $D[l]$ distinct targets are detected in the $\hat{\alpha}_l$-th frequency bin, provided that 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

$$\mathcal{T}_l \triangleq \left\{ \left(\hat{C}_i\left[l\right], \hat{F}_i\left[l\right], \hat{\alpha}_i\left[l\right], \hat{F}_{\mathrm{V},i}[l], \hat{F}_{\mathrm{H},i}[l] \right); i = 0, 1, \ldots, D[l]-1 \right\}, \tag{5.27}$$

or in the set

$$\mathcal{T}_l \triangleq \left\{ \left(\hat{C}_i\left[l\right], \hat{F}_i\left[l\right], \hat{\alpha}_i\left[l\right], \hat{F}_{\mathrm{H},i}[l] \right); i = 0, 1, \ldots, D[l]-1 \right\}, \tag{5.28}$$

with $l = 0, 1, \ldots, L_b - 1$, in the case of 3D and 2D radar imaging, respectively; here, $\hat{C}_i\left[l\right]$, $\hat{F}_i\left[l\right]$ and $\hat{\alpha}_i\left[l\right]$ denote the estimates of the complex amplitude $C_i\left[l\right]$, of the normalised frequency $F_i\left[l\right]$ and of the frequency bin $\alpha_i\left[l\right]$, respectively, characterizing the i-th target detected *on the reference* VA and in the l-th frequency bin, whereas $\hat{F}_{\mathrm{H},i}[l]$ and $\hat{F}_{\mathrm{V},i}[l]$ represent the estimates of the normalised horizontal spatial frequency $F_{\mathrm{H},i}[l]$ and of the normalised vertical spatial frequency $F_{\mathrm{V},i}[l]$, respectively, 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*.

5.4.2 Some considerations on target detection and cancellation in the angular domain

It is worth pointing out that the STDAEC algorithm represents the most complicated part of the processing accomplished by all the blocks appearing in Fig. 5.5. The derivation of this algorithm relies on the fact that: a) each target provides an *additive contribution* to the spectra evaluated on all the VAs; b) *periodic variations* are observed in the phase of this contribution if we move horizontally or vertically along the considered virtual array (see Fig. 5.4). 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 frequency bin does not change from antenna to antenna, the complex amplitude $C_i[p,q,l]$ observed on the (p,q) VA can be expressed as (see (5.9) and (5.10))

$$C_i[p,q,l] = C_i[l] \exp\left[j \frac{4\pi}{\lambda} \left[d_{\mathrm{VH}} \left(p - p_R\right) \cos(\phi_i[l]) \sin(\theta_i[l]) + d_{\mathrm{VV}} \left(q - q_R\right) \sin(\phi_i[l]) \right] \right]; \tag{5.29}$$

here, $\lambda = c/f_0$ is the wavelength associated with the start frequency, $(p_R,\, q_R)$ is the couple of integers identifying the selected *reference* VA, $\theta_i[l]$, $\phi_i[l]$ and $R_i[l]$ are the azimuth, the elevation

and the range, respectively, characterizing the considered target, and $C_i[l]$ is its complex amplitude observed on the reference antenna. If (5.29) holds, the rate of the phase variations observed in the complex amplitudes $\{C_i[p,q,l]\}$ for a given l is proportional to (see (5.18) and (5.19))

$$F_{\mathrm{H},i}[l] \triangleq \frac{2d_{\mathrm{VH}}}{\lambda}\cos(\phi_i[l])\sin(\theta_i[l]) \tag{5.30}$$

and

$$F_{\mathrm{V},i}[l] \triangleq \frac{2d_{\mathrm{VV}}}{\lambda}\sin(\phi_i[l]), \tag{5.31}$$

if we move along an HULA and a *vertical* ULA (VULA), respectively. In fact, the quantity $F_{\mathrm{H},i}[l]$ ($F_{\mathrm{V},i}[l]$) represents the normalised horizontal (vertical) *spatial frequency* characterizing the i-th target detected in the l-th frequency bin; if both these frequencies are known, the elevation and the azimuth of this target can be evaluated as

$$\phi_i[l] = \arcsin\left(\frac{\lambda}{2d_{\mathrm{VV}}}F_{\mathrm{V},i}[l]\right) \tag{5.32}$$

and

$$\theta_i[l] = \arcsin\left(\frac{\lambda}{2d_{\mathrm{VH}}\cos(\phi_i[l])}F_{\mathrm{H},i}[l]\right), \tag{5.33}$$

respectively.

Moreover, in the derivation of the STDAEC algorithm, the following two techniques have been 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 frequency bin and, in particular, to identify targets having similar spatial coordinates. It is important to keep in mind that the frequencies associated with distinct targets detected in the same frequency 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 frequency 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 normalised frequency is exactly a multiple of the fundamental frequency

$$F_{\mathrm{FFT}} = 1/N_0, \tag{5.34}$$

consequently, such a contribution is spread over multiple adjacent frequency bins (i.e., spectral leakage is observed)

Spatial folding – As already stated above, the frequency associated with a target detected in the l-th frequency 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 frequency 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 frequencies associated with a target detected in a given frequency bin have been estimated (see (5.30) and (5.31)), the spectra computed on multiple VAs can be combined *in a constructive fashion* by

1) taking a *reference* VA (identified by $(p,q) = (p_R, q_R)$; see Fig. 5.4), and compensating for the phase differences, estimated for that target, between the reference VA and the other VAs of the whole array, or

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

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 that affects the VAs and b) combining, in a

constructive fashion, the contributions of all the targets different from the one which the employed spatial frequencies 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 frequency bin or in a bin close to it. When this peak is detected, a refined estimate of the frequency (and, consequently, of the range) and the complex amplitude characterizing the target for which folding has been accomplished can be computed by identifying its position. On the contrary, if no peak is found, 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 chapter, when folding is employed, the following terminology is adopted:

Vertical folding – This refers to the case in which folding involves a reference 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 virtual URA; a single folded spectrum is computed in this case.

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 that refers to the l-th frequency bin, the *estimates* $\hat{F}_{\mathrm{H},i}[l]$ and $\hat{F}_{\mathrm{V},i}[l]$ of the frequencies $F_{\mathrm{H},i}[l]$ 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*.

All the mathematical details about vertical and overall folding can be found in the next paragraph.

5.4.3 Detailed description of the range & angle serial cancellation algorithms for a frequency modulated continuous wave radar system endowed with a two-dimensional antenna array

In the following, the RASCA-FR3 is described first; then, the (minor) modifications required to obtain the RASCA-FC3 from it are illustrated. The RASCA-FR3 processing is divided in three tasks, each associated with one of the blocks appearing in Fig. 5.5 (the i-th task is denoted Ti); a description of each task is provided below. Various details about the techniques employed in these tasks, omitted here to ease the understanding of the overall flow of the algorithm, are provided in Section 5.5.

T1 – FFT processing

The processing accomplished within this task can be summarized as follows. Given the vector $\mathbf{x}_z^{(v)}$, the N-dimensional vectors

$$\mathbf{x}_1^{(v)} \triangleq \left[x_{1,0}^{(v)}, x_{1,1}^{(v)}, \ldots, x_{1,N-1}^{(v)}\right]^T \tag{5.35}$$

and

$$\mathbf{x}_2^{(v)} \triangleq \left[x_{2,0}^{(v)}, x_{2,1}^{(v)}, \ldots, x_{2,N-1}^{(v)}\right]^T \tag{5.36}$$

are evaluated for $v = 0, 1, \ldots, N_{\mathrm{VR}} - 1$; here,

$$x_{m,n}^{(v)} \triangleq n^m \, x_{z,n}^{(v)} \tag{5.37}$$

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

$$\mathbf{x}_{0,\mathrm{ZP}}^{(v)} = \left[(\mathbf{x}_z^{(v)})^T \; \mathbf{0}_{(M-1)N}^T\right]^T, \tag{5.38}$$

$$\mathbf{x}_{1,\mathrm{ZP}}^{(v)} = \left[(\mathbf{x}_1^{(v)})^T \; \mathbf{0}_{(M-1)N}^T\right]^T \tag{5.39}$$

and

$$\mathbf{x}_{2,\mathrm{ZP}}^{(v)} = \left[(\mathbf{x}_2^{(v)})^T \ \mathbf{0}_{(M-1)N}^T \right]^T, \tag{5.40}$$

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

$$N_0 \triangleq M\,N. \tag{5.41}$$

Finally, the N_0-dimensional vectors

$$\begin{aligned}\mathbf{X}_m^{(v)} =& \left[X_{m,0}^{(v)}, X_{m,1}^{(v)}, ..., X_{m,N_0-1}^{(v)} \right]^T = \mathbf{X}_m\,[p,q] \\ =& \left[X_{m,0}\,[p,q]\,, X_{m,1}\,[p,q]\,, ..., X_{m,N_0-1}\,[p,q] \right]^T \\ \triangleq& \mathrm{DFT}_{N_0}\left[\mathbf{x}_{m,\mathrm{ZP}}^{(v)} \right],\end{aligned} \tag{5.42}$$

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

$$X_{m,k}^{(v)} \triangleq \frac{1}{N} \sum_{n=0}^{N-1} n^m\, x_{z,n}^{(v)} \exp\left(-j \frac{2\pi n k}{N_0} \right), \tag{5.43}$$

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

T2 – RPE

The processing accomplished within this task consists of the three consecutive steps listed below (the i-th step is denoted T2-Si in the following); each step is associated with one of the blocks included in the RPE, as shown in Fig. 5.5.

T2-S1) VA *selection* – In this step, the set $\bar{\mathcal{S}}_{\mathrm{FFT}}$ (5.23) is built. This requires generating the set $\mathcal{S}_A$ (5.24), i.e. a set of N_A integers that identifies the selected VAs. In our computer simulations, the elements of $\mathcal{S}_A$ have been generated by *randomly* extracting N_A distinct integers from the set $\{0, 1, ..., N_{\mathrm{VR}} - 1\}$.

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

$$\mathbf{X}_m^{(v_k)}[0] \triangleq \mathbf{X}_m^{(v_k)}, \tag{5.44}$$

with $m = 0,\ 1,\ 2$, and the iteration index i to 0. Then, the STDREC iterations are 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 each is denoted 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 normalised frequency $F_i^{(v_k)}$ and the complex amplitude $C_i^{(v_k)}$ associated with it. Note that, generally speaking, the normalised frequency $F_i^{(v_k)}$ is not a multiple of the fundamental frequency F_{FFT} (5.34), that characterizes the FFT processing executed in **T1**; for this reason, it can be expressed as

$$F_i^{(v_k)} = F_{c,i}^{(v_k)} + \delta_i^{(v_k)}\ F_{\mathrm{DFT}}, \tag{5.45}$$

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 an algorithm, dubbed *single frequency estimator*[6] (SFE) and whose

[6]Note that our general description of the SFE includes the computation of three DFTs, that, in this case, are already evaluated in **T1**.

detailed description is provided in Paragraph 5.5.1. In short, the SFE computes the estimates $\hat{C}_i^{(v_k)}$, $\hat{F}_{c,i}^{(v_k)}$, $\hat{\delta}_i^{(v_k)}$,

$$\hat{\alpha}_i^{(v_k)} = \left\lfloor \hat{F}_{c,i}^{(v_k)} / F_{\mathrm{DFT}} \right\rfloor \tag{5.46}$$

and

$$\hat{F}_i^{(v_k)} = \hat{F}_{c,i}^{(v_k)} + \hat{\delta}_i^{(v_k)} \; F_{\mathrm{DFT}} \tag{5.47}$$

of the parameters $C_i^{(v_k)}$, $F_{c,i}^{(v_k)}$, $\delta_i^{(v_k)}$, $\alpha_i^{(v_k)}$ 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 frequency 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 $\mathcal{S}_{v_k}$ (5.25), 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 computed on the basis of (5.110)–(5.112) (see Paragraph 5.5.3) and cancelled from the triad itself. Cancellation consists in the computation of the new *residual* triad

$$\begin{aligned} \mathbf{X}_m^{(v_k)}[i+1] =& \left[X_{m,0}^{(v_k)}[i+1], ..., X_{m,N_0-1}^{(v_k)}[i+1] \right]^T \\ \triangleq& \mathbf{X}_m^{(v_k)}[i] - \mathbf{C}_{X_m}^{(v_k)}[i], \end{aligned} \tag{5.48}$$

with $m = 0,\ 1,\ 2$.

STDREC-S3) *Computation of the residual energy in the frequency domain* – The energy

$$E_{i+1}^{(v_k)} \triangleq \left\| \mathbf{X}_0^{(v_k)}[i+1] \right\|^2 = \sum_{p=0}^{N_0-1} \left| X_{0,p}^{(v_k)}[i+1] \right|^2 \tag{5.49}$$

characterizing the residual spectrum vector $\mathbf{X}_0^{(v_k)}[i+1]$ (5.48) is computed and compared with the positive threshold T_{STDREC} (which may depend on range, i.e. on the detected frequency). 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 sets $\{\mathcal{S}_{v_k}\}$ (5.25) evaluated in the previous step. Its output is represented by the set $\mathcal{S}_{\mathrm{RPE}}$ (5.26), whose elements (i.e., the L_b couples $\{(\hat{\alpha}_l, E_{b,l})\}$) are evaluated as follows. If we define the set

$$\mathcal{A}_b^{(v_k)} \triangleq \{\hat{\alpha}_i^{(v_k)}; i = 0, 1, \ldots, L_k - 1\}, \tag{5.50}$$

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

$$\mathcal{A}_b \triangleq \{\hat{\alpha}_l; l = 0, 1, \ldots, L_b - 1\} \tag{5.51}$$

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

$$E_{b,l} = \frac{1}{N_{b,l}} \sum_{k=0}^{N_A-1} \sum_{i=0}^{L_k-1} \left| \hat{C}_i^{(v_k)} \right|^2 \delta \left[\hat{\alpha}_i^{(v_k)} - \hat{\alpha}_l \right], \tag{5.52}$$

where

$$N_{b,l} = \sum_{k=0}^{N_A-1} \sum_{i=0}^{L_k-1} \delta \left[\hat{\alpha}_i^{(v_k)} - \hat{\alpha}_l \right] \tag{5.53}$$

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 T3-Si in the following); each step is associated with one of the blocks contained in the SPE represented in Fig. 5.5.

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., frequency 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 STDAEC-Sr in the following) and is initialised by

1. Setting its iteration index i to 0.
2. Setting
$$\mathbf{X}^{(0)}[l] \triangleq \mathbf{X}[l], \tag{5.54}$$
where
$$\mathbf{X}[l] \triangleq [X_{0,\hat{\alpha}_l}[p,q]], \tag{5.55}$$
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 frequency 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 $N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ matrix
$$\mathbf{X}^{(i)}[l] \triangleq \left[X_l^{(i)}[p,q]\right], \tag{5.56}$$
is processed to detect the strongest target contributing to it, and to compute the estimates $\hat{\theta}_i[l]$, $\hat{\phi}_i[l]$ and $\hat{C}_i[l]$ of $\theta_i[l]$, $\phi_i[l]$) and $C_i[l]$, respectively (note that this target represents the i-th one detected in the considered frequency bin, since $(i-1)$ targets have been detected in the previous recursions). This result is achieved by executing a novel iterative detection and estimation algorithm called *single target detection and angular estimation* (STDAE), whose description is provided after illustrating the overall structure of the RASCA-FR3 to ease reading.

STDAEC-S2) *Target cancellation* – The contribution $\mathbf{C}_{X_0}^{(i)}[l]$, given by the i-th target detected in the l-th frequency bin, to the vector $\mathbf{X}^{(i)}[l]$ (5.56) is computed on the basis of (5.124)–(5.125) (see Paragraph 5.5.3) and is cancelled. Cancellation consists in the computation of the new *residual* vector
$$\mathbf{X}^{(i+1)}[l] \triangleq \mathbf{X}^{(i)}[l] - \mathbf{C}_{X_0}^{(i)}[l]. \tag{5.57}$$

STDAEC-S3) *Residual energy test* – The energy
$$E^{(i+1)}[l] \triangleq \left\|\mathbf{X}^{(i+1)}[l]\right\|^2 = \sum_{p=0}^{N_{\mathrm{VH}}-1}\sum_{q=0}^{N_{\mathrm{VV}}-1}\left|X_l^{(i+1)}[p,q]\right|^2 \tag{5.58}$$
of the residual spectrum vector $\mathbf{X}^{(i+1)}[l]$ (5.57) is compared with the positive threshold T_{STDAEC} (which may depend on angular coordinates). 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 frequency 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 targets information acquired from the $\hat{\alpha}_l$-th frequency bin are collected in the set $\mathcal{T}_l$ (5.27).

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 frequency bin are computed as (see (5.15), (5.32) and (5.33))
$$\hat{R}_i[l] = \frac{c}{2\mu}\hat{f}_i[l], \tag{5.59}$$

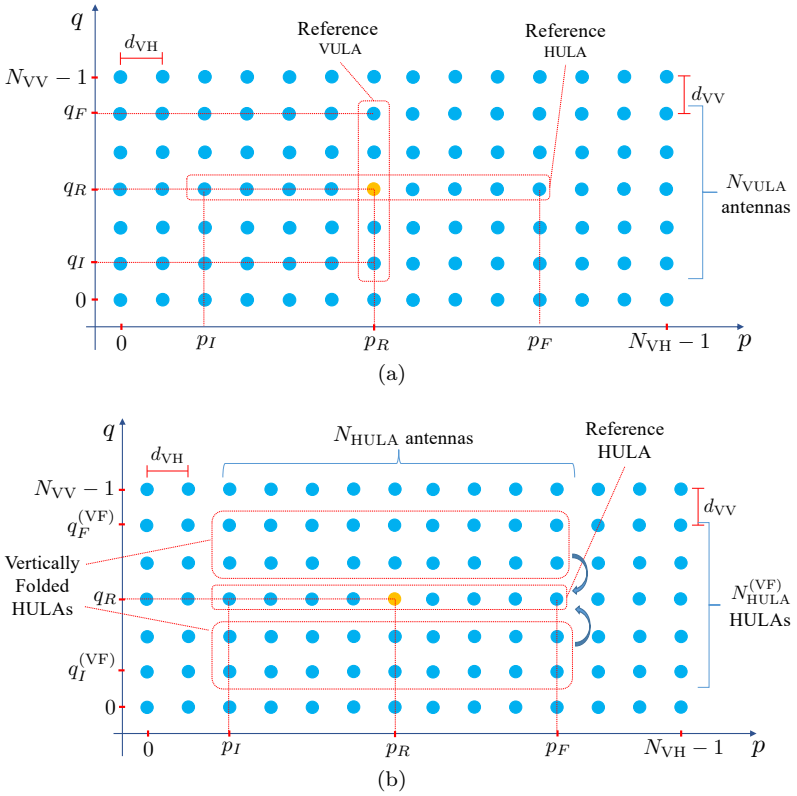


Figure 5.6: Example of reference VULA and reference HULA including the reference antenna (a) and representation of a set of vertically folded HULAs (b).

$$\hat{\phi}_i[l] = \arcsin\left(\frac{\lambda}{2d_{VV}}\hat{F}_{V,i}[l]\right) \tag{5.60}$$

and

$$\hat{\theta}_i[l] = \arcsin\left(\frac{\lambda}{2d_{VH}\cos(\hat{\phi}_i[l])}\hat{F}_{H,i}[l]\right), \tag{5.61}$$

respectively; here, $\hat{f}_i[l] = F_i[l]\, f_s$ (see (5.8)). Finally, these information are fused to generate the overall set $\mathcal{I}_t$ (5.20), describing the generated radar image; in general, this image is a cloud of $\hat{L}$ points. The set $\mathcal{I}_t$ (5.20) results from the union of all the sets $\{\mathcal{I}_t^{(l)}\}$, where

$$\mathcal{I}_t^{(l)} \triangleq \left\{\left(\hat{R}_i[l], \hat{\theta}_i[l], \hat{\phi}_i[l], \left|\hat{C}_i[l]\right|\right); i = 0, 1, \ldots, D[l]-1\right\}, \tag{5.62}$$

with $l = 0, 1, \ldots, L_b - 1$.

This concludes our description of the RASCA-FR3.

Let us focus now on the most complicated part of the STDAEC algorithm, i.e. on the STDAE algorithm. This algorithm makes use of the so called *spatial folding* (see the previous paragraph). The exploitation of this procedure in the STDAE algorithm requires:

1. Selecting a *reference* VULA, that consists of N_{VULA} adjacent and vertically aligned VAs (with $N_{VULA} \leq N_{VV}$), within the virtual array; in the following, we assume, without any loss of generality, that the reference VULA includes the reference antenna and, consequently, is identified by $p = p_R$ and $q = q_I, q_I + 1, \ldots, q_F$ (with $q_I \leq q_R \leq q_F$), so that $N_{VULA} = q_F - q_I + 1$ (see Fig. 5.6a).

2. Selecting a *reference* HULA, that consists of N_{HULA} adjacent and horizontally aligned VAs; in the following, we assume, without any loss of generality, that the reference HULA is the horizontal ULA containing the reference antenna and, consequently, is identified by $p = p_I, p_I + 1, \ldots, p_F$ (with $p_I \leq p_R \leq p_F$) and $q = q_R$, so that $N_{HULA} = p_F - p_I + 1$ (see Fig. 5.6a).

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, without any loss of generality, that these HULAs, called *vertically folded* HULAs, correspond to $q = q_I^{(\mathrm{VF})}$, $q_I^{(\mathrm{VF})}+1$, ..., q_R-1, q_R+1 , ..., $q_F^{(\mathrm{VF})}$, with $q_I^{(\mathrm{VF})} < q_R < q_F^{(\mathrm{VF})}$, as illustrated in Fig. 5.6b; note that the overall number of involved HULAs is $N_{\mathrm{HULA}}^{(\mathrm{VF})} = q_F^{(\mathrm{VF})} - q_I^{(\mathrm{VF})} + 1$.

The STDAE algorithm consists in the four steps described below (its r-th step is denoted STDAE-Sr in the following).

STDAE-S1) FFT *processing on the reference* VULA *and vertical frequency estimation* – 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

$$\begin{aligned}\mathbf{S}_{\mathrm{VULA},0}^{(i)}[l] &= \left[S_{0,0}^{(i)}[l], S_{0,1}^{(i)}[l], ..., S_{0,N_{\mathrm{VULA}}-1}^{(i)}[l]\right]^T \\ &\triangleq \left[X_l^{(i)}[p_R, q_I], ..., X_l^{(i)}[p_R, q_F]\right]^T,\end{aligned} \tag{5.63}$$

that is processed by the *complex single frequency estimator*[7] (CSFE). This algorithm detects the i-th (strongest target) appearing in $\hat{\alpha}_l$-th frequency bin and computes the estimates $\hat{C}_{\mathrm{V},i}[l]$ and $\hat{F}_{\mathrm{V},i}[l]$ of the parameters $C_i[l]$ and $F_{\mathrm{V},i}[l]$ (see (5.31)), respectively. Note that the quantity $\hat{C}_{\mathrm{V},i}[l]$ is not exploited in the following since, it represents a *preliminary estimate* of $C_i[l]$.

It is worth pointing out that the execution of the CSFE entails:

a) The evaluation of the N_{VULA}-dimensional vector

$$\mathbf{S}_{\mathrm{VULA},k}^{(i)}[l] \triangleq \left[S_{k,0}^{(i)}[l], S_{k,1}^{(i)}[l], ..., S_{k,N_{\mathrm{VULA}}-1}^{(i)}[l]\right]^T, \tag{5.64}$$

with $k = 1$ and 2; here,

$$S_{k,p}^{(i)}[l] \triangleq p^k \; X_{\mathrm{VULA},p}^{(i)}[l] = p^k \; X_l^{(i)}[p_R, q_I + p] \tag{5.65}$$

with $p = 0$, 1, ..., $N_{\mathrm{VULA}} - 1$.

b) The computation of an $\bar{N}_0$ order FFT of the vector $\bar{\mathbf{S}}_{\mathrm{VULA},k}^{(i)}[l]$, that represents a zero padded version of the vector $\mathbf{S}_{\mathrm{VULA},k}^{(i)}[l]$ (with $k = 0, 1$ and 2); here,

$$\bar{N}_0 \triangleq \bar{M} \cdot N_{\mathrm{VULA}} \tag{5.66}$$

and $\bar{M}$ represents the adopted oversampling factor. This produces the vector

$$\begin{aligned}\mathbf{s}_{\mathrm{VULA},k}^{(i)}[l] &= \left[s_{k,0}^{(i)}[l], s_{k,1}^{(i)}[l], ..., s_{k,\bar{N}_0-1}^{(i)}[l]\right]^T \\ &\triangleq \mathrm{DFT}_{\bar{N}_0}\left[\bar{\mathbf{S}}_{\mathrm{VULA},k}^{(i)}[l]\right],\end{aligned} \tag{5.67}$$

with $k = 0$, 1 and 2. Note that the m-th element of the vector $\mathbf{s}_{\mathrm{VULA},k}^{(i)}[l]$ can be expressed as

$$s_{k,m}^{(i)}[l] \triangleq \frac{1}{N_{\mathrm{VULA}}} \sum_{p=0}^{N_{\mathrm{VULA}}-1} S_{k,p}^{(i)}[l] \exp\left(-j\frac{2\pi pm}{\bar{N}_0}\right), \tag{5.68}$$

with $m = 0$, 1, ..., $\bar{N}_0 - 1$.

STDAE-S2) *Vertical folding* – The estimate $\hat{F}_{\mathrm{V},i}[l]$ of the normalised vertical frequency $F_{\mathrm{V},i}[l]$ (5.31) is employed to compensate for the phase difference between each of the HULAs selected for vertical folding and the reference HULA (i.e., for the phase differences along the

[7] A detailed description of this estimator is provided in Paragraph 5.5.2. Note that this algorithm represents the complex counterpart of the SFE, in the sense that the former is fed by a complex sequence, whereas the latter by a real one.

vertical direction), so that the spectral information associated with all these HULAs can be combined (i.e., summed) in a constructive fashion. To this aim, the *phase rotation factor*

$$R_i^{(\mathrm{VF})}[l,q] \triangleq \left[\exp\left(-j\Delta\psi_i^{(\mathrm{VF})}[l]\right)\right]^{q-q_R}, \tag{5.69}$$

with

$$\Delta\psi_i^{(\mathrm{VF})}[l] \triangleq 2\pi\hat{F}_{\mathrm{V},i}[l], \tag{5.70}$$

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

$$\mathbf{X}_i^{(\mathrm{VF})}[l] = \mathbf{X}^{(i)}[l,q_R] + \sum_{\substack{q=q_I^{(\mathrm{VF})} \\ q\neq q_R}}^{q_F^{(\mathrm{VF})}} \mathbf{X}^{(i)}[l,q]\ R_i^{(\mathrm{VF})}[l,q], \tag{5.71}$$

that collects the values taken on by the N_{HULA} vertically folded spectra referring to the $\hat{\alpha}_l$-th frequency bin; here,

$$\mathbf{X}^{(i)}[l,q] \triangleq \left[X_l^{(i)}[p_I,q], X_l^{(i)}[p_I+1,q], ..., X_l^{(i)}[p_F,q]\right]^T, \tag{5.72}$$

is a N_{HULA}-dimensional row vector extracted from the q-th row of the matrix $\mathbf{X}^{(i)}[l]$ (5.56).

STDAE-S3) FFT *processing and horizontal frequency estimation* – The processing accomplished in this step is very similar to that carried out in **STDAE-S1**. In fact, the only difference is represented by the fact that the N_{VULA}-dimensional vector $\mathbf{S}_{\mathrm{VULA},0}^{(i)}[l]$ (5.63) is replaced by the N_{HULA}-dimensional vector $\mathbf{X}_i^{(\mathrm{VF})}[l]$ (5.71) generated in the previous step. Therefore, in this case, the CSFE algorithm is exploited to compute the estimate $\hat{F}_{\mathrm{H},i}[l]$ of the horizontal frequency $F_{\mathrm{H},i}[l]$ (5.30) and a new estimate, denoted $\hat{C}_{\mathrm{H},i}[l]$, of the complex amplitude $C_i[l]$ associated with the i-th target. Note that: a) in general, an order different from $\bar{N}_0$ (5.66) can be selected for the three DFTs computed by the CSFE algorithm in this step; b) the quantity $\hat{C}_{\mathrm{H},i}[l]$ is not exploited in the following since, it represents a preliminary estimate of $C_i[l]$; c) the estimates $\hat{C}_{\mathrm{V},i}[l]$ and $\hat{C}_{\mathrm{H},i}[l]$ can be significantly different if multiple targets having similar horizontal frequencies or similar vertical frequencies contribute to the considered frequency bin.

STDAE-S4) *Overall folding and frequency/amplitude estimation* – In this step, the angular information i.e., the frequencies $\hat{F}_{\mathrm{V},i}[l]$ and $\hat{F}_{\mathrm{H},i}[l]$ computed in **STDAE-S2** and **STDAE-S3**, respectively, are exploited to accomplish overall folding[8]; 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

$$\mathbf{X}_{0,\mathrm{OF}}[l] \triangleq \sum_{p=0}^{N_{\mathrm{VH}}-1}\sum_{q=0}^{N_{\mathrm{VV}}-1} \mathbf{X}_0[p,q]\ R_i^{(\mathrm{HV})}[l,p,q], \tag{5.73}$$

where

$$R_i^{(\mathrm{HV})}[l,p,q] \triangleq R_i^{(\mathrm{VF})}[l,q]\ R_i^{(\mathrm{HF})}[l,p] \tag{5.74}$$

is a phase rotation factor, $R_i^{(\mathrm{VF})}[l,q]$ is expressed by (5.69),

$$R_i^{(\mathrm{HF})}[l,p] \triangleq \left[\exp\left(-j\Delta\psi_i^{(\mathrm{HF})}[l]\right)\right]^{p-p_R} \tag{5.75}$$

and

$$\Delta\psi_i^{(\mathrm{HF})}[l] \triangleq 2\pi\hat{F}_{\mathrm{H},i}[l]; \tag{5.76}$$

[8]As already mentioned above, a portion of the whole virtual array can be exploited to mitigate the impact of the estimation errors affecting these spatial frequencies.

note that $R_i^{(\mathrm{HV})}[l,p,q] = 1$ if $p = p_R$ and $q = q_R$. Given $\mathbf{X}_{0,\mathrm{OF}}[l]$ (5.73), the sequence of the absolute values of its elements is analysed to verify the presence of a peak in the $\hat{\alpha}_l$-th frequency bin or in a bin close to it. To this aim, after evaluating

$$\hat{\alpha}_{\mathrm{OF}} \triangleq \arg \max_{\tilde{a}\in\{0,1,\ldots,N_0-1\}} |\mathbf{X}_{0,\mathrm{OF}}[\tilde{a}]|\,, \tag{5.77}$$

the quantity $d_{\hat{\alpha}}[l] \triangleq |\hat{\alpha}_{\mathrm{OF}} - \hat{\alpha}_l|$ is compared with the positive threshold T_{OF}. If $d_{\hat{\alpha}}[l]$ exceeds T_{OF}, the presence of a ghost target is detected; otherwise, the N_0-dimensional vector

$$\mathbf{X}_{m,\mathrm{OF}}[l] \triangleq \sum_{p=0}^{N_{\mathrm{VH}}-1} \sum_{q=0}^{N_{\mathrm{VV}}-1} \mathbf{X}_m[p,q]\ R_i^{(\mathrm{HV})}[l,p,q]\,, \tag{5.78}$$

is computed for $m = 1$ and 2, and the CSFE algorithm[9] is run to estimate, on the basis of the triad ($\mathbf{X}_{0,\mathrm{OF}}[l]$, $\mathbf{X}_{1,\mathrm{OF}}[l]$, $\mathbf{X}_{2,\mathrm{OF}}[l]$), the *final estimates* $\hat{F}_i[l]$ and $\hat{C}_i[l]$ of the parameters $F_i[l]$ and $C_i[l]$, respectively; these parameters characterize the i-th target detected in the $\hat{\alpha}_l$-th frequency bin. Note that the integer part (see (5.46))

$$\hat{\alpha}_i[l] \triangleq \left\lfloor \hat{F}_i[l]/F_{\mathrm{DFT}} \right\rfloor \tag{5.79}$$

of $\hat{F}_i[l]$ does not necessarily coincide with $\hat{\alpha}_l$ but, if it differs, it is certainly close to $\hat{\alpha}_l$. If $\hat{\alpha}_i[l]$ is different from $\hat{\alpha}_l$ and appears in one of the couples forming the set $\mathcal{S}_{\mathrm{RPE}}$ (5.26), it is discarded, because the corresponding frequency bin is already being analysed by one of the other STDAEC algorithms. Otherwise, the new couple

$$(\hat{\alpha}_i[l]\,, E_{b,L_b})\,, \tag{5.80}$$

where $E_{b,L_b} \triangleq |\hat{C}_i[l]|^2$, is added to the set $\mathcal{S}_{\mathrm{RPE}}$ and the number of its elements (i.e., L_b) is increased by one. This means that an additional STDAEC algorithm is run on the (new) $\hat{\alpha}_i[l]$-th bin.

This concludes our description of the STDAE algorithm and, consequently, of the RASCA-FR3, whose overall structure is summarised in Algorithm 7.

The RASCA-FC3 can be easily obtained from RASCA-FR3 by: a) replacing $C_i^{(v_k)}$ and $\hat{C}_i^{(v_k)}$ with $A_i^{(v_k)}$ and $\hat{A}_i^{(v_k)}$, respectively (see **STDREC-S1**); b) replacing the SFE with the CSFE in **STDREC-S1**; c) computing the vectors of the triad ($\mathbf{C}_{X_0}^{(v_k)}[i]$, $\mathbf{C}_{X_1}^{(v_k)}[i]$, $\mathbf{C}_{X_2}^{(v_k)}[i]$) on the basis of (5.120)–(5.122) (see Paragraph 5.5.3) in **STDREC-S2**; d) replacing $C_i[l]$ and $\hat{C}_i[l]$ with $\hat{A}_i[l]$ and $A_i[l]$, respectively (see **STDAEC-S1**); e) replacing $\hat{C}_{\mathrm{V},i}[l]$ and $\hat{C}_{\mathrm{H},i}[l]$ with $\hat{A}_{\mathrm{V},i}[l]$ and $\hat{A}_{\mathrm{H},i}[l]$, respectively (see **STDAE-S1** and **STDAE-S3**).

Additional comments

The structure of the RASCA-FR3 (RASCA-FC3) 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 vectors $\{\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)}\}$ is computed by executing a N_0 order FFT. The vector $\mathbf{X}_0^{(v)}$ collects N_0 equally spaced samples of the spectrum of the sequence $\{x_{z,n}^{(v)}\}$ acquired on the v-th VA (see (5.37), (5.42) and (5.14)). 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 $N_A < N_{\mathrm{VR}}$) is not recommended.

[9]Note that our general description of the CSFE includes the computation of three order N_0 DFTs, that, in this case, are already available, being represented by $\{\mathbf{X}_{m,\mathrm{OF}}[l];\ m = 0,\ 1,\ 2\}$.

Algorithm 7: *Range & angle serial cancellation algorithm* for an FMCW radar system (real case)

T1 – FFT Processing:
Compute the vectors $\mathbf{x}_{0,\mathrm{ZP}}^{(v)}$, $\mathbf{x}_{1,\mathrm{ZP}}^{(v)}$ and $\mathbf{x}_{2,\mathrm{ZP}}^{(v)}$ according to (5.38)–(5.40); then, compute the triad $\{\mathbf{X}_0^{(v)}, \mathbf{X}_1^{(v)}, \mathbf{X}_2^{(v)}\}$ according to (5.42).

T2 – RPE:
S1) Extract N_A VAs from all the available VAs; then, build the set $\bar{\mathcal{S}}_{\mathrm{FFT}}$ (5.23).
for $k = 0$ *to* $N_A - 1$ **do**
  S2) Set $\mathbf{X}_m^{(v_k)}[0] \triangleq \mathbf{X}_m^{(v_k)}$ for $m = 0, 1, 2$ (see (5.44)); then, set the iteration index $i = 0$ and compute the initial energy $E_0^{(v_k)}$ according to (5.49).
  while $E_i^{(v_k)} > T_{\mathrm{STDREC}}$ **do**
    STDREC-S1) Compute the couple $(\hat{C}_i^{(v_k)}, \hat{F}_i^{(v_k)})$ running the SFE algorithm on the triad $(\mathbf{X}_0^{(v_k)}[i], \mathbf{X}_1^{(v_k)}[i], \mathbf{X}_2^{(v_k)}[i])$.
    STDREC-S2) Compute the vectors $(\mathbf{C}_{X_0}^{(v_k)}[i], \mathbf{C}_{X_1}^{(v_k)}[i], \mathbf{C}_{X_2}^{(v_k)}[i])$ according to (5.110)–(5.112); then, compute the new residual triad $(\mathbf{X}_0^{(v_k)}[i+1], \mathbf{X}_1^{(v_k)}[i+1], \mathbf{X}_2^{(v_k)}[i+1])$ according to (5.48).
    STDREC-S3) Compute the residual energy $E_{i+1}^{(v_k)}$ according to (5.49).
  end
end
S3) Build the set $\mathcal{S}_{\mathrm{RPE}}$ (5.26) (see (5.51) and (5.53)).

T3 – SPE:
S1) Set the iteration index $i = 0$ and set the initial vector $\mathbf{X}^{(0)}[l]$ according to (5.54); then, compute the initial energy $E^{(0)}[l]$ according to (5.58).
Parallel For $l = 0$ *to* $l = L_b - 1$ **do**
  while $E^{(i)}[l] > T_{\mathrm{STDAEC}}$ **do**
    STDAEC-S1) Compute the couple $(\hat{C}_{\mathrm{V},i}[l], \hat{F}_{\mathrm{V},i}[l])$ running the CSFE algorithm fed by the vector $\mathbf{S}_{\mathrm{VULA},0}^{(i)}[l]$ evaluated according to (5.63). Then, compute the phase rotation factor $R_i^{(\mathrm{VF})}[l,q]$ and the matrix $\mathbf{X}_i^{(\mathrm{VF})}[l]$ according to (5.69) and (5.71), respectively. Then, run the CSFE algorithm to compute the couple $(\hat{C}_{\mathrm{H},i}[l], \hat{F}_{\mathrm{H},i}[l])$ and compute the phase rotation factor $R_i^{(\mathrm{HF})}[l,p]$ according to (5.75). Finally, compute the vectors $\{\mathbf{X}_{m,\mathrm{OF}}[l]; m = 0, 1, 2\}$ according to (5.73) and (5.78) and run the CSFE algorithm fed by the set $\{\mathbf{X}_{m,\mathrm{OF}}[l]; m = 0, 1, 2\}$ to evaluate the couple $(\hat{C}_i[l], \hat{F}_i[l])$.
    STDAEC-S2) Compute the vector $\mathbf{C}_{X_0}^{(i)}[l]$ according to (5.124)–(5.125); then compute the new residual vector $\mathbf{X}^{(i+1)}[l]$ according to (5.57).
    STDAEC-S3) Compute the residual energy $E^{(i+1)}[l]$ according to (5.58).
  end
  S2) Compute $\hat{R}_i[l]$, $\hat{\phi}_i[l]$, $\hat{\theta}_i[l]$ according to (5.59)–(5.61).
end

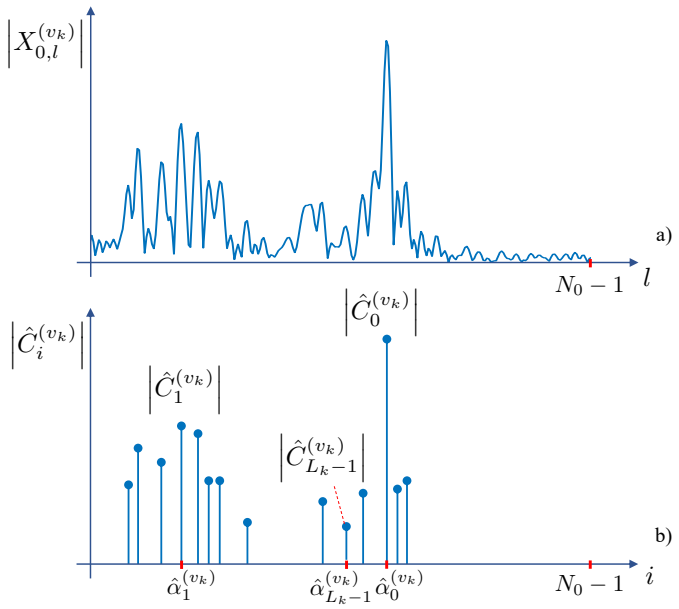


Figure 5.7: Representation of: a) the absolute value of the elements of the vector $\mathbf{X}_0^{(v_k)}$; b) the corresponding discrete amplitude-range profile generated by the STDREC algorithm (the RASCA-FC3 is considered).

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 normalised frequency $F_i^{(v_k)}$ and of the complex amplitude $C_i^{(v_k)}$ ($A_i^{(v_k)}$) (see (5.7) and (5.13)) plays an important role in this step, since these parameters are exploited in the serial cancellation procedure based on (5.48). In particular, ignoring the frequency residual $\delta_i^{(v_k)}$ of the normalised frequency $F_i^{(v_k)}$ (5.45) in this procedure (i.e., assuming that $\hat{F}_i^{(v_k)} = \hat{\alpha}_i^{(v_k)}$; see (5.46)) may result in a significant *error accumulation*.

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 triads $\{(\hat{\alpha}_i^{(v_k)}, \hat{F}_i^{(v_k)}, \hat{C}_i^{(v_k)});\ i = 0, 1, \ldots, L_k - 1\}$ ($\{(\hat{\alpha}_i^{(v_k)}, \hat{F}_i^{(v_k)}, \hat{A}_i^{(v_k)});\ i = 0, 1, \ldots, L_k - 1\}$), 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 *frequency bins* (i.e., to N_0 distinct *range bins*) and is usually *dense* in the presence of multiple extended targets, as illustrated in Fig. 5.7-a) (where the absolute value of its elements is represented). The STDREC allows to extract a *discrete frequency* (i.e., *range*) *profile* from the vector $\mathbf{X}_0^{(v_k)}$, as illustrated in Fig. 5.7-b). 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., frequency bins). The *range profile* characterizing the

v_k-th VA is described by the set of L_k couples $\mathcal{S}_{v_k} = \{(\hat{\alpha}_i^{(v_k)}, \hat{C}_i^{(v_k)});\ i = 0,\ 1,\ \ldots,\ L_k - 1\}$ ($\mathcal{S}_{v_k} = \{(\hat{\alpha}_i^{(v_k)}, \hat{A}_i^{(v_k)});\ i = 0,\ 1,\ \ldots,\ L_k - 1\}$), with $k = 0,\ 1,\ \ldots\ ,\ N_A - 1$; the parameter $\hat{\alpha}_i^{(v_k)}$ identifies the frequency bin associated with the i-th target detected on the considered VA, whereas the absolute value of $\hat{C}_i^{(v_k)}$ ($\hat{A}_i^{(v_k)}$) 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.

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 SFE (CSFE) algorithm in parallel. Each of these instances is initialised with the frequency corresponding to 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 (5.42)). In this case, a constraint is set on the minimum spacing between the $m_i^{(v_k)}$ detected frequencies 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 frequencies are discarded. It is also worth stressing that, if a *cluster* of $m_i^{(v_k)}$ distinct frequencies is estimated, each of the components of the triad $(\mathbf{C}_{X_0}^{(v_k)}[i], \mathbf{C}_{X_1}^{(v_k)}[i], \mathbf{C}_{X_2}^{(v_k)}[i])$ appearing in the RHS of (5.48) consists of the sum of $m_i^{(v_k)}$ terms, each associated with one of these frequencies.

g) The STDREC algorithm employed in the RASCA-FR3 (RASCA-FC3) represents an instance of the *single frequency estimation and cancellation* (*complex single frequency estimation and cancellation*) algorithm derived in [1] for the estimation of multiple overlapped *real* (*complex*) tones. For this reason, in the case of complex received signals, it can be replaced by one of the *multiple tone estimators* available in the technical literature, like the CFH algorithm [45], the algorithm developed by *Ye* and *Aboutanios* in [43, 44] and the algorithm derived by *Serbes* in [46] (the last two algorithms are denoted Alg-YA and Alg-S, respectively, in the following). In fact, all these algorithms are recursive and rely on a serial cancellation procedure since, within each recursion, they detect a single tone, estimate its parameters and subtract its contribution from the residual signal emerging from the previous iteration.

h) The estimates generated by the STDREC algorithm are potentially biased if the parameters of the SFE (CSFE) executed in its first step are not properly selected (see [1]). In principle, this bias can be arbitrarily reduced by increasing the overall number of iterations and/or re-estimations accomplished by the SFE (CSFE). However, we found out that, in the case of complex received signal, a computationally efficient alternative to this approach is represented by running an additional step (i.e., STDREC-S4) after that the first three steps of the STDREC algorithm has been carried out. In this final step, the Alg-YA is run after initializing it with the estimates of the normalised frequencies and the associated complex amplitudes generated by the STDREC. The hybrid technique that results 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 CSFE proposed in [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 frequencies (see (5.30) and (5.31)) of multiple targets contributing to the $\hat{\alpha}_l$-th frequency bin can be detected by first computing a 2D

DFT of the matrix $\mathbf{X}[l]$ (5.55) and, then, by looking for 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 FFT 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 FFTs. Consequently, relevant spatial frequencies are estimated by searching for the peaks of 1D amplitude spectra (i.e., in the absolute values of the elements of the vectors $\mathbf{S}^{(i)}_{\mathrm{VULA},0}[l]$ and $\mathbf{X}^{(\mathrm{VF})}_{i}[l]$); in other words, an AM approach is adopted. Note that this approach allow us to mitigate the overall computational complexity and to detect weak targets hidden by close strong targets through successive cancellations.

b) In **STDAE-S1**, each of the three vectors $\{\mathbf{s}^{(i)}_{\mathrm{VULA},k}[l];\, k = 0,\, 1,\, 2\}$ is computed by executing a $\bar{N}_0$ order FFT (see (5.67)). 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 (5.65)). On the other hand, the k-th vector $\mathbf{s}^{(i)}_{\mathrm{VULA},k}[l]$ (with $k = 1$ and 2) 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-S3** is very similar to that carried out in **STDAE-S1**. In fact, the only difference is represented by the fact that the N_{VULA}-dimensional vector $\mathbf{S}^{(i)}_{\mathrm{VULA},0}[l]$ (5.63) is replaced by the N_{HULA}-dimensional vector $\mathbf{X}^{(\mathrm{VF})}_{i}[l]$ (5.71) generated in **STDAE-S2**. Therefore, in this case, the CSFE is exploited to estimate the horizontal frequency $F_{\mathrm{H},i}[l]$ and, again, the complex amplitude $C_i[l]$ ($A_i[l]$) associated with the i-th target.

d) Similarly as the STDREC algorithm, the STDAEC algorithm can also be considered as an instance of the CSFEC algorithm mentioned at point g) of **T2-S2**. Therefore, in principle, it can be replaced by the CFH algorithm [45], the Alg-YA [43, 44] or the Alg-S [46]. Moreover, a further (and final) step, based the Alg-YA can be added to the STDAEC algorithm to mitigate its estimation bias.

e) 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 CSFE algorithm in parallel.

Our final comments concern the use of RASCA-FR3 and RASCA-FC3 in FMCW radar systems whose virtual antenna array is not an URA; for instance, in our experimental work (see Section 5.9), a colocated MIMO FMCW radar equipped with the virtual receive array shown in Fig. 5.8 has been employed. Note that the first two processing tasks of the RASCAs are carried out on an antenna-by-antenna basis; therefore, they are not influenced by the shape of the considered virtual array. However, this shape influences the way spatial folding is accomplished in **T3**. More specifically, as far as the last point is concerned, the following considerations can be made:

1) The array structure represented in Fig. 5.8 can be treated as an URA if its gaps are zero-padded.

2) The reference VULA should be selected in a way to maximize the number of non-zero vertically aligned VAs and, consequently, the number of VAs contributing to the estimation of the elevation angle, as illustrated in Fig. 5.8.

3) The reference HULA should be selected in the middle of the antenna array; this mitigates the effects of the errors affecting the estimate of normalised vertical frequency in the vertical folding procedure (do not forget that such errors may have a significant impact on the contributions of the HULAs that are farther from the reference HULA; see (5.69)).

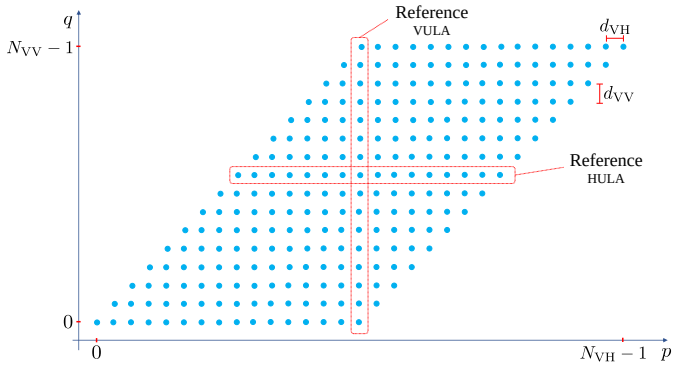


Figure 5.8: Virtual array considered in our experimental work.

4) The vertical folding accomplished by the STDAE algorithm involves VULAs of different sizes. More specifically, in the i-th iteration of the STDAEC algorithm, vertical folding is accomplished by computing the N_{HULA}-dimensional vector (see (5.56) and **STDAE-S2**)

$$\mathbf{X}_i^{(\mathrm{VF})}[l] = \left[X_i^{(\mathrm{VF})}[p_I, l], X_l^{(i)}[p_I+1, l], ..., X_l^{(i)}[p_F, l]\right]^T \tag{5.81}$$

where

$$X_i^{(\mathrm{VF})}[p, l] = \frac{1}{N_V[p]} \sum_{\substack{q=q_I[p] \\ q \neq q_R}}^{q_F[p]} X_l^{(i)}[p, q] \; R_i^{(\mathrm{VF})}[l, q] + \frac{X_l^{(i)}[p, q_R]}{N_V[p]}, \tag{5.82}$$

with $p = p_I$, p_I+1, ..., p_F; here, $R_i^{(\mathrm{VF})}[l, q]$ is expressed by (5.69), $q_I[p]$ ($q_F[p]$) is the index identifying the first (last) antenna of the p-th VULA and $N_{\mathrm{V}}[p]$ is the overall number of VAs which that VULA consists of.

5.4.4 Range & angle serial cancellation algorithms for a radar system endowed with a one-dimensional antenna array

The algorithms described in the previous paragraph can be easily adapted to the case in which the considered colocated MIMO radar system is equipped with a single ULA and, consequently, can be exploited for 2D imaging only; this leads to RASCA-FR2 and RASCA-FC2. The changes made in RASCA-FR3 and RASCA-FC3 to obtain RASCA-FR2 and RASCA-FC2, respectively, 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 $\mathbf{X}^{(i)}[l]$ (5.56) is replaced by the N_{VH}-dimensional vector

$$\mathbf{X}^{(i)}[l] \triangleq \left[X_l^{(i)}[p]\right], \tag{5.83}$$

   collecting the spectral information available on the whole virtual receive array and referring to the $\hat{\alpha}_l$-th frequency bin only.

2. The spatial frequency $\hat{F}_{\mathrm{V},i}[l]$ is unavailable and, therefore, it is not included in the set $\mathcal{T}_l$ (5.27); note that the elevation angle $\hat{\phi}_i[l]$ (5.60) is not estimated in this case.

5.5 Description of Various Algorithms Employed in the Proposed Embodiments

In this section, various mathematical details about the techniques employed in the RASCAs are provided.

Since the processing accomplished in **T1** of the RASCAs has been fully analysed in the previous section, in this paragraph we provide a detailed description of: a) the SFE (see **T2-S2**); b) the CSFE (see **T2-S2** and **T3-S1**); c) the target cancellation procedures employed in **T2-S3** and **T3-S1**.

5.5.1 Single frequency estimator

In this Paragraph, the SFE derived in [1] is summarized. This algorithm processes the samples of the real sequence $\{x_{r,n};\, n = 0,\, 1,\, ...,\, N-1\}$, whose n-th element is

$$\begin{aligned} x_{r,n} &= a\cos\left(2\pi n\, F\, + \psi\right) + w_{r,n} \\ &= C\exp\left(j2\pi n\, F\right) + C^{*}\exp\left(-j2\pi n\, F\right) + w_{r,n}, \end{aligned} \tag{5.84}$$

with $n = 0,\, 1,\, ...,\, N-1$, and generates an estimate of the *normalised frequency* F and of the *complex amplitude*

$$C \triangleq \frac{1}{2} a\exp\left(j\psi\right) \tag{5.85}$$

of the real tone appearing in the RHS of (5.84); here, N is the overall number of elements of the sequence $\{x_{r,n}\}$, a and ψ are the tone amplitude and phase, respectively, and $\{w_{r,n};\, n = 0,\, 1,\, ...,\, N-1\}$ is a real AWGN sequence. This algorithm is initialised by

1) Evaluating: a) the vector

$$\mathbf{X}_0 = \left[X_{0,0}, X_{0,1}^{(v)}, ..., X_{0,N_0-1}^{(v)}\right]^T \triangleq \mathrm{DFT}_{N_0}\left[\mathbf{x}_{0,\mathrm{ZP}}^{(v)}\right], \tag{5.86}$$

where the DFT order N_0 is defined by (5.41),

$$\mathbf{x}_{0,\mathrm{ZP}} \triangleq \left[\left(\mathbf{x}_0\right)^T\ \mathbf{0}_{(M-1)N}^T\right]^T, \tag{5.87}$$

M is the *oversampling factor* and

$$\mathbf{x}_0 \triangleq \left[x_{r,0}, x_{r,1}, ..., x_{r,N-1}\right]^T; \tag{5.88}$$

b) the initial coarse estimate $\hat{F}_c^{(0)}$ of F as

$$\hat{F}_c^{(0)} = \hat{\alpha}\, F_{\mathrm{DFT}}, \tag{5.89}$$

where the integer $\hat{\alpha}$ is computed as

$$\hat{\alpha} = \arg \max_{\tilde{\alpha}\in\{0,1,...,N_0/2-1\}} \left|\bar{X}_{0,\tilde{\alpha}}\right|; \tag{5.90}$$

c) the quantity

$$\hat{\rho}^{(0)} \triangleq \frac{\hat{F}_c^{(0)}}{F_{\mathrm{DFT}}} = \hat{\alpha}; \tag{5.91}$$

d) the initial estimate $\hat{C}^{(0)}$ of C as

$$\hat{C}^{(0)} = G\big(\hat{F}_c^{(0)}\big) \tag{5.92}$$

where

$$G\big(\tilde{F}\big) \triangleq \frac{\bar{X}(\tilde{F}) - \bar{X}^{*}(\tilde{F}) g(\tilde{F})}{1 - \left|g(\tilde{F})\right|^2}, \tag{5.93}$$

$$\bar{X}(\tilde{F}) \triangleq \frac{1}{N}\sum_{n=0}^{N-1} x_{r,n}\exp\left(-j2\pi n\tilde{F}\right) \tag{5.94}$$

and

$$g(\tilde{F}) \triangleq \frac{1}{N}\sum_{n=0}^{N-1}\exp\left(-j4\pi n\tilde{F}\right); \tag{5.95}$$

e) the spectral coefficients $\bar{X}_{1,\hat{\alpha}}$ and $\bar{X}_{2,\hat{\alpha}}$, and the coefficients $\{K_p(2\hat{\alpha});\ p=1,\ 2\}$ and $\{b(\hat{\alpha}),\ c(\hat{\alpha})\}$ on the basis of the definitions

$$\bar{X}_{k,\rho} \triangleq \frac{1}{N}\sum_{n=0}^{N-1} x_{k,n}\exp\left(-j\frac{2\pi n\rho}{N_0}\right), \tag{5.96}$$

$$K_p(x) \triangleq \frac{1}{N}\sum_{n=0}^{N-1} g_p[n]\exp\left(-j\frac{2\pi nx}{N_0}\right), \tag{5.97}$$

$$b(\rho) \triangleq -\Re\left\{\hat{C}^*\bar{X}_{2,\rho}\right\} + 2\Re\left\{\left(\hat{C}^*\right)^2 K_2(2\rho)\right\} \tag{5.98}$$

and

$$c(\rho) \triangleq \Im\left\{\hat{C}^*\bar{X}_{1,\rho}\right\} - \Im\left\{\left(\hat{C}^*\right)^2 K_1(2\rho)\right\}, \tag{5.99}$$

respectively; f) the initial estimate $\hat{\Delta}^{(0)}$ of Δ as

$$\hat{\Delta}^{(0)} = P\big(\hat{\rho}^{(0)}\big), \tag{5.100}$$

where

$$P(\tilde{\rho}) \triangleq -c(\tilde{\rho})/b(\tilde{\rho}); \tag{5.101}$$

g) the first fine estimate $\hat{F}^{(0)}$ of F as

$$\hat{F}^{(0)} = \hat{F}_c^{(0)} + \frac{\hat{\Delta}^{(0)}}{2\pi} \tag{5.102}$$

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 $\hat{C}^{(i-1)}$ of F and C, respectively, and produces the new estimates $\hat{F}^{(i)}$ and $\hat{C}^{(i)}$ of the same quantities (with $i = 1, 2, ..., N_{\mathrm{SFE}}$, where N_{SFE} represents the overall number of iterations); the procedure employed for the evaluation of $\hat{F}^{(i)}$ and $\hat{C}^{(i)}$ consists of the two steps described below (the p-th step is denoted SFE-Sp).

SFE-S1) - The new estimate $\hat{\Delta}^{(i)}$ of Δ is computed as[10] (see (5.100)–(5.101))

$$\hat{\Delta}^{(i)} = P(\hat{\rho}^{(i-1)}) = -c\big(\hat{\rho}^{(i-1)}\big)/b\big(\hat{\rho}^{(i-1)}\big); \tag{5.103}$$

in the evaluation of the coefficients $\{b(\rho),\ c(\rho)\}$ appearing in the RHS of (5.101), $\hat{C} = \hat{C}^{(i-1)}$ and

$$\rho = \hat{\rho}^{(i-1)} \triangleq \hat{F}^{(i-1)}/F_{\mathrm{DFT}} \tag{5.104}$$

are assumed. Then,

$$\hat{F}^{(i)} = \hat{F}^{(i-1)} + \hat{\Delta}^{(i)}/(2\pi) \tag{5.105}$$

is evaluated.

SFE-S2) - The new estimate $\hat{C}^{(i)}$ of $\hat{C}$ is evaluated as $\hat{C}^{(i)} = G(\hat{F}^{(i)})$ (see (5.92)–(5.93)). Moreover, the index i 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 $\hat{F} = \hat{F}^{(N_{\mathrm{SFE}})}$ and $\hat{C} = \hat{C}^{(N_{\mathrm{SFE}})}$ of F and C, respectively, become available.

[10]The quantities $\{\bar{X}_{k,\rho}; k = 1, 2\}$ required in the computation of the coefficients $b(\rho)$ and $c(\rho)$ can be also evaluated by means of the interpolation-based method illustrated in [1, Sect. III, p. 12]. In our work, *barycentric interpolation* has been always used [61]; in the following, the parameter I represents the interpolation order. These considerations hold also for the CSFE described below.

5.5.2 Complex single frequency estimator

All the results illustrated in the previous paragraph refer to the real sequence $\{x_{r,n}\}$, whose n-th element is expressed by (5.84). However, a similar estimation method (namely, the CSFE) has been developed for the complex counterpart, i.e. for a complex sequence $\{x_{c,n};\ n = 0,\ 1,\ ...,\ N-1\}$, whose n-th element is

$$x_{c,n} = A \exp\left(j2\pi n\, F\right) + w_{c,n}, \tag{5.106}$$

with $n = 0,\ 1,\ ...,\ N-1$. Here, A is the complex amplitude of the single tone appearing in the RHS of the last equation, $\{w_{c,n};\ n = 0,\ 1,\ ...,\ N-1\}$ is a complex AWGN sequence and all the parameters have exactly the same meaning as that illustrated for (5.84). The description of the CSFE is similar to that illustrated for the SFE in the previous paragraph, the only differences being represented by the fact that: a) the parameter C (5.85) is replaced by A; b) (5.92) is replaced by

$$\tilde{A} = \hat{A} = \bar{X}(\hat{F}), \tag{5.107}$$

where $\bar{X}(\hat{F})$ is computed according to (5.94) (in which $x_{r,n}$ is replaced by $x_{c,n}$ (5.106)); c) (5.98) and (5.99) are replaced by

$$b\left(\rho\right) \triangleq \Re\{\hat{A}^* \bar{X}_{2,\rho}\} \tag{5.108}$$

and

$$c\left(\rho\right) \triangleq -\Im\{\hat{A}^* \bar{X}_{1,\rho}\}, \tag{5.109}$$

respectively.

5.5.3 Target cancellation procedures employed in FMCW radar systems

In **T2** of the RASCA-FR2 and RASCA-FR3 (and, in particular, in **STDREC-S2**; see (5.48)), a target cancellation procedure is used in combination with the SFE. This procedure requires the evaluation of the triad $(\mathbf{C}_{X_0}^{(v)}[i],\ \mathbf{C}_{X_1}^{(v)}[i],\ \mathbf{C}_{X_2}^{(v)}[i])$, that represents the contribution given by the i-th (i.e., by the last) *point target* detected on the v-th VA. If $\hat{F}_i^{(v)}$ and $\hat{C}_i^{(v)}$ denote the estimates of the normalised frequency and the complex amplitude, respectively, characterizing this target, the expressions

$$\mathbf{C}_{X_0}^{(v)}[i] = \hat{C}_i^{(v)}\ \bar{\mathbf{W}}_0^{(v)}[i] + \left(\hat{C}_i^{(v)}\right)^* \left(\bar{\mathbf{W}}_{0,c}^{(v)}[i]\right), \tag{5.110}$$

$$\mathbf{C}_{X_1}^{(v)}[i] = \hat{C}_i^{(v)}\ \bar{\mathbf{W}}_1^{(v)}[i] + \left(\hat{C}_i^{(v)}\right)^* \left(\bar{\mathbf{W}}_{1,c}^{(v)}[i]\right) \tag{5.111}$$

and

$$\mathbf{C}_{X_2}^{(v)}[i] = \hat{C}_i^{(v)}\ \ \bar{\mathbf{W}}_2^{(v)}[i] + \left(\hat{C}_i^{(v)}\right)^* \left(\bar{\mathbf{W}}_{2,c}^{(v)}[i]\right) \tag{5.112}$$

are employed; here, $\bar{\mathbf{W}}_k^{(v)}[i]$ denotes the N_0 order DFT of the vector

$$\bar{\mathbf{w}}_k^{(v)}[i] \triangleq \left[0, 1^k \cdot \bar{w}_i^{(v)}, 2^k \cdot \left(\bar{w}_i^{(v)}\right)^2, \ldots, (N-1)^k \cdot \left(\bar{w}_i^{(v)}\right)^{N-1}, 0, \ldots, 0\right]^T, \tag{5.113}$$

with $k = 0,\ 1$ and 2, $\bar{\mathbf{W}}_{k,c}^{(v)}[i]$ the N_0 order DFT of the vector $(\bar{\mathbf{w}}_k^{(v)}[i])^*$,

$$\bar{w}_i^{(v)} \triangleq \exp(j2\pi \bar{F}_i^{(v)}) \tag{5.114}$$

and

$$\bar{F}_i^{(v)} \triangleq \hat{f}_i^{(v)} T_s \tag{5.115}$$

is the normalised frequency associated with the frequency $\hat{f}_v^{(i)}$. It is important to point out that an efficient method can be used for the computation of the vectors $\bar{\mathbf{W}}_k^{(v)}[i]$ and $\bar{\mathbf{W}}_{k,c}^{(v)}[i]$ appearing in the RHS of (5.110)–(5.112) (with $k = 0,\ 1$ and 2); note that, for any k, these vectors represent

the N_0 order DFTs of the sequences $\{n^k\ (\bar{w}_i^{(v)})^n;\ n=0,\ 1,\ ...,\ N-1\}$ and $\{n^k\ ((\bar{w}_i^{(v)})^*)^n;\ n=0,\ 1,\ ...,\ N-1\}$, respectively. In fact, the l-th element of the vectors $\bar{\mathbf{W}}_k^{(v)}[i]$ and $\bar{\mathbf{W}}_{k,c}^{(v)}[i]$ is given by

$$\begin{aligned}\bar{W}_k^{(v)}[i,l] =& \frac{1}{N}\sum_{n=0}^{N-1} n^k\ \left(\bar{w}_i^{(v)}\right)^n \exp\left(-j\frac{2\pi l}{N_0}n\right)\\ =& \frac{1}{N}\sum_{n=0}^{N-1} n^k (q[l])^n\end{aligned} \tag{5.116}$$

and

$$\begin{aligned}\bar{W}_{k,c}^{(v)}[i,l] =& \frac{1}{N}\sum_{n=0}^{N-1} n^k\ \left(\left(\bar{w}_i^{(v)}\right)^*\right)^n \exp\left(-j\frac{2\pi l}{N_0}n\right)\\ =& \frac{1}{N}\sum_{n=0}^{N-1} n^k (q_c[l])^n,\end{aligned} \tag{5.117}$$

respectively, where

$$q[l] \triangleq \exp\left(j2\pi\left(\bar{F}_i^{(v)} - \frac{l}{N_0}\right)\right) \tag{5.118}$$

and

$$q_c[l] \triangleq \exp\left(j2\pi\left(-\bar{F}_i^{(v)} - \frac{l}{N_0}\right)\right). \tag{5.119}$$

Therefore, the identities listed in [1, eqs. (84)-(85) and (145)] can be exploited for an efficient computation of the RHSs of (5.116) and (5.117).

A target cancellation procedure is also employed in **T2** of the RASCA-FC2 and RASCA-FC3; however, in this case, the CSFE is adopted in place of the SFE, and the vectors $\mathbf{C}_{X_0}^{(v)}[i]$, $\mathbf{C}_{X_1}^{(v)}[i]$ and $\mathbf{C}_{X_2}^{(v)}[i]$ are evaluated as

$$\mathbf{C}_{X_0}^{(v)}[i] = \hat{A}_i^{(v)}\ \bar{\mathbf{W}}_0^{(v)}[i], \tag{5.120}$$

$$\mathbf{C}_{X_1}^{(v)}[i] = \hat{A}_i^{(v)}\ \bar{\mathbf{W}}_1^{(v)}[i] \tag{5.121}$$

and

$$\mathbf{C}_{X_2}^{(v)}[i] = \hat{A}_i^{(v)}\ \bar{\mathbf{W}}_2^{(v)}[i]; \tag{5.122}$$

respectively; here, $\bar{\mathbf{W}}_k^{(v)}[i]$ denotes the N_0 order DFT of the vector

$$\bar{\mathbf{w}}_k^{(v)}[i] \triangleq \left[0, 1^k\cdot\bar{w}_i^{(v)}, 2^k\cdot\left(\bar{w}_i^{(v)}\right)^2, \ldots, (N-1)^k\cdot\left(\bar{w}_i^{(v)}\right)^{N-1}, 0, \ldots, 0\right]^T, \tag{5.123}$$

with $k=0$, 1 and 2, and $\bar{w}_i^{(v)}$ is still expressed by (5.114). The vector $\bar{\mathbf{W}}_k^{(v)}[i]$ appearing in (5.110)–(5.112) (with $k=0$, 1 and 2) can be efficiently computed following the same approach illustrated above for the SFE.

The CSFE is also employed in **T3-S1** and, in particular, in **STDAEC-S2** of the RASCA-FR2, RASCA-FR3, RASCA-FC2 and RASCA-FC3. In this case, the cancellation procedure requires the evaluation of the contribution

$$\mathbf{C}_{X_0}^{(i)}[l] = \left[C_{X_0}^{(i)}[p,q,l]\right] \tag{5.124}$$

given by the i-th (i.e., by the last) target detected in the l-th frequency bin to the whole array (see (5.57)). Here, we focus on the target cancellation procedure employed in the above mentioned RASCAs. In this case, if $\hat{A}_i[l]$, $\hat{F}_{\mathrm{V},i}[l]$ and $\hat{F}_{\mathrm{H},i}[l]$ denote the estimates of the complex amplitude, the

normalised vertical spatial frequency and the normalised horizontal spatial frequency, respectively, characterizing the i-th target, the expression

$$C_{X_0}^{(i)}[p,q,l] = \hat{A}_i[l] \exp\left\{ j2\pi \left[(p-p_R)\hat{F}_{\mathrm{H},i}[l] + (q-q_R)\hat{F}_{\mathrm{V},i}[l] \right] \right\}, \tag{5.125}$$

is employed for any VA (i.e., for any p and q).

Finally, it is important to mention that the cancellation procedure adopted in STDREC algorithm aims at removing the contribution of a single target in each of its iterations. If a *cluster* of $m_i^{(v)}$ distinct frequencies is estimated by the SFE (CSFE) in the i-th iteration of the above mentioned algorithm, each of the components of the triad ($\mathbf{C}_{X_0}^{(v)}[i]$, $\mathbf{C}_{X_1}^{(v)}[i]$, $\mathbf{C}_{X_2}^{(v)}[i]$) consists of the sum of $m_i^{(v)}$ terms and each term is evaluated on the basis of (5.110)–(5.112) ((5.120)–(5.122)).

5.6 Limitations

In this section, some technical limitations that have emerged in the implementation of our algorithms on commercial radar devices are illustrated and the solutions we have devised to mitigate their impact are described.

5.6.1 Unequal response of virtual antennas

The derivation of the RASCAs for FMCW radar systems relies on the assumption that the real (complex) sample sequence made available by the v-th VA is expressed by (5.6) ((5.12)). The adopted signal models hold if the amplitudes of the L overlapped oscillations contributing to the useful component of the received signal do not change from antenna to antenna. However, our experiments accomplished on commercial colocated radar devices 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. We 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. It 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.

It important to point out that, in principle, the presence of this phenomenon can be accounted for in the development of target detection and estimation algorithms by including its effects in the received signal model. For instance, (5.6) can be generalised as

$$x_{r,n}^{(v)} = \sum_{l=0}^{L-1} \alpha_v(\theta_l,\phi_l)\ a_l \cos\left(2\pi n F_l^{(v)} + \psi_l^{(v)} \right) + w_{r,n}^{(v)}, \tag{5.126}$$

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. Consequently, the complex amplitude associated with the l-th target detectable on the considered VA becomes (see (5.7))

$$C_l^{(v)}(\theta_l,\phi_l) \triangleq \frac{1}{2} a_l\ \alpha_v(\theta_l,\phi_l) \exp\left(j\ \psi_l^{(v)} \right). \tag{5.127}$$

Neglecting the presence of the factor $\alpha_v(\theta_l,\ \phi_l)$ in the development of our algorithms has the following implication: an error is introduced by the STDAEC algorithm in its cancellation procedure (see **STDAEC-S2** in Paragraph 5.5). Note, in particular, that the estimate $\hat{C}_i[l]$ 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 (5.124)–(5.125)). Note that, 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 $\hat{C}_i[l]$ of the complex gain $C_i[l]$ with

$$\hat{C}_i[v, l] \triangleq \hat{C}_i[l]\ \alpha_v(\hat{\theta}_i, \hat{\phi}_i). \tag{5.128}$$

in the evaluation of the term $\mathbf{C}_{X_0}^{(i)}[l]$ appearing in (5.124)–(5.125). Estimating the function $\alpha_v(\theta, \phi)$, however, is a time consuming task, since it requires a proper measurement setup and an anechoic chamber. We believe that this problem can be circumvented by: a) exploiting *deep learning techniques* [83] in the SPE; b) adopting a *data-driven approach* [84], [85]. This solution 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 data, without prior knowledge about the parametric representation of the considered problem. Note that a fundamental role is played by the adopted training procedure since it makes the involved network able to generate correct predictions on the basis of never seen data available at its input.

In practice, the adoption of the proposed approach requires modifying the STDAEC technique employed in the RASCAs (see Fig. 5.5) 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 frequency bin (with $l = 0, 1, \ldots, L_b - 1$), so that accurate cancellation becomes possible.

The use of this solution in our radar systems is not investigated in the following, since it is out of the scope of this work.

5.6.2 Antenna coupling

In our description of the SFE and the CSFE (see Section 5.5), it has been implicitly assumed that the minimum frequency of the useful component contained in the observed data sequence can be arbitrarily small. Unluckily, this is not always true. For instance, in commercial colocated FMCW MIMO radar systems, a strong interference is observed in the lower portion of the spectrum evaluated on all the receive antennas. This phenomenon, known as *mutual coupling* [86], is due to the electromagnetic coupling that originates from the small distance between adjacent transmit and receive antennas [82]. Its impact can be mitigated resorting to various methods based on calibration measurements [87]. Because of mutual coupling, any target whose range is below a certain threshold cannot be detected by our algorithms in a reliable fashion.

5.7 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 FFT 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 [70]. 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 [12]. In the remaining part of this section, a brief description is provided for both types.

The inner structure of both types of algorithms is described by the block diagram shown in Fig. 5.9. The processing accomplished by the blocks this diagram consists of, can be summarized as follows. Each vector of the set $\{\mathbf{x}_z^{(v)}\}$, collecting N_{VR} vectors (see (5.14)), undergoes, after ZP,

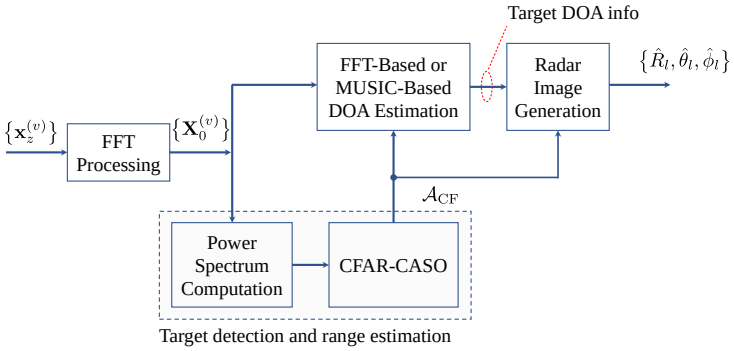


Figure 5.9: Block diagram describing the overall structure of the FFT-BAs and the MUSIC-BAs.

a N_0 order FFT; this produces a set of N_0-dimensional vectors $\{\mathbf{X}_0^{(v)}\}$ (see (5.38), (5.42) and (5.43)). Based on this set of vectors, the N_0-dimensional power spectrum

$$\mathbf{P}_0 = [P_{0,0}, P_{0,1}, ..., P_{0,N_0-1}]^T \tag{5.129}$$

is computed; here,

$$P_{0,i} \triangleq \frac{1}{N_{\mathrm{VR}}} \sum_{v=0}^{N_{\mathrm{VR}}-1} \left(X_{0,i}^{(v)}\right)^2 \tag{5.130}$$

with $i = 0, 1, ..., N_0 - 1$. The vector $\mathbf{P}_0$ (5.129) feeds the *cell-averaging smallest of - constant false alarm rate* (CFAR-CASO) algorithm developed in [88]. Based on this algorithm, a target is detected in the i-th frequency bin if

$$P_{0,i} > T_{\mathrm{CFAR}}, \tag{5.131}$$

where $i \in \{i_m, i_m + 1, ..., i_M\}$. Here,

$$T_{\mathrm{CFAR}} = K_0 \min\left(\bar{P}_l\,,\, \bar{P}_u\right) \tag{5.132}$$

represents a decision threshold, K_0 is a real parameter whose value is selected on the basis of the required *false alarm rate*, and

$$\bar{P}_l = \frac{1}{C_s} \sum_{k=i-(G_s+C_s)}^{i-(G_s+1)} P_{0,k} \tag{5.133}$$

and

$$\bar{P}_u = \frac{1}{C_s} \sum_{k=i+G_s+1}^{i+G_s+C_s} P_{0,k} \tag{5.134}$$

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 frequency 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 $\bar{P}_l$ (5.133) and $\bar{P}_u$ (5.134), 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

$$P_{0,i} > P_{l,u} \tag{5.135}$$

is also required to be satisfied together with the condition (5.131), 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

$$\mathcal{A}_{\mathrm{CF}} = [\hat{\alpha}_0, \hat{\alpha}_1, ..., \hat{\alpha}_{L_b-1}]^T \tag{5.136}$$

where $\hat{\alpha}_l$ represents the index of the frequency 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.

FFT-*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 frequency bin and estimating their azimuth. In this case, azimuth estimation consists of the following two steps:

1) The N_{VH}-dimensional column vector (see (5.83))

$$\mathbf{X}\left[l\right] \triangleq \left[X_{0,\hat{\alpha}_l}^{(0)}, X_{0,\hat{\alpha}_l}^{(1)}, ..., X_{0,\hat{\alpha}_l}^{(N_{\mathrm{VH}}-1)}\right]^T, \tag{5.137}$$

collecting the spectral information available on the whole array and referring to the $\hat{\alpha}_l$-th frequency bin (with $l = 0$, 1, ..., $L_b - 1$) is applied to an $\bar{N}_0$ order FFT algorithm; let $\mathbf{s}[l] = [s_0[l]$, $s_1[l]$, ..., $s_{\bar{N}_0-1}[l]]^T$ denote the $\bar{N}_0$-dimensional FFT output.

2) The dominant peaks[11] 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

$$\hat{\theta}_i\left[l\right] = \arcsin\left(h_{\bar{N}_0}\left[k_i\left[l\right]\right]\right) \tag{5.138}$$

where

$$h_{\bar{N}_0}\left[x\right] \triangleq 2\left(x - \bar{N}_0/2\right)/\bar{N}_0\ . \tag{5.139}$$

Let us now consider the case in which the URA represented in Fig. 5.4 is employed for resolving the targets associated with each frequency bin, and estimating their azimuth and elevation. The algorithm employed in this case involves the $N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ matrix $\mathbf{X}[l] \triangleq [X_{0,\hat{\alpha}_l}[p,q]]$ (5.55), collecting the spectral information available on the whole array for the $\hat{\alpha}_l$-th frequency 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 $\mathbf{s}_{\mathrm{VULA},0}[l] = \left[s_{0,0}\left[l\right], s_{0,1}\left[l\right], ..., s_{0,\bar{N}_0-1}\left[l\right]\right]^T$ on the basis of (5.63); here, p_R represents the column index of the reference antenna in the considered URA (see Fig. 5.4).

2) The dominant peaks of the sequence $\{[|s_{0,r}\left[l\right]|;\ r = 0, 1, ..., \bar{N}_0 - 1\}$ 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 frequency bin), the estimate of the elevation $\hat{\phi}_i[l]$ of the associated target is evaluated as

$$\hat{\phi}_i\left[l\right] = \arcsin\left(h_{\bar{N}_0}\left[r_i\left[l\right]\right]\right). \tag{5.140}$$

3) The 2D 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

$$\bar{S}_{k,r}\left[l\right] \triangleq \frac{1}{N_{\mathrm{VR}}} \sum_{q=0}^{N_{\mathrm{VV}}-1} \sum_{p=0}^{N_{\mathrm{VH}}-1} X_{0,\hat{\alpha}_l}\left[p,q\right] \cdot \exp\left(-j\frac{2\pi}{\lambda}\psi_{r,k}\right), \tag{5.141}$$

where

$$\psi_{r,k} \triangleq q\, h_{\bar{N}_0}[r]\, d_{\mathrm{VV}} + p\, h_{\bar{N}_0}[k]\, d_{\mathrm{VH}}. \tag{5.142}$$

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

[11] 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 [70].

$k = k_{m,i}[l]$ (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

$$\hat{\theta}_{i,r_i[l]}[l] = \arcsin\left(\frac{h_{\bar{N}_0}[k_{m,i}[l]]}{\cos(\hat{\phi}_i[l])}\right), \tag{5.143}$$

where $\hat{\phi}_i[l]$ is expressed by (5.140); consequently, the angular coordinates of the i-th target detected in the $\hat{\alpha}_l$-th frequency 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

$$\hat{L} = \sum_{l=0}^{L_b-1} \sum_{i=0}^{L_v[l]-1} L_h[i,l]. \tag{5.144}$$

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 frequency 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

$$\mathbf{R}_X[l] = \mathbf{X}[l]\ \mathbf{X}[l]^H \tag{5.145}$$

is computed; here, $\mathbf{X}[l]$ is defined by (5.137).

2) The $\bar{N}_0$-dimensional *pseudo-spectrum* $\mathbf{P}_{\mathrm{MU}}^{(l)}$ is evaluated; its k-th element is given by

$$\mathcal{P}_{\mathrm{MU}}^{(l)}[k] = \frac{1}{\mathbf{a}^H[k]\ \mathbf{Q}_{N_{\mathrm{VR}}}\mathbf{Q}_{N_{\mathrm{VR}}}^H\mathbf{a}[k]} \tag{5.146}$$

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 $\mathbf{R}_X[l]$ (5.145) and $\mathbf{a}[k]$ is a N_{VH}-dimensional steering vector, whose n-th element $a_n[k]$ is given by

$$a_n[k] = \exp\left(j\pi n\, h_{\bar{N}_0}[k]\right), \tag{5.147}$$

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 (5.138)–(5.139).

Let us consider now the case in which the uniform rectangular array shown in Fig. 5.4 is employed for resolving the targets associated with each frequency bin, and estimating their azimuth and elevation. In this case, the adopted procedure involves the $N_{\mathrm{VH}} \times N_{\mathrm{VV}}$ matrix $\mathbf{X}[l] \triangleq [X_{0,\hat{\alpha}_l}[p,q]]$ (5.55) 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 $\mathbf{R}_X[l]$ (5.145) and that the $\bar{N}_0$-dimensional vector $\mathbf{P}_{\mathrm{MU}}^{(\mathrm{VULA})}[l]$ is computed on the basis of (5.146)–(5.147) (note that N_{VR} 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 (5.140).

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

Table 5.2: Acronyms adopted for the FFT-based and MUSIC-based algorithms.

Alg. \ Radar	Complex FMCW	Real FMCW
FFT-BA (2D)	FFT-FC2	FFT-FR2
MUSIC-BA (2D)	MUSIC-FC2	MUSIC-FR2
FFT-BA (3D)	FFT-FC3	FFT-FR3
MUSIC-BA (3D)	MUSIC-FC3	MUSIC-FR3

computed according to (5.145) (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 (5.146). Note that, in this case, N_{VR} 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

$$a_n[k] = \exp\left(j\pi n\, h_{\bar{N}_0}[k] \cos\big(\hat{\phi}_i[l]\big)\right) \tag{5.148}$$

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

$$\hat{\theta}_{i,r_i[l]}[l] = \arcsin\left(h_{\bar{N}_0}[k_{m,i}[l]]\right). \tag{5.149}$$

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 (5.144).

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. The acronyms adopted in the following for these types of algorithms are summarized in Table 5.2.

5.8 Computational complexity

The computational cost of the algorithms described in Sections 5.4 and 5.7 has been carefully assessed in terms of *floating point operations* (flops) to be executed in the detection of L targets[12]. Various details about the method we adopted for the evaluation of the computational cost of each algorithm are provided in Appendix J for the RASCA-FC3 only. Our analysis leads to the conclusion that the overall cost of this algorithm and RASCA-FC2 is approximately of order $\mathcal{O}(M_{\mathrm{R-FC3}})$ and $\mathcal{O}(M_{\mathrm{R-FC2}})$, respectively, where (see (J.5))

$$\begin{aligned} M_{\mathrm{R-FC3}} = {} & 24 N_{\mathrm{VR}}\, N_0 \log_2(N_0) + 26 N_A\, K_{\mathrm{T}_2}\, N_0 \\ & + L_b\, K_{\mathrm{T}_3}\left(18 N_{\mathrm{VH}}\, N_{\mathrm{VV}}\, N_0 + 16 \bar{N}_0 \log_2\left(\bar{N}_0\right)\right) \end{aligned} \tag{5.150}$$

and

$$\begin{aligned} M_{\mathrm{R-FC2}} = {} & 24 N_{\mathrm{VH}}\, N_0 \log_2(N_0) + 26 N_A\, K_{\mathrm{T}_2} N_0 \\ & + L_b\, K_{\mathrm{T}_3}\left(18 N_{\mathrm{VH}}\, N_0 + 8 \bar{N}_0 \log_2(\bar{N}_0)\right); \end{aligned} \tag{5.151}$$

[12] In the remaining part of this section, the overall number of estimated targets ($\hat{L}$) is assumed to be equal to L, for simplicity.

here, K_{T_2} (K_{T_3}) represents the overall number of iterations carried out by the STDREC (STDAEC) algorithm.

In evaluating the cost of the FFT-based and MUSIC-based algorithms described in Section 5.7, we have assumed that: a) the cost due to the computation of the eigenvalue decomposition of a $d\times d$ matrix is $\mathcal{O}(d^3)$; b) the computational effort required to find the dominant peaks in a sequence is negligible. Based on these assumptions, it can be shown that the computational complexity of the FFT-FC3, FFT-FC2, MUSIC-FC3 and MUSIC-FC2 algorithms are approximately of order $\mathcal{O}(M_{\mathrm{F-FC3}})$, $\mathcal{O}(M_{\mathrm{F-FC2}})$, $\mathcal{O}(M_{\mathrm{M-FC3}})$ and $\mathcal{O}(M_{\mathrm{M-FC2}})$, respectively, where

$$\begin{aligned} M_{\mathrm{F-FC3}} =& 8N_{\mathrm{VR}}N_0\log_2(N_0) \\ &+ 8L_b\left(\bar{N}_0^2\log_2(\bar{N}_0^2)+\bar{N}_0\log_2(\bar{N}_0)\right), \end{aligned} \tag{5.152}$$

$$M_{\mathrm{F-FC2}} = 8\big(N_{\mathrm{VH}}N_0\log_2(N_0)+L_b\bar{N}_0\log_2(\bar{N}_0)\big), \tag{5.153}$$

$$\begin{aligned} M_{\mathrm{M-FC3}} =& 8N_{\mathrm{VR}}N_0\log_2(N_0)+L_b\bar{N}_0(N_{\mathrm{VV}}^3+N_{\mathrm{VH}}^3) \\ &+ 16L_b\bar{N}_0(N_{\mathrm{VV}}^2+N_{\mathrm{VH}}^2) \end{aligned} \tag{5.154}$$

and

$$M_{\mathrm{M-FC2}} = 8N_{\mathrm{VH}}N_0\log_2(N_0)+L_b\bar{N}_0(N_{\mathrm{VH}}^3+16N_{\mathrm{VH}}^2). \tag{5.155}$$

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 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 $\mathcal{A}_{\mathrm{CF}}$ (5.136), collecting the indices of the frequency bins in which *at least* one target has been detected, may include *ghost* targets; as evidenced by our computer simulations, the impact of this phenomenon on the overall computation time may not be negligible. Despite this, some interesting insights on how the complexity is influenced by the overall number of targets can be obtained by comparing the computational costs (5.150), (5.152) and (5.154) ((5.151), (5.153) and (5.155)) in two specific scenarios. The first scenario we take into consideration refers to the case in which the mutual distance between the targets is *above* the range resolution of the employed radar system, so that $K_{\mathrm{T}_2}=L$, $K_{\mathrm{T}_3}=1$ and $L_b=L$ can be assumed in the RHS of (5.150)–(5.155). In our second scenario, instead, the targets form clusters, each of which consists of four targets having the *same* range, but different angular coordinates; for this reason, $K_{\mathrm{T}_2}=L/4$, $K_{\mathrm{T}_3}=4$ and $L_b=L/4$ can be assumed in the RHS of (5.150)–(5.155). Moreover, the following parameters have been chosen for both scenarios: a) $N_{\mathrm{VR}}=256$; b) $N_0=1024$; c) $N_A=10$ d) $N_{\mathrm{VV}}=16$; e) $N_{\mathrm{VH}}=16$; f) $\bar{N}_0=32$. The dependence of the complexity M_{alg} on L is represented in Fig. 5.10a (Fig. 5.10b) for the first (second) scenario; here, alg denotes the algorithm which this complexity refers to. From these figures it is easily inferred that:

a) The RASCAs require the largest computational effort in both the considered scenarios for any value of L; for instance, $M_{\mathrm{RASCA-FC3}}$ is approximately 4.1 (4.6) times greater than $M_{\mathrm{MUSIC-FC3}}$ in the first (second) scenario for $L=8$.

b) The ratio between $M_{\mathrm{RASCA-FC2}}$ and $M_{\mathrm{MUSIC-FC2}}$ is approximately 2.4 for any value of L in the first scenario, but it increases with L in the second scenario; for instance, $M_{\mathrm{RASCA-FC2}}$ is 3.9 (4.6) times greater than $M_{\mathrm{MUSIC-FC2}}$ for $L=12$ ($L=36$).

c) The computational cost estimated for the MUSIC-BAs in the first scenario is larger than that referring to the second scenario for $L\geq 4$; for instance, if $L=20$, the value of $M_{\mathrm{MUSIC-FC3}}$ ($M_{\mathrm{MUSIC-FC2}}$) evaluated in first scenario is 1.33 (2.5) times larger than that found in the second scenario .

Finally, it is important to stress that:

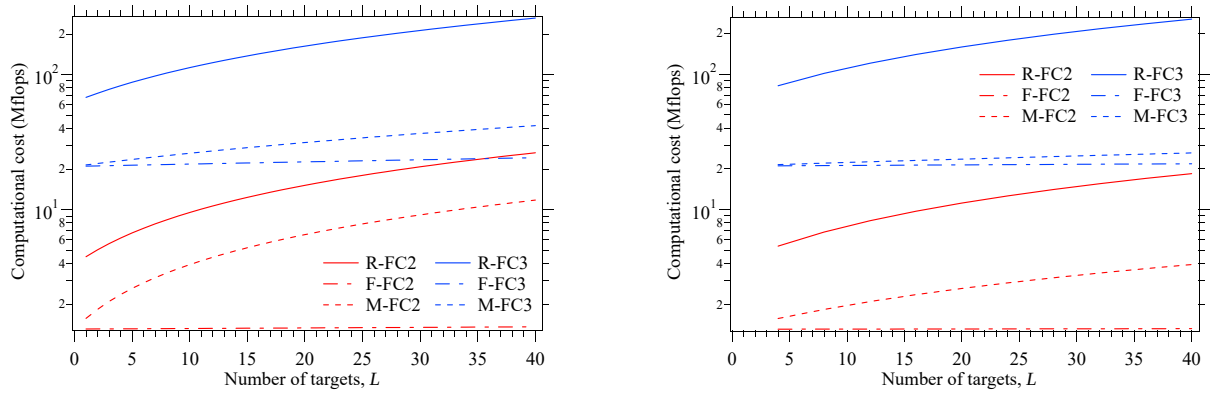


Figure 5.10: Computational cost versus overall number of targets for: a) the first scenario; b) the second scenario. The RASCA-FC3, RASCA-FC2, FFT-FC3, FFT-FC2, MUSIC-FC3 and MUSIC-FC2 are considered.

a) in real world scenarios, the overall number of targets detected by the FFT-BAs and the MUSIC-BAs may be greater than the true number of targets, since some targets are detected multiple times; this may have a significant impact on the overall computational effort required by these algorithms.

b) The computational complexity of the RASCA-FR3 (RASCA-FR2), the FFT-FR3 (FFT-FR2), and the MUSIC-FR3 (MUSIC-FR2) is of the same order as the RASCA-FC3 (RASCA-FC2), the FFT-FC3 (FFT-FC2) and the MUSIC-FC3 (MUSIC-FC2), respectively.

5.9 Numerical results

In this section, the accuracy of the RASCAs is assessed on the basis of both synthetically generated and experimental data, and is compared with that provided by various FFT-BAs and MUSIC-BAs.

5.9.1 Numerical results based on synthetically generated measurements

In this paragraph, the accuracy achieved by the RASCA-FC3, the FFT-FC3 and the MUSIC-FC3 in the generation of 3D radar images is assessed. The performance of these algorithms has been evaluated in a colocated MIMO FMCW radar system providing both the in-phase and quadrature components of all its received signals, and equipped with an URA consisting of $N_T = 16$ TX and $N_R = 16$ RX antennas; therefore, the available virtual array is made of $16 \cdot 16 = 256$ VAs with inter-antenna spacing $d_{\mathrm{VV}} = d_{\mathrm{VH}} = \lambda/4$. The other relevant parameters of the considered radar system are: a) chirp slope $\mu = 4 \cdot 10^{13}$ Hz/s; b) bandwidth $B = 2.5$ GHz; c) central frequency $f_0 = 77$ GHz; d) sampling frequency $f_s = 8$ MHz; e) number of samples per chirp $N = 512$. Note that, in principle, the available antenna array allows us to achieve the range resolution

$$\Delta R = \frac{c}{2B} \cong 6 \text{ cm}, \tag{5.156}$$

the azimuthal resolution

$$\Delta\theta = \frac{\lambda}{2d_{\mathrm{VH}}(N_{\mathrm{HULA}} - 1)} \cong 7.45^\circ \tag{5.157}$$

and the elevation resolution

$$\Delta\phi = \frac{\lambda}{2d_{\mathrm{VV}}(N_{\mathrm{VULA}} - 1)} \cong 7.45^\circ. \tag{5.158}$$

Table 5.3: *Root mean square error* $\bar{\varepsilon}_X$, *peak error* $\hat{\varepsilon}_X$ and *detection rate* R_D evaluated in the two simulation scenarios defined in Paragraph 5.9.1. Target range, azimuth and elevation are taken into consideration.

		$\bar{\varepsilon}_X$			$\hat{\varepsilon}_X$			
Scenario	Algorithm	R (m)	θ (°)	ϕ (°)	R (m)	θ (°)	ϕ (°)	R_{D} (%)
S1	RASCA-FC3	0.01	1.56	0.79	0.02	9.86	2.64	100
	FFT-FC3	0.02	2.07	1.05	0.04	14.34	3.06	71
	MUSIC-FC3	0.02	1.74	0.83	0.04	34.03	13.48	70
S2	RASCA-FC3	0.01	1.54	0.79	0.02	8.01	2.31	100
	FFT-FC3	0.02	2.05	1.05	0.04	14.78	2.95	70
	MUSIC-FC3	0.02	1.53	0.81	0.04	7.56	2.03	0.72

The considered radar system is assumed to operate in the presence of $L = 10$ targets, whose echoes have unit amplitude. The range, the azimuth and the elevation of each target are sequentially generated at the beginning of each run. Moreover, the range R_k, the azimuth θ_k and the elevation ϕ_k of the k-th target (with $k = 1, 2, ..., 10$) have been randomly evaluated in a way that: a) they belong to the intervals [1, 10] m, $[-\pi/3, \pi/3]$ rad and $[-\pi/3, \pi/3]$ rad, respectively; b) the minimum spacing between the k-th target and the previously generated $(k-1)$ targets is not smaller than ΔR (5.156), $\Delta\theta$ (5.157) and $\Delta\phi$ (5.158) in the range, azimuth and elevation dimensions, respectively (scenario **S1**) or is not smaller than ΔR (5.156) in the range domain, but can be arbitrarily small in the azimuth and elevation dimensions (this scenario is denoted **S2**). In our computer simulations, the following values have been selected for the parameters of the RASCA-FC3:

a) FFT Processing: $M = 2$;

b) RPE: $N_A = 10$, $N_{\mathrm{CSFE}} = 10$, $I = 7$ and $T_{\mathrm{STDREC}} = 0.001 \cdot E_0^{(v_k)}$ (see (5.49));

c) SPE: $\bar{M} = 2$, $N_{\mathrm{CSFE}} = 10$, $I = 7$ and $T_{\mathrm{STDAEC}} = 0.001 \cdot E^{(0)}[l]$ (see (5.58)), $T_{\mathrm{OF}} = 0$.

In addition, the following values have been selected for the parameters of the FFT-FC3 and the MUSIC-FC3: $C_s = 3$, $G_s = 2$ and $K_0 = 1.5$. The SNR $\triangleq 1/\sigma^2$ has been assumed to be equal to 10 dB and the following performance indices have been evaluated to assess estimation accuracy: a) The *detection rate* (R_{D}) defined as the percentage of simulation runs in which the considered algorithm detects all the targets; b) the *root mean square error* (RMSE)

$$\bar{\varepsilon}_X \triangleq \sqrt{N_m^{-1} \sum_{k=0}^{N_m-1} \left[X_k - \hat{X}_k\right]^2}; \tag{5.159}$$

c) the *peak error*

$$\hat{\varepsilon}_X \triangleq \max_k \left|X_k - \hat{X}_k\right|; \tag{5.160}$$

here, X_i and $\hat{X}_i$ represent the exact value of a parameter X and its corresponding estimate, whereas N_m represents the overall number of synthetically generated values of X; note that, if all the targets are detected by the considered algorithm in each run,

$$N_m \triangleq N_r\, L \tag{5.161}$$

where N_r is the overall number of simulation runs. In our work, the performance of the above mentioned algorithms has been assessed by: a) evaluating the detection rate for both the considered scenarios; b) ignoring the failure events (i.e., the events in which not all the targets have been detected) in the evaluation of all the RMSEs. The three performance indices defined above have been assessed on the basis of the estimates generated by executing $N_r = 500$ runs; the resulting values are summarised in Table 5.3. From these results it is easily inferred that:

a) The RASCA-FC3 achieves the lowest RMSEs in range, azimuth and elevation (range, elevation) estimation in the first (second) scenario; for instance, the RMSE $\bar{\varepsilon}_\theta$ characterising the RASCA-FC3 is about 1.3 (1.3) times smaller than the corresponding RMSE obtained for the FFT-FC3 in the first (second) scenario.

b) The RASCA-FC3 exhibits the lowest peak errors in range, azimuth and elevation (range) in the first (second) scenario; for instance, its peak error $\hat{\varepsilon}_R$ is 2 times smaller than the corresponding RMSE obtained for the FFT-FC3 and the MUSIC-FC3 in both scenarios.

c) All the considered algorithms achieve an excellent accuracy in both scenarios, since the RMSEs evaluated for range, azimuth and elevation are smaller than the corresponding resolutions given above.

d) The FFT-FC3 and the MUSIC-FC3 are outperformed by the RASCA-FC3 in terms of detection rate; in fact, the value of this parameter is about 70 % for the first two algorithms, but is equal to 100 % for the RASCA-FC3, since the last algorithm has been able to detect all the targets in every simulation run in both scenarios.

5.9.2 Numerical results based on experimental measurements

In this paragraph, we first describe the radar devices employed in our measurement campaigns and the adopted experimental setup. Then, we analyse: 1) the accuracy achieved by our RPE (and, in particular, by the STDREC algorithm) in range and phase estimation on multiple antennas of the same array in the presence of a single target and of multiple targets; 2) the accuracy of the 2D (3D) images generated by RASCA-FR2 (RASCA-FR3) and RASCA-FC2 (RASCA-FC3) in the presence of multiple targets.

Employed radar devices and adopted experimental setup

A measurement campaign has been accomplished to acquire a data set through two FMCW MIMO radars, all operating in the E-band. The first FMCW device, dubbed **TI FMCW** radar in the following, is the TIDEP-01012 *Cascade mmWave* radar (see Fig. 5.11-a)). It is manufactured by *Texas Instrument Inc.* [64], classified as a *long range radar* (LLR) and provides both the in-phase and quadrature components of received signals (i.e., *complex measurements*). Its main parameters are: a) chirp slope $\mu = 4 \cdot 10^{13}$ Hz/s; b) bandwidth $B_1 = 2.5$ GHz; c) central frequency $f_c = 77$ GHz; d) sampling frequency $f_s = 8$ MHz; e) number of samples per chirp $N = 512$. Moreover, it is endowed with a planar array made of $N_T = 12$ TX and $N_R = 16$ RX antennas (each consisting of an array of four patch elements), as shown in Fig. 5.11-a). The corresponding virtual array consists of $12 \cdot 16 = 192$ VAs; however, only 134 of them are available, since the remaining 58 VAs overlap with the other elements of the virtual array. As shown in Fig. 5.11-b) (where each VA is represented by a small blue circle), the virtual array has the following characteristics:

1. the non-overlapped VAs form an *horizontal* ULA (HULA_1), consisting of $N_{\text{HULA}_1} = 86$ VAs and three smaller HULAs, each made of 16 equally-spaced VAs;
2. the inter-antenna spacing of all the HULAs is $d_{\text{VH}} = \lambda/4$;
3. the vertical spacing of the three smaller HULAs is not uniform, since $d_{\text{VV}_1} = \lambda/4$, $d_{\text{VV}_2} = \lambda$ and $d_{\text{VV}_3} = 3\lambda/2$ (see Fig. 5.11-b)).

This virtual antenna array allows us to achieve a range, azimuth and elevation resolution equal to $\Delta R_1 = 5.8$ cm (see (5.156)), $\Delta\theta_1 = 1.35°$ (see (5.157)), and $\Delta\phi_1 = 16.4°$ (see (5.158)), respectively; note that the elevation resolution is coarser than the azimuth one since $N_{\text{VV}} = 7$ equally aligned antennas ($d_{\text{VV}} = d_{\text{VV}_1} = \lambda/4$) are assumed along the vertical direction (this is equivalent to considering an elevation aperture $D_y = 3\lambda$ along the vertical direction; see [82]). In our work, on the one hand, a central portion of the first HULA (contained inside the red rectangle

appearing in Fig. 5.11-b)), consisting of $N_{\mathrm{VV}} = 16$ antennas, has been exploited for 2D imaging, in order to guarantee a fair comparison with the other two radar devices. On the other hand, the whole array have been employed for 3D imaging.

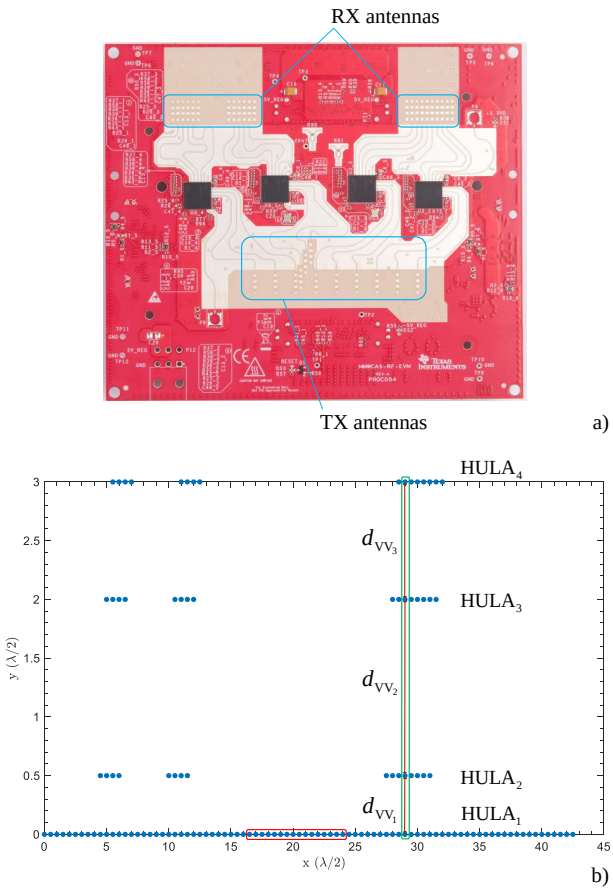


Figure 5.11: Representation of: a) the TI FMCW radar (the physical TX/RX antennas are contained inside the three rounded rectangles); b) the corresponding virtual array (the lower rounded rectangle contains the portion of HULA_1 employed for 2D imaging, whereas the green one the vertical array chosen as a reference for 3D imaging).

The second FMCW device, dubbed **Inras FMCW** radar in the following (see Fig. 5.12-a)), is a modular system manufactured by *Inras GmbH* [89] and consisting of: a) the so called *Radar Log* board; b) an RF front-end including multiple TX/RX antennas and *monolithic microwave integrated circuits* (MMIC) operating at 77 GHz. This system is classified as a LLR and its main parameters are: a) chirp slope $\mu = 9.7656 \cdot 10^{12}$ Hz/s; b) bandwidth $B_2 = 2.5$ GHz; c) central frequency $f_c = 77$ GHz; d) sampling frequency $f_s = 8$ MHz; e) number of samples per chirp $N = 2048$. Unlike the TI FMCW radar, this device provides only the in-phase component of the RF received signals and, consequently, *real measurements*. Moreover, 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, as shown in Fig. 5.12-a). The resulting virtual array, consisting of $N_{\mathrm{VR}} = 16 \cdot 16 = 256$ VAs is shown in Fig. 5.12-b). As it can be inferred from the last figure, the virtual array has the following characteristics:

1. It consists of 16 HULAs, each of which is made of 16 antennas with inter-antenna spacing $d_{\mathrm{VH}} = \lambda/4$.
2. The vertical distance between each couple of its adjacent HULAs is $d_{\mathrm{VV}} = \lambda/2$; this entailes the unambiguous elevation range $[-45°, 45°]$.
3. Its shape is not rectangular (the horizontal shift of adjacent HULAs is equal to $\lambda/4$).

This virtual array allows us to achieve the same range resolution as the TI FMCW radar, and azimuth and elevation resolutions equal to $\Delta\theta_2 = 7.6°$ and $\Delta\phi_2 = 3.8°$, respectively (see (5.157)–(5.158)). In our work, the HULA contained inside the red rectangle appearing in Fig. 5.12-b) (the whole array) has been exploited for 2D (3D) imaging.

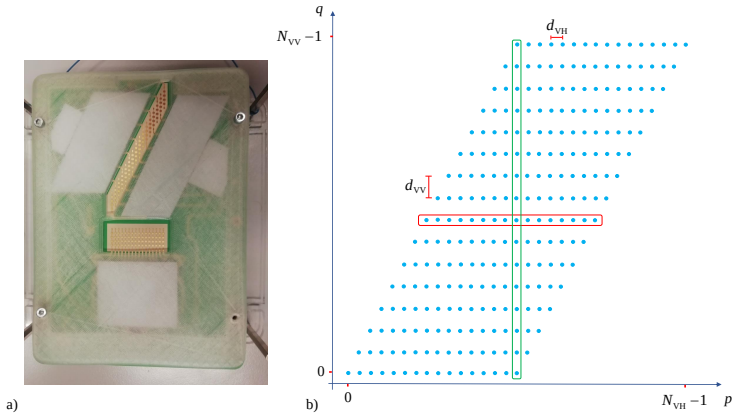


Figure 5.12: Representation of: a) the physical array of the Inras FMCW radar; b) the corresponding virtual array (the red rounded rectangle contains the HULA employed for 2D imaging, whereas the green one the vertical array chosen as a reference for 3D imaging).

Our 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). Each of the employed radar devices has been mounted on an horizontal wooden bar together with a pico-flexx camera manufactured by *PMD Technologies Inc.* [65] and has been lifted by a tripod at an height of roughly 1.60 m from ground, as shown in Fig. 5.13. The employed camera is based on a near-infrared *vertical cavity surface emitting laser*, 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 $62° \times 45°$).

In each measurement campaign, the experiments have been repeated for all the radar devices exactly in the same conditions.

It is important to point out that:

a) in all the radar systems, the 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) the data processing has been accomplished in the MATLAB environment;

d) all our detection and estimation algorithms have been run on a desktop computer equipped with a single i7 processor.

Range and amplitude estimation

In this paragraph, the accuracy of the STDREC algorithm employed by the RPE is analysed for two specific static scenarios. The first scenario is characterized by a single detectable target (a small metal disk[13] having a diameter equal to 5.5 cm) placed in ten different positions. The target range R and azimuth θ have been selected in the interval [1.0, 3.0] m, with a step of 0.5 m and $[-40°, 40°]$. The range and azimuth of the considered targets are listed in Table 5.4 for all the employed 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

[13]Each target is hung from the ceiling: a nylon thread has been used for suspending it.

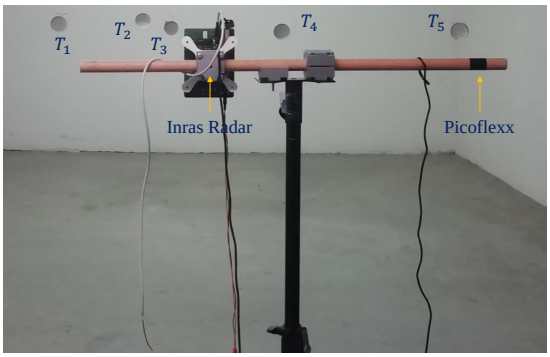


Figure 5.13: Experimental set-up developed for our measurement campaigns. The radar device (the Inras FMCW radar in this case) and a reference sensor (pico-flexx) are mounted on a wooden bar. A group of metal targets, placed at the same height as our sensors, is also visible.

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 5.5 (the data referring to the i-th target are collected in the column identified by T_i, with $i = 1, 2, ..., 9$). Each target has been sequentially added in our scenario; this has allowed us 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*.

Prior knowledge of L has been assumed during the processing; moreover, the following values have been selected for the parameters of the STDREC algorithm[14]:

TI FMCW radar - $N = 512$, $M = 4$, $N_0 = N \cdot M = 2048$, $N_{\text{CSFE}} = 5$ and $I = 7$.

Inras FMCW radar - $N = 2048$, $M = 1$, $N_0 = N \cdot M = 2048$, $N_{\text{SFE}} = 5$ and and $I = 7$.

Note that: a) the value of the oversampling factor (M) has been selected in way to guarantee approximately the same value of N_0 in all cases, i.e. roughly the same resolution in the spectral analysis of radar signals; b) the values of the parameters N_{SFE} and N_{CSFE} are all equal and large enough so that accurate range estimation is achieved by the STDREC algorithm.

The accuracy of range estimates has been assessed by evaluating the RMSE $\bar{\varepsilon}_R$ and the peak error $\hat{\varepsilon}_R$, expressed by (5.159)–(5.160) with $X = R$, $X_i = R_i$ and $\hat{X}_i$. Since the RCS of the considered targets was unknown, our 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. The phases $\{\psi^{(v)};\, v = 1, 2, ..., 16\}$ estimated by the STDREC algorithm over the considered reference HULA (consisting of 16 VAs; see the red rounded rectangles appearing in Figs. 5.11-b) and 5.12-b) and associated with a target placed at approximately[15] the same azimuth angle with respect to the centre of the radars is shown in Fig. 5.14. Since the distance d_{VH} between adjacent virtual channels is constant, the (unwrapped) estimated phases exhibit a linear dependence on the index of the virtual channel, as illustrated in Section 5.2 (see, in particular, (5.10) and (5.11)). 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 (5.18) with $\phi = 0$); this is confirmed by the results shown in Fig. 5.14. To assess the quality of the estimated phases, their RMSE $\bar{\varepsilon}_\psi$ has been evaluated in all the scenarios; in doing so, the linear fitting of the 16 phases $\{\psi^{(v)}\}$ has been taken as a reference with respect to which the error of each of them has been computed.

The estimate of the target range generated by the STDREC algorithm for each of the $N_m = 10$ distinct positions considered in the first scenario are listed in Table 5.4; 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 5.4 are also represented in Fig. 5.15. The errors $\bar{\varepsilon}_R$ and $\hat{\varepsilon}_R$, the mean of $\bar{\varepsilon}_\psi$ (denoted

[14]Note that in this case the stopping criterion based on eq. (5.49) has not be employed, since the overall number of targets is known.

[15]The exact range of this target can be found in the T_7 (T_3) column for the TI FMCW radar (Inras FMCW radar) in Table 5.4.

Table 5.4: Exact positions (range and azimuth) of the considered target, estimated ranges and RMSEs evaluated for the phase fitting over the considered sixteen virtual channels of each device (first experimental scenario). All our radar devices are taken into consideration.

Method		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Exact (TI FMCW)	R (m)	1.2	1.25	1.5	1.6	2.16	2.23	2.85	2.55	2.96	3.25
	θ (°)	-25	40	-10	30	-18	24	-14	21	-18	25
STDREC	$\hat{R}$ (m)	1.166	1.219	1.523	1.643	2.184	2.278	2.885	2.582	2.932	3.267
	$\bar{\varepsilon}_\psi$ (rad)	0.8	0.597	0.592	0.5	0.521	0.524	0.576	0.574	0.513	0.506
Exact (Inras FMCW)	R (m)	1.35	1.3	1.73	1.72	2.2	2.1	2.71	2.67	3.2	3.2
	θ (°)	-25	35	-13	30	-5	15	-10	26	-9	20
STDREC	$\hat{R}$ (m)	1.38	1.32	1.74	1.74	2.25	2.14	2.76	2.7	3.24	3.24
	$\bar{\varepsilon}_\psi$ (rad)	0.18	0.18	0.2	0.22	0.67	0.79	0.16	0.21	0.16	0.3

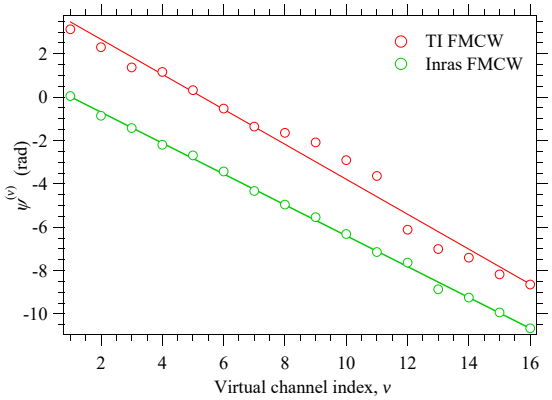


Figure 5.14: Unwrapped phase of the complex gain versus the index of the virtual channel of the reference HULA; a single target is assumed (first experimental scenario). The phase estimates generated by the STDREC algorithm in our three radar systems are identified by red, blue and green circles, whereas straight lines represent their linear fitting.

Table 5.5: Exact ranges of the nine coins characterizing our second experimental scenario. The range estimates evaluated by the Alg-YA, the Alg-S and the STDREC, the HSTDREC and the CFH algorithm are also provided.

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 (TI FMCW)	1.900	1.966	2.015	2.113	2.158	2.242	2.377	2.441	2.516
HSTDREC (TI FMCW)	1.906	1.971	2.016	2.118	2.158	2.238	2.378	2.444	2.514
STDREC (Inras FMCW)	1.920	1.980	2.040	2.100	2.220	2.280	2.460	2.520	2.580
CFH (TI FMCW)	1.947	2.017	2.077	2.161	2.241	2.338	2.374	2.435	2.518
Alg-YA (TI FMCW)	2.022	2.054	2.161	2.257	2.294	2.339	2.416	2.447	2.514
Alg-S (TI FMCW)	2.020	2.142	2.142	2.142	2.153	2.236	2.383	2.433	2.522

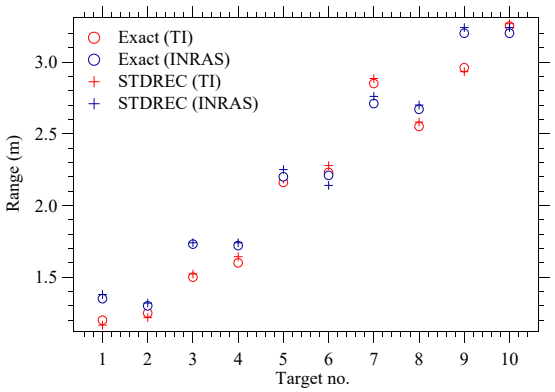


Figure 5.15: Representation of the ranges estimated by the STDREC algorithm (first experimental scenario). All our radar devices are considered.

Table 5.6: *Root mean square error* $\bar{\varepsilon}_R$, *peak error* $\hat{\varepsilon}_R$, *mean error* $\bar{\varepsilon}_{m,\psi}$ and CT evaluated for the STDREC algorithm in our first experimental scenario.

Method	$\bar{\varepsilon}_R$ (m)	$\hat{\varepsilon}_R$ (m)	$\bar{\varepsilon}_{m,\psi}$ (rad)	CT (msec)
TI FMCW	0.033	0.048	0.570	0.3
INRAS FMCW	0.035	0.050	0.30	0.4

$\bar{\varepsilon}_{m,\psi}$ 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 5.6.

The results referring to the first scenario lead us to the following conclusions:

1. In all the considered cases, the STDREC is able to accurately estimate the range and the phase characterizing each 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 our devices.
3. The Inras FMCW radar achieves the lowest $\bar{\varepsilon}_{m,\psi}$.
4. The CTs are always in the order of few milliseconds.

Let us focus on the second scenario. In this case, our range estimates have been generated by: a) the STDREC algorithm for all the radar devices; b) the HSTDREC algorithm for the TI FMCW; c) the Alg-YA, the Alg-S and the CFH algorithm for the TI FMCW. The obtained results are listed in Table 5.5. The errors $\bar{\varepsilon}_R$ and $\hat{\varepsilon}_R$, and the CT obtained in this case are listed in Table 5.7. From these results it can be inferred that:

1. In the case of the TI FMCW radar, all the considered algorithms achieve comparable accuracy. However, the STDREC and the HSTDREC algorithms, unlike all the other algorithms, achieve the lowest RMSE and peak error.
2. The HSTDREC algorithm is not more accurate than the STDREC algorithm; moreover, these algorithms are characterized by similar CTs.
3. The estimated RMSEs and peak errors are in the order of the resolution of our radar devices, but a little bit higher in the Inras FMCW radar systems. This is mainly due to the poorer estimates evaluated for the targets T_8 and T_9, since, in our specific experiment, the energy received from these targets has been found to be lower than that coming from the others. This problem is not so evident in the case of the TI FMCW radar, whose RMSE and peak errors are very low.

Table 5.7: *Root mean square error* $\bar{\varepsilon}_R$, *peak error* $\hat{\varepsilon}_R$ and *computation time* (CT) evaluated in our second experimental scenario.

Method	$\bar{\varepsilon}_R$ (m)	$\hat{\varepsilon}_R$ (m)	CT (msec)
STDREC (TI FMCW)	0.03	0.07	20
HSTDREC (TI FMCW)	0.03	0.07	20
STDREC (Inras FMCW)	0.07	0.10	40
CFH (TI FMCW)	0.07	0.12	15
Alg-YA (TI FMCW)	0.11	0.18	40
Alg-S (TI FMCW)	0.08	0.24	45

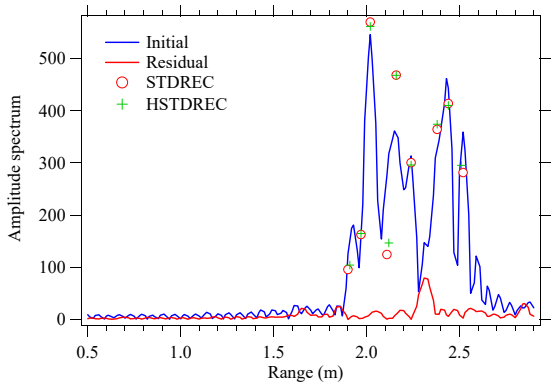


Figure 5.16: Representation of the initial amplitude spectrum of the signal observed on the central virtual channel (blue line) and of the final *residual* amplitude spectrum generated by the STDREC algorithm (red line). The TI FMCW radar operating in our second experimental scenario is considered; moreover, the target positions estimated by STDREC (HSTDREC) are represented by red circles (green crosses).

Finally, we would like to stress that the accuracy of STDREC and HSTDREC algorithms can be related to the accuracy of the estimation and cancellation procedure they accomplish. This is exemplified by Fig. 5.16, where the *initial amplitude spectrum* of the signal received on the central virtual channel of the TI FMCW radar in the second scenario and *its* (weak) *residual*, resulting from the cancellation of the spectral contributions due to the detected targets, are shown. Here, the range and amplitude of the targets estimated by the STDREC (HSTDREC) are also represented by red circles (green crosses).

Two-dimensional and three-dimensional imaging

In this paragraph, the accuracy of the 2D and 3D images generated by the RASCAs is assessed. Two different groups of experiments have been carried out. The first (second) group of experiments has allowed us 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 for all our radar devices. In the first group of experiments, the following choices have been made:

1. The targets have been placed at the same height. Their range and azimuth belong to the intervals [2.2, 2.7] m and [−15 °, 30°] , respectively (see Table 5.8).

2. The measurements have been acquired through a virtual ULA, consisting of 16 VAs, in all the considered radar systems.

As far as the second group of experiments is concerned, the following choices have been made:

1. The range, azimuth and elevation of the targets have been selected in the intervals [1.9, 2.8] m, $[-30°, 35°]$ and $[-10°, 10°]$, respectively (see Table 5.11).

2. The measurements have been acquired through the whole virtual array of each of our radar devices.

The following values have been selected for the parameters of the RASCAs: a) $N_A = 16$ ($N_A = 10$) in the RPE employed in 2D (3D) imaging; b) $N_{\mathrm{CSFE}} = N_{\mathrm{SFE}} = 5$ in both the STDREC and the STDAEC algorithms; c) the threshold $T_{\mathrm{OF}} = 0$ has been selected in the STDAE-S4 algorithm; d) the values of the parameters N_0 and M are equal to those employed for the STDREC in the previous paragraph; e) the oversampling factor is $\bar{M} = 16$ ($\bar{M} = 7$) for Inras FMCW radar (for the TI FMCW), so that the FFT order is $\bar{N}_0 = 256$ ($\bar{N}_0 = 602$). Moreover, the following values have been selected for the parameters[16] of the FFT-BAs and the MUSIC-BAs: $C_s = 6$, $G_s = 6$ and $K_0 = 2$. Prior knowledge of L has been assumed and the threshold T_{STDAEC} has been selected in the range $[0.01, 0.9] \cdot E^{(0)}[l]$ (5.58) (the value of this threshold has been adjusted on the basis of the SNR characterizing the received signal and the overall number of detectable targets).

The estimates of range and azimuth generated by the RASCAs on the basis of the measurements acquired in our first group of experiments are listed in Table 5.8, 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 5.9. In the last table, the values of RMSE, peak error and CT for the employed FFT-BAs and MUSIC-BAs are also provided. These results lead to the following conclusions:

1. All the range and azimuth errors are comparable with the resolution of our devices.

2. The RASCAs always outperform the other algorithms and require a lower CT.

3. The highest range (azimuth) peak errors and RMSEs are found in the case of the Inras FMCW radar; the TI FMCW radar, instead, achieves the lowest range and azimuth errors. This is mainly due to the differences in the SNR available at the receive side of distinct radar devices is different.

The good accuracy achieved by the RASCAs is also evidenced by Fig. 5.17, where a *range-azimuth map* [81], generated though standard 2D FFT processing of the measurements acquired through the Inras FMWC radar, is represented as a contour plot[17]; in the same figure, the exact position of the five targets employed in our first group of experiments and their estimates evaluated by all the considered algorithms are shown.

Let us consider now on the results obtained for our second group of experiments. The estimates of range, azimuth and elevation generated by the RASCAs are listed in Table 5.11, whereas the values of RMSE, peak error and CT evaluated on the basis of this table are listed in Table 5.10. In the last table, the errors characterizing the FFT-based and MUSIC-based algorithm for 3D imaging are also provided. From these results it can be inferred that:

1. The RMSEs 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. Moreover, these errors are smaller than the ones characterizing the FFT-based and MUSIC-based algorithms.

2. The azimuth and elevation estimates computed on the basis of the measurements acquired through the TI and Inras FMCW radars are reasonably good.

[16] Our simulations have evidenced that small changes in the value of these parameters do not significantly influence the detection probability and the estimation accuracy of the considered algorithms.

[17] 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 our reference system.

Table 5.8: Exact range and azimuth of the five targets considered in our first group of experiments and corresponding estimates generated by the RASCAs.

Exp.	Method	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-FC2	R (m)		2.52			
		θ (°)		-5.1			
	RASCA-FR2	R (m)		2.6			
		θ (°)		-3.9			
2)	RASCA-FC2	R (m)		2.53	2.40		
		θ (°)		-5.2	8.1		
	RASCA-FR2	R (m)		2.6	2.55		
		θ (°)		-4.3	11.3		
3)	RASCA-FC2	R (m)	2.27	2.53	2.41		
		θ (°)	-14.0	-5.2	9.8		
	RASCA-FR2	R (m)		2.63	2.55		2.33
		θ (°)		-4.3	11.3		28.2
4)	RASCA-FC2	R (m)	2.32	2.53	2.42		2.26
		θ (°)	-13.7	-5.2	11.2		29.4
	RASCA-FR2	R (m)	2.30	2.64	2.42		2.22
		θ (°)	-12.5	-4.1	11.9		31.2
5)	RASCA-FC2	R (m)	2.31	2.53	2.42	2.56	2.27
		θ (°)	-12.4	-5.2	11.2	17.0	26.3
	RASCA-FR2	R (m)	2.34	2.47	2.39	2.66	2.36
		θ (°)	-17.3	-6.7	8.1	16.4	29.5

Table 5.9: *Root mean square error* $\bar{\varepsilon}_X$, *peak error* $\hat{\varepsilon}_X$, and *computation time* (CT) evaluated on the basis of our first group of measurements. Target range and azimuth are taken into consideration.

Method	$\bar{\varepsilon}$		$\hat{\varepsilon}$		CT
	R (m)	θ (°)	R (m)	θ (°)	(sec)
RASCA-FC2	0.04	1.2	0.05	1.5	0.3
FFT-FC2	0.06	1.6	0.11	2.8	0.4
MUSIC-FC2	0.05	1.4	0.09	2.4	0.5
RASCA-FR2	0.09	1.1	0.12	1.8	0.4
FFT-FR2	0.13	1.8	0.16	2.65	0.45
MUSIC-FR2	0.13	1.6	0.16	2.49	0.45

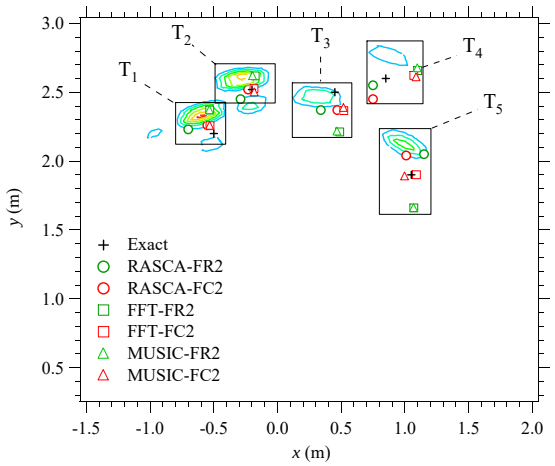


Figure 5.17: Representation of the range-azimuth map (in $x-y$ coordinates) computed on the basis of the measurements acquired through the Inras FMCW radar in the presence of five targets. The exact position of each target and its estimate obtained through the RASCAs (shown in Table 5.8) and the other algorithms are also shown. The rectangles allow to delimit the region in which the position of each target and its estimates are located.

Table 5.10: *Root mean square error* $\bar{\varepsilon}_X$, *peak error* $\hat{\varepsilon}_X$ and *computation time* (CT) evaluated on the basis of our second group of measurements. Target range, azimuth and elevation are taken into consideration.

Method	$\bar{\varepsilon}$			$\hat{\varepsilon}$			CT
	R (m)	θ (°)	ϕ (°)	R (m)	θ (°)	ϕ (°)	(sec)
RASCA-FC3	0.05	2.8	1.9	0.07	3.9	2.5	2.0
FFT-FC3	0.07	3.0	1.5	0.08	4.2	2.1	1.5
MUSIC-FC3	0.07	3.9	3.2	0.08	6.4	3.8	1.6
RASCA-FR3	0.06	2.0	2.3	0.08	3.0	3.0	2.6
FFT-FR3	0.1	1.0	2.1	0.17	1.5	3.2	1.1
MUSIC-FR3	0.1	1.0	2.2	0.15	1.5	3.3	1.3

3. The average CT is in the order of few seconds for all the proposed algorithms; the lowest (highest) average CT is found in the case of the TI FMCW radars (Inras FMCW radar). This is mainly due to the fact that the STDREC algorithm employed in the case of the Inras FMCW requires an higher computational effort with respect to its counterpart employed with the TI FMCW radar.

4. In general, the CT of RASCAs is higher than that required by the proposed FFT-based or MUSIC-based methods. This is mainly due to the recursive cancellation procedure that is not employed by the other two methods. However, we believe that this cancellation procedure plays a fundamental role in the detection of weak targets and allows to achieve a good estimation accuracy.

The good accuracy and resolution provided by the RASCAs are highlighted by Fig. 5.18, where the exact positions of the five targets employed in our second group of experiments and their estimates produced by all the considered algorithms are shown; note that, unlike FFT-based and MUSIC-based algorithms, the RASCAs achieve good accuracy even in the presence of closely spaced targets, like T_4 and T_5.

Table 5.11: Exact range, azimuth and elevation of the five targets considered in our second group of experiments and corresponding estimates generated by the RASCAs.

Exp.	Method	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-FC3	R (m)	1.89				
		θ (°)	-25.8				
		ϕ (°)	-4.0				
	RASCA-FR3	R (m)	2.04				
		θ (°)	-29.3				
		ϕ (°)	-7.5				
2)	RASCA-FC3	R (m)	1.89				2.12
		θ (°)	-25.2				37.8
		ϕ (°)	-3.4				-2.3
	RASCA-FR3	R (m)	2.0				2.1
		θ (°)	-27.0				34.2
		ϕ (°)	-8.0				-2.5
3)	RASCA-FC3	R (m)	1.89	2.43			2.12
		θ (°)	-25.2	-8.6			37.8
		ϕ (°)	-3.4	1.1			-3.4
	RASCA-FR3	R (m)	2.0	2.45			2.1
		θ (°)	-29	-10			31.5
		ϕ (°)	-4.0	5.0			-3.0
4)	RASCA-FC3	R (m)	1.89	2.44		2.48	2.12
		θ (°)	-25.2	-8.6		20.1	37.3
		ϕ (°)	-3.2	-0.6		-6.9	-2.9
	RASCA-FR3	R (m)	2.0	2.4		2.45	2.1
		θ (°)	-29.5	-12.0		18.3	33.5
		ϕ (°)	-4.0	5.0		5.0	-3.0
5)	RASCA-FC3	R (m)	1.89	2.44	2.83	2.48	2.12
		θ (°)	-25.2	-8.6	3.2	20.1	37.3
		ϕ (°)	-3.2	-0.6	-2.87	-6.9	-2.9
	RASCA-FR3	R (m)	1.98	2.4	2.65	2.4	2.13
		θ (°)	-27.6	-15.0	0.4	18.3	37.3
		ϕ (°)	-5.8	5	-1.1	-3	-6.2

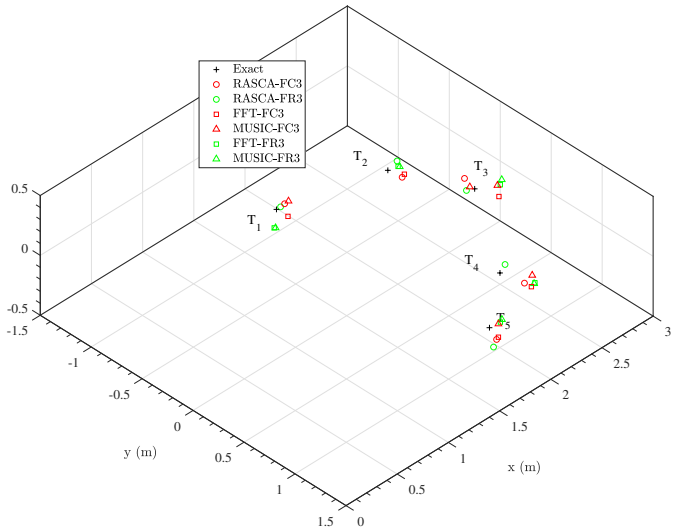


Figure 5.18: Representation of a 3D scenario characterized by five targets. The exact position of each target and the corresponding estimates evaluated by all the considered algorithms are shown (all our radar systems are considered).

5.10 Conclusions

In this chapter, four novel algorithms, dubbed *range & angle serial cancellation algorithms* (RASCAs), have been developed for the detection and the estimation of multiple targets in colocated MIMO radar systems. All these algorithms can be seen as instances of a general approach to target detection and estimation, and exploit new methods for the estimation of multiple overlapped real and complex tones. As evidenced by our computer simulations run on both synthetically generated data and measurements acquired through commercial devices, the devised algorithms are able to generate accurate 2D and 3D radar images in the presence of multiple closely spaced targets and outperform other algorithms based on the computation of multiple FFTs or on the MUSIC for DOA estimation.

F

Approximate expressions for the cost functions of the CSFE and the SFE

In this appendix, the derivation of the approximate MSEs $\varepsilon_{\text{CSFE}}(\tilde{\Delta}, \hat{A})$ (4.40) and $\varepsilon_{\text{SFE}}(\tilde{\Delta}, \hat{C})$ (4.75) is sketched.

F.1 Complex Single Frequency Estimator

To begin, we rewrite eq. (4.23) as

$$\begin{aligned}\varepsilon_n(\tilde{F}, \tilde{A}) = \tilde{A}_R^2 + \tilde{A}_I^2 - 2(x_{c,n}^{(R)}\tilde{A}_R + x_{c,n}^{(I)}\tilde{A}_I)\cos(\tilde{\phi}_n) \\ - 2(x_{c,n}^{(I)}\tilde{A}_R - x_{c,n}^{(R)}\tilde{A}_I)\sin(\tilde{\phi}_n) + |x_{c,n}|^2,\end{aligned} \tag{F.1}$$

where $\tilde{A}_R \triangleq \Re\{\tilde{A}\}$, $\tilde{A}_I \triangleq \Im\{\tilde{A}\}$, $x_{c,n}^{(R)} \triangleq \Re\{x_{c,n}\}$, $x_{c,n}^{(I)} \triangleq \Im\{x_{c,n}\}$ and $\tilde{\phi}_n$ is defined by eq. (4.37) (see also eq. (4.38)). Substituting the RHS of eq. (4.37) in that of eq. (F.1) produces, after some manipulation,

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

Finally, substituting the approximations

$$\cos\left(n\tilde{\Delta}\right) \simeq 1 - \frac{1}{2}n^2\tilde{\Delta}^2, \tag{F.3}$$

and

$$\sin\left(n\tilde{\Delta}\right) \simeq n\tilde{\Delta} - n^3\frac{\tilde{\Delta}^3}{6} \tag{F.4}$$

in the RHS of eq. (F.2) and, then, the resulting expression in the RHS of eq. (4.20) produces eq. (4.40).

F.2 Single Frequency Estimator

A similar procedure is followed in the derivation of eq. (4.75). First of all, we rewrite eq. (4.75) as

$$\begin{aligned}\varepsilon_n(\tilde{F}, \hat{C}) = x_{r,n}^2 + 2\left[\hat{C}_R^2 + \hat{C}_I^2\right] \\ -4\hat{C}_R\left[x_{r,n}\cos(\hat{\theta}_n)\cos(n\tilde{\Delta}) - x_{r,n}\sin(\hat{\theta}_n)\sin(n\tilde{\Delta})\right] \\ +4\hat{C}_I\left[x_{r,n}\sin(\hat{\theta}_n)\cos(n\tilde{\Delta}) + x_{r,n}\cos(\hat{\theta}_n)\sin(n\tilde{\Delta})\right] \\ +2\left(\hat{C}_R^2 - \hat{C}_I^2\right)\left[\cos(2\hat{\theta}_n)\cos(2n\tilde{\Delta}) - \sin(2\hat{\theta}_n)\sin(2n\tilde{\Delta})\right] \\ -4\hat{C}_R\hat{C}_I\left[\sin(2\hat{\theta}_n)\cos(2n\tilde{\Delta}) + \cos(2\hat{\theta}_n)\sin(2n\tilde{\Delta})\right]\end{aligned} \tag{F.5}$$

using standard trigonometric formulas. If the normalized frequency F_{DFT} (4.29) is small enough (i.e., if the FFT order N_0 is large enough), the trigonometric functions $\cos(kn\tilde{\Delta})$ and $\sin(kn\tilde{\Delta})$ appearing in the RHS of eq. (F.5) (with $k = 1$ and 2) can be approximated as[1]

$$\sin\left(n\tilde{\Delta}\right) \simeq n\tilde{\Delta}, \tag{F.6}$$

$$\cos\left(n\tilde{\Delta}\right) \simeq 1 - \frac{1}{2}n^2\tilde{\Delta}^2, \tag{F.7}$$

$$\sin\left(2n\tilde{\Delta}\right) \simeq 2n\tilde{\Delta} - \frac{4}{3}n^3\tilde{\Delta}^3 \tag{F.8}$$

and

$$\cos\left(2n\tilde{\Delta}\right) \simeq 1 - 2n^2\tilde{\Delta}^2. \tag{F.9}$$

Substituting the RHSs of eqs. (F.6)–(F.9) in that of eq. (F.5) and, then, substituting the resulting approximate expression in the RHS of eq. (4.20) yields eq. (4.75).

[1] Note that the approximation adopted for $\sin(2n\tilde{\Delta})$ ($\cos(2n\tilde{\Delta})$) is more accurate than the one used for $\sin(n\tilde{\Delta})$ ($\cos(n\tilde{\Delta})$), since the width of the interval of its argument is twice that of the last function.

G

Cancellation Procedures employed in the CSFE/CSFEC and the SFE/SFEC

G.1 Spectrum cancellation employed in CSFE and SFE

In this paragraph, the expression of the vector $\bar{\mathbf{C}}_0(\hat{A}_l[r-1], \hat{F}_l[r-1])$ appearing in the RHS of eq. (4.92) is derived. First of all, let us take into consideration the CSFE; in this case, $\bar{\mathbf{C}}_0(\cdot,\cdot)$ is computed to cancel the contribution of the sequence (see eq. (4.22))

$$s_n\left(\bar{F}_l, \bar{A}_l\right) \triangleq \bar{A}_l \exp\left(j2\pi n \bar{F}_l\right) = \bar{A}_l\, \bar{w}_l^n \tag{G.1}$$

to the vector $\mathbf{X}_0$ (4.30); here,

$$\bar{w}_l \triangleq \exp(j2\pi \bar{F}_l). \tag{G.2}$$

Since $\mathbf{X}_0$ is the N_0-th order DFT of the zero-padded vector $\mathbf{x}_{0,\mathrm{ZP}}$ (4.31) (where the vector $\mathbf{x}_0$ collects the elements of the complex sequence $\{x_{c,n};\, n = 0,\, 1,\, ...,\, N-1\}$), it is easy to prove that

$$\bar{\mathbf{C}}_0(\hat{A}_l[r-1], \hat{F}_l[r-1]) = \bar{A}_l\ \ \bar{\mathbf{W}}_0^{(l)}, \tag{G.3}$$

where $\bar{\mathbf{W}}_0^{(l)}$ denote the N_0-th order DFT of the vector

$$\bar{\mathbf{w}}_0^{(l)} \triangleq \left[1, \bar{w}_l, \bar{w}_l^2, ..., \bar{w}_l^{N-1}, 0, ..., 0\right]^T. \tag{G.4}$$

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

$$\begin{aligned} \bar{W}_0^{(l)}[m] &= \frac{1}{N} \sum_{n=0}^{N-1} \bar{w}_l^n \exp\left(-j\frac{2\pi m}{N_0} n\right) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \left(q\,[m]\right)^n, \end{aligned} \tag{G.5}$$

where

$$q\,[m] \triangleq \exp\left(j2\pi\left(\hat{F} - \frac{m}{N_0}\right)\right). \tag{G.6}$$

Therefore, the identity

$$\sum_{n=0}^{N-1} q^n = \frac{q^N - 1}{q-1}, \tag{G.7}$$

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

Similar considerations can be formulated for the SFE. However, in this case, eq. (G.1) is replaced by (see eq. (4.64))

$$\begin{aligned} s_n\left(\bar{F}_l, \bar{C}_l\right) &\triangleq \bar{C}_l \exp\left(j2\pi n \bar{F}_l\right) + \bar{C}_l^* \exp\left(-j2\pi n \bar{F}_l\right) \\ &= \bar{C}_l\ \left(\bar{w}_l\right)^n + \bar{C}_l^*\, \left(\bar{w}_l^*\right)^n, \end{aligned} \tag{G.8}$$

where $\bar{w}_l$ is still defined by eq. (G.2). Consequently, eq. (G.3) is replaced by

$$\bar{\mathbf{C}}_0(\hat{A}_l[r-1], \hat{F}_l[r-1]) = \bar{C}_l\, \bar{\mathbf{W}}_0^{(l)} + \bar{C}_l^* \bar{\mathbf{W}}_{0,c}^{(l)}, \tag{G.9}$$

where $\bar{\mathbf{W}}_0^{(l)}$ is the N_0-dimensional vector defined above and $\bar{\mathbf{W}}_{0,c}^{(l)}$ denote the N_0-th order DFT of the vector $(\bar{\mathbf{w}}_0^{(l)})^*$. Therefore, the m-th element of the vector $\bar{\mathbf{W}}_{0,c}^{(l)}$ is given by

$$\begin{aligned} \bar{W}_{0,c}^{(l)}[m] &= \frac{1}{N} \sum_{n=0}^{N-1} (\bar{w}_l^*)^n \exp\left(-j\frac{2\pi m}{N_0} n\right) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} (q_c\,[m])^n\,, \end{aligned} \tag{G.10}$$

where

$$q_c\,[m] \triangleq \exp\left(j2\pi\left(-\hat{F} - \frac{m}{N_0}\right)\right). \tag{G.11}$$

Again, the identity (G.7) can be exploited for an efficient computation of all the elements of both the vectors $\bar{\mathbf{W}}_0^{(l)}$ and $\bar{\mathbf{W}}_{0,c}^{(l)}$.

G.2 Leakage cancellation employed in CSFEC and SFEC

In this paragraph, the expression of the quantity $\bar{X}_{\mathrm{lk},k}(\hat{A}_l\,[r-1]\,, \hat{F}_l\,[r-1]\,, \hat{F}_{c,r}\,[r])$ appearing in the RHS of eqs. (4.96) and (4.99) is derived for the CSFEC and the SFEC algorithms. In the first case, this quantity is computed to cancel the contribution of the sequence $\{s_n(\bar{F}_l, \bar{A}_l)\}$ (G.1) to $\bar{X}_{k,\rho[r]}$ for $k = 0$, 1, 2 and 3. Since $\bar{X}_{k,\rho[r]}$ is defined by eq. (4.42), it is not difficult to show that

$$\bar{X}_{\mathrm{lk},k}\left(\hat{A}_l\,[r-1]\,, \hat{F}_l\,[r-1]\,, \hat{F}_{c,r}\,[r]\right) = \bar{A}_l\, \bar{W}_k^{(l)}(\hat{F}_{c,r}\,[r]), \tag{G.12}$$

where

$$\begin{aligned} \bar{W}_k^{(l)}(\hat{F}_{c,r}\,[r]) &= \frac{1}{N} \sum_{n=0}^{N-1} n^k\, \bar{w}_l^n \exp\left(-j2\pi n \hat{F}_{c,r}\,[r]\right) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} n^k\, (\bar{q}[r])^n \end{aligned} \tag{G.13}$$

and

$$\bar{q}[r] \triangleq \exp\left(j2\pi\left(\bar{F}_l - \hat{F}_{c,r}\,[r]\right)\right). \tag{G.14}$$

Therefore, the identities (G.7) and (4.84)–(4.85) can be exploited for an efficient computation of all the terms appearing in the RHS of eqs. (4.96) and (4.99) for $k = 0$, 1, 2 and 3.

In the second case, the only modification with respect to the first one consists in adding the term $\bar{C}_l^* \bar{W}_{k,c}^{(l)}(\hat{F}_{c,r}[r])$ to the RHS of eq. (G.12). Note that

$$\begin{aligned} \bar{W}_{k,c}^{(l)}(\hat{F}_{c,r}\,[r]) &= \frac{1}{N} \sum_{n=0}^{N-1} n^k\, (\bar{w}_l^*)^n \exp\left(-j2\pi n \hat{F}_{c,r}\,[r]\right) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} n^k\, (\bar{q}_c[r])^n \end{aligned} \tag{G.15}$$

where

$$\bar{q}_c[r] \triangleq \exp\left(-j2\pi\left(\bar{F}_l + \hat{F}_{c,r}\,[r]\right).\right) \tag{G.16}$$

Therefore, the identities (G.7) and (4.84)–(4.85) can be exploited for an efficient computation of all the terms appearing in the RHS of eqs. (4.96) and (4.99) even if the SFEC algorithm is used.

H

Derivation of the bias of the CSFE

In this Appendix, the bias of the CSFE algorithm is analysed in the case in which the parameter Δ is computed on the basis of eq. (4.52). To begin, let us assume that: a) the complex sequence $\{x_{c,n}\}$ is not affected by noise, so that (see eqs. (4.18) and (4.28))

$$x_{c,n} = A \exp\left(j2\pi n F\right) = A \exp\left(j2\pi n \left(F_c + \delta\, F_{\mathrm{DFT}}\right)\right); \tag{H.1}$$

b) $\hat{\alpha} = \alpha$ (see eqs. (4.54) and (4.55)). Then, it is easy to show that (see eq. (4.42))

$$\bar{X}_{1,\rho^{(i-1)}} = \frac{A}{N_0} \sum_{n=0}^{N-1} n \, \exp\left(j2\pi n \vartheta\right) \tag{H.2}$$

and

$$\bar{X}_{2,\rho^{(i-1)}} = \frac{A}{N_0} \sum_{n=0}^{N-1} n^2 \, \exp\left(j2\pi n \vartheta\right) \tag{H.3}$$

where $\vartheta_i \triangleq (\delta - \hat{\delta}_i) F_{\mathrm{DFT}}$ and $\hat{\delta}_i \triangleq (\hat{F}^{(i)} - \hat{F}_c^{(0)})/F_{\mathrm{DFT}}$. The identities (4.84) and (4.85) allow us to rewrite eqs. (H.2) and (H.3) as

$$\bar{X}_{1,\rho^{(i-1)}} = \frac{1}{N_0} A \frac{(N-1) q_i^{N+1} - N q_i^N + q_i}{(q_i - 1)^2} \tag{H.4}$$

and

$$\bar{X}_{2,\rho^{(i-1)}} = \frac{1}{N_0} A \frac{(N-1)^2 q_i^{N+2} + \left(-2N^2 + 2N + 1\right) q_i^{N+1} + N^2 q_i^N - q_i^2 - q_i}{(q_i - 1)^3} \tag{H.5}$$

respectively, where $q_i \triangleq \exp\left(j2\pi\vartheta_i\right)$. If the normalised frequency F_{DFT} (4.29) is small enough (i.e., if the FFT order N_0 is large enough), the quantity q_i can be approximated through its truncated Taylor series $1 + j2\pi\vartheta_i - 2\pi^2\vartheta_i^2 - j4\pi^3\vartheta_i^3/3$. Then, substituting this approximation in the RHS of eqs. (H.4) and (H.5) and, then, the resulting expressions in the RHSs of eqs. (4.49) and (4.50) yields, after some manipulation,

$$b\left(\rho^{(i-1)}\right) \triangleq \Re\{\hat{A}^* \bar{X}_{2,\rho^{(i-1)}}\} = F_{\mathrm{DFT}} \frac{|A|^2}{6} \left(2N - 1\right) N(N-1). \tag{H.6}$$

and

$$c\left(\rho^{(i-1)}\right) \triangleq -\Im\{\hat{A}^* \bar{X}_{1,\rho^{(i-1)}}\} = -F_{\mathrm{DFT}} |A|^2 \frac{\pi}{3} \frac{\left(2N - 1\right) N(N-1)\vartheta_i}{1 + 4\pi^2\vartheta_i^2}, \tag{H.7}$$

respectively. Then, substituting the RHSs of the last two equations in that of eq. (4.52) gives

$$\hat{\Delta}^{(i)} = \frac{2\pi\vartheta_i}{1 + 4\pi^2\vartheta_i^2}. \tag{H.8}$$

Note that ϑ_i gets smaller as i increases if the CSFE converges; therefore, the last expression can be approximated as

$$\hat{\Delta}^{(i)} \simeq 2\pi\vartheta_i = 2\pi \left(\delta - \hat{\delta}_i\right) F_{\mathrm{DFT}} = \Delta - \hat{\Delta}_i \tag{H.9}$$

where $\hat{\Delta}_i \triangleq 2\pi\hat{\delta}_i F_{\mathrm{DFT}}$. This proves that the CSFE is unbiased.

I

Computational complexity of single tone estimators

In this Appendix, the computational complexity, in terms of flops, is assessed for the CSFE and the SFE developed in Paragraphs 4.3.1–4.3.2.

I.1 Complex Single Frequency Estimator

The overall computational cost of each this algorithm can be expressed as

$$\mathcal{C}_{\text{CSFE}} = \mathcal{C}_{i,\text{CSFE}} + N_{\text{CSFE}}\,\mathcal{C}_{r,\text{CSFE}}, \tag{I.1}$$

where $\mathcal{C}_{i,\text{CSFE}}$, $\mathcal{C}_{r,\text{CSFE}}$ and N_{CSFE} represent the cost of its initialization, that of each of its iterations and the overall number of iterations, respectively. The general criteria adopted in estimating the computational cost of both $\mathcal{C}_{i,\text{CSFE}}$ and $\mathcal{C}_{r,\text{CSFE}}$ are the same as those illustrated in [90] and can be summarised as follows:

- $4d-2$ flops are required to compute the inner product $\mathbf{u}_c^T\,\mathbf{v}$ of a $d\times 1$ complex column vector and a $d\times 1$ real column vector;
- $6d+2(d-1)$ flops are required to compute the inner product $\mathbf{u}_c^T\,\mathbf{v}_c$ of two $d\times 1$ complex vectors;
- d flops are required to find the maximum element of a vector $\mathbf{v}\in\mathbb{R}^{1\times d}$;
- $4d^2+14d-8$ flops are required to compute an interpolation based on the elements of a complex vector $\mathbf{v}\in\mathbb{C}^{1\times d}$;

The cost $\mathcal{C}_{i,\text{CSFE}}$ is evaluated as

$$\mathcal{C}_{i,\text{CSFE}} \triangleq \mathcal{C}_{\mathbf{X}_0} + \mathcal{C}_{\hat{\alpha}} + \mathcal{C}_{X_{k,\hat{\alpha}}} + \mathcal{C}_{\hat{\Delta}}, \tag{I.2}$$

where: a) $\mathcal{C}_{\mathbf{X}_0} = 8N_0\log_2 N_0$ is the contribution due to the computation of the vector $\mathbf{X}_0$ (4.30); b) $\mathcal{C}_{\hat{\alpha}} = 4N_0$ is the contribution due to the computation of $\hat{\alpha}$ on the basis of eq. (4.55); c) $\mathcal{C}_{X_{k,\hat{\alpha}}} = 33N+15$ is the contribution due to computation of $\bar{X}_{k,\hat{\alpha}}$ on the basis of eq. (4.42); d) $\mathcal{C}_{\hat{\Delta}} = 29$ is the contribution due to the computation of the coefficients of the quadratic equation (4.47) and to the evaluation of its solution $\hat{\Delta}$. The cost $\mathcal{C}_{r,\text{CSFE}}$, instead, is evaluated as

$$\mathcal{C}_{r,\text{CSFE}} \triangleq \mathcal{C}_{\bar{X}} + \mathcal{C}_{\hat{\rho}} + \mathcal{C}_{\hat{A}} + \mathcal{C}_{X_{k,\hat{\rho}}} + \mathcal{C}_{\hat{\Delta}} \tag{I.3}$$

where: a) $\mathcal{C}_{\bar{X}} = 6N+4$ is the contribution due to the computation of $\bar{X}(\hat{F})$ on the basis of eq. (4.27); b) $\mathcal{C}_{\hat{\rho}} = 1$ is due to the evaluation of $\hat{\rho}^{(i-1)}$ on the basis of eq. (4.58); c) $\mathcal{C}_{\hat{A}} = 6N+2$ is the contribution due to the evaluation of $\hat{A}$ on the basis of eq. (4.33); d) $\mathcal{C}_{X_{k,\hat{\rho}}} = 33N+15$ is the contribution due to the evaluation of the quantities $\{\bar{X}_{k,\hat{\rho}^{(i-1)}}\}$ on the basis of eq. (4.42); e) $\mathcal{C}_{\hat{\Delta}} = 29$ is the contribution due to the computation of the coefficients of the quadratic equation (4.47) and to the evaluation of its solution $\hat{\Delta}$. An interpolation technique can be employed to compute the quantities $\bar{X}(\hat{F})$ and $\{\bar{X}_{k,\hat{\rho}^{(i-1)}}\}$. In this case, $\mathcal{C}_{\bar{X}} = 4I^2+14I-8$ flops and

$\mathcal{C}_{X_{k,\hat{\rho}}} = 3(4I^2 + 14I - 8)$ flops are needed for the computation of $\bar{X}(\hat{F})$ and $\bar{X}_{k,\hat{\rho}^{(i-1)}}$, respectively, if a barycentric interpolation technique is used. Substituting the terms appearing in the RHSs of eqs. (I.2) and (I.3) with their expressions and, then, the resulting expressions in the RHS of eq. (I.1) gives

$$\begin{aligned}\mathcal{C}_{\text{CSFE}} = {} & 4N_0 + 8N_0 \log_2 N_0 + 33N + 44 \\ & + N_{\text{CSFE}}(39N + 49).\end{aligned} \tag{I.4}$$

From the last expression, eq. (4.101) can be easily inferred.

I.2 Single Frequency Estimator

The overall computational cost of each this algorithm can be expressed as

$$\mathcal{C}_{\text{SFE}} = \mathcal{C}_{i,\text{SFE}} + N_{\text{SFE}}\, \mathcal{C}_{r,\text{SFE}}, \tag{I.5}$$

where $\mathcal{C}_{i,\text{SFE}}$, $\mathcal{C}_{r,\text{SFE}}$ and N_{SFE} represent the cost of its initialization, that of each of its iterations and the overall number of iterations, respectively. The general criteria adopted in estimating the computational cost of both $\mathcal{C}_{i,\text{SFE}}$ and $\mathcal{C}_{r,\text{SFE}}$ are the same as those illustrated in the previous section. The cost $\mathcal{C}_{i,\text{SFE}}$ is evaluated as

$$\mathcal{C}_{i,\text{SFE}} \triangleq \mathcal{C}_{\mathbf{X}_0} + \quad \mathcal{C}_{\hat{\alpha}} + \mathcal{C}_{K_p} + \mathcal{C}_{X_{k,\hat{\alpha}}} + \mathcal{C}_{\hat{\Delta}}, \tag{I.6}$$

where: a) $\mathcal{C}_{\mathbf{X}_0} = 8N_0 \log_2 N_0$ is the contribution due to the computation of the vector $\mathbf{X}_0$ (4.30); b) $\mathcal{C}_{\hat{\alpha}} = 4N_0$ is the contribution due to the computation of $\hat{\alpha}$ on the basis of eq. (4.78); c) $\mathcal{C}_{K_p} = 6\log_2 N + 151$ is the contribution due to the evaluation of the quantities $\{K_1(2\hat{\alpha}), K_2(2\hat{\alpha}), K_3(2\hat{\alpha})\}$ on the basis of eq. (4.76); d) $\mathcal{C}_{X_{k,\hat{\alpha}}} = 14N + 10$ is the contribution due to computation of $\bar{X}_{k,\hat{\alpha}}$ on the basis of eq. (4.42); e) $\mathcal{C}_{\hat{\Delta}} = 65$ is the contribution due to the computation of the coefficients of the quadratic equation (4.47) and to the evaluation of its solution $\hat{\Delta}$. The cost $\mathcal{C}_{r,\text{SFE}}$, instead, is evaluated as

$$\mathcal{C}_{r,\text{SFE}} \triangleq \mathcal{C}_g + \mathcal{C}_{\bar{X}} + \mathcal{C}_{\hat{\rho}} + \mathcal{C}_{\hat{C}} + \mathcal{C}_{X_{k,\hat{\rho}}} + \mathcal{C}_{K_p} + \mathcal{C}_{\hat{\Delta}} \tag{I.7}$$

where: a) $\mathcal{C}_g = 15$ is the contribution due to the computation of $g(\hat{F})$ on the basis of eq. (4.74); b) $\mathcal{C}_{\bar{X}} = 6N + 4$ is the contribution due to the computation of $\bar{X}(\hat{F})$ on the basis of eq. (4.27); c) $\mathcal{C}_{\hat{\rho}} = 1$ is due to the evaluation of $\hat{\rho}^{(i-1)}$ on the basis of eq. (4.58); d) $\mathcal{C}_{\hat{C}} = 17$ is the contribution due to the evaluation of the complex amplitude $\hat{C}$ on the basis of eq. (4.73); e) $\mathcal{C}_{X_{k,\hat{\rho}}} = 14N + 10$ is the contribution due to the evaluation of the quantities $\{\bar{X}_{k,\hat{\rho}^{(i-1)}}\}$ on the basis of eq. (4.42); f) the cost $\mathcal{C}_{K_p} = 6\log_2 N + 151$ is the contribution due to the evaluation of the quantities $\{K_p(2\hat{\rho}^{(i-1)})\}$ on the basis of eq. (4.76) g) $\mathcal{C}_{\hat{\Delta}} = 65$ is the contribution due to the computation of the coefficients of the quadratic equation (4.47) and to the evaluation of its solution $\hat{\Delta}$. It is worth mentioning that, as explained in Paragraph 4.3.2, an interpolation technique can be used to compute the quantities $\bar{X}(\hat{F})$ and $\{\bar{X}_{k,\hat{\rho}^{(i-1)}}\}$. In this case, $\mathcal{C}_{\bar{X}} = 4I^2 + 14I - 8$ flops and $\mathcal{C}_{X_{k,\hat{\rho}}} = 3(4I^2 + 14I - 8)$ flops are needed for the computation of $\bar{X}(\hat{F})$ and $\{\bar{X}_{k,\hat{\rho}^{(i-1)}}\}$, respectively, if a *barycentric interpolation technique* is used [61]. Substituting the terms appearing in the RHSs of eqs. (I.6) and (I.7) with their expressions and, then, the resulting expressions in the RHS of eq. (I.5) yields

$$\begin{aligned}\mathcal{C}_{\text{SFE}} = {} & 4N_0 + 14N + 8N_0 \log_2 N_0 + 6\log_2 N + 226 \\ & + N_{\text{SFE}}(20N + 6\log_2 N + 269).\end{aligned} \tag{I.8}$$

From the last expression, eq. (4.102) can be easily inferred.

J

Computational complexity of the RASCA-FC3

In this Appendix, the computational complexity, in terms of flops, is assessed for the RASCA-FC3 developed in Paragraph 5.4. The overall computational cost of this algorithm can be expressed as

$$\mathcal{C}_{\text{FC3}} = N_{\text{VR}}\,\mathcal{C}_{\text{T}_1} + N_A\,K_{\text{T}_2}\,\mathcal{C}_{\text{T}_2} + L_b\,K_{\text{T}_2}\,\mathcal{C}_{\text{T}_3} + \mathcal{C}_{\text{T}_{\text{sc}}}, \tag{J.1}$$

where $\mathcal{C}_{\text{T}_1}$ is the contribution due to the first task of the the RASCA-FC3, K_{T_2} (K_{T_3}) represents the overall number of iterations carried out by the STDREC (STDAEC) algorithm, $\mathcal{C}_{\text{T}_2}$ ($\mathcal{C}_{\text{T}_3}$) is the contribution due to a single iteration of the STDREC (STDAEC) executed on a single VA (on the whole virtual array for a given frequency bin) and $\mathcal{C}_{\text{T}_{\text{sc}}}$ is the contribution due to the computation of the spatial coordinates of the overall image. The general criteria adopted in estimating the computational costs appearing in the RHS of (J.1) are illustrated in [90] and can be summarised as follows:

- $4d-2$ flops are required to compute the inner product $\mathbf{u}_c^T\,\mathbf{v}$ of the d-dimensional complex column vector $\mathbf{u}_c$ and the d-dimensional real column vector $\mathbf{v}$;
- $6d+2(d-1)$ flops are required to compute the inner product $\mathbf{u}_c^T\,\mathbf{v}_c$ of the d-dimensional complex vectors $\mathbf{u}_c$ and $\mathbf{v}_c$;
- d flops are required to find the largest element of d-dimensional real vector $\mathbf{v}$;
- $4d^2+14d-8$ flops are required to compute an interpolation based on the elements of the d-dimensional complex vector $\mathbf{v}$.
- $8d\log_2(d)$ flops are required to compute the FFT of the d-dimensional complex vector $\mathbf{v}$.

The expressions of the computational costs associated with each of the three tasks of the RASCA-FC3 are illustrated below.

$\mathbf{T}_1$ - The cost $\mathcal{C}_{\text{T}_1}$ can be expressed as

$$\mathcal{C}_{\text{T}_1} \triangleq \mathcal{C}_{\mathbf{x}_{k,\text{ZP}}} + \mathcal{C}_{\mathbf{X}_k}, \tag{J.2}$$

where: a) $\mathcal{C}_{\mathbf{x}_{\text{ZP}}} = 4\,N$ is the contribution due to the computation of the vectors $\{\mathbf{x}_{k,\text{ZP}}^{(v)};\ k=0,1,2\}$ (see (5.38)–(5.40)); b) $\mathcal{C}_{\mathbf{X}} = 24N_0\log_2 N_0$ is the contribution due to the computation of the vectors $\{\mathbf{X}_k^{(v)};\ k=0,1,2\}$ (see (5.42)).

$\mathbf{T}_2$ - The computational cost of this task is mainly due to its main algorithm, i.e., to the STDREC algorithm. The cost $\mathcal{C}_{\text{T}_2}$ can be expressed as

$$\mathcal{C}_{\text{T}_2} \triangleq \mathcal{C}_{\text{CSFE}} + \mathcal{C}_{\mathbf{C}_{X_k}} + \mathcal{C}_E, \tag{J.3}$$

where: a) $\mathcal{C}_{\text{CSFE}} = 4N_{\text{CSFE}}I^2$ is the cost originating from the CSFE[1] employed in **STDREC-S1**; b) $\mathcal{C}_{\mathbf{C}_{X_k}} = 18N_0$ is the contribution due to the computation of the vectors $(\mathbf{C}_{X_0}^{(v_k)}[i], \mathbf{C}_{X_1}^{(v_k)}[i], \mathbf{C}_{X_2}^{(v_k)}[i])$

[1] Note that, in this case, the cost of the CSFE does not account for the evaluation of three DFTs, since these have been already evaluated in **T1**.

(see (5.120)–(5.122)); c) $\mathcal{C}_E = 8N_0 - 2$ is the contribution due to the computation of the residual energy (see (5.49)).

$\mathbf{T}_3$ - The cost $\mathcal{C}_{\mathrm{T}_3}$ can be expressed as

$$\mathcal{C}_{\mathrm{T}_3} \triangleq \mathcal{C}_{\mathrm{CSFE_V}} + \mathcal{C}_{\mathbf{X}^{(\mathrm{VF})}} + \mathcal{C}_{\mathrm{CSFE_H}} + \mathcal{C}_{\mathbf{X}_{\mathrm{OF}}} + \mathcal{C}_{\mathrm{CSFE_{OF}}} + \mathcal{C}_E, \tag{J.4}$$

where: a) $\mathcal{C}_{\mathrm{CSFE_V}} = 8\bar{N}_0 \log_2(\bar{N}_0) + 4N_{\mathrm{CSFE}}I^2$ is the cost originating from the CSFE employed in **STDAE-S1**; b) $\mathcal{C}_{\mathbf{X}^{(\mathrm{VF})}} = 6N_{\mathrm{VV}}N_{\mathrm{VH}} + 2N_{\mathrm{VV}}$ is the contribution due to the computation of the vertically folded spectrum $\mathbf{X}_i^{(\mathrm{VF})}[l]$ (5.71) in **STDAE-S2**; c) $\mathcal{C}_{\mathrm{CSFE_H}} = 8\bar{N}_0 \log_2(\bar{N}_0) + 4N_{\mathrm{CSFE}}I^2$ is the cost originating from the CSFE employed in **STDAE-S3**; d) $\mathcal{C}_{\mathbf{X}_{\mathrm{OF}}} = 6N_{\mathrm{VV}}N_{\mathrm{VH}} + 18N_{\mathrm{VV}}N_{\mathrm{VH}}N_0$ is the contribution due to the computation of the overall folded spectrum $\{\mathbf{X}_{m,\mathrm{OF}}[l]; m = 0, 1, 2\}$ (see (5.73) and (5.78)); e) $\mathcal{C}_{\mathrm{CSFE_{OF}}} = 4N_{\mathrm{CSFE}}I^2$ is the cost due to the CSFE[2] in **STDAE-S4**; f) $\mathcal{C}_E = 6N_{\mathrm{VV}}N_{\mathrm{VH}}$ is the contribution due to the computation of the residual energy in **STDAEC-S3** (see (5.58)).

Finally, the cost $\mathcal{C}_{\mathrm{T_{sc}}} = 5L$ is required to generate the overall point cloud.

Based on the results illustrated above, (J.1) can be rewritten as

$$\begin{aligned}\mathcal{C}_{\mathrm{FC3}} =& N_{\mathrm{VR}}(4N + 24N_0 \log_2(N_0)) \\ &+ N_{\mathrm{T}_2}(4N_{\mathrm{CSFE}}I^2 + 26N_0 - 2) \\ &+ N_{\mathrm{T}_3}(12N_{\mathrm{CSFE}}I^2 + 2N_{\mathrm{VV}} + 18N_{\mathrm{VH}}N_{\mathrm{VV}} \\ &+ 18N_{\mathrm{VH}}N_{\mathrm{VV}}N_0 + 16\bar{N}_0 \log_2(\bar{N}_0) + 5L,\end{aligned} \tag{J.5}$$

where $N_{\mathrm{T}_2} \triangleq N_A\, K_{\mathrm{T}_2}$ and $N_{\mathrm{T}_3} \triangleq L_b\, K_{\mathrm{T}_2}$.

[2]Note that, in this case, the cost of the CSFE does not account for the evaluation of three DFTs, since these are made available by overall folding.

Bibliography

[1] P. Di Viesti, A. Davoli, G. Guerzoni, and G. M. Vitetta, "Novel Methods for Approximate Maximum Likelihood Estimation of Multiple Superimposed Undamped Tones and Their Application to Radar Systems," Jul. 2021. [Online]. Available: https://doi.org/10.36227/techrxiv.15054321

[2] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive Radars: A Review of Signal Processing Techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017.

[3] T. Schmidl and D. Cox, "Robust Frequency and Timing Synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45, no. 12, pp. 1613–1621, Dec. 1997.

[4] B. Fleury, M. Tschudin, R. Heddergott, D. Dahlhaus, and K. Ingeman Pedersen, "Channel Parameter Estimation in Mobile Radio Environments Using the SAGE Algorithm," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 3, pp. 434–450, Mar. 1999.

[5] Y. Baoguo, K. B. Letaief, R. S. Cheng, and Z. Cao, "Channel estimation for OFDM transmission in multipath fading channels based on parametric channel modeling," *IEEE Trans. Commun.*, vol. 49, no. 3, pp. 467–479, 2001.

[6] J. Liang, J. He, and W. Yu, "3D Parametric Channel Estimation for Multi-User Massive-MIMO OFDM Systems," in *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2020, pp. 1–5.

[7] L. Xu, C. Qian, F. Gao, W. Zhang, and S. Ma, "Angular Domain Channel Estimation for mmWave Massive MIMO With One-Bit ADCs/DACs," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 969–982, 2021.

[8] P. Whittle, "The Simultaneous Estimation of a Time Series Harmonic Components and Covariance Structure," p. 15, 1952.

[9] A. M. Walker, "On the Estimation of a Harmonic Component in a Time Series with Stationary Independent Residuals," *Biometrika*, vol. 58, no. 1, pp. 21–36, 1971.

[10] D. C. Rife and R. R. Boorstyn, "Multiple Tone Parameter Estimation from Discrete-Time Observations," *Bell Syst. Tech. J.*, vol. 55, no. 9, pp. 1389–1410, Nov. 1976.

[11] 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.

[12] R. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[13] R. Roy and T. Kailath, "ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[14] J. Capon, "High-Resolution Frequency-Wavenumber Spectrum Analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, Aug. 1969.

[15] J. Li and P. Stoica, "An Adaptive Filtering Approach to Spectral Estimation and SAR Imaging," *IEEE Trans. Signal Process.*, vol. 44, no. 6, pp. 1469–1484, Jun. 1996.

[16] T. Yardibi, J. Li, P. Stoica, M. Xue, and A. B. Baggeroer, "Source Localization and Sensing: A Nonparametric Iterative Adaptive Approach Based on Weighted Least Squares," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 1, pp. 425–443, Jan. 2010.

[17] P. Gough, "A Fast Spectral Estimation Algorithm Based on the FFT," *IEEE Trans. Signal Process.*, vol. 42, no. 6, pp. 1317–1322, Jun. 1994.

[18] 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.

[19] J. Rissanen, "A Universal Prior for Integers and Estimation by Minimum Description Length," *Ann. Stat.*, vol. 11, no. 2, pp. 416–431, Jun. 1983.

[20] D. C. Rife and G. A. Vincent, "Use of the Discrete Fourier Transform in the Measurement of Frequencies and Levels of Tones," *Bell Syst. Tech. J.*, vol. 49, no. 2, pp. 197–228, Feb. 1970.

[21] B. Quinn, "Estimating Frequency by Interpolation Using Fourier Coefficients," *IEEE Trans. Signal Process.*, vol. 42, no. 5, pp. 1264–1268, May 1994.

[22] ——, "Estimation of Frequency, Amplitude, and Phase from the DFT of a Time Series," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 814–817, Mar. 1997.

[23] E. Jacobsen and P. Kootsookos, "Fast, Accurate Frequency Estimators [DSP Tips Tricks]," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 123–125, May 2007.

[24] Ç. Candan, "A Method For Fine Resolution Frequency Estimation From Three DFT Samples," *IEEE Signal Process. Lett.*, vol. 18, no. 6, pp. 351–354, Jun. 2011.

[25] ——, "Analysis and Further Improvement of Fine Resolution Frequency Estimation Method From Three DFT Samples," *IEEE Signal Process. Lett.*, vol. 20, no. 9, pp. 913–916, Sep. 2013.

[26] U. Orguner and Ç. Candan, "A Fine-Resolution Frequency Estimator Using an Arbitrary Number of DFT Coefficients," *Signal Processing*, vol. 105, pp. 17–21, Dec. 2014.

[27] C. Yang and G. Wei, "A Noniterative Frequency Estimator With Rational Combination of Three Spectrum Lines," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 5065–5070, Oct. 2011.

[28] X. Liang, A. Liu, X. Pan, Q. Zhang, and F. Chen, "A New and Accurate Estimator With Analytical Expression for Frequency Estimation," *IEEE Commun. Lett.*, vol. 20, no. 1, pp. 105–108, Jan. 2016.

[29] L. Fan and G. Qi, "Frequency Estimator of Sinusoid Based on Interpolation of Three DFT Spectral Lines," *Signal Processing*, vol. 144, pp. 52–60, Mar. 2018.

[30] L. Fan, G. Qi, J. Xing, J. Jin, J. Liu, and Z. Wang, "Accurate Frequency Estimator of Sinusoid Based on Interpolation of FFT and DTFT," *IEEE Access*, vol. 8, pp. 44 373–44 380, 2020.

[31] 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.

[32] T. Abatzoglou, "A Fast Maximum Likelihood Algorithm for Frequency Estimation of a Sinusoid Based on Newton's Method," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 33, no. 1, pp. 77–89, Feb. 1985.

[33] Y. V. Zakharov and T. C. Tozer, "Frequency Estimator with Dichotomous Search of Periodogram Peak," *Electron. Lett.*, vol. 35, no. 19, pp. 1608–1609, Sep. 1999.

[34] E. Aboutanios, "A Modified Dichotomous Search Frequency Estimator," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 186–188, Feb. 2004.

[35] 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.

[36] E. Aboutanios, "Estimating the Parameters of Sinusoids and Decaying Sinusoids in Noise," *IEEE Instrum. Meas. Mag.*, vol. 14, no. 2, pp. 8–14, Apr. 2011.

[37] Y. Liu, "Generalization of Iterative Fourier Interpolation Algorithms for Single Frequency Estimation," *Digit. Signal Process.*, p. 9, 2011.

[38] I. Djurovic, "Estimation of the Sinusoidal Signal Frequency Based on the Marginal Median DFT," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2043–2051, May 2007.

[39] J.-R. Liao and C.-M. Chen, "Analysis and Reduction of Estimation Bias for an Iterative Frequency Estimator of Complex Sinusoid," in *Proc. of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6138–6142.

[40] J.-R. Liao and S. Lo, "Analytical Solutions for Frequency Estimators by Interpolation of DFT Coefficients," *Signal Processing*, vol. 100, pp. 93–100, Jul. 2014.

[41] Y. V. Zakharov, V. M. Baronkin, and T. C. Tozer, "DFT-Based Frequency Estimators with Narrow Acquisition Range," *IEE Proc. - Commun.*, vol. 148, no. 1, pp. 1–7, Feb. 2001.

[42] C.-F. Huang, H.-P. Lu, and W.-H. Chieng, "Estimation of Single-Tone Signal Frequency with Special Reference to a Frequency-Modulated Continuous Wave System," *Meas. Sci. Technol.*, vol. 23, no. 3, Mar. 2012.

[43] 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.

[44] ——, "Rapid Accurate Frequency Estimation of Multiple Resolved Exponentials in Noise," *Signal Processing*, vol. 132, pp. 29–39, Mar. 2017.

[45] A. Serbes and K. Qaraqe, "A Fast Method for Estimating Frequencies of Multiple Sinusoidals," *IEEE Signal Process. Lett.*, vol. 27, pp. 386–390, 2020.

[46] A. Serbes, "Fast and Efficient Estimation of Frequencies," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 4054–4066, Jun. 2021.

[47] J. A. Hogbom, "Aperture Synthesis with Non-Rectangular Distribution of Interferometric Baselines," *Astron Astrophys Suppl*, vol. 15, no. 3, pp. 417–426, Jun. 1974.

[48] 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, no. 6, pp. 52–69, Sep. 2017.

[49] 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.

[50] A. Serbes, "Fast and Efficient Sinusoidal Frequency Estimation by Using the DFT Coefficients," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2333–2342, Mar. 2019.

[51] T. Grandke, "Interpolation Algorithms for Discrete Fourier Transforms of Weighted Signals," *IEEE Trans. Instrum. Meas.*, vol. 32, no. 2, pp. 350–355, Jun. 1983.

[52] M. Macleod, "Fast DFT-Domain Algorithms for Near-Optimal Tonal Detection and Frequency Estimation," *Institute of Acoustics Proceedings*, vol. 13, pp. 102–109, 1991.

[53] D. Belega and D. Petri, "Sine-Wave Parameter Estimation by Interpolated DFT Method Based on New Cosine Windows with High Interference Rejection Capability," *Digital Signal Processing*, vol. 33, pp. 60–70, Oct. 2014.

[54] K. Duda, L. B. Magalas, M. Majewski, and T. P. Zielinski, "DFT-Based Estimation of Damped Oscillation Parameters in Low-Frequency Mechanical Spectroscopy," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 11, pp. 3608–3618, Nov. 2011.

[55] M. Feder and E. Weinstein, "Parameter Estimation of Superimposed Signals Using the EM Algorithm," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 4, pp. 477–489, Apr. 1988.

[56] J. Fessler and A. Hero, "Space-Alternating Generalized Expectation-Maximization Algorithm," *IEEE Trans. Signal Process.*, vol. 42, no. 10, pp. 2664–2677, Oct. 1994.

[57] P. Bloomfield, *Fourier Analysis of Time Series: An Introduction.* New York: Wiley, 1976.

[58] I. Ziskind and M. Wax, "Maximum Likelihood Localization of Multiple Sources by Alternating Projection," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 36, no. 10, pp. 1553–1560, Oct. 1988.

[59] A. Davoli, G. Guerzoni, and G. M. Vitetta, "Machine Learning and Deep Learning Techniques for Colocated MIMO Radars: A Tutorial Overview," *IEEE Access*, vol. 9, pp. 33 704–33 755, 2021.

[60] J. Gamba, *Radar Signal Processing for Autonomous Driving.* Springer Singapore, 2020.

[61] J. Selva, "Efficient Wideband DOA Estimation Through Function Evaluation Techniques," *IEEE Trans. Signal Process.*, vol. 66, no. 12, pp. 3112–3123, Jun. 2018.

[62] S. Djukanović and V. Popović-Bugarin, "Efficient and accurate detection and frequency estimation of multiple sinusoids," *IEEE Access*, vol. 7, pp. 1118–1125, 2019.

[63] Y.-X. Yao and S. Pandit, "Cramer-Rao Lower Bounds for a Damped Sinusoidal Process," *IEEE Trans. Signal Process.*, vol. 43, no. 4, pp. 878–885, Apr. 1995.

[64] "Imaging radar using cascaded mmWave sensor reference design - TIDEP-01012." [Online]. Available: https://www.ti.com/tool/TIDEP-01012

[65] "CamBoard pico flexx." [Online]. Available: https://pmdtec.com/picofamily/

[66] P. Di Viesti, A. Davoli, G. Guerzoni, and G. M. Vitetta, "Novel deterministic detection and estimation algorithms for colocated multiple-input multiple-output radars," *IEEE Access*, vol. 10, pp. 2216–2255, 2022.

[67] A. M. Haimovich, R. S. Blum, and L. J. Cimini, "MIMO Radar with Widely Separated Antennas," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 116–129, 2008.

[68] E. Fishler, A. Haimovich, R. Blum, R. Cimini, D. Chizhik, and R. Valenzuela, "Performance of MIMO radar systems: advantages of angular diversity," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, 2004, pp. 305–309 Vol.1.

[69] M. A. Richards, *Fundamentals of Radar Signal Processing.* New York: McGraw-Hill, 2005.

[70] S. Rao, "MIMO Radar," *Texas Instruments - Application Report SWRA554A*, July 2018.

[71] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, "Overview of Bayesian Sequential Monte Carlo Methods for Group and Extended Object Tracking," *Digital Signal Processing*, vol. 25, pp. 1–16, Feb. 2014.

[72] D. Oh and J. Lee, "Low-Complexity Range-Azimuth FMCW Radar Sensor Using Joint Angle and Delay Estimation Without SVD and EVD," *IEEE Sensors Journal*, vol. 15, no. 9, pp. 4799–4811, Sep. 2015.

[73] J.-J. Lin, Y.-P. Li, W.-C. Hsu, and T.-S. Lee, "Design of an FMCW radar baseband signal processing system for automotive application," *SpringerPlus*, vol. 5, no. 42, pp. 1–16, Jan. 2016.

[74] 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.

[75] A. A. Gorji, R. Tharmarasa, W. D. Blair, and T. Kirubarajan, "Multiple Unresolved Target Localization and Tracking Using Colocated MIMO Radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2498–2517, Jul. 2012.

[76] J. Li and P. Stoica, "MIMO Radar with Colocated Antennas," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 106–114, Sep. 2007.

[77] L. Xu, J. Li, and P. Stoica, "Target Detection and Parameter Estimation for MIMO Radar Systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 927–939, Jul. 2008.

[78] S. Zuther, M. Biggel, M. M. Muntzinger, and K. Dietmayer, "Multi-Target Tracking for Merged Measurements of Automotive Narrow-Band Radar Sensors," in *2009 12th International IEEE Conference on Intelligent Transportation Systems*, Oct. 2009, pp. 1–6.

[79] J. Li and P. Stoica, Eds., *MIMO Radar Signal Processing.* Hoboken, NJ: J. Wiley & Sons, 2009.

[80] L. L. Scharf and C. Demeure, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis.* Addison-Wesley Pub. Co, 1991.

[81] 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, Sep. 2019.

[82] S. Sun, A. P. Petropulu, and H. V. Poor, "MIMO Radar for Advanced Driver-Assistance Systems and Autonomous Driving: Advantages and Challenges," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 98–117, Jul. 2020.

[83] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. New York, NY: Springer, 2006.

[84] O. Simeone, "A Very Brief Introduction to Machine Learning With Applications to Communication Systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, 2018.

[85] N. Shlezinger, R. Fu, and Y. C. Eldar, "Deep Soft Interference Cancellation for MIMO Detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8881–8885.

[86] C. M. Schmid, S. Schuster, R. Feger, and A. Stelzer, "On the Effects of Calibration Errors and Mutual Coupling on the Beam Pattern of an Antenna Array," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 8, pp. 4063–4072, Aug. 2013.

[87] C. M. Schmid, C. Pfeffer, R. Feger, and A. Stelzer, "An FMCW MIMO Radar Calibration and Mutual Coupling Compensation Approach," in *2013 European Radar Conference*, Oct. 2013, pp. 13–16.

[88] 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.

[89] "INRAS RadarLog device." [Online]. Available: http://www.inras.at/en/products/radarlog.html

[90] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., ser. Johns Hopkins Studies in the Mathematical Sciences. Baltimore: Johns Hopkins University Press, 1996.

Six

Conclusions and Future Research Activities

In this chapter, some conclusions and suggestions for future research activities are illustrated.

Conclusions

In this dissertation, two lines of research have been presented. First, a FG approach has been adopted to develop new solutions to the filtering and smoothing problems. First, this approach has allowed us to provide a new perspective on the filtering problem for CLG SSMs; this has led to a new interpretation of *marginalized particle filter* (MPF) as a *forward only* message passing algorithm over a proper FG and to the derivation of a new filtering algorithm, called *dual* MPF (DMPF). Then, it has been exploited to devise a family of filtering algorithms, collectively dubbed *multiple Bayesian filtering*, that involve multiple interconnected Bayesian filters running in parallel. An example of this family, based on the interconnection of an extended Kalman filter with a particle filter, has been derived for CLG SSMs. Finally, the conceptual approach proposed for filtering has been exploited to develop new smoothing algorithms based on the interconnection of two Bayesian filters in the forward pass and of two backward information filters in the backward pass. This has allowed us to derive a new approximate method for Bayesian smoothing, called *double Bayesian smoothing*. All the obtained results have confirmed that FGs represent a powerful tool to: a) describe and interpret known algorithms as message passing; b) derive novel iterative algorithms in cases in which the graphical model underlying the considered problem is not cycle-free. Simulation results obatined for specific SSMs have evidenced that the developed filtering techniques perform closely to other well known filtering methods, but are appreciably faster or offer a better tracking capability. As far as the new smoothers are concerned, they have been compared, in terms of both accuracy and execution time, with other smoothing algorithms for specific dynamic models; our simulation results have evidenced that the devised algorithms can achieve a better complexity-accuracy tradeoff and a better tracking capability.

Secondly, novel methods for 2D and 3D radar imaging have been developed. These methods rely on two new algorithms, dubbed *single frequency estimator* (SFE) and *complex* SFE (CSFE), for detecting and estimating a single real or complex tone. Such methods can be exploited to estimate multiple tones through a serial cancellation procedure. The accuracy and robustness of the devised single tone and multiple tone estimators have been assessed by means of extensive computer simulations. In particular, our numerical results have evidenced that these estimators outperform all the other related estimators available in the technical literature in terms of probability of convergence and accuracy, in the presence of multiple tones. The application of these methods to a colocated MIMO FMCW radar system has led to the development of four novel algorithms, dubbed *range & angle serial cancellation algorithms* (RASCAs), for the detection and the estimation of multiple targets in colocated MIMO FMCW radar systems. All these algorithms can be seen as instances of a general approach to target detection and estimation. Our computer simulations, run on both synthetically generated data and measurements acquired through commercial devices, have evidenced that the RASCAs are able to generate accurate 2D and 3D radar images in the presence of multiple closely spaced targets.

Future research activities

Future research activities in line with the work illustrated in this thesis include:

a) *The application of multiple Bayesian filtering to multiple target tracking in MIMO radar systems* – Recently, factor graphs and the SPA have been exploited to solve the problem of multi target tracking in the cases of both point targets and extended targets [1, 2]. Even though the computational complexity of this methods is lower than that of other methods appeared in the literature, it is still high. Multiple Bayesian filtering might represent a useful tool to develop simpler filtering methods having a scalable and parallelizable structure.

b) *The development of 2D and 3D radar imaging for dynamic propagation scenarios* – The algorithms for 2D and 3D radar imaging (namely, the RASCAs) have been developed under the assumption of a static or slowly changing propagation scenario. The same conceptual approach as that employed in the development of the SFE and the CSFE could be be exploited to devise algorithms for the detection of multiple point targets, and the estimation of both their spatial coordinates and radial velocity. Multi target tracking algorithms, in turn, can leverage on speed information to improve their accuracy.

c) *The development of 2D and 3D radar imaging for MIMO OFDM and OTFS radars* – Recently, substantial attention has been paid to the possibility of combining sensing and communications in vehicular applications. In ref. [3], a joint radar parameter estimation based on the *orthogonal time frequency space* (OTFS) modulation has been developed; the numerical results shown in that manuscript lead to the conclusion that radars employing that modulation format should achieve an estimation accuracy similar to that provided by state-of-the art FMCW radars. Future efforts should be devoted to the development of novel algorithms, based on the approach illustrated in the derivation of the SFE and the CSFE, for the detection of multiple targets and the estimation of their parameters in OTFS-based radar systems.

Bibliography

[1] F. Meyer, T. Kropfreiter, J. L. Williams, R. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message Passing Algorithms for Scalable Multitarget Tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[2] F. Meyer and J. Williams, "Scalable Detection and Tracking of Geometric Extended Objects," *IEEE Transactions on Signal Processing*, pp. 1–1, 2021.

[3] L. Gaudio, M. Kobayashi, G. Caire, and G. Colavolpe, "On the Effectiveness of OTFS for Joint Radar Parameter Estimation and Communication," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 5951–5965, 2020.