UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

ROBUST TECHNIQUES FOR MULTIPLE TARGET
TRACKING AND FULLY ADAPTIVE RADAR

(TÉCNICAS ROBUSTAS PARA SEGUIMIENTO DE
MÚLTIPLES BLANCOS Y RADAR ADAPTATIVO)

TESIS DOCTORAL

Luis Antonio Úbeda Medina

Ingeniero de Telecomunicación

2018

DEPARTAMENTO DE SEÑALES, SISTEMAS Y
RADIOCOMUNICACIONES

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE MADRID

ROBUST TECHNIQUES FOR MULTIPLE TARGET
TRACKING AND FULLY ADAPTIVE RADAR

(TÉCNICAS ROBUSTAS PARA SEGUIMIENTO DE
MÚLTIPLES BLANCOS Y RADAR ADAPTATIVO)

TESIS DOCTORAL

Autor:

Luis Antonio Úbeda Medina

Ingeniero de Telecomunicación

Directores:

Jesús Grajal de la Fuente

Catedrático de Universidad del Dpto. de Señales, Sistemas y Radiocomunicaciones

Universidad Politécnica de Madrid

Ángel Froilán García Fernández

Lecturer in the Department of Electrical Engineering and Electronics

University of Liverpool

2018

TESIS DOCTORAL

ROBUST TECHNIQUES FOR MULTIPLE TARGET TRACKING AND FULLY
ADAPTIVE RADAR

(TÉCNICAS ROBUSTAS PARA SEGUIMIENTO DE MÚLTIPLES BLANCOS
Y RADAR ADAPTATIVO)

AUTOR: Luis Antonio Úbeda Medina

DIRECTORES: Jesús Grajal de la Fuente

Ángel Froilán García Fernández

Tribunal nombrado por el Magfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid,
el día __ de _____ de 2018.

PRESIDENTE:

SECRETARIO:

VOCAL:

VOCAL:

VOCAL:

SUPLENTE:

SUPLENTE:

Realizado el acto de defensa y lectura de la Tesis el día __ de _____ de 2018.

En la E.T.S. de Ingenieros de Telecomunicación.

Calificación:

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

*‘One, remember to look up at the stars
and not down at your feet.*

*Two, never give up work. Work gives
you meaning and purpose and life is
empty without it.*

*Three, if you are lucky enough to find
love, remember it is rare and don’t throw
it away.’*

Stephen Hawking

Agradecimientos

Además del contenido académico, esta tesis oculta entre sus líneas la narración de una historia. No es otra que la de los diferentes eventos de mi biografía personal que han tenido lugar en todo este tiempo. Pequeños detalles en cada capítulo hablan de las diferentes personas que han inspirado algunas de las ideas que aquí se recogen. La bibliografía, que recopila las diferentes publicaciones en las que he tenido la suerte de ser coautor, no solamente cuenta las horas de trabajo en equipo que he compartido con excelentes compañeros, deja ver también los viajes por el mundo para acudir a congresos, las personas conocidas y las experiencias vividas en ellos. Las variaciones en el ritmo de mi producción académica hablan de los sucesos, algunos felices y otros no tanto, que han influido en mi vida en este tiempo. Desafortunadamente, la narración de cada uno de estos episodios en este documento es tan sutil que resulta difícil apreciarla. Con estas palabras pretendo corregir en lo posible esta circunstancia y poner en relevancia al resto de protagonistas de esta historia, que merecen también el reconocimiento de su contribución a este trabajo.

Es necesario comenzar reconociendo el crédito que merecen mis directores de tesis, Jesús y Ángel. Agradezco sinceramente la confianza depositada y la oportunidad que me dieron para realizar esta tesis, sus infinitas aportaciones a este trabajo, así como la comprensión y actitud positiva que han tenido conmigo en este tiempo ante todo tipo de circunstancias. Así mismo, quiero explicitar mi agradecimiento al Programa Propio de I+D+i de la Universidad Politécnica de Madrid por la financiación parcial de la realización de esta tesis.

Quiero también agradecer la acogida durante todos estos años a todos los miembros del Grupo de Microondas y Radar. Es un orgullo haber pertenecido a esta familia que ha hecho más llevadero el desarrollo de este trabajo. Agradecer, por tanto, el apoyo a mis compañeros durante estos años: Rodrigo, Álvaro, Jordi, Javier, Chema y Carmen. Y por supuesto, a los actuales y pasados miembros del GIT: Diego, Víctor, Gorka, Carlos, Federico, Mario, Clara, Carlos Jr. y Daniel. A Fernando y Marta, además de por todo lo anterior, gracias por acompañarme en la memorable cena de gala en OKC. Me gustaría mencionar especialmente aquí a Alejandro, que ha demostrado una vez

tras otra una confianza ciega en mí.

Sin salir de la escuela, y de manera muy especial, quiero agradecer a los miembros *honorarios* de la Cátedra Orange su increíble apoyo durante estos años. Carlos, Iker, J. y Pedro han inspirado durante numerosas conversaciones soluciones a algunos de los problemas surgidos durante el desarrollo de la tesis. Pero, sobre todo, han cuidado de mí en los momentos más duros de todo este tiempo, algo que nunca podré agradecer lo suficiente.

Siento un especial orgullo en poder afirmar que todos los amigos mencionados hace años en los agradecimientos de mi Proyecto Fin de Carrera han seguido a mi lado formando parte de esta historia. A Héctor, que me acogió en Chicago en una semana que atesoro como uno de los mejores recuerdos de este tiempo. También a todos los compañeros de carrera, a los *Nikelaos*, a Miriam, Paloma, Sonia, Miguel, Daniel y Kapil. Todos ellos me han apoyado y empujado en los mejores y, sobre todo, en los peores momentos de estos años. Muchísimas gracias.

A mi familia: José María, Azucena, Carlos, Ana, Carlos Jr. y Ana Paula. Gracias por el apoyo y el ánimo que me habéis mostrado todo este tiempo. A mi padrino José, a mi tía Esther y, por supuesto, a mi abuela Lourdes: estoy y estaré agradecido por cómo me habéis transmitido siempre vuestro orgullo por mis andanzas.

Finalmente, si hubiera que destacar por su importancia algo de entre todo aquello que he podido llegar a aprender en el tiempo en que se ha desarrollado esta tesis, no tendría nada que ver con sistemas dinámicos ni filtros de partículas. De entre todas, la lección más importante me la han enseñado mis padres y Paula. Su valentía y lucha durante estos últimos años es un ejemplo que valoro más allá de lo que sé expresar con palabras. Relacionado con ello, agradecer muy especialmente el apoyo infinito y constante en todo este tiempo de mi hermana, Ana, así como de Sergio. Es una suerte y un orgullo tenerlos cerca a todos. Quiero, por último, recordar aquí también a Pilar, que me regaló lo mejor que tenía y nunca dejaré de cuidarlo.

A mi padre y a mi madre.

A Ana.

A Paula.

*Por todo el tiempo que os he robado
para realizar esta tesis y por estar a mi
lado en esta lucha, que no es sino la
menor de las que hemos venido librando.*

Juntos.

Resumen

La primera parte de esta tesis se centra en el desarrollo de algoritmos robustos para los problemas de filtrado recursivo en sistemas dinámicos con alta dimensionalidad usando un enfoque bayesiano. La segunda parte de esta tesis se centra en el desarrollo de técnicas bayesianas robustas para gestión de sensores.

Una gran cantidad de problemas en ciencia e ingeniería necesitan de la estimación del estado de un sistema, que cambia a lo largo del tiempo, basándose en una secuencia de medidas ruidosas. El filtrado recursivo bayesiano se lleva a cabo calculando en cada instante la función densidad de probabilidad (FDP) del estado del sistema, dadas las medidas recibidas por los sensores hasta el momento. Esta FDP a posteriori contiene toda la información necesaria para la caracterización del estado del sistema.

Sin embargo, generalmente, la FDP a posteriori no se puede calcular de forma analítica, por lo que es común recurrir a aproximaciones, como las provistas por los filtros de partículas. Desafortunadamente, el rendimiento de estas aproximaciones generalmente se ve severamente afectado de manera negativa cuando la dimensión del espacio en el que puede tomar valores el estado del sistema es alta. Este efecto se conoce como la maldición de la dimensionalidad. La Parte I de esta tesis está dedicada, por tanto, al desarrollo de diferentes filtros de partículas con un rendimiento robusto en problemas de filtrado en espacios de alta dimensionalidad.

Una estrategia comúnmente usada en este tipo de problemas es considerar una partición del espacio de estados, de manera que un filtro de partículas pueda obtener muestras de cada componente de la partición de manera independiente. Siguiendo esta estrategia, y considerando el uso de la técnica de filtrado auxiliar, se propone un primer filtro que es capaz de superar las limitaciones de algoritmos presentados anteriormente en la literatura. Además, se presenta un segundo método que considera la inclusión de una etapa adicional de remuestreo por componentes, que se demuestra útil en casos de alta dimensionalidad del espacio de estados, pagando, no obstante, el precio de la pérdida de diversidad en las muestras de cada componente. Se propone, por tanto, un tercer filtro que decide de manera adaptativa en cada instante de tiempo si es necesario hacer el remuestreo de cada componente del estado. Este último filtro es por tanto

un método robusto, con un buen rendimiento para espacios de dimensionalidad tanto baja como alta, ya que realiza el remuestreo por componentes cuando es necesario, intentando, no obstante, favorecer la diversidad de las muestras mientras es posible.

Una estrategia alternativa para superar la maldición de la dimensionalidad es el uso de filtros múltiples, en los que se aproxima la FDP marginal a posteriori para cada una de las componentes de una partición del estado usando filtros diferentes. El uso de filtros múltiples requiere, sin embargo, del uso de un procedimiento adicional de marginalización, que generalmente se realiza también de manera aproximada.

En esta tesis se detalla la manera de incluir el filtrado auxiliar en un filtro múltiple con una aproximación de primer orden de la integral de marginalización. Se presenta además un filtro que, utilizando métodos de integración mediante cuadratura, es capaz de calcular una aproximación de segundo orden a las integrales de marginalización. Adicionalmente, se presenta otra variante del filtro que, además de usar esta aproximación, incluye el uso de filtrado auxiliar. El rendimiento de los diferentes algoritmos presentados se prueba en un problema de seguimiento de múltiples blancos, obteniendo una mejora notable en el rendimiento respecto a otros filtros similares en la literatura.

Es común que las medidas recibidas puedan depender de algún parámetro de los sensores, que puede ser modificado. En estos casos, la FDP a posteriori del estado dependerá también de los parámetros seleccionados, de manera que es necesario seleccionarlos adecuadamente para maximizar el rendimiento y mejorar la estimación. Este problema de gestión de sensores es tratado en la Parte II de esta tesis.

Enfoques recientes al problema de la gestión de sensores, como el del radar completamente adaptativo (FAR), están inspirados en la visión de la neurociencia sobre los problemas de la cognición y la toma de decisiones. El enfoque del FAR recoge en una arquitectura simple y compacta los principales conceptos de este tipo de planteamientos. Las implementaciones en la literatura de un sistema FAR para los problemas de seguimiento de un blanco y detección y seguimiento simultáneos de un blanco son revisadas, mientras que también se presenta la especialización del sistema para el problema de seguimiento de un número finito y conocido de blancos.

En la parte final de esta tesis, se muestra cómo las implementaciones de un sistema FAR para los problemas de seguimiento de un blanco y detección y seguimiento simultáneos de un blanco previamente revisadas, presentan ciertos problemas de robustez en escenarios que entrañan dificultades, haciendo que caiga severamente el rendimiento del sistema. Estos problemas de robustez son primero caracterizados de manera exhaustiva, para finalmente proponer nuevos algoritmos robustos dentro del esquema de un sistema FAR. Los métodos presentados se prueban mediante

simulaciones en escenarios que entrañan dificultad, demostrando que tienen un rendimiento fiable en este tipo de situaciones.

Abstract

This Ph.D. thesis is concerned with the development of robust methods for high-dimensional recursive Bayesian filtering and sensor management in dynamic systems.

Many problems in science and engineering require the estimation of the state of a system, that changes over time, using a sequence of available noisy measurements. Recursive Bayesian filters sequentially compute at each time step the probability density function (PDF) of the state of a dynamic system given all the received sensor measurements. This posterior PDF includes all the information of interest about the state for estimation purposes.

However, except for a very limited class of models, the posterior PDF cannot be generally computed in closed form, and approximations, such as particle filters, are necessary. Unfortunately, the performance of these approximations generally severely degrades when the dimension of the space in which the state of the system takes values is high, an effect which is commonly referred to as the curse of dimensionality. Part I of this thesis focuses on the development of different particle filtering techniques which can robustly tackle the filtering of high-dimensional states.

A useful strategy to overcome the curse of dimensionality is to consider a partition of the state space, so that samples from each component of the partition can be drawn independently in a particle filter. Following this strategy, the auxiliary parallel partition (APP) method is proposed, which overcomes limitations of previous partitioned particle filters in the literature by considering the use of auxiliary particle filtering. A second filter, which additionally incorporates a component-resampling (target-resampling) stage, is also presented. This filter, the target-resampling APP (TRAPP) is shown to be useful when the dimension of the state space is high, at the cost of a loss in the diversity of the samples of each component. Thus, a third method is considered, the adaptive TRAPP (ATRAPP), which adaptively decides if target-resampling is needed in each component. This makes ATRAPP a robust algorithm, with a reliable performance regardless of the dimension of the state space, performing target resampling when necessary and favoring sample diversity when possible.

An alternative strategy to beat the curse of dimensionality is to make use of multiple filtering, where the marginal posterior PDF of each component of the state is individually estimated using a different filter. Multiple filters, however, require of a marginalization procedure, which generally also needs to be computed in an approximated form.

The inclusion of auxiliary particle filtering along with a first-order approximation to the marginalization procedure in the multiple auxiliary particle filter (MAPF) is detailed. In addition, the sigma-point multiple particle filter (SP-MPF) is presented, which, making use of sigma-point integration methods, computes a second-order approximation to the required marginalization procedure. Finally, auxiliary filtering is also considered within this setting in the sigma-point multiple auxiliary particle filter (SP-MAPF). The presented algorithms are shown to have an outstanding performance with respect to previous methods in the literature in a multiple target tracking scenario.

It is often the case in which the received measurements depend on some parameters of the sensors which can be tuned. The posterior PDF of the state in such cases therefore depends on the selected sensor parameters, so that they need to be carefully chosen to maximize performance. This sensor-management problem is the focus of Part II of this thesis.

Recent approaches to the sensor management problem, such as the fully adaptive radar (FAR), are inspired by the neuroscience approach to decision making and cognition. The FAR framework gathers in a simple and compact architecture the main concepts of this novel approach. Implementations in the literature of the FAR framework for the problems of single target tracking and simultaneous single target detection and tracking are first reviewed, and the specialization of the FAR for the problem of multiple target tracking with a fixed and known number of targets is presented.

In the final part of this thesis, the reviewed FAR implementations in the literature for the problems of single target tracking and simultaneous single target detection and tracking are shown to suffer robustness issues when applied to difficult scenarios. These robustness issues are first thoroughly characterized and alternative robust novel methods within the FAR framework are presented. The proposed methods are shown to reliably perform in these difficult scenarios.

Contents

Contents	xi
List of Figures	xvii
List of Tables	xxiii
List of algorithms	xxv
Nomenclature	xxviii
1 Introduction	1
1.1 Robust techniques for Bayesian dynamic state filtering and sensor management	1
1.2 Contributions	3
1.3 Thesis organization	5
I High-dimensional filtering and multiple target tracking	7
2 Bayesian recursive filtering	9
2.1 Dynamic Systems	10
2.2 Dynamic state estimation	11
2.2.1 Bayesian and non Bayesian estimation	11
2.2.2 Batch and recursive estimation	12
2.2.3 Smoothing, filtering and prediction	12
2.3 Recursive Bayesian filtering	14

2.4	Gaussian filtering with linear models	15
2.5	Gaussian filtering with nonlinear models	18
2.5.1	Extended Kalman filtering	18
2.5.2	Sigma-point Kalman filtering	20
2.5.2.1	Sigma-point integration methods	20
2.5.2.2	Unscented Kalman filtering	21
2.5.3	Discussion on Kalman filtering performance in nonlinear systems	23
2.6	Grid-based filtering methods	24
2.6.1	Approximate grid-based filtering methods	25
2.7	Particle filters	26
2.7.1	Sequential importance sampling	26
2.7.1.1	The particle degeneracy problem	29
2.7.2	Sampling importance resampling	30
2.7.2.1	The cost of resampling	31
2.7.3	Marginal particle filter	33
2.7.4	Optimal importance density	35
2.7.5	Auxiliary particle filter	36
2.8	Curse of dimensionality	37
2.8.1	Rao-Blackwellization	39
2.9	Partitioned particle filters	40
2.9.1	Independent partition and parallel partition particle filters . . .	42
2.10	Multiple particle filtering	49
2.10.1	Multiple particle filter	51
2.10.2	C-PF-PROP	53
3	Auxiliary particle filters for high-dimensional filtering	55
3.1	Auxiliary parallel partition particle filter	56
3.2	Target-resampling auxiliary parallel partition particle filter	63
3.3	On target-resampling	69
3.4	Adaptive target-resampling auxiliary parallel partition particle filter . .	70

3.5	Experimental results	75
3.5.1	Dynamic model	76
3.5.2	Measurement model	77
3.5.3	APP and TRAPP simulation results	79
3.5.4	ATRAPP simulation results	86
3.5.4.1	Constant threshold	86
3.5.4.2	Variable threshold	89
3.6	Conclusions	93
4	Multiple auxiliary particle filter and second-order multiple particle filters	95
4.1	Multiple auxiliary particle filter	96
4.2	Second-order sigma-point multiple particle filters	100
4.2.1	Sigma-point MPF: A sigma-point-based second-order approximation	101
4.2.2	Sigma-point multiple auxiliary particle filter	104
4.3	On the computational complexity of multiple particle filters	107
4.4	Experimental results	110
4.4.1	Dynamic and measurement model	112
4.4.2	Simulation results	113
4.4.2.1	MAPF	117
4.4.2.2	SP-MPF and SP-MAPF	119
4.5	Conclusions	131
II	Sensor management and fully adaptive radar	133
5	Fully adaptive radar: fundamentals and applications	135
5.1	The fully adaptive radar framework	137
5.1.1	Resource management in FAR	141
5.2	Single target tracking in fully adaptive radar	141

5.2.1	The predicted conditional Cramér-Rao lower bound	142
5.2.1.1	PC-CRLB for resource management in a sensor network	145
5.3	Simultaneous target detection and tracking in fully adaptive radar . . .	146
5.3.1	Bayesian recursion for simultaneous target detection and tracking	146
5.3.2	Estimation for target detection and tracking	148
5.3.3	Predicted Bayes risk for target detection and tracking	150
5.4	Multiple target tracking with fixed and known number of targets in fully adaptive radar	151
5.4.1	Experimental results	154
5.4.1.1	Dynamic and measurement model	155
5.4.1.2	Simulation results	156
5.5	Conclusions	156
6	Robust approach to FAR	161
6.1	Robustness issues and limitations of the minimization of PC-CRLB for single target tracking	162
6.1.1	Performance of the PC-CRLB minimization with a large resource budget	164
6.1.2	Performance of the PC-CRLB minimization with a scarce resource budget	166
6.1.3	Practical limitations of the FAR framework	168
6.2	A robust sigma-point approach to single target tracking in FAR	171
6.2.1	Linear-Gaussian approximation for the FAR optimization proce- dure	171
6.2.2	Simulations	175
6.3	Suitability of PC-CRLB minimization for simultaneous target detection and tracking	180
6.4	Bernoulli-Gaussian approach to simultaneous target detection and tracking in the FAR framework	184
6.4.1	Bernoulli-Gaussian approximation	185
6.4.1.1	Specification of the estimator using the Bernoulli- Gaussian approximation	187

6.4.1.2	FAR controller policy using the Bernoulli-Gaussian approximation	188
6.4.2	Note on the accuracy of the Bernoulli-Gaussian approximation .	190
6.4.3	Experimental results	191
6.4.3.1	Dynamic and measurement model	192
6.4.3.2	Simulation results	194
6.5	Conclusions	197
7	Conclusions and future work	201
7.1	Conclusions	201
7.2	Future work	203
7.2.1	High-dimensional filtering	203
7.2.2	Fully adaptive radar	205
A	Additional material and related mathematical derivations	207
A.1	Multivariate Gaussian distributions and related mathematical deriva- tions	207
A.1.1	Multivariate Gaussian PDF	207
A.1.2	PDF of a linearly transformed multivariate Gaussian PDF . . .	207
A.1.2.1	Linear transformation	207
A.1.2.2	Linear transformation with additive Gaussian noise . .	208
A.1.2.3	Affine transformation with additive Gaussian noise . .	209
A.1.3	PDF of jointly Gaussian multivariate variables	210
A.1.3.1	Conditional PDFs	211
A.2	Partitioned particle filter PDF	213
A.3	Minimum mean squared error estimation of a multitarget state using a PF	214
A.4	Sensor EFIM expression for resource management in a sensor network .	217
A.5	Sensor EFIM for received power measurement model	218
A.6	Sensor EFIM for angle of arrival measurement models	219
A.7	Sensor EFIM for additive received power measurement models	220

A.8	Analysis of the PC-CRLB minimization as a function of the available resource budget	221
A.8.1	Angular measurement model	222
A.8.1.1	Derivation of solution bounds	226
A.8.2	Received power measurement model	227
A.8.2.1	Derivation of solution bounds	232
A.9	Metrics for multiple target tracking	233
A.9.1	OSPA metric	234
A.9.2	GOSPA metric	234
	Bibliography	237

List of Figures

2.1	Schematic representation of the hidden Markov model	12
2.2	Schematic representation of the smoothing, filtering and prediction problems	13
2.3	Particle approximation of a nonlinearly transformed density	27
2.4	The SIR PF algorithm for $N = 10$ particles	32
2.5	Sample Importance Resampling PF scheme	33
2.6	Auxiliary sampling in APF	38
2.7	IP and PP sampling scheme	45
3.1	APP sampling procedure (I)	58
3.2	APP sampling procedure (II)	59
3.3	APP sampling procedure (III)	60
3.4	APP sampling scheme	61
3.5	TRAPP sampling procedure (I)	64
3.6	TRAPP sampling procedure (II)	65
3.7	TRAPP sampling scheme	67
3.8	Simulation results on the necessity of resampling techniques	71
3.9	8 simulated trajectories for MTT filters test	75
3.10	OSPA error of JA, IP, PP, APP and TRAPP with respect to the number of particles for 1 target	77
3.11	OSPA error of JA, IP, PP, APP and TRAPP with respect to the number of particles for 3 targets	78
3.12	OSPA error of JA, IP, PP, APP and TRAPP with respect to the number of particles for 6 targets	80

3.13	OSPA error of JA, IP, PP, APP and TRAPP with respect to the number of particles for 8 targets	81
3.14	OSPA error of JA, IP, PP, APP and TRAPP with respect to the number of targets for 300 particles	83
3.15	OSPA error of PP, APP and TRAPP with respect to the number of particles and larger measurement noise	84
3.16	OSPA error of JA, IP, PP, APP and TRAPP with respect to the execution time with 6 targets	85
3.17	APP, TRAPP and ATRAPP OSPA error with respect to the number of particles for 2 and 8 targets	87
3.18	RMS error of JA, IP, PP, APP, TRAPP and ATRAPP with respect to the number of particles for 6 targets	88
3.19	Variable target-resampling threshold for ATRAPP with respect to the number of targets	89
3.20	APP, TRAPP and ATRAPP OSPA error with respect to the number of targets	91
3.21	APP, TRAPP and ATRAPP OSPA error with respect to time	92
3.22	OSPA error of PP, APP, TRAPP and ATRAPP with respect to the execution time with 6 targets	93
4.1	SP-MPF particles and sigma-points in a scenario with 3 different targets	110
4.2	OSPA error of PP, APP, TRAPP, ATRAPP, MPF, MAPF, SP-MPF and SP-MAPF with respect to the number of particles for 1 target with additive received power measurements	113
4.3	OSPA error of PP, APP, TRAPP, ATRAPP, MPF, MAPF, SP-MPF and SP-MAPF with respect to the number of particles for 4 targets with additive received power measurements	114
4.4	OSPA error of PP, APP, TRAPP, ATRAPP, MPF, MAPF, SP-MPF and SP-MAPF with respect to the number of particles for 6 targets with additive received power measurements	114
4.5	OSPA error of PP, APP, TRAPP, ATRAPP, MPF, MAPF, SP-MPF and SP-MAPF with respect to the number of particles for 8 targets with additive received power measurements	115

4.6	OSPA error of PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF with respect to the number of particles for 1 target with RSSI measurements	115
4.7	OSPA error of PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF with respect to the number of particles for 4 targets with RSSI measurements	116
4.8	OSPA error of PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF with respect to the number of particles for 6 targets with RSSI measurements	116
4.9	OSPA error of PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF with respect to the number of particles for 8 targets with RSSI measurements	117
4.10	PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF OSPA error with respect to the number of targets	120
4.11	Performance comparison of MPF, C-PF-PROP, SP-MPF and direct marginalization filter.	122
4.12	MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF effective sample size with respect to time	123
4.13	PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF OSPA error with respect to time with 3 targets in the scenario with RSSI measurements	124
4.14	PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF RMS error with respect to the number of particles with 3 and 6 targets using RSSI measurements	126
4.15	PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF execution time with respect to the number of particles for 6 targets with additive received power measurements	127
4.16	PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF OSPA error with respect to the execution time with 6 targets with RSSI measurements	128
4.17	PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF execution time with respect to the number of particles for 6 targets with RSSI measurements	129

4.18	PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF OSPA error with respect to the execution time with 6 targets with RSSI measurements	130
5.1	FAR framework representation	138
5.2	MTT FAR scenario and trajectories	155
5.3	FAR resource distributions for 1 and 2 targets in the scenario	157
5.4	FAR resource distributions for 3 and 4 targets in the scenario	158
5.5	FAR OSPA error performance in a MTT scenario	159
6.1	Simulated single target tracking FAR scenario	163
6.2	FAR large budget case. PC-CRLB minimization	165
6.3	FAR scarce budget case. PC-CRLB minimization	167
6.4	RMS error with respect to the available budget. Robustness issues of FAR for single target tracking	168
6.5	Resource distribution in a FAR scenario for different available budgets for the PC-CRLB minimization method	169
6.6	Single target tracking FAR resulting trajectories	170
6.7	RMS error with respect to the available budget. Solution to the robustness issues of FAR for single target tracking	174
6.8	FAR scarce budget case. Gaussian approximation	176
6.9	Single target tracking FAR resulting trajectories	177
6.10	FAR large budget case. Gaussian approximation	178
6.11	Resource distribution in a FAR scenario for a scarce budgets for the PC-CRLB minimization and the proposed method	179
6.12	Measurement model, sensor EFIM and sensor likelihood	183
6.13	Target trajectory and sensors for the simultaneous target and tracking simulations	191
6.14	Average GOSPA error at each time step with a high resource budget for different controller policies	193
6.15	Average probability of existence at each time step with with a high resource budget for different controller policies	195

6.16	Average resources assigned to sensor $\mathbf{s}_7$ in the scenario in Figure 6.13 at each time step with a high resource budget for different controller policies	196
6.17	Average GOSPA error at each time step with a low resource budget for different controller policies	196
6.18	Average probability of existence at each time step with a low resource budget for different controller policies	198
6.19	Average probability of occurrence of a cardinality error at each time step with a low resource budget for different controller policies	198
6.20	Average resources assigned to sensor $\mathbf{s}_7$ in the scenario in Figure 6.13 at each time step with a low resource budget for different controller policies	199
A.1	Simulated target trajectories of a 2 target scenario	214
A.2	Estimated target trajectories of a 2 target scenario using a PF	216

List of Tables

2.1	Feature comparison between IP and PP algorithms	44
3.1	Feature comparison between IP, PP, APP and TRAPP algorithms . . .	65
3.2	Importance sampling function of the j -th component of the partitioned state for IP, PP, APP and TRAPP algorithms	66
4.1	Principal features of the different multiple particle filtering algorithms	108
4.2	MPF, MAPF, C-PF-PROP, SP-MPF, SP-MAPF OSPA error with respect to the measurement noise variance	119
6.1	Principal features of the different sensor management strategies	180

List of Algorithms

2.1	IP procedure	47
2.2	PP procedure	48
3.1	APP procedure	62
3.2	TRAPP procedure	68
3.3	ATRAPP procedure	73
3.4	target_ATRAPP subroutine for the j -th target	74
4.1	PF for the j -th component in MAPF	99
4.2	PF for the j -th component in SP-MPF	103
4.3	PF for the j -th component in SP-MAPF	106

List of Acronyms

APF	Auxiliary particle filter
APP	Auxiliary parallel partition
ATRAPP	Adaptive target-resampling auxiliary parallel partition
EKF	Extended Kalman filter
FAR	Fully adaptive radar
GOSPA	Generalized optimal sub-pattern assignment
IJOID	Independent joint optimal importance density
IP	Independent partition
JA PF	Jointly auxiliary particle filter
KF	Kalman filter
LMMSE	Linear minimum mean squared error
MAPF	Multiple auxiliary particle filter
MC	Monte Carlo
MMSE	Minimum mean squared error
MPF	Multiple particle filter
MTT	Multiple target tracking
OID	Optimal importance density
OSPA	Optimal sub-pattern assignment
PDF	Probability density function

PF	Particle filter
PMF	Point mass filter
PP	Parallel partition
RMS	Root mean square
RSSI	Received signal strength indicator
SIR	Sequential importance resampling
SIS	Sequential importance sampling
SP-MAPF	Sigma-point multiple auxiliary particle filter
SP-MPF	Sigma-point multiple particle filter
TRAPP	Target-resampling auxiliary parallel partition
UKF	Unscented Kalman filter
UT	Unscented transform

Chapter 1

Introduction

1.1 Robust techniques for Bayesian dynamic state filtering and sensor management

Initially motivated by aerospace and defense applications such as radar [11], sonar [11], navigation [123, 131], air traffic control [105], and the space exploration since the 1960s [66], the fields of dynamic state filtering and sensor management have been topics of great interest during the last decades. In addition, these research areas have paved the way for the flourishing of other disciplines such as meteorology [174], oceanography [174], autonomous vehicles [129], robotics [162], remote sensing [182], biomedical research [141] or space debris tracking [115]. In particular, the problem of sensor management is receiving an increasing attention during the past few years, as it is recognized as an enabling technology for new generations of agile sensors [74].

Many problems of science require the estimation of the state of a system that changes over time using a sequence of available noisy measurements. Thus, the estimation of the state of a dynamic system is a widely studied field, with a vast body of knowledge of related contributions in the literature. In this thesis, we focus on the Bayesian approach to estimation, which requires the availability of at least two models [7]: a dynamic model which describes the evolution of the state with time, and a measurement model which relates the state of the system with the measurements received by sensors. These two models are often available in a probabilistic form, making the Bayesian approach a rigorous and general framework for dynamic state estimation problems, as it is ideally suited for the problem of updating the available information about the system as new measurements are received.

Unfortunately, closed-form optimal solutions with a finite number of parameters to the Bayesian dynamic state filtering problem do not generally exist except for a very

limited class of problems [152], so that one generally has to rely on approximations. The performance of these approximations depends on several factors, with attention being often dedicated primarily to the form of the dynamic and measurement models. Nevertheless, a key factor for performance of these types of methods is the dimension of the space in which the state of the system can take values. If the dimension of the state space is high, as a general rule, the performance of approximated Bayesian filtering techniques severely drops. Common examples of high-dimensional state estimation problems are the tracking of multiple targets in surveillance applications or meteorological estimation in large areas. There is therefore a great interest in the development of robust Bayesian filtering methods which can accurately provide estimates in high-dimensional state spaces. In Part I of this thesis, this problem is tackled, proposing different methods for high-dimensional Bayesian filtering. These methods were conceived with an eye on their application to the problem of multiple target tracking (MTT). However, we wish to emphasize that these methods can be used in a wide range of applications for high-dimensional filtering besides MTT. With this purpose in mind, the proposed methods are first presented and derived in a general setting, and then, their performance is illustrated in MTT scenarios.

The sensor management problem applies to the frequent case in which measurements received by sensors depend on some parameters that can be tuned by the system controller. A common example of this kind of setting is a dynamic system in which the precision of a measurement received by a sensor depends on the amount of resources (e.g., observation time, or energy) allocated to it, while the total budget of resources is shared among all sensors. Other common settings of this type are dynamic systems in which the position of sensors can be altered, or whose measurement model at each time step can be chosen from a catalog of available sensing modes. The posterior PDF and, consequently, the accuracy in the estimation of the state in such problems will generally depend on the selected sensor parameters, so that they need to be carefully chosen. As previously mentioned, this problem has gained interest over the years as more sophisticated and versatile sensors have been made available.

The sensor management problem has been previously tackled in the literature making use of a wide variety of approaches, including, among others, control theory [162], partially observed Markov decision processes [162] or game theory [147]. Some recent contributions to the sensor management problem in the field of versatile radar systems are, however, inspired by the neuroscience approach to decision making and cognition [17–20, 59, 67, 70, 71, 73, 74]. Among these, the formulation of the fully adaptive radar framework [17–19] stands out as it gathers the fundamental concepts of this latter approach to the sensor management problem in a simple and compact architecture.

Different objective-specialized fully adaptive radar implementations in the literature are reviewed in the second part of this thesis, showing that, unfortunately, some robustness issues can show up using these implementations in some difficult scenarios, leading the system to severe misperformance. Part II of this thesis is therefore devoted to the characterization of these robustness issues and the proposal of some robust novel methods within the fully adaptive radar framework which adequately perform in such situations.

1.2 Contributions

In this section, the different contributions of this thesis are briefly summarized. Contributions in Part I of this thesis are related to the problem of robust high-dimensional state filtering, starting with those related to the use of partitioned particle filters:

1. Different methods and techniques to tackle the difficult problem of high-dimensional filtering are reviewed in Chapter 2. In particular, some limitations of the parallel partition (PP) particle filter [50] are highlighted in Section 2.9.1. Following the previous discussion on the drawbacks of the PP particle filter, contributions in this thesis start in Section 3.1 by presenting a novel method, the auxiliary PP (APP) particle filter. The APP method is able to overcome the discussed limitations of the PP filter. In addition, it incorporates the use of auxiliary filtering, which added to the various techniques already employed in the PP method allows for the remarkable performance of the APP particle filter.
2. APP is further improved in high dimensional state spaces by making use of an additional processing stage referred to as a target-resampling step. Considering this technique, an alternative novel method, namely the target-resampling APP (TRAPP) particle filter, is proposed in Section 3.2.
3. The application of the above mentioned target-resampling method is analyzed in Section 3.3. It is shown how this technique does not come without a cost, as it induces a loss of particle diversity. Thus, the use of TRAPP is only preferred to APP in some situations, depending on the problem at hand. With the aim of providing a method that can autonomously decide if the target-resampling stage of TRAPP is advised, Section 3.4 introduces an additional method for high-dimensional filtering, the adaptive TRAPP (ATRAPP). This filter monitors particle diversity by means of the effective sample size at each time step, and

adaptively decides if target-resampling is advisable in some components of the state.

The rest of the contributions related to high-dimensional filtering in Part I of this thesis are within the framework of multiple particle filtering [15, 26, 31, 40, 42, 184, 185]. Different multiple particle filtering methods in the literature are analyzed in Sections 2.10.1 and 2.10.2, highlighting some of their limitations. Several novel methods which are able of overcoming these limitations are proposed in this thesis:

4. First, we consider the use of auxiliary filtering within the multiple particle filtering framework. This approach, coined as the multiple auxiliary PF (MAPF), has already been used by a previous work in the literature [30], but lacked of a detailed derivation of the method, which is provided in this thesis in Section 4.1.
5. Two additional novel methods in the field of multiple particle filtering are respectively presented in Sections 4.2.1 and 4.2.2 of this thesis. These methods make use of a second-order approximation to the marginalization integrals required in this type of filters. This second-order approximation can be efficiently computed using sigma-point integration methods, giving rise to the sigma-point MPF (SP-MPF) and the sigma-point MAPF (SP-MAPF) algorithms, with the latter considering the use of auxiliary filtering. The performance of the presented methods is shown to be remarkable with respect to previous approaches in the literature.

Contributions in Part II of this thesis are related to the field of the fully adaptive radar (FAR) approach to the problem of sensor management:

6. After reviewing the application of the FAR framework to the problems of single target tracking and simultaneous single target detection and tracking, we tackle in Section 5.4 of this thesis the application of the framework to the problem of multiple target tracking with a fixed and known number of targets, illustrating it through an application in a sensor network with resource constraints.
7. The use of the FAR framework for the above described applications can nonetheless suffer from some robustness issues when employed in difficult scenarios. In particular, in Section 6.1, we analyze the robustness problems resulting from the use of previously reviewed FAR solution to the problem of single target tracking, while Section 6.3 points out the difficulties that may arise from the application of the reviewed solution in the literature to the problem of simultaneous single target detection and tracking.

8. The two final contributions of this thesis are related to the above mentioned robustness issues in the application of the FAR framework. A novel method based on an enabling linear-Gaussian approximation is proposed in Section 6.2 which is shown to reliably tackle the problem of single target tracking in the FAR framework.
9. Finally, a Bernoulli-Gaussian approximation is proposed in order to tackle the problem of simultaneous single target detection and tracking within the FAR framework, showing through simulations that it can successfully overcome the previously discussed limitations of solutions to this problem in the literature.

1.3 Thesis organization

Part I, which spans Chapters 2-4, is concerned with the contributions in the field of high-dimensional filtering.

- Chapter 2 reviews the basics of Bayesian filtering of dynamic systems. The general Bayesian filtering problem is first considered, followed by a closed-form solution for linear/Gaussian systems: the celebrated Kalman filter [81]. Then, some widespread approximations to the Bayesian filter based on Kalman filtering and particle filters [7,68] are also covered. In particular, we analyze the difficulties of filtering high-dimensional state spaces [32] and review some useful strategies and particle filter (PF) implementations for this purpose.
- Chapter 3 builds on the particle filters for high-dimensional state spaces reviewed in the final part of Chapter 2 and addresses some of its limitations making use of auxiliary particle filtering. Contributions 1 to 3 are covered in this chapter. Some contents of this chapter have been published in [163] and [164].
- Chapter 4 is devoted to an alternative approach to high-dimensional filtering coined as multiple filtering [26, 40, 42]. Contributions 4 and 5 are covered in this chapter. Some contents of this chapter have been published in [165] and submitted for publication in [167].

Part II is devoted to the contributions in the field of sensor management and fully adaptive radar:

- Chapter 5 introduces the general framework employed in the second part of this thesis, i.e., the fully adaptive radar (FAR) approach to sensor management

[17–19]. The analysis of various applications within this framework is also addressed. Contribution 6 is covered in this chapter. The contents in the final part of this chapter have been published in [169].

- Chapter 6 addresses the robustness analysis of the different FAR implementations presented in Chapter 5 and proposes alternative strategies to overcome some identified robustness issues. Contributions 7 to 9 are covered in this chapter. Some contents of this chapter have been published in [168] and [166].
- Chapter 7 Includes conclusions and future work.

Part I

High-dimensional filtering and multiple target tracking

Chapter 2

Bayesian recursive filtering

In this chapter, the concept of Bayesian recursive filtering of a dynamic system is introduced and some relevant methods in this field are explained. The concept of a dynamic system and its state is introduced in Section 2.1, emphasizing the great utility of this mathematical abstraction, which has allowed to estimate, model and predict the behavior of a vast plethora of physical systems. Section 2.2 poses the notion of state estimation of a dynamic system and proposes a classification of the different problems and approaches that can be considered related to this topic.

Within the classification provided in Section 2.2, the main interest in this thesis is put in the problem of recursive Bayesian filtering of the state of a dynamic system, which is further covered in Section 2.3. Considering some common assumptions, a first approach to the solution of this problem, namely the Kalman filter, is explained in Section 2.4. This method owes its name to its developer, the Hungarian-born American Rudolph E. Kalman [93], and is definitely the most well-known contribution in this field, having been used in a vast number of applied mathematics, engineering and science applications. Section 2.5 explains different extensions of the Kalman filter to nonlinear systems.

Provided that the state of the dynamic system can only take a finite set of discrete values, a closed-form solution to the recursive Bayesian filtering problem can also be obtained. This type of solutions, commonly referred to as grid-based methods are explained in Section 2.6. The extension of this kind of methods to the case in which the state of the dynamic system can take values in a continuous space is also covered, thus introducing the notion of sampling of the state space. This concept is further exploited in Section 2.7, considering the use of importance sampling techniques to the problem, and resulting in the derivation of the so-called particle filter methods, which are a key tool along this thesis. Some important considerations that apply for the use

of particle filters are also covered in Section 2.7.

It is often the case in which a method designed to tackle some problem of interest related to a dynamic system sees a high degradation of its performance if the dimension of the space in which the state takes values is high. This effect, which is particularly severe in filtering techniques, is commonly referred to as the curse of dimensionality and is treated in Section 2.8. Unfortunately, high-dimensional state spaces are quite common so that there is a great interest in the design of robust filtering techniques that can adequately handle the effects of the curse of dimensionality. Sections 2.9 and 2.10 are devoted to describing some techniques that can help to such purpose. Different particle filtering algorithms that implement these techniques and that can be useful in high-dimensional state filtering are also covered in these two sections.

2.1 Dynamic Systems

The field of dynamic systems attempts to describe, model and predict the change or evolution through time of physical systems. This fruitful field is at the root of some highly celebrated branches of science such as estimation theory [13,84,85], optimization [108], robotics [162], chaos theory [160,186], control theory [82,151,170,193], or complex systems [159,170,192], among others. Dynamic systems have been widely used in a plethora of areas of knowledge such as astrodynamics, avionics, navigation, biology, chemistry, meteorology, economics, applied mathematics or physics.

In dynamic systems, all information of interest about the system is gathered in its state. At any given time k , the state of the system is described by its state vector $\mathbf{x}^k$ which can take any particular value in the state space, which has dimension n_x . The evolution of the state through time is often modeled by differential equations (for the continuous time case) or difference equations (for the discrete-time case). In this work, this latter approach is selected. Thus, changes in the system are described by the state-transition equation

$$\mathbf{x}^k = \mathbf{f}^k(\mathbf{x}^{k-1}, \mathbf{w}^{k-1}, \mathbf{u}^{k-1}) \quad (2.1)$$

where $\mathbf{f}^k(\cdot, \cdot, \cdot)$ might be a nonlinear function, $\mathbf{w}^{k-1}$ is the process noise at time $k-1$ and $\mathbf{u}^{k-1}$ is an exogenous input to the system, often referred to as control vector, and which is usually assumed to be known. Vectors $\mathbf{w}^{k-1}$ and $\mathbf{u}^{k-1}$ are here considered to also take values in a space of dimension n_x although other configurations are possible, and quite common in the case of $\mathbf{u}^{k-1}$. The control vector $\mathbf{u}^{k-1}$ is a key element in the field of control theory, and its design and characterization has a deep impact in areas

like astrodynamics, avionics or chemistry. Nonetheless, in this thesis it is assumed that there is no such known input to the system in the form of a control vector, so that the evolution of the state will then be characterized by a simplified version of (2.1)

$$\mathbf{x}^k = \mathbf{f}^k(\mathbf{x}^{k-1}, \mathbf{w}^{k-1}). \quad (2.2)$$

With the aim of gathering some information about the state of the dynamic system at time k , some noisy measurements are collected from sensors and the measurement vector $\mathbf{z}^k$ is built. The vector $\mathbf{z}^k$ takes values in the measurement space, of dimension n_z . The relation between the measurements and the state is mediated through the measurement equation

$$\mathbf{z}^k = \mathbf{h}^k(\mathbf{x}^k, \mathbf{v}^k) \quad (2.3)$$

where $\mathbf{h}^k(\cdot, \cdot)$ might be also a nonlinear function, and $\mathbf{v}^k$ is the measurement noise at time k which also takes values in a space of dimension n_z .

Note that (2.2) and (2.3) describe an unobserved (hidden) Markov process of order one. It is therefore assumed that the state $\mathbf{x}^k$ is *complete* [162], meaning that the dependence of future states of the system on past states and past measurements is mediated through the state $\mathbf{x}^k$. Thus, the system can be graphically represented as shown in Figure 2.1.

2.2 Dynamic state estimation

In this thesis, recursive Bayesian filtering algorithms for estimation of the state of a dynamic system are considered. However, before further discussions are due, we find it necessary to contextualize this type of estimation methods.

2.2.1 Bayesian and non Bayesian estimation

Two distinct approaches are usually taken in estimation problems [13, 85]. The non-Bayesian (or classical) approach considers the quantity of interest as a deterministic, yet unknown, parameter. On the other hand, the Bayesian approach, considers the quantity of interest as a random variable, and aims to estimate its posterior probability density function (PDF), making use of some available prior knowledge. There has been a great historical controversy [114] on whether the Bayesian approach is sound, with the main criticism centered in the definition, suitability and availability of the chosen

Figure 2.1: Schematic representation of the hidden Markov model described by (2.2) and (2.3). The process is Markov and the measurements are conditionally independent given the states.

prior information. Nonetheless, when repeatability of the measurement procedure is not possible, or the amount of measurements is small, the Bayesian approach usually yields much better results, making it advisable.

2.2.2 Batch and recursive estimation

The second distinction among different estimation approaches is concerned with the strategy in processing the received measurements. Note in equation (2.3) that measurements are sequentially made available at each time step.

The batch estimation approach proceeds by first gathering and storing a sequence of received measurements $\mathbf{z}^l, \dots, \mathbf{z}^{l+b-1}$ in a batch $\mathbf{z}^{l:l+b-1}$, where b stands for the batch size. Once the batch of measurements is available, an estimate of the state of the dynamic system at any desired time step is computed processing all the data in the batch.

Recursive estimation, on the other hand, proceeds by sequentially computing at each time step estimates of the state of the dynamic system, using $\mathbf{z}^{1:k}$, the available measurements up to the current time step k . It is a common approach to use recursive estimation along with a Bayesian approach, as the posterior PDF computed at time $k-1$ can then be used as prior knowledge for the estimation of the posterior PDF at time k .

2.2.3 Smoothing, filtering and prediction

Given $\mathbf{z}^{1:k}$, a sequence of measurements $\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^k$ up to the current time k , and a state $\mathbf{x}^l$ of interest which is to be estimated, three different state estimation problems

Figure 2.2: Schematic representation of the smoothing, filtering and prediction problems. The objective is to estimate the system state at time l with available measurements up to time k .

can be posed [142].

If $l > k$, the interest lies in the estimation of a future state and the problem is called a *prediction* problem. The prediction problem is of course of great interest, being useful in areas such as meteorology, chemistry, stock market forecasting or astrodynamics, among others.

If $l = k$, the interest is in estimating the current state of the dynamic system and the problem is referred to as a *filtering* problem. The filtering problem is also of high importance as it serves to approach the common situation of online inferring the present state of a dynamic system given available measurements up to the current time, which for example naturally arises in areas such as navigation or surveillance.

Finally, if $l < k$, the interest is in estimating a past state and the estimation problem is referred to as the *smoothing* problem. The smoothing problem is also of great interest as the state of the dynamic system is not only conditional on measurements up to and including time k , but also on future measurements. Hence, as more information on which to base the estimation is made available when future measurements are gathered, smoothed estimates are usually more accurate than filtered estimates.

In this thesis, the focus is put on the filtering problem, although analyzed and presented methods could be accommodated to fit the smoothing and prediction problems. A schematic view of these three different problems is shown in Figure 2.2.

2.3 Recursive Bayesian filtering

In this section, recursive Bayesian filtering [2, 75, 161] in the context of dynamic systems is reviewed. As previously stated, in the Bayesian setting, the objective of the filtering problem is to approximate the state posterior probability density function $p(\mathbf{x}^k | \mathbf{z}^{1:k})$, which gathers all the information of interest about the state of the system at a given time k . Following the usual steps of Bayesian filtering, the posterior PDF of the state at time k based on the measurements up to time k , can be obtained by recursively applying two steps: prediction and update [7]. The PDF of the state at time 0, $p(\mathbf{x}^0)$, is assumed to be known.

In the prediction step, the prior PDF at time k , which denotes the PDF of the current state given the measurements up to time $k - 1$, is computed via the Chapman-Kolmogorov equation

$$p(\mathbf{x}^k | \mathbf{z}^{1:k-1}) = \int p(\mathbf{x}^k | \mathbf{x}^{k-1}) p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}) d\mathbf{x}^{k-1} \quad (2.4)$$

where the transition PDF, $p(\mathbf{x}^k | \mathbf{x}^{k-1})$, is obtained from (2.2). Once the measurement at time k is available, the update step makes use of Bayes' rule to provide the posterior at time k

$$\begin{aligned} p(\mathbf{x}^k | \mathbf{z}^{1:k}) &= \frac{p(\mathbf{z}^k | \mathbf{x}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1})}{\int p(\mathbf{z}^k | \mathbf{x}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}) d\mathbf{x}^k} \\ &\propto p(\mathbf{z}^k | \mathbf{x}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}) \end{aligned} \quad (2.5)$$

where $\propto$ indicates proportionality and the PDF $p(\mathbf{z}^k | \mathbf{x}^k)$ is the likelihood of the state given the measurement, which is obtained from the measurement equation (2.3). The denominator in (2.5) is a normalization constant, which is needed so that the integral of the posterior sums up to 1

$$\int p(\mathbf{x}^k | \mathbf{z}^{1:k}) d\mathbf{x}^k = 1. \quad (2.6)$$

The optimal solution to the general nonlinear filtering problem was obtained by Kushner [94] and Stratonovich. Unfortunately, the above recursion is not generally tractable, and can only be analytically solved if [152]:

1. The marginalization over $\mathbf{x}^{k-1}$ in (2.4) is analytically tractable
2. The resulting posterior PDF $p(\mathbf{x}^k | \mathbf{z}^{1:k})$ has the same functional form as in the previous step $p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1})$, allowing the procedure to be iterated.

These conditions are only satisfied for a very limited class of models [21,34,35], allowing for the derivation of exact filters in those cases. The dimension of such filters does not grow itself when new measurements arrive, meaning that the problem has *fixed finite dimensional sufficient statistics*. If these conditions are not met, the solution is generally described by an infinite set of parameters [161], making its exact computation intractable, except for some special cases [35]. As a result, some approximations are often used to tackle the general nonlinear filtering problem in order to compute a solution that comprises only a finite set of parameters, and thus can be computed in finite time in a finite-memory machine.

In the remainder of this chapter, some solutions in the literature to the Bayesian recursive filtering problem are presented. First, some Gaussian techniques are considered in Sections 2.4 and 2.5. These types of algorithms assume a fixed functional form of the posterior PDF, that matches a normal distributed random variable, in order to simplify the computation of the above presented Bayes filter recursion. In contrast to these Gaussian techniques, in Section 2.6 some nonparametric Bayes filters are also reviewed that do not need to rely on any particular form of the posterior PDF. Instead, they make use of different discrete decompositions of the state space to approximate the posterior. The advantage of these latter methods lies in their ability to approximate general posterior PDFs, with the property of convergence to the true posterior PDF when the state space is approximated with a high enough number of samples. Therefore, a trade-off between accuracy and computational burden is directly accomplished. These kinds of generally applicable methods broaden the possibilities for the development of algorithms that can take advantage of the characteristics of the problem and the structure of the state space, in order to achieve better filtering approximations to the posterior PDF in a computationally affordable manner. Some of these techniques are considered in Sections 2.7, 2.9 and 2.10. The difficulties arising in the filtering of high-dimensional state spaces are described in Section 2.8. The methods in Sections 2.9 and 2.10 are specifically designed for filtering of high-dimensional state spaces.

2.4 Gaussian filtering with linear models

The PDF of a linearly transformed Gaussian distributed random variable is also Gaussian distributed. In addition, the PDF of a sum of independent Gaussian distributed random variables will be also Gaussian distributed [85] (see Appendix A.1). Using these properties, if the involved dynamic and measurement models in (2.2) and (2.3) are linear transformations of the state with additive independent and Gaussian

distributed noises, and the posterior PDF at time $k - 1$ is Gaussian distributed, then the posterior PDF at time k will also be Gaussian. Thus, the conditions for the tractability of the Bayesian filter aforementioned in the previous section hold, and a recursive closed-form solution to the filtering problem can be provided. This approach to the Bayes filter is commonly known as the Kalman filter (KF) (or Kalman-Bucy for the continuous-time version) [81].

If the above described conditions hold, (2.2) and (2.3) become [7]

$$\mathbf{x}^k = \mathbf{F}^k \mathbf{x}^{k-1} + \mathbf{w}^k \quad (2.7)$$

$$\mathbf{z}^k = \mathbf{H}^k \mathbf{x}^k + \mathbf{v}^k \quad (2.8)$$

where the matrices $\mathbf{F}^k$ and $\mathbf{H}^k$ are known (although they can vary at each time step), the process noise $\mathbf{w}^k$ is Gaussian distributed with zero mean and covariance matrix $\mathbf{Q}^k$, and the measurement noise is Gaussian distributed with zero mean and covariance matrix $\mathbf{R}^k$. The noise vectors are assumed to be independent in each time step, and a known Gaussian prior PDF of the state at time 0 is also assumed

$$p(\mathbf{x}^0) = \mathcal{N}(\mathbf{x}^0; \bar{\mathbf{x}}^0, \Sigma^0) \quad (2.9)$$

where $\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \Sigma)$ stands for a Gaussian PDF with mean $\bar{\mathbf{x}}$ and covariance matrix Σ evaluated at $\mathbf{x}$. Under the above assumptions, the dynamic model and likelihood function can be written as

$$p(\mathbf{x}^k | \mathbf{x}^{k-1}) = \mathcal{N}(\mathbf{x}^k; \mathbf{F}^k \mathbf{x}^{k-1}, \mathbf{Q}^k) \quad (2.10)$$

$$p(\mathbf{z}^k | \mathbf{x}^k) = \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \mathbf{x}^k, \mathbf{R}^k). \quad (2.11)$$

The assumption of the posterior PDF at time $k - 1$ to be Gaussian-distributed, implies that it can be fully described by a finite set of parameters consisting of its mean vector (of dimension n_x) and its symmetric covariance matrix of size $n_x \times n_x$ (whose dimension, or number of different components is $n_x \frac{n_x+1}{2}$). As previously mentioned, under the linear transformation in (2.7) and (2.8), the posterior at time k also takes the same functional form and can also be described by a finite set of parameters.

The Kalman Filter obtains a recursive analytical solution to the system given by (2.9), (2.10) and (2.11) by applying the recursion in (2.4) and (2.5). Assuming the posterior PDF at time $k - 1$ is given by

$$p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}) = \mathcal{N}(\mathbf{x}^{k-1}; \bar{\mathbf{x}}^{k-1}, \Sigma^{k-1}). \quad (2.12)$$

The prediction step yields the prior PDF

$$p(\mathbf{x}^k | \mathbf{z}^{1:k-1}) = \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{x}}^{k|k-1}, \Sigma^{k|k-1}) \quad (2.13)$$

with (see A.1.2)

$$\bar{\mathbf{x}}^{k|k-1} = \mathbf{F}^k \bar{\mathbf{x}}^{k-1} \quad (2.14)$$

$$\Sigma^{k|k-1} = \mathbf{F}^k \Sigma^{k-1} (\mathbf{F}^k)^T + \mathbf{Q}^k. \quad (2.15)$$

Given the above Gaussian prior, and the linear measurement model in (2.8), the PDF of the measurement $\mathbf{z}^k$ will also have a Gaussian PDF (see A.1.2)

$$p(\mathbf{z}^k | \mathbf{z}^{1:k-1}) = \mathcal{N}(\mathbf{z}^k; \bar{\mathbf{z}}^{k|k-1}, \mathbf{S}^k) \quad (2.16)$$

with

$$\bar{\mathbf{z}}^{k|k-1} = \mathbf{H}^k \bar{\mathbf{x}}^{k|k-1} \quad (2.17)$$

$$\mathbf{S}^k = \mathbf{H}^k \Sigma^{k|k-1} (\mathbf{H}^k)^T + \mathbf{R}^k. \quad (2.18)$$

Given that $\mathbf{x}^k$ and $\mathbf{z}^k$ are jointly Gaussian random variables, their cross covariance can also be computed as (see A.1.2)

$$\Psi^k = \Sigma^{k|k-1} (\mathbf{H}^k)^T. \quad (2.19)$$

Once the measurement arrives, the update step can finally perform the conditioning of the joint PDF of $\mathbf{x}^k$ and $\mathbf{z}^k$, over the received measurement $\mathbf{z}^k$ to yield the posterior PDF (see A.1.3.1)

$$p(\mathbf{x}^k | \mathbf{z}^{1:k}) = \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{x}}^k, \Sigma^k) \quad (2.20)$$

where

$$\bar{\mathbf{x}}^k = \bar{\mathbf{x}}^{k|k-1} + \Psi^k (\mathbf{S}^k)^{-1} (\mathbf{z}^k - \bar{\mathbf{z}}^{k|k-1}) \quad (2.21)$$

$$\Sigma^k = \Sigma^{k|k-1} - \Psi^k (\mathbf{S}^k)^{-1} (\Psi^k)^T. \quad (2.22)$$

The above Kalman filter recursion computes the optimum estimation of the state $\mathbf{x}^k$ of the system in the sense of the minimum mean squared error (MMSE) estimator.

2.5 Gaussian filtering with nonlinear models

As previously stated, the Kalman filter can be used to optimally estimate the mean and covariance of the posterior PDF when the assumptions of linearity and Gaussian distributed random variables are met. However, Kalman's original work [81] was not limited to the linear-Gaussian case and it can still be used to approximate the first two moments of the posterior PDF [149] under nonlinear or non-Gaussian models. In fact, when $\mathbf{w}^k$ and $\mathbf{v}^k$ are non-Gaussian, the Kalman filter recursion is still the optimal linear minimum mean squared error (LMMSE) estimator [85], while some modifications of the filter can be used if $\mathbf{w}^k$ and $\mathbf{v}^k$ are correlated or not white noises. In addition, some approximations can be made, which allow for the utilization of the Kalman filter recursion when dealing with nonlinear measurement or dynamic models.

There exist several such approximations to the Kalman filter recursion, among these, the extended Kalman filter and sigma-point Kalman filters [4–6, 77] are reviewed in this section because of their widespread use.

2.5.1 Extended Kalman filtering

When the dynamics or the measurement equations of the dynamic system are nonlinear or non Gaussian, a closed-form solution to the recursion described by equations (2.4) and (2.5) cannot be generally derived. The extended Kalman filter (EKF) [7, 34, 149] is one way of being able to use the KF recursion in Section 2.4 in nonlinear problems. This is achieved by performing a local linearization of the involved nonlinear functions in (2.2) and (2.3) using a first-order Taylor series expansion.

The EKF assumes the posterior at each time step to be approximately Gaussian distributed,

$$p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}) \approx \mathcal{N}(\mathbf{x}^{k-1}; \bar{\mathbf{x}}^{k-1}, \Sigma^{k-1}) \quad (2.23)$$

and as with the Kalman filter, its sufficient statistics (mean vector and covariance matrix) are propagated to the next time step using the dynamic model in (2.2). Consider a first-order Taylor series expansion of $\mathbf{f}^k(\mathbf{x}^{k-1})$ around $\bar{\mathbf{x}}^{k-1}$ to approximate the dynamic function with a linear model, i.e.,

$$\mathbf{f}^k(\mathbf{x}^{k-1}) \approx \mathbf{f}^k(\bar{\mathbf{x}}^{k-1}) + \hat{\mathbf{F}}^k \cdot (\mathbf{x}^{k-1} - \bar{\mathbf{x}}^{k-1}) \quad (2.24)$$

where $\hat{\mathbf{F}}$ is the Jacobian of $\mathbf{f}^k(\mathbf{x}^{k-1})$ evaluated at $\bar{\mathbf{x}}^{k-1}$, which is given by

$$\hat{\mathbf{F}}^k = \left[\nabla_{\mathbf{x}^{k-1}} \left(\mathbf{f}^k(\mathbf{x}^{k-1})^T \right)^T \right] \bigg|_{\mathbf{x}^{k-1} = \bar{\mathbf{x}}^{k-1}}. \quad (2.25)$$

Then, the predicted mean and covariance matrix are approximated as

$$\bar{\mathbf{x}}^{k|k-1} \approx \mathbf{f}^k(\bar{\mathbf{x}}^{k-1}) \quad (2.26)$$

$$\Sigma^{k|k-1} \approx \hat{\mathbf{F}}^k \Sigma^{k-1} (\hat{\mathbf{F}}^k)^T + \mathbf{Q}^k. \quad (2.27)$$

Note that this implies that the Jacobian matrix must be calculated. Using the same approach, the measurement model is approximated by a linear model by using a first-order Taylor series expansion around $\bar{\mathbf{x}}^{k|k-1}$ as

$$\mathbf{h}^k(\mathbf{x}^k) \approx \mathbf{h}^k(\bar{\mathbf{x}}^{k|k-1}) + \hat{\mathbf{H}}^k \cdot (\mathbf{x}^k - \bar{\mathbf{x}}^{k|k-1}) \quad (2.28)$$

with

$$\hat{\mathbf{H}}^k = \left[\nabla_{\mathbf{x}^k} \left(\mathbf{h}^k(\mathbf{x}^k)^T \right)^T \right] \Big|_{\mathbf{x}^k = \bar{\mathbf{x}}^{k|k-1}} \quad (2.29)$$

so that the measurement mean and the measurement covariance and cross-covariance matrices can be computed as

$$\bar{\mathbf{z}}^{k|k-1} \approx \mathbf{h}^k(\bar{\mathbf{x}}^{k|k-1}) \quad (2.30)$$

$$\mathbf{S}^k \approx \hat{\mathbf{H}}^k \Sigma^{k|k-1} (\hat{\mathbf{H}}^k)^T + \mathbf{R}^k \quad (2.31)$$

$$\Psi^k \approx \Sigma^{k|k-1} (\hat{\mathbf{H}}^k)^T, \quad (2.32)$$

noting that the Jacobian matrix for the measurement model must also be computed.

Once the new measurement is available, the update procedure finally computes the mean vector and covariance matrix of the posterior at time k by means of conditioning with respect to the received measurement $\mathbf{z}^k$ (see A.1.3.1) as

$$\bar{\mathbf{x}}^k \approx \bar{\mathbf{x}}^{k|k-1} + \Psi^k (\mathbf{S}^k)^{-1} (\mathbf{z}^k - \bar{\mathbf{z}}^{k|k-1}) \quad (2.33)$$

$$\Sigma^k \approx \Sigma^{k|k-1} - \Psi^k (\mathbf{S}^k)^{-1} (\Psi^k)^T. \quad (2.34)$$

If the nonlinearities of the system are mild, the EKF can perform sufficiently well, however if those nonlinearities are more severe, its performance rapidly drops down, as the first-order linearization cannot adequately approximate the nonlinearity. With the aim of tackling this problem, different versions of the EKF have been proposed that also take into account further terms of the Taylor series expansion in the linearization procedure [137]. However, as higher-order terms are taken into account, the EKF computational complexity drastically rises.

Note that the linearization process assumes that the implied Jacobian matrices (and other higher-order computations in the case of considering further terms of the Taylor

series expansion) actually exist, which is not always true, as the state transition or measurement functions might not be differentiable and can even contain discontinuities. Alternative formulations of the KF recursion, as the sigma-point Kalman filter methods can deal with these situations.

2.5.2 Sigma-point Kalman filtering

In this section, sigma-point Kalman filtering is introduced. First, sigma-point integration methods are briefly discussed in Section 2.5.2.1, and their use in the KF recursion are illustrated in the unscented KF method (UKF), described in Section 2.5.2.2.

2.5.2.1 Sigma-point integration methods

In this section we briefly review the use of sigma-point methods [79, 142], which are numerical integration tools that are useful for computing the moments of a distribution that undergoes a nonlinear transformation. In this context it is worth highlighting that, in contrast to the previously described EKF, which based on a Taylor series expansion attempts to linearize the functions involved in the problem, sigma-point methods directly attempt to approximate the moments of the nonlinearly transformed distribution.

Considering a random variable $\mathbf{x}$ with Gaussian PDF with mean $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{P}$

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{P}) \quad (2.35)$$

which is transformed through a function $\mathbf{z} = \mathbf{h}(\mathbf{x})$. The first two moments of $\mathbf{z}$ can be obtained as

$$\bar{\mathbf{z}} = \int \mathbf{h}(\mathbf{x}) \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{P}) d\mathbf{x} \quad (2.36)$$

$$\mathbf{S} = \int (\mathbf{h}(\mathbf{x}) - \bar{\mathbf{z}})(\mathbf{h}(\mathbf{x}) - \bar{\mathbf{z}})^T \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{P}) d\mathbf{x}. \quad (2.37)$$

Sigma-point methods are an efficient way to approximate (2.36) and (2.37). To do so, this methods consider the use of a set of m sigma-points with their corresponding weights, $\mathcal{X} = \{(\mathcal{X}^1, w^1), \dots, (\mathcal{X}^m, w^m)\}$ to compute the approximation

$$\bar{\mathbf{z}} \approx \sum_{i=1}^m w^i \mathbf{h}(\mathcal{X}^i) \quad (2.38)$$

$$\mathbf{S} \approx \sum_{i=1}^m w^i (\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}}) \cdot (\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}})^T \quad (2.39)$$

where the weighted sigma-points $\mathcal{X}$ are selected so that their first two moments match the mean $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{P}$.

Different approaches exist in order to cover more general transformations [142]. Relevant to this thesis is the case where the variable of interest $\mathbf{z}$ consists of an additive transformation $\mathbf{z} = \mathbf{h}(\mathbf{x}) + \mathbf{q}$, with $\mathbf{q} \sim \mathcal{N}(\mathbf{q}; \mathbf{0}, \mathbf{Q})$. In such case, the approximation to the joint PDF of $\mathbf{x}$ and $\mathbf{z}$ is given by

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}; \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{z}} \end{bmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{S} \end{pmatrix} \right) \quad (2.40)$$

where

$$\bar{\mathbf{z}} \approx \sum_{i=1}^m w^i \mathbf{h}(\mathcal{X}^i) \quad (2.41)$$

$$\mathbf{S} \approx \mathbf{Q} + \sum_{i=1}^m w^i (\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}}) \cdot (\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}})^T \quad (2.42)$$

$$\mathbf{C} \approx \sum_{i=1}^m w^i (\mathcal{X}^i - \bar{\mathbf{x}}) \cdot (\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}})^T. \quad (2.43)$$

There are many ways of selecting sigma-points and their corresponding weights, and the choice of the most appropriate sigma-point method thus depends on the specifics of the problem, with important factors being the form of the nonlinear function $\mathbf{h}(\mathbf{x})$, or the support and form of the PDF $p(\mathbf{x})$. Some commonly used methods for selection of sigma-points are the unscented transform [79], the Gauss-Hermite quadrature rule [5], or the cubature rule [4] among others [77, 95, 96, 142]. The use of each of these methods gives rise to a different Kalman filter method for nonlinear filtering, namely the unscented Kalman filter [79, 80, 116], the Gauss-Hermite Kalman filter [5], or the cubature Kalman filter [4]. In the next section, the unscented Kalman filter is reviewed in detail.

2.5.2.2 Unscented Kalman filtering

The unscented Kalman filter (UKF) was originally proposed by Julier and Uhlman [79, 80, 116] based on the idea on which the unscented transform (UT) is founded: *it is easier to approximate a probability distribution than it is to approximate an*

arbitrary nonlinear function or transformation. The UT therefore approximates in a computationally efficient way the moments of a distribution that has undergone a nonlinear transformation.

The UKF assumes that the posterior PDF at each time step to be approximately Gaussian distributed

$$p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}) \approx N(\mathbf{x}^{k-1}; \bar{\mathbf{x}}^{k-1}, \Sigma^{k-1}). \quad (2.44)$$

The selection of weights and sigma-points is based on the UT so that $m = 2n_x + 1$ sigma-points $(\mathcal{X}_0^{k-1}, \dots, \mathcal{X}_{2n_x}^{k-1})$ are drawn with associated weights $(w_0^{k-1}, \dots, w_{2n_x}^{k-1})$ so that for $i = 1, \dots, n_x$ [80]

$$\mathcal{X}_0^{k-1} = \bar{\mathbf{x}}^{k-1} \quad (2.45)$$

$$\mathcal{X}_i^{k-1} = \bar{\mathbf{x}}^{k-1} + \left(\sqrt{\frac{n_x}{1-w_0}} \Sigma^{k-1} \right)_i \quad (2.46)$$

$$w_i = \frac{1-w_0}{2n_x} \quad (2.47)$$

$$\mathcal{X}_{i+n_x}^{k-1} = \bar{\mathbf{x}}^{k-1} - \left(\sqrt{\frac{n_x}{1-w_0}} \Sigma^{k-1} \right)_i \quad (2.48)$$

$$w_{i+n_x} = \frac{1-w_0}{2n_x} \quad (2.49)$$

where n_x is the dimension of the state space, w_0 is a parameter of the UKF that represents the weight of the sigma-point associated to the mean of the prior distribution, and $\left(\sqrt{\mathbf{P}} \right)_i$ stands for the i -th row or column of the matrix square root of $\mathbf{P}$. If $\mathbf{A} = \sqrt{\mathbf{P}}$ is of the form $\mathbf{A}\mathbf{A}^T = \mathbf{P}$ or $\mathbf{A}^T\mathbf{A} = \mathbf{P}$, the sigma-points are formed respectively from the rows or columns of $\mathbf{A}$. In this case, the square root matrix $\mathbf{A}$ can be efficiently computed via the Cholesky decomposition [61] of $\mathbf{P}$. Note that this selection of sigma-points effectively matches the first two moments of $p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1})$, $\bar{\mathbf{x}}^{k-1}$ and Σ^{k-1} .

In the prediction step, the sigma-points are propagated through the nonlinear state transition function, and based on the UT, the statistics of the transformed distribution are computed

$$\mathcal{X}_i'^{k|k-1} = \mathbf{f}^k(\mathcal{X}_i'^{k-1}) \quad (2.50)$$

$$\bar{\mathbf{x}}^{k|k-1} = \sum_{i=0}^{2n_x} w_i \mathcal{X}_i'^{k|k-1} \quad (2.51)$$

$$\Sigma^{k|k-1} = \sum_{i=0}^{2n_x} w_i \left(\mathcal{X}_i'^{k|k-1} - \bar{\mathbf{x}}^{k|k-1} \right) \left(\mathcal{X}_i'^{k|k-1} - \bar{\mathbf{x}}^{k|k-1} \right)^T + \mathbf{Q}^k. \quad (2.52)$$

As the process noise has not been previously taken into account, the sigma-points have to be redrawn before the UT over the measurement function is considered, so that the new sigma-points can match the first two moments of the predicted density, $\bar{\mathbf{x}}^{k+1|k}$ and $\Sigma^{k+1|k}$ as [69, 187]

$$\mathcal{X}_0^{k|k-1} = \bar{\mathbf{x}}^{k|k-1} \quad (2.53)$$

$$\mathcal{X}_i^{k|k-1} = \bar{\mathbf{x}}^{k|k-1} + \left(\sqrt{\frac{n_x}{1-w_0} \Sigma^{k|k-1}} \right)_i \quad (2.54)$$

$$\mathcal{X}_{i+N_x}^{k|k-1} = \bar{\mathbf{x}}^{k|k-1} - \left(\sqrt{\frac{n_x}{1-w_0} \Sigma^{k|k-1}} \right)_i. \quad (2.55)$$

The new sigma-points are this time propagated through the nonlinear measurement function, and based on the UT, the statistics of the transformed posterior distribution can be approximated to be included in the KF mean and covariance computation in (2.21) and (2.22)

$$\mathcal{Z}_i^{k|k-1} = \mathbf{h}^k \left(\mathcal{X}_i^{k|k-1} \right) \quad (2.56)$$

$$\bar{\mathbf{z}}^{k|k-1} = \sum_{i=0}^{2n_x} w_i \mathcal{Z}_i^{k|k-1} \quad (2.57)$$

$$\mathbf{S}^k = \sum_{i=0}^{2n_x} w_i \left(\mathcal{Z}_i^{k|k-1} - \bar{\mathbf{z}}^{k|k-1} \right) \left(\mathcal{Z}_i^{k|k-1} - \bar{\mathbf{z}}^{k|k-1} \right)^T + \mathbf{R}^k \quad (2.58)$$

$$\Psi^k = \sum_{i=0}^{2n_x} w_i \left(\mathcal{X}_i^{k|k-1} - \bar{\mathbf{x}}^{k|k-1} \right) \left(\mathcal{Z}_i^{k|k-1} - \bar{\mathbf{z}}^{k|k-1} \right)^T. \quad (2.59)$$

The UKF commonly outperforms the EKF [80], in addition, it does not require the computation of the Jacobian matrices of the involved functions and can be used even when those functions are not differentiable or contain some discontinuities.

2.5.3 Discussion on Kalman filtering performance in nonlinear systems

The accuracy of the above described methods to approximate the KF recursion under nonlinear dynamic or measurement functions depends on different factors that should be taken into account. First, the Gaussian approximation in (2.23) and (2.44) needs to hold. If the the posterior at the previous time step is unimodal, this approximation can be reasonable. However, if dealing with multimodal densities, the Gaussian approximation will not be accurate and other filtering methods would be required to adequately tackle the state estimation.

On the other hand, for the EKF and the sigma-point KFs to achieve a good estimation performance, the potential nonlinearities of the dynamic model are generally required to be quite mild in the region covered by the posterior at the previous time step. The same principle applies for the nonlinearities of the measurement model with respect to the prior PDF. In addition it should be noted that the accuracy of these KF variants deteriorates as the variance of the process noise and measurement noise decreases, with this deterioration being much slower for the UKF than for the EKF as shown in [119]. In such cases where the noise variances are low, alternatives such as the truncated UKF should be considered [52].

2.6 Grid-based filtering methods

When dealing with a discrete state space formed by a finite number, N^s , of possible states, i.e., $\mathbf{x}^{k-1} \in \{\mathbf{x}_1^{k-1}, \dots, \mathbf{x}_{N^s}^{k-1}\}$, the conditions for tractability of the nonlinear Bayesian filtering problem exposed in Section 2.3 hold and the posterior PDF can be optimally computed with a finite set of parameters. These kinds of filters are often referred to as grid-based methods [7, 68].

For each possible state $\mathbf{x}_i^{k-1}$ at time $k-1$, the conditional probability of the state given the measurements up to time $k-1$ is

$$p(\mathbf{x}^{k-1} = \mathbf{x}_i^{k-1} | \mathbf{z}^{1:k-1}) \triangleq \omega_i^{k-1}. \quad (2.60)$$

The posterior PDF at time $k-1$ can therefore be written a

$$p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}) = \sum_{i=1}^{N^s} \omega_i^{k-1} \delta(\mathbf{x}^{k-1} - \mathbf{x}_i^{k-1}) \quad (2.61)$$

where $\delta(\cdot)$ stands for the Dirac delta function. Assuming that the transition probabilities $p(\mathbf{x}_i^k | \mathbf{x}_j^{k-1})$ are known, the prediction step in (2.4) yields

$$p(\mathbf{x}^k | \mathbf{z}^{1:k-1}) = \sum_{i=1}^{N^s} \omega_i^{k|k-1} \delta(\mathbf{x}^k - \mathbf{x}_i^k) \quad (2.62)$$

with

$$\omega_i^{k|k-1} \triangleq \sum_{j=0}^{N^s} \omega_j^{k-1} p(\mathbf{x}_i^k | \mathbf{x}_j^{k-1}). \quad (2.63)$$

The update procedure makes use of (2.5) to form the posterior at time k

$$p(\mathbf{x}^k | \mathbf{z}^{1:k}) = \sum_{i=1}^{N^s} \omega_i^k \delta(\mathbf{x}^k - \mathbf{x}_i^k) \quad (2.64)$$

$$\omega_i^k \triangleq \frac{\omega_i^{k|k-1} p(\mathbf{z}^k | \mathbf{x}_i^k)}{\sum_{j=1}^{N^s} \omega_j^{k|k-1} p(\mathbf{z}^k | \mathbf{x}_j^k)}. \quad (2.65)$$

In contrast with the EKF and UKF, the presented grid-based filter is not constrained to any particular functional form of the dynamic and measurement equations as long as there is a finite number of possible states. In addition, it makes use of the *true* nonlinear functions in equations (2.2) and (2.3) to optimally compute the posterior.

2.6.1 Approximate grid-based filtering methods

The assumption of a discrete state space with a finite number of possible states does not usually hold for a great variety of problems of interest, so optimal solutions cannot be computed for them based on these methods. Nevertheless, grid-based approximations as the point mass filter (PMF) can be of use. A PMF [1, 7, 68, 90] makes use of SMC techniques to sample the state space in order to approximate the above grid-based recursion.

The PMF does not need to restrict to any specific functional form of the posterior, and furthermore the selected set of points can be displayed over the state space in any (deterministic) layout of interest. However, in most conventional problems, points are selected to form a regular grid over the state space, so the accuracy of the approximation to the posterior is limited by the grid resolution. In fact, grid resolution is the main design parameter of the PMF, and can be adaptively tuned.

The bottleneck of grid-based approximations like the PMF relies on the inefficiency of representing a high-dimensional state space with a deterministic grid, rapidly giving rise to the curse of dimensionality (see Section 2.8). If the state space is not finite in extent, then using a grid-based approach also requires some truncation of the state space. In addition, the complexity of the weight computation procedure of a grid-based method using (2.63) and (2.65) is $\mathcal{O}((N^s)^2)$, so the application of this technique can be unaffordable because of its computational burden when high accuracy (with a high number of grid points) is desired in the approximation to the posterior.

2.7 Particle filters

Particle filters (PFs) implement the Bayes filter in Section 2.3 based on a discrete approximation to the posterior PDF using, as for the PMF, sequential Monte Carlo techniques [7, 28, 41, 68, 136, 162]. However, both approaches differ in the way the samples, called particles, are generated and in the way they populate the state space. While the PMF samples the state space at a deterministic (and usually regular) grid, PFs try to represent the posterior PDF by sampling the state space in a more efficient way, making use of importance sampling [43]. Particles are therefore drawn according to a proposed importance density, that is intended to imitate the true posterior PDF at every time step, this way sampling the state space with a stochastic grid.

PFs are a nonparametric technique, meaning that the posterior PDF is not assumed to be of any particular functional form, so they can approximate a wider space of distributions than, for example, Gaussian PDFs can. As a result, any nonlinear transformation over the involved densities can be considered and the resulting distribution can be approximated using a high enough number of particles in an accurate way. Figure 2.3 illustrates the estimation of a nonlinearly transformed density via a particle approximation.

The computation of the approximation to the posterior PDF for a given precision using a PF has a very low computational burden compared to that of the PMF, as the regions with negligible probability are not assigned a high number of particles. However, for the same reason, the PMF can generally estimate with a higher accuracy long tailed densities, as all the regions of the support of the posterior PDF are equally considered.

2.7.1 Sequential importance sampling

The most basic PF, the sequential importance sampling (SIS) [43, 136] filter, is a sequential Monte Carlo (MC) method which recursively approximates the posterior PDF at every time step [106] by making use of importance sampling.

Importance sampling is an MC variance reduction technique that is based on the idea that some values of the random value under study in the MC simulation are more influent than others when estimating a certain parameter. Thus, if for a given sample size these *important* values are emphasized by sampling them more frequently, then the variance of the estimator can be reduced.

The most determinant design aspect when using an importance sampling technique lies on the selection of an adequate function, or importance sampling density, to sample

Figure 2.3: Particle approximation of a nonlinearly transformed density. A set of particles is drawn from the proposed density $p(x)$ in the lower right part of the figure. These samples are transformed by the nonlinear function $y = f(x)$. As a result, the outcoming samples are distributed according to $p(y)$.

from. However, the use of a biased distribution to enhance the sampling of higher *important* values of the distribution results in a biased estimator. In order to correct this bias, a weighting of the output samples is performed according to the principle of importance sampling before the estimation of the parameter of interest.

The SIS PF indirectly approximates $p(\mathbf{x}^{k-1}|\mathbf{z}^{1:k-1})$, by considering the propagation of $p(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1})$, the posterior PDF of the complete sequence of states from time 0 to $k-1$, $\mathbf{x}^{0:k-1} = [(\mathbf{x}^0)^T, \dots, (\mathbf{x}^{k-1})^T]^T$. The rationale behind this procedure is to avoid the computation of the Chapman-Kolmogorov integral in (2.4), which would increase the computational cost of the filter as explained in Section 2.7.3.

Thus, if an importance density $q(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1})$ is considered to draw a set of N particles (samples), $\mathbf{x}_i^{0:k-1}$ with $i = 1, \dots, N$, such that

$$p(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1}) > 0 \rightarrow q(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1}) > 0, \quad (2.66)$$

a particle approximation to the posterior of the state trajectory can be built as

$$p(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1}) \simeq \sum_{i=1}^N \omega_i^{k-1} \delta(\mathbf{x}^{0:k-1} - \mathbf{x}_i^{0:k-1}) \quad (2.67)$$

where, according to the principle of importance sampling [7]

$$\omega_i^{k-1} \propto \frac{p(\mathbf{x}_i^{0:k-1}|\mathbf{z}^{1:k-1})}{q(\mathbf{x}_i^{0:k-1}|\mathbf{z}^{1:k-1})}. \quad (2.68)$$

Particle weights are required to sum up to 1, so that both the posterior PDF $p(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1})$ and the importance density function $q(\mathbf{x}_i^{0:k-1}|\mathbf{z}^{1:k-1})$ are only required to be known up to a normalizing constant, as weights are normalized using

$$\sum_{i=1}^N \omega_i^{k-1} = 1. \quad (2.69)$$

If the importance sampling density is chosen so that it factors as

$$q(\mathbf{x}^{0:k}|\mathbf{z}^{1:k}) = q(\mathbf{x}^k|\mathbf{x}^{0:k-1}, \mathbf{z}^{1:k}) q(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1}), \quad (2.70)$$

particles can then be obtained recursively, as a sample $\mathbf{x}_i^{0:k} \sim q(\mathbf{x}^{0:k}|\mathbf{z}^{1:k})$ of the whole trajectory can be formed by augmenting the already available samples $\mathbf{x}_i^{0:k-1} \sim q(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1})$ with $\mathbf{x}_i^k \sim q(\mathbf{x}^k|\mathbf{x}_i^{0:k-1}, \mathbf{z}^{1:k})$. Following Bayes' rule and applying the Markov property in the prediction equation

$$p(\mathbf{x}^{0:k}|\mathbf{z}^{1:k}) = \frac{p(\mathbf{z}^k|\mathbf{x}^k) p(\mathbf{x}^{0:k}|\mathbf{z}^{1:k-1})}{p(\mathbf{z}^k|\mathbf{z}^{1:k-1})} \quad (2.71)$$

$$\begin{aligned} &= \frac{p(\mathbf{z}^k|\mathbf{x}^k) p(\mathbf{x}^k|\mathbf{x}^{0:k-1}, \mathbf{z}^{1:k-1}) p(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1})}{p(\mathbf{z}^k|\mathbf{z}^{1:k-1})}, \\ &\propto p(\mathbf{z}^k|\mathbf{x}^k) p(\mathbf{x}^k|\mathbf{x}^{k-1}) p(\mathbf{x}^{0:k-1}|\mathbf{z}^{1:k-1}). \end{aligned} \quad (2.72)$$

The update of the weights to achieve an unbiased estimation at time k must now be considered. Making use of (2.68), (2.70) and (2.72) the updated weights can finally be computed as

$$\begin{aligned} \omega_i^k &\propto \frac{p(\mathbf{x}_i^{0:k}|\mathbf{z}^{1:k})}{q(\mathbf{x}_i^{0:k}|\mathbf{z}^{1:k})} \\ &\propto \frac{p(\mathbf{z}^k|\mathbf{x}_i^k) p(\mathbf{x}_i^k|\mathbf{x}_i^{k-1}) p(\mathbf{x}_i^{0:k-1}|\mathbf{z}^{1:k-1})}{q(\mathbf{x}_i^k|\mathbf{x}_i^{0:k-1}, \mathbf{z}^{1:k}) q(\mathbf{x}_i^{0:k-1}|\mathbf{z}^{1:k-1})} \end{aligned}$$

$$\propto \frac{p(\mathbf{z}^k | \mathbf{x}_i^k) p(\mathbf{x}_i^k | \mathbf{x}_i^{k-1})}{q(\mathbf{x}_i^k | \mathbf{x}_i^{0:k-1}, \mathbf{z}^{1:k})} \omega_i^{k-1}. \quad (2.73)$$

The approximation to the posterior PDF of the state trajectory can therefore be expressed as

$$p(\mathbf{x}^{0:k} | \mathbf{z}^{1:k}) \simeq \sum_{i=1}^N \omega_i^k \delta(\mathbf{x}^{0:k} - \mathbf{x}_i^{0:k}). \quad (2.74)$$

The SIS PF sequentially approximates the posterior PDF of the whole trajectory of the state, $p(\mathbf{x}^{0:k} | \mathbf{z}^{1:k})$. An approximation to the posterior PDF of the state at the current time step, $p(\mathbf{x}^k | \mathbf{z}^{1:k})$, can be easily obtained by marginalization, integrating out the states at previous time instants as

$$\begin{aligned} p(\mathbf{x}^k | \mathbf{z}^{1:k}) &= \int p(\mathbf{x}^{0:k} | \mathbf{z}^{1:k}) d\mathbf{x}^{0:k-1} \\ &\simeq \int \sum_{i=1}^N \omega_i^k \delta(\mathbf{x}^{0:k} - \mathbf{x}_i^{0:k}) d\mathbf{x}^{0:k-1} \\ &\simeq \sum_{i=1}^N \omega_i^k \delta(\mathbf{x}^k - \mathbf{x}_i^k). \end{aligned} \quad (2.75)$$

In addition, if the importance sampling function is chosen to satisfy

$$q(\mathbf{x}^k | \mathbf{x}^{0:k-1}, \mathbf{z}^{1:k}) = q(\mathbf{x}^k | \mathbf{x}^{k-1}, \mathbf{z}^k) \quad (2.76)$$

the complete particle trajectories can be discarded and the only information that needs to be stored is the current state $\mathbf{x}_i^k$ and weight ω_i^k of all the particles. The resulting algorithm complexity is linear with the number of particles, $\mathcal{O}(N)$.

2.7.1.1 The particle degeneracy problem

It has been shown [44, 106] that an importance density with the form in (2.70) can only increase the variance of the importance weights over time. This means that, for a finite number of particles, the performance of the SIS algorithm decays for long time estimations. Only a small proportion of the weights will contain almost all the probability mass while most weights will end up being negligible, implying that most particles will almost not contribute to the state estimation.

Particle degeneracy is a consequence of sampling on a very high-dimensional state-space (see Section 2.8), i.e., the complete trajectory of the state $\mathbf{x}^{0:k}$, which grows larger with every time step. A reasonable indicator of the degeneracy is given by the effective sample size N_{eff} [88]. One cannot generally exactly evaluate N_{eff} , however a

commonly accepted estimate is given by [7]

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (\omega_i^k)^2}, \quad (2.77)$$

although other possible estimates can also be considered [110, 112].

The effective sample size can take values between 1 and N , with these extremes taking place in the following cases:

- If $\omega_i^k = \frac{1}{N}$ for $i = 1, \dots, N$, then

$$\begin{aligned} \hat{N}_{eff} &= \frac{1}{\sum_{i=1}^N \left(\frac{1}{N}\right)^2} \\ &= N \end{aligned}$$

- If $\omega_i^k = 0$ for $i = 1, \dots, j-1, j+1, \dots, N$ and $\omega_j^k = 1$, then

$$\begin{aligned} \hat{N}_{eff} &= \frac{1}{\sum_{i=1}^N (\omega_i^k)^2} \\ &= 1. \end{aligned}$$

Small N_{eff} , therefore, indicates severe degeneracy. In these cases, one must rely on some strategy as the resampling of high weighted samples to overcome the degeneracy of particles in SIS.

2.7.2 Sampling importance resampling

The sampling importance resampling (SIR) PF [63], also called bootstrap filter, managed to overcome the aforementioned degeneracy problem that prevented the applicability of particle filters to most real engineering problems, becoming one of the most widespread methods for nonlinear recursive state estimation. The SIR PF takes the prior PDF as the importance sampling density

$$q(\mathbf{x}^k | \mathbf{x}^{0:k-1}, \mathbf{z}^{1:k}) = p(\mathbf{x}^k | \mathbf{x}^{k-1}). \quad (2.78)$$

In addition, the SIR PF prevents the degeneracy problem by performing a resampling [98] of the particles at every time step. This means that the particles with low importance weights are eliminated and particles with high importance weights are replicated to avoid the increase in the weight variance. After the resampling is performed the representation of the posterior at time k is no longer given by the

set of weighted particles $\{(\mathbf{x}_i^k, \omega_i^k)\}$ for $i = 1, \dots, N$ and an alternative set of particles with uniform weights $\{(\mathbf{x}_i^{*k}, 1/N)\}$ is used to represent the desired posterior, where each particle, $\mathbf{x}_i^{*k}$, is chosen from the original set of particles $\mathbf{x}_i^k$, with probability ω_i^k . There are quite a few ways of implementing the resampling step, however, systematic resampling is the most commonly used due to its low computational burden [136].

Under the above stated conditions, the weight update equation in (2.73) becomes

$$\begin{aligned}
 \omega_i^k &\propto \frac{p(\mathbf{x}_i^{0:k} | \mathbf{z}^{1:k})}{q(\mathbf{x}_i^{0:k} | \mathbf{z}^{1:k})} \\
 &= \frac{p(\mathbf{z}^k | \mathbf{x}_i^k) p(\mathbf{x}_i^k | \mathbf{x}_i^{k-1})}{q(\mathbf{x}_i^k | \mathbf{x}_i^{0:k-1}, \mathbf{z}^{1:k})} \omega_i^{k-1} \\
 &= \frac{p(\mathbf{z}^k | \mathbf{x}_i^k) p(\mathbf{x}_i^k | \mathbf{x}_i^{k-1})}{p(\mathbf{x}_i^k | \mathbf{x}_i^{k-1})} \frac{1}{N} \\
 &\propto p(\mathbf{z}^k | \mathbf{x}_i^k).
 \end{aligned} \tag{2.79}$$

The SIR PF was the first practical particle-based filter approach. However, resampling implies the loss of diversity among particles, which generally degrades the performance of the filter. In addition, when narrow likelihoods are considered, only a small number of particles are not assigned negligible weights, resulting in only a few particles being resampled after every iteration, so the loss of particle diversity in these cases severely affects the performance of the filter. Latter approaches as the auxiliary particle filter (see Section 2.7.5) avoid the use of resampling, generally resulting in better approximations to the posterior PDF. For the sake of completeness, the steps of the SIR algorithm, whose computational complexity is also $\mathcal{O}(N)$, are illustrated in Figures 2.4 and 2.5.

2.7.2.1 The cost of resampling

Although resampling allows for a decrease in the effects of the particle degeneracy problem (see Section 2.7.1.1), it does not come without a cost. Using resampling has two main drawbacks: the limitation of parallelization of the particle filtering algorithms and the so called particle impoverishment problem.

The use of resampling introduces a step in the particle filter which requires availability of global information, as knowledge of all particle weights is required in the resampling procedure. In addition, the next population of particles is generated in this step. As a consequence, parallelized schemes of particle filters cannot avoid at least a centralized computation in the resampling step, which limits the efficiency of such approaches.

Figure 2.4: The SIR PF algorithm for $N = 10$ particles. Particle size is proportional to its associated weight. Particles are first drawn according to the prior density $p(\mathbf{x}^k | \mathbf{x}^{k-1})$. The update of weights ω_i^k is performed using (2.79). Resampling of particles is made according to the computed weights and finally the resampled particles are once again propagated according to the prior at time $k + 1$.

On the other hand, resampling leads to those particles having high relative weights ω_i^k to be statistically selected multiple times. This induces a loss in diversity among particles, which will have many repeated samples. This problem, known as sample impoverishment, results in a performance decrease of the particle filter which will be severe if the process noise is small [7]. Another consequence is that diversity in particle trajectories is dramatically reduced, which can make smoothing estimates (see Section 2.2.3) to degenerate.

It is worth noting that quite a big effort has been put into solving parallelization [33, 72, 177] and particle impoverishment issues [58, 60, 111, 124] coming from resampling in particle filters, although we do not further review this approaches as this is not in the scope of this thesis.

Figure 2.5: Sample Importance Resampling PF scheme. Particles from the posterior at time $k-1$ are propagated using the dynamic model. Once $\mathbf{z}^k$ arrives, particle weights at time k are computed using (2.79). Particles at time k with higher weights are finally resampled to avoid particle degeneracy.

2.7.3 Marginal particle filter

It has been shown in Section 2.7.1 how the SIS and SIR particle filters produce samples of the posterior PDF $p(\mathbf{x}^k | \mathbf{z}^{1:k})$ by sampling the whole trajectory $\mathbf{x}^{0:k}$ and then discarding the particle states of the previous time steps. Although this is arguably the most widespread procedure to obtain samples from the posterior PDF, it has some drawbacks which are ultimately related to the curse of dimensionality (see Section 2.8). The dimension of the underlying sampled state space grows with every time step, which after a few steps results in a poor representation of the state trajectories, leading to particle degeneracy as explained in Section 2.7.1.1. Although the SIR particle filter can help to alleviate this problem as seen in Section 2.7.2 by considering a resampling stage, it has some non-negligible drawbacks covered in the previous section.

The marginal particle filter is an alternative which can avoid the above described problems by directly producing samples from the (marginal) posterior PDF $p(\mathbf{x}^k | \mathbf{z}^{1:k})$ [86]. Assume the availability of a particle posterior PDF at time $k-1$

$$p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}) \simeq \sum_{j=1}^N \omega_j^{k-1} \delta(\mathbf{x}^{k-1} - \mathbf{x}_j^{k-1}), \quad (2.80)$$

making use of the Chapman-Kolmogorov equation in (2.4) one can approximate the prior PDF as

$$\begin{aligned} p(\mathbf{x}^k | \mathbf{z}^{1:k-1}) &= \int p(\mathbf{x}^k | \mathbf{x}^{k-1}) p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}) d\mathbf{x}^{k-1} \\ &\simeq \sum_{j=1}^N \omega_j^{k-1} p(\mathbf{x}^k | \mathbf{x}_j^{k-1}), \end{aligned} \quad (2.81)$$

while after the arrival of measurement $\mathbf{z}^k$, Bayes' rule in (2.5) yields the approximation to the posterior PDF

$$\begin{aligned} p(\mathbf{x}^k | \mathbf{z}^{1:k}) &\propto p(\mathbf{z}^k | \mathbf{x}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}) \\ &\simeq p(\mathbf{z}^k | \mathbf{x}^k) \sum_{j=1}^N \omega_j^{k-1} p(\mathbf{x}^k | \mathbf{x}_j^{k-1}). \end{aligned} \quad (2.82)$$

Note that in contrast to (2.72), the integral of the prediction step in (2.81) causes the presence of a weighted sum of N densities $p(\mathbf{x}^k | \mathbf{x}_j^{k-1})$ in the posterior PDF in (2.82). The selection of any importance density can be considered as long as it has the appropriate support. Nonetheless, consider for convenience that it takes a similar form to the posterior [86]

$$q(\mathbf{x}^k | \mathbf{z}^{1:k}) = \sum_{j=1}^N \omega_j^{k-1} q(\mathbf{x}^k | \mathbf{x}_j^{k-1}, \mathbf{z}^k), \quad (2.83)$$

which results in the i -th particle weight update equation

$$\begin{aligned} \omega_i^k &= \frac{p(\mathbf{x}_i^k | \mathbf{z}^{1:k})}{q(\mathbf{x}_i^k | \mathbf{z}^{1:k})} \\ &\propto \frac{p(\mathbf{z}^k | \mathbf{x}_i^k) \sum_{j=1}^N \omega_j^{k-1} p(\mathbf{x}_i^k | \mathbf{x}_j^{k-1})}{\sum_{j=1}^N \omega_j^{k-1} q(\mathbf{x}_i^k | \mathbf{x}_j^{k-1}, \mathbf{z}^k)}. \end{aligned} \quad (2.84)$$

The weight variance of the marginal particle filter (see 2.7.4) is lower than that of particle filters sampling the whole state trajectory $\mathbf{x}^{0:k}$ [86], which suggests a better

performance for a given number of particles and a lower necessity of performing particle resampling. However, computing the weight of each particle requires the sum of N terms in (2.84), so that the complexity of the marginal particle filter is $\mathcal{O}(N^2)$, which can result in a prohibitively computational effort for a high number of particles.

2.7.4 Optimal importance density

The choice of a good importance density is a key element in the design of a particle filter, often determining the performance of the algorithm. It is therefore of interest to use the best possible importance density. Typical importance sampling applications usually consider as an optimality criterion the minimization of the variance of an estimator based on the samples produced by the importance density [156]. However, it should be noted that, after a few steps in the filtering procedure, the variance of any estimator based on a particle filter will be dominated by the degeneracy problem (see Section 2.7.1.1), as for a finite number of particles this problem can rapidly lead the filter to divergence. Thus, the optimality criterion in choosing the importance density is often based on the reduction of the variance of unnormalized particle weights in (2.68), as this will maximize the effective sampling size in (2.77), thus diminishing the risk of divergence of the filter due to the particle degeneracy.

The importance density which minimizes the unnormalized particle weight variance of the i -th particle at time k , conditioned to the particle trajectory $\mathbf{x}_i^{0:k-1}$ and the received measurements $\mathbf{z}^{1:k}$ has been shown to be [44, 107, 191]

$$\begin{aligned} q_{opt}(\mathbf{x}^k | \mathbf{x}_i^{k-1}, \mathbf{z}^k) &= p(\mathbf{x}^k | \mathbf{x}_i^{k-1}, \mathbf{z}^k) \\ &= \frac{p(\mathbf{z}^k | \mathbf{x}^k) p(\mathbf{x}^k | \mathbf{x}_i^{k-1})}{p(\mathbf{z}^k | \mathbf{x}_i^{k-1})}. \end{aligned} \quad (2.85)$$

Thus, using (2.85) into (2.73), weights of particles sampled from the optimal importance density are computed as

$$\omega_i^k = p(\mathbf{z}^k | \mathbf{x}_i^{k-1}) \omega_i^{k-1} \quad (2.86)$$

which has the interesting property of producing the same weight without regard of the sample $\mathbf{x}_i^k$ that will be generated [68], meaning that the variance of particle weights is unaffected by the sampling procedure. Unfortunately, the use of the optimal importance density has some important drawbacks which heavily limit its use: it requires sampling from $p(\mathbf{x}^k | \mathbf{x}_i^{k-1}, \mathbf{z}^k)$, and computing particle weights in (2.86)

requires to evaluate

$$p(\mathbf{z}^k | \mathbf{x}_i^{k-1}) = \int p(\mathbf{z}^k | \mathbf{x}^k) p(\mathbf{x}^k | \mathbf{x}_i^{k-1}) d\mathbf{x}^k \quad (2.87)$$

at least up to a normalizing constant. These two conditions are not generally satisfied, so that alternative, easier to use importance sampling densities, such as the one employed by the SIR particle filter in (2.78) are considered.

It is worth noting that the current measurement is used by the optimal importance density in the drawing of samples, a feature which is usually interesting to imitate by other sampling functions whenever the use of the optimal importance sampling density is not possible. Nonetheless, using other importance sampling function will add variance to the weights, decreasing the effective sample size in (2.77).

2.7.5 Auxiliary particle filter

The auxiliary particle filter (APF) [28, 78, 130] makes use of an alternative importance sampling function, that considers the measurement at time k at the sampling step, in an attempt to approach the way the optimal importance density (OID) would obtain samples. The APF achieves this sampling in a higher dimension by taking into account an auxiliary variable, $a \in \{1, \dots, N\}$, which is an index on the mixture

$$p(\mathbf{x}^k | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{x}^k) \sum_{i=1}^N p(\mathbf{x}^k | \mathbf{x}_i^{k-1}) \omega_i^{k-1}, \quad (2.88)$$

therefore sampling from the posterior PDF $p(\mathbf{x}^k, a | \mathbf{z}^{1:k})$. Applying Bayes' rule this posterior can be expressed as

$$\begin{aligned} p(\mathbf{x}^k, a | \mathbf{z}^{1:k}) &\propto p(\mathbf{z}^k | \mathbf{x}^k) p(\mathbf{x}^k, a | \mathbf{z}^{1:k-1}) \\ &= p(\mathbf{z}^k | \mathbf{x}^k) p(\mathbf{x}^k | a, \mathbf{z}^{1:k-1}) p(a | \mathbf{z}^{1:k-1}) \\ &= p(\mathbf{z}^k | \mathbf{x}^k) p(\mathbf{x}^k | \mathbf{x}_a^{k-1}) \omega_a^{k-1}. \end{aligned} \quad (2.89)$$

The APF obtains a set of N samples, $(\mathbf{x}_i^k, a_i)$ for $i = 1, \dots, N$, from (2.89), and then it discards the indices, a_i , to obtain a sample $\mathbf{x}_i^k$ from the marginalized density $p(\mathbf{x}^k | \mathbf{z}^{1:k})$. The importance density for the APF is chosen to satisfy the proportionality

$$q(\mathbf{x}^k, a | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mu_a^k) p(\mathbf{x}^k | \mathbf{x}_a^{k-1}) \omega_a^{k-1} \quad (2.90)$$

where μ_a^k stands for some characterization of $\mathbf{x}^k$ given $\mathbf{x}_a^{k-1}$. Common approaches are the mean $\mu_a^k = \mathbb{E}[\mathbf{x}^k | \mathbf{x}_a^{k-1}]$ or a sample $\mu_a^k \sim p(\mathbf{x}^k | \mathbf{x}_a^{k-1})$. By considering

$$q(\mathbf{x}^k, a | \mathbf{z}^{1:k}) \propto q(a | \mathbf{z}^{1:k}) q(\mathbf{x}^k | a, \mathbf{z}^{1:k}) \quad (2.91)$$

and defining

$$q(\mathbf{x}^k | a, \mathbf{z}^{1:k}) \triangleq p(\mathbf{x}^k | \mathbf{x}_a^{k-1}), \quad (2.92)$$

it follows from (2.90), (2.91) and (2.92) that

$$\lambda_a \triangleq q(a | \mathbf{z}^{1:k}) = p(\mathbf{z}^k | \mu_a^k) \omega_a^{k-1} \quad (2.93)$$

where λ_a are called the first-stage weights. Each sample is therefore assigned a final weight according to

$$\begin{aligned} \omega_i^k &\propto \frac{p(\mathbf{x}_i^k, a_i | \mathbf{z}^{1:k})}{q(\mathbf{x}_i^k, a_i | \mathbf{z}^{1:k})} \\ &= \frac{p(\mathbf{z}^k | \mathbf{x}_i^k) p(\mathbf{x}_i^k | \mathbf{x}_{a_i}^{k-1}) \omega_{a_i}^{k-1}}{p(\mathbf{z}^k | \mu_{a_i}^k) p(\mathbf{x}_i^k | \mathbf{x}_{a_i}^{k-1}) \omega_{a_i}^{k-1}} \\ &= \frac{p(\mathbf{z}^k | \mathbf{x}_i^k)}{p(\mathbf{z}^k | \mu_{a_i}^k)}. \end{aligned} \quad (2.94)$$

The APF can be seen as a sampling importance resampling scheme where the sampling and resampling steps are interchanged in an attempt to favor the generation of particles that are more likely to survive in the next time step. In this sense, APF generally outperforms SIR as its importance density is closer to the optimal choice. A graphical explanation on how APF works can be found in Figure 2.6.

If the process noise is small, $p(\mathbf{x}^k | \mathbf{x}_i^{k-1})$ is sufficiently well characterized by $\mu_{a_i}^k$ also making the APF less sensible to outliers. However, for some specific cases, when the prior density is much broader than the likelihood, a single point does not describe the density $p(\mathbf{x}^k | \mathbf{x}_i^{k-1})$ sufficiently well and the use of the APF approach may degrade the performance of the estimation. The APF algorithm computational complexity is, as for the SIR PF, $\mathcal{O}(N)$.

2.8 Curse of dimensionality

Some applications require dealing with high-dimensional spaces, which is commonly troublesome as a plethora of problems of different nature arise as the dimension increases, causing the performance of algorithms which can adequately tackle low-

Figure 2.6: Auxiliary sampling in the APF. Once $\mathbf{z}^k$ arrives, the first stage weights, $\lambda_i = p(\mathbf{z}^k | \mu_i^k) \omega_i^{k-1}$ (represented by their color, with a hotter color for higher λ_i) are computed. For the sake of simplicity, uniform particle weights at time $k-1$, ω_i^{k-1} , are considered. Particles at time k are formed from particles with higher predicted first-stage weights.

dimension problems to severely drop when applied to high-dimension spaces. This is usually referred to as the curse of dimensionality [32].

In the case of filtering methods which sample the state space, such as the PMF or PFs, the effects of the curse of dimensionality are important, as the sampling of high-dimensional state spaces can be very low efficient. As an example, note that if N samples are necessary to evenly sample a hypercube of dimension 1 (i.e., a line segment) with a required resolution, N^2 samples will be required to sample a hypercube of dimension 2 (i.e., a square) to keep the required sampling resolution. In general, N^d samples will be required to keep the sampling resolution in a d -dimensional hypercube. Difficulties in sampling high-dimensional state spaces will be a capital issue in this thesis, as importance sampling is a fundamental building block of particle filters. Thus, the algorithms presented in the following chapters are designed so as to try to soften

its effects.

Other well-known curse of dimensionality effects can be found in optimization problems (where the difficulty in looking for an optimal value in the search space exponentially grows with its dimension) machine learning problems (where the amount of required data to train a neural network rapidly increases with the number of neurons to train in a given architecture), clustering problems (as distance functions in high dimensional spaces can lose sensitivity to small changes in a single variable), among others.

The next section presents a common approach to tackle the curse of dimensionality in state filtering, known as Rao-Blackwellization which aims to reduce the dimension of the state space.

2.8.1 Rao-Blackwellization

As previously argued, the performance of a particle filter can severely decay if dealing with a high-dimensional state space due to the curse of dimensionality [32]. It is therefore of interest to keep the dimensionality of the state space which a particle filter deals with as low as possible. Rao-Blackwellization is a common technique often used along particle filters to achieve this point [183]. The main idea behind Rao-Blackwellization is to identify those components of the state that admit the use of an optimal closed-form algorithm such as a Kalman Filter, while only feeding the particle filter with the rest of the components of the state that cannot be optimally estimated with such algorithms.

The use of Rao-Blackwellization reduces the dimension which the particle filter has to deal with, however, the possibility of using this technique is dependent on the application at hand, as not all state spaces admit such division of its components. Rao-Blackwellization therefore requires that the state can be partitioned as

$$\mathbf{x}^k = \left[(\mathbf{x}_l^k)^T, (\mathbf{x}_{nl}^k)^T \right]^T \quad (2.95)$$

such that

$$p(\mathbf{x}_l^k, \mathbf{x}_{nl}^{0:k} | \mathbf{z}^{1:k}) = p(\mathbf{x}_l^k | \mathbf{x}_{nl}^{0:k}, \mathbf{z}^{1:k}) p(\mathbf{x}_{nl}^{0:k} | \mathbf{z}^{1:k}) \quad (2.96)$$

where computation of the conditional PDF $p(\mathbf{x}_l^k | \mathbf{x}_{nl}^{0:k}, \mathbf{z}^{1:k})$ admits the use of a closed-form expression. A Rao-Blackwellized particle filter thus consists on using a particle representation to approximate $p(\mathbf{x}_{nl}^{0:k} | \mathbf{z}^{1:k})$ and analytically computing $p(\mathbf{x}_l^k | \mathbf{x}_{nl}^{0:k}, \mathbf{z}^{1:k})$ for each particle using the appropriate closed-form method. The dimensionality which

the particle filter deals with is therefore only that of $\mathbf{x}_{nl}^{0:k}$ so that the effects of the curse of dimensionality in the particle filter can be mitigated.

A common case in which Rao-Blackwellization can be applied is when conditional linear-Gaussian models [49, 183] are involved, so that given $\mathbf{x}_{nl}^{0:k}$ and $\mathbf{z}^{1:k}$, the relation with $\mathbf{x}_l^k$ is linear and Gaussian, and the conditional PDF $p(\mathbf{x}_l^k | \mathbf{x}_{nl}^{0:k}, \mathbf{z}^{1:k})$ can be estimated by means of Kalman filtering. Thus, given a particle representation of $p(\mathbf{x}_{nl}^{0:k} | \mathbf{z}^{1:k})$, a Kalman filter in each particle is run to estimate $p(\mathbf{x}_l^k | \mathbf{x}_{nl}^{0:k}, \mathbf{z}^{1:k})$.

The posterior PDF $p(\mathbf{x}_l^k, \mathbf{x}_{nl}^k | \mathbf{z}^{1:k})$, can be approximated by discarding the particle states prior to time k , so that considering the use of a Kalman filter on each particle, the posterior can be approximated as

$$p(\mathbf{x}_l^k, \mathbf{x}_{nl}^k | \mathbf{z}^{1:k}) = \sum_{i=1}^N \omega_i^k \delta(\mathbf{x}_{nl}^k - \mathbf{x}_{nl,i}^k) \mathcal{N}(\mathbf{x}_l^k; \bar{\mathbf{x}}_{l,i}^k, \Sigma_{l,i}^k) \quad (2.97)$$

where $\mathbf{x}_{nl,i}^k$ is the state of the i -th particle representing the nonlinear component of the state in (2.95) with associated weight ω_i^k , while $\bar{\mathbf{x}}_{l,i}^k$ and $\Sigma_{l,i}^k$ are respectively the mean and covariance matrix of the output of the Kalman filter applied to $\mathbf{x}_{nl}^{0:k}$ and $\mathbf{z}^{1:k}$.

2.9 Partitioned particle filters

In this section partitioned particle filters such as the independent partition (IP) PF or the parallel partition (PP) PF are reviewed. As previously explained in Section 2.8, the curse of dimensionality can be a limiting factor in the use of particle filters due to the difficulty of sampling high-dimensional state spaces. As explained in Section 2.8.1, taking advantage of the structure of the state space allows Rao-Blackwellization to reduce the dimension which the particle filter has to sample from, mitigating the effects of the curse of dimensionality. The same principle, i.e., exploiting some structure of the state space, is used by partitioned particle filters to also tackle high-dimensional filtering problems while appeasing the curse of dimensionality. The distinct feature of partitioned particle filters is that the structure of the state space is assumed to be such that it can be partitioned as

$$\mathbf{X}^k = [(\mathbf{x}_1^k)^T, \dots, (\mathbf{x}_t^k)^T]^T \quad (2.98)$$

where t is assumed to be fixed and known and T stands for the vector transpose.

It should be noted that the use of a capital letter $\mathbf{X}^k$ is considered so as to easily distinguish between the complete state of the dynamic system (capital letter) and the

elements of the partition (small letter $\mathbf{x}_j^k$). This notation is used in the rest of this thesis with the purpose of easily distinguishing if a state space partitionable as in (2.98) is being considered. Particle $\mathbf{X}_i^k$ is therefore

$$\mathbf{X}_i^k = [(\mathbf{x}_{1,i}^k)^T, \dots, (\mathbf{x}_{t,i}^k)^T]^T \in \mathbb{R}^{n \cdot t} \quad (2.99)$$

where $\mathbf{x}_{j,i}^k$ is the part of the particle that corresponds to target j , which is referred to as a subparticle. Each particle is therefore formed by t subparticles. To describe the proposed methods, we find the following nomenclature useful. In the prediction and update steps from time $k-1$ to time k , particles and subparticles at time $k-1$ are referred to as *parent* particles and subparticles, while particles and subparticles at time k are referred to as *child* particles and subparticles.

As in the particle filter methods reviewed in previous sections, the posterior PDF of the state trajectory $p(\mathbf{X}^{0:k-1}|\mathbf{z}^{1:k-1})$ is approximated by sequentially propagating a set of N particles with their associated weights $\{(\mathbf{X}_1^{0:k-1}, \omega_1^{k-1}), \dots, (\mathbf{X}_N^{0:k-1}, \omega_N^{k-1})\}$ as

$$p(\mathbf{X}^{0:k-1}|\mathbf{z}^{1:k-1}) \simeq \sum_{i=1}^N \omega_i^{k-1} \delta(\mathbf{X}^{0:k-1} - \mathbf{X}_i^{0:k-1}), \quad (2.100)$$

allowing for an approximation of the posterior PDF of the state at time $k-1$, by integrating out the previous states as

$$p(\mathbf{X}^{k-1}|\mathbf{z}^{1:k-1}) \simeq \sum_{i=1}^N \omega_i^{k-1} \delta(\mathbf{X}^{k-1} - \mathbf{X}_i^{k-1}). \quad (2.101)$$

Derivation of partitioned filters in the literature [50, 128] rely on a specific dynamic model (usually, the nearly constant velocity model [12]), however, this kind of filters are applicable to a broad kind of dynamic models of the form

$$p(\mathbf{X}^k|\mathbf{X}^{k-1}) = \prod_{j=1}^t p(\mathbf{x}_j^k|\mathbf{x}_j^{k-1}), \quad (2.102)$$

while the predicted PDF can be expressed as

$$\begin{aligned} p(\mathbf{X}^k|\mathbf{z}^{1:k-1}) &= \int p(\mathbf{X}^k|\mathbf{X}^{k-1}) p(\mathbf{X}^{k-1}|\mathbf{z}^{1:k-1}) d\mathbf{X}^{k-1} \\ &\simeq \int p(\mathbf{X}^k|\mathbf{X}^{k-1}) \sum_{i=1}^N \omega_i^{k-1} \delta(\mathbf{X}^{k-1} - \mathbf{X}_i^{k-1}) d\mathbf{X}^{k-1} \\ &= \sum_{i=1}^N \omega_i^{k-1} p(\mathbf{X}^k|\mathbf{X}_i^{k-1}) \end{aligned} \quad (2.103)$$

$$= \sum_{i=1}^N \omega_i^{k-1} \prod_{j=1}^t p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}).$$

Once the measurement $\mathbf{z}^k$ has arrived, making use of Bayes' rule, the posterior of the state at time k can be written as

$$p(\mathbf{X}^k | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{X}^k) \sum_{i=1}^N \omega_i^{k-1} \prod_{j=1}^t p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}). \quad (2.104)$$

Samples from this PDF are obtained from an importance sampling $q(\mathbf{X}^k | \mathbf{z}^{1:k})$. The update of the weights is performed according to the principle of importance sampling as [7]

$$\begin{aligned} \omega_i^k &= \frac{p(\mathbf{X}_i^k | \mathbf{z}^{1:k})}{q(\mathbf{X}_i^k | \mathbf{z}^{1:k})} \\ &\propto \frac{p(\mathbf{z}^k | \mathbf{X}_i^k) \sum_{i=1}^N \omega_i^{k-1} \prod_{j=1}^t p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1})}{q(\mathbf{X}_i^k | \mathbf{z}^{1:k})}. \end{aligned} \quad (2.105)$$

Many different possibilities exist in the selection of $q(\mathbf{X}^k | \mathbf{z}^{1:k})$ to approximate the posterior PDF $p(\mathbf{X}^k | \mathbf{z}^{1:k})$ with each one defining a different particle filter. Two such approximations in the literature are presented in the following section, the independent partition and the parallel partition particle filters, which are often used in the first part of this thesis as baseline methods for performance comparison. The algorithms presented in Chapter 3 of this thesis fall within the framework of partitioned particle filters.

2.9.1 Independent partition and parallel partition particle filters

In this section the IP [92] and PP [50] methods are reviewed. The IP and PP methods are intrinsically designed for being able of competently dealing with high-dimensional state spaces. The IP and PP algorithms were conceived for the problem of multiple target tracking, which adequately suits the condition stated in (2.98), with the state of the j -th target at time k being represented by $\mathbf{x}_j^k$, while $\mathbf{X}^k$ gathers the state of the t targets in the scenario. In this context, the PP method is an improvement of the IP method to account for nearby targets in the sampling procedure, which can be applied to any dynamic and measurement model.

The IP [92] and PP [50] methods make the following two assumptions:

- Assumption 1: The state of the different components in the partition in (2.98) are *a priori* independent.
- Assumption 2: The particle weights are uniformly distributed, i.e., $\omega_i^{k-1} = \frac{1}{N}$ for $i = 1, 2, \dots, N$.

Under this two assumptions, the prior and posterior PDFs in (2.103) and (2.104) can be respectively written as (see Appendix A.2 and [50] for further details on this point)

$$p(\mathbf{X}^k | \mathbf{z}^{1:k-1}) \propto \prod_{j=1}^t \sum_{i=1}^N p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}) \quad (2.106)$$

$$p(\mathbf{X}^k | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{X}^k) \prod_{j=1}^t \sum_{i=1}^N p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}) \quad (2.107)$$

where $\mathbf{x}_{j,i}^{k-1}$ is the state vector of the i -th particle for target j at time $k-1$.

It should be noted that even though assumptions like Assumption 1 are quite common [25, 46, 125, 178], it does not necessarily hold, as conditioning on measurements can make the different marginal PDFs of the components of the partition to be dependent, so that this assumption can introduce a non-negligible bias in the estimation of the state. Nevertheless, it has been shown that independence assumptions like Assumption 1 improve the performance of particle filters for a low enough number of particles or a high enough number of components in (2.98) [190], therefore allowing for a trade-off between bias and variance of the state estimator. On the other hand, it must be noted that Assumption 2 imposes a particle resampling step after every iteration to guarantee that weights are evenly distributed [7].

IP and PP methods obtain samples from the posterior by making use of an auxiliary variable [130] for each target, therefore sampling in a higher dimension from

$$p(\mathbf{X}^k, \mathbf{a} | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{X}^k) \prod_{j=1}^t p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}) \quad (2.108)$$

where $\mathbf{a} = [a_1, a_2, \dots, a_t]^T \in \{1, 2, \dots, N\}^t$, and a_j is an index in the mixture of the j -th target in (2.107). The selection of the auxiliary variable is sound because $p(\mathbf{X}^{k+1}, \mathbf{a} | \mathbf{z}^{1:k+1}) \geq 0$ and by integrating (summing) (2.108) over $\mathbf{a}$, one gets the posterior. The use of an auxiliary variable decreases the computational burden of IP and PP and enables subparticle crossover.

A subparticle is the fragment of a particle that represents the state of a component of the partition in (2.98). Subparticle crossover, which is a desirable property in

Table 2.1: Feature comparison between IP and PP algorithms

MTT filter	auxiliary filtering in component sampling	target-resampling	sampling accounts for other components	avoids particle resampling
IP	×	✓	×	×
PP	×	✓	✓	×

partitioned filters, refers to the formation of a particle at time k from the propagation of subparticles that belonged to different particles at time $k - 1$. IP and PP perform the crossover between subparticles via a target-resampling step.

The importance density function for IP and PP is

$$q(\mathbf{X}^k, \mathbf{a} | \mathbf{z}^{1:k}) = \prod_{j=1}^t q_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \quad (2.109)$$

$$q_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto b_j(\mathbf{x}_j^k) p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}) \quad (2.110)$$

where the state of each component $\mathbf{x}_j^k$ can be independently sampled because of the independence assumption in Assumption 1. These filters only differ in the definition of their first-stage weights or predicted component likelihood $b_j(\mathbf{x}_j^k)$

- IP

$$b_j(\mathbf{x}_j^k) \propto p(\mathbf{z}^k | \mathbf{x}_j^k) \quad (2.111)$$

- PP

$$b_j(\mathbf{x}_j^k) \propto p(\mathbf{z}^k | \hat{\mathbf{X}}_{-\{j\}}^k, \mathbf{x}_j^k) \quad (2.112)$$

where

$$\hat{\mathbf{X}}_{-\{j\}}^k = [(\hat{\mathbf{x}}_1^k)^T, \dots, (\hat{\mathbf{x}}_{j-1}^k)^T, (\hat{\mathbf{x}}_{j+1}^k)^T, \dots, (\hat{\mathbf{x}}_t^k)^T]^T \quad (2.113)$$

incorporates the prediction of the state at time k of every component of the partition of the state except for the j -th, averaged over all the particles as

$$\hat{\mathbf{x}}_j^k = \sum_{i=1}^N \omega_i^{k-1} \cdot \mathbf{x}_{j,i}^{k|k-1} \quad (2.114)$$

where $\mathbf{x}_{j,i}^{k|k-1}$ is the prediction of the j -th target state at time k of the i -th subparticle at time $k - 1$ [49]. This way, the PP method outperforms IP method when there are

Figure 2.7: Subparticle crossover in IP and PP for $t = 3$ components in the partitioned state for the IP and PP algorithms using two particles. The particles at time k can be formed from the propagation of subparticles belonging to different particles at time $k - 1$. Once $\mathbf{z}^k$ arrives, the predicted component likelihoods, $b_j(\mathbf{x}_j^k)$, are computed. Particles at time k are more likely to be formed by subparticles with higher predicted component likelihoods.

targets moving in close proximity as IP does not take other targets into consideration in the sampling step. Once $b_j(\mathbf{x}_{j,i}^k)$ is available, the crossover between subparticles is performed, resulting in the formation of particles, $\mathbf{X}_i^k$, from those subparticles that show a higher predicted target likelihood value. A schematic example of this procedure is shown in Figure 2.7.

Making use of (2.108) and (2.109), the updated weights are obtained according to the principle of importance sampling as

$$\omega_i^k = \frac{p(\mathbf{X}_i^k, \mathbf{a}_i | \mathbf{z}^{1:k})}{q(\mathbf{X}_i^k, \mathbf{a}_i | \mathbf{z}^{1:k})}$$

$$\begin{aligned}
& \propto \frac{p(\mathbf{z}^k | \mathbf{X}_i^k) \prod_{j=1}^t p(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,a_j^i}^{k-1})}{\prod_{j=1}^t b_j(\mathbf{x}_{j,i}^k) p(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,a_j^i}^{k-1})} \\
& = \frac{p(\mathbf{z}^k | \mathbf{X}_i^k)}{\prod_{j=1}^t b_j(\mathbf{x}_{j,i}^k)}.
\end{aligned} \tag{2.115}$$

where $\mathbf{a}_i = [a_1^i, \dots, a_t^i]^T$ is the i -th sample from the auxiliary variable $\mathbf{a}$.

It is not generally feasible or computationally affordable to sample directly from the desired importance density in (2.110). IP and PP try to accomplish this by making use of the sampling/importance resampling method described in [8], drawing samples $(\mathbf{x}_j^*, a_j^*)$ for $p = 1, 2, \dots, M$ from a first-pass approximation of $q_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k})$, defined as

$$h_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}) \tag{2.116}$$

and computing the first-stage weights

$$r(\mathbf{x}_{j,p}^*, a_{j,p}^*) \propto \frac{q_j(\mathbf{x}_{j,p}^*, a_{j,p}^* | \mathbf{z}^{1:k})}{h_j(\mathbf{x}_{j,p}^*, a_{j,p}^* | \mathbf{z}^{1:k})} = b_j(\mathbf{x}_{j,p}^*), \tag{2.117}$$

allowing the generation of N samples of $q_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k})$ from the distribution defined by $(\mathbf{x}_{j,p}^*, a_{j,p}^*)$ for $p = 1, 2, \dots, M$ with probability $r(\mathbf{x}_{j,p}^*, a_{j,p}^*)$. The pseudocode for the PP and IP methods is respectively provided in Algorithms 2.1 and 2.2, and some features of interest of PP and IP are shown in Table 2.1.

It should be noted that another great advantage of the PP method is that it can always be applied to approximate the prediction (2.4) and update (2.5) equations, i.e., it can be satisfactorily applied to any high-dimensional filtering problem that meets Assumptions 1 and 2 besides multiple target tracking. In the multiple target tracking problem the likelihood for a given target, $p(\mathbf{z}^k | \mathbf{x}_j^k)$, is usually defined so IP method can be generally applied in multiple target tracking problems. However, other high-dimensional filtering problems do not admit the use of the IP method, as the likelihood for a given component of the state, $p(\mathbf{z}^k | \mathbf{x}_j^k)$, might not be defined.

Neither IP nor PP sample from the OID and they both need a final particle resampling stage, in order to meet Assumption 2 after every time step, which generally causes a degradation of performance due to a loss of diversity among the particles. In addition, both IP and PP boil down to the SIR PF when $t = 1$. These two latter problems are tackled in this thesis in Chapter 3, resulting in PF methods for high-dimensional filtering problems which generalize the APF, gathering all the outstanding properties of the APF while appeasing the curse of dimensionality that would arise from its direct application over the complete state space with high-dimensional problems.

Algorithm 2.1 IP procedure

- Initialize particles according to $p(\mathbf{X}^0)$ with uniform weights, $\omega_i^0 = \frac{1}{N}$

for $k = 1, \dots, k_{end}$ **do** ▷ k_{end} is the final time step

- Collect new measurements $\mathbf{z}^k$ from sensors

for $j = 1, \dots, t$ **do**

for $i = 1, \dots, N$ **do**

- Draw a sample $\mathbf{x}_{j,i}^*$ using $p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1})$

- Compute $b_j(\mathbf{x}_{j,i}^*) = p(\mathbf{z}^k | \mathbf{x}_{j,i}^*)$

end for

for $i = 1, \dots, N$ **do**

- Normalize $b_j(\mathbf{x}_{j,i}^*) = \frac{b_j(\mathbf{x}_{j,i}^*)}{\sum_{i=1}^N b_j(\mathbf{x}_{j,i}^*)}$

end for

for $i = 1, \dots, N$ **do** ▷ Target-resampling

- Sample an index p from the distribution defined by $(b_j(\mathbf{x}_{j,1}^*), \dots, b_j(\mathbf{x}_{j,N}^*))$

- Set $\mathbf{x}_{j,i}^k = \mathbf{x}_{j,p}^*$

end for

end for

for $i = 1, \dots, N$ **do** ▷ Weight Update

- Compute $\omega_i^k = \frac{p(\mathbf{z}^k | \mathbf{X}_i^k)}{\prod_{j=1}^t b_j(\mathbf{x}_{j,i}^k)}$

end for

for $i = 1, \dots, N$ **do**

- Normalize $\omega_i^k = \frac{\omega_i^k}{\sum_{i=1}^N \omega_i^k}$

end for

for $i = 1, \dots, N$ **do** ▷ Global resampling

- Sample an index r from the distribution defined by $(\omega_1^k, \dots, \omega_N^k)$

- Set $\mathbf{X}_i^k = \mathbf{X}_r^k$

end for

for $i = 1, \dots, N$ **do**

- Set $\omega_i^k = \frac{1}{N}$

end for

end for

Algorithm 2.2 PP procedure

- Initialize particles according to $p(\mathbf{X}^0)$ with uniform weights, $\omega_i^0 = \frac{1}{N}$

for $k = 1, \dots, k_{end}$ **do** $\triangleright k_{end}$ is the final time step

- Collect new measurements $\mathbf{z}^k$ from sensors

for $j = 1, \dots, t$ **do**

for $i = 1, \dots, N$ **do**

- Draw a sample $\mathbf{x}_{j,i}^*$ using $p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1})$

end for

- Compute $\hat{\mathbf{x}}_j^k = \sum_{i=1}^N \omega_i^{k-1} \cdot \mathbf{x}_{j,i}^*$

end for

for $j = 1, \dots, t$ **do**

for $i = 1, \dots, N$ **do**

- Compute $b_j(\mathbf{x}_{j,i}^*) = p(\mathbf{z}^k | \hat{\mathbf{X}}_{-\{j\}}^k, \mathbf{x}_{j,i}^*)$

end for

for $i = 1, \dots, N$ **do**

- Normalize $b_j(\mathbf{x}_{j,i}^*) = \frac{b_j(\mathbf{x}_{j,i}^*)}{\sum_{i=1}^N b_j(\mathbf{x}_{j,i}^*)}$

end for

for $i = 1, \dots, N$ **do** $\triangleright$ Target-resampling

- Sample an index p from the distribution defined by $(b_j(\mathbf{x}_{j,1}^*), \dots, b_j(\mathbf{x}_{j,N}^*))$

- Set $\mathbf{x}_{j,i}^k = \mathbf{x}_{j,p}^*$

end for

end for

for $i = 1, \dots, N$ **do** $\triangleright$ Weight Update

- Compute $\omega_i^k = \frac{p(\mathbf{z}^k | \mathbf{X}_i^k)}{\prod_{j=1}^t b_j(\mathbf{x}_{j,i}^k)}$

end for

for $i = 1, \dots, N$ **do**

- Normalize $\omega_i^k = \frac{\omega_i^k}{\sum_{i=1}^N \omega_i^k}$

end for

for $i = 1, \dots, N$ **do** $\triangleright$ Global resampling

- Sample an index r from the distribution defined by $(\omega_1^k, \dots, \omega_N^k)$

- Set $\mathbf{X}_i^k = \mathbf{X}_r^k$

end for

for $i = 1, \dots, N$ **do**

- Set $\omega_i^k = \frac{1}{N}$

end for

end for

2.10 Multiple particle filtering

As previously stated, directly computing or approximating (2.4) and (2.5) can be a difficult task due to the possible nonlinearities in (2.2) and (2.3), specially when the dimensionality of the involved spaces is high. As in partitioned particle filters in Section 2.9, multiple filters [15, 26, 31, 40, 42, 184, 185] address this problem by considering that the state of the dynamic system can be partitioned into t components as in (2.98) [31]. However, instead of approximating the posterior PDF of the joint state in (2.104), multiple filters aim to separately approximate the marginal posterior PDF $p(\mathbf{x}_j^k | \mathbf{z}^{1:k})$ with a different filter for each component of the partition in (2.98). MPFs therefore need to approximate marginalization integrals to compute each marginal posterior PDF. Computation of the involved marginalization integrals does not generally admit an analytical result, and approximations via Monte Carlo sampling can result in computationally expensive methods [113]. Thus, it is of high interest in multiple filtering to develop tractable, accurate approximations to deal with these marginalization integrals. The multiple PFs presented in [15, 26, 42] follow the same strategy but avoid the computational burden of [113] by means of additional approximations of the involved densities. In order to do so, it is assumed that the dynamic model is such that the state transition PDF can be written as in (2.102). While using (2.102) is not a constraint of multiple filters, which can be adapted to consider a broader family of dynamic models [40, 42], it is still a common assumption in multiple filtering [15, 26, 31, 39], and we adopt it so as to simplify their derivation and focus on how to correctly perform the update step, which is generally the key point of multiple filtering. In addition, we assume that

- A1: The posterior PDF is independent so that

$$p(\mathbf{X}^{k-1} | \mathbf{z}^{1:k-1}) = \prod_{j=1}^t p(\mathbf{x}_j^{k-1} | \mathbf{z}^{1:k-1}). \quad (2.118)$$

It should be noted that, due to the correlation arising from processing the measurements at time $k-1$, the posterior PDF is not generally independent as indicated by A1. Nevertheless, it is shown in [190] that, for a Gaussian posterior, the posterior independence assumption improves performance for a high enough state dimension or a low enough number of particles. This assumption has been demonstrated to be beneficial in other high-dimensional particle filters [50, 128, 164]. Making use of (2.118)

and (2.102), the predicted PDF in (2.4) can be written as

$$\begin{aligned} p(\mathbf{X}^k | \mathbf{z}^{1:k-1}) &= \prod_{j=1}^t \left[\int p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{k-1} | \mathbf{z}^{1:k-1}) d\mathbf{x}_j^{k-1} \right] \\ &= \prod_{j=1}^t p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1}) \end{aligned} \quad (2.119)$$

where $p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1})$ is the predicted density of the j -th component of the state at time k based on measurements up to time $k-1$.

As previously mentioned, multiple filters separately approximate the marginal posterior PDF $p(\mathbf{x}_j^k | \mathbf{z}^{1:k})$ for each component of the partition in (2.98) using an individual filter. To explain this procedure, we define vector

$$\mathbf{X}_{-\{j\}}^k = [(\mathbf{x}_1^k)^T, \dots, (\mathbf{x}_{j-1}^k)^T, (\mathbf{x}_{j+1}^k)^T, \dots, (\mathbf{x}_t^k)^T]^T \quad (2.120)$$

in which the state of the j -th component has been removed from $\mathbf{X}^k$. Using (2.5) and (2.119), the marginal posterior PDF of $\mathbf{x}_j^k$ can then be computed as

$$\begin{aligned} p(\mathbf{x}_j^k | \mathbf{z}^{1:k}) &= \int p(\mathbf{X}^k | \mathbf{z}^{1:k}) d\mathbf{X}_{-\{j\}}^k \\ &\propto p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1}) \int p(\mathbf{z}^k | \mathbf{X}^k) p(\mathbf{X}_{-\{j\}}^k | \mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k \end{aligned} \quad (2.121)$$

where using (2.119)

$$p(\mathbf{X}_{-\{j\}}^k | \mathbf{z}^{1:k-1}) = \prod_{l=1, l \neq j}^t p(\mathbf{x}_l^k | \mathbf{z}^{1:k-1}). \quad (2.122)$$

Approximating these marginal posterior distributions instead of the high-dimensional joint state posterior PDF generally results in a better performing particle filter [15, 26, 40, 42]. Unfortunately, the computation of (2.121) is not usually analytically tractable and approximations such as those provided by PFs or alternative filtering methods [31, 184] are commonly used.

Multiple PFs use an individual PF to approximate each of the marginal posterior PDFs of the different components of the partition in (2.98). However, in order to efficiently achieve its purpose, the j -th PF of a multiple PF approximates the marginal posterior PDF of $\mathbf{x}_j^k$ in an indirect way by considering the whole sequence of states up to time k , $\mathbf{x}_j^{0:k}$. The advantage of this approach is that it is no longer necessary to

compute the integral in the prediction step, see (2.4), so that [7, 136]

$$p(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k-1}) = p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{0:k-1} | \mathbf{z}^{1:k-1}) \quad (2.123)$$

and

$$p(\mathbf{X}^{0:k} | \mathbf{z}^{1:k-1}) = \prod_{j=1}^t p(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k-1}) \quad (2.124)$$

where assumption A1 has been extended to the state sequence.

We also denote $\mathbf{X}_{-\{j\}}^{0:k} = (\mathbf{X}_{-\{j\}}^0, \dots, \mathbf{X}_{-\{j\}}^k)$, which represents the sequence of states up to time k of all the components of the partition except for the j -th. Given that the measurements only depend on the current state of the system, the marginal posterior PDF can be written as

$$\begin{aligned} p(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k}) &\propto \int p(\mathbf{z}^k | \mathbf{X}^k) \prod_{l=1}^t [p(\mathbf{x}_l^{0:k} | \mathbf{z}^{1:k-1})] d\mathbf{X}_{-\{j\}}^{0:k} \\ &= p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{0:k-1} | \mathbf{z}^{1:k-1}) \int p(\mathbf{z}^k | \mathbf{X}^k) p(\mathbf{X}_{-\{j\}}^k | \mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k. \end{aligned} \quad (2.125)$$

MPFs use a different particle representation for each marginal posterior PDF. Therefore, we have t different PFs with a set of N weighted samples (particles) $\{(\mathbf{x}_{j,1}^{0:k}, \omega_{j,1}^k), \dots, (\mathbf{x}_{j,N}^{0:k}, \omega_{j,N}^k)\}$, where $\mathbf{x}_{j,i}^{0:k}$ stands for the i -th particle of the j -th filter at time k , with associated weight $\omega_{j,i}^k$. Then, the marginal posterior PDF of $\mathbf{x}_j^{0:k}$ is approximated as

$$p(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k}) \approx \sum_{i=1}^N \omega_{j,i}^k \delta(\mathbf{x}_j^{0:k} - \mathbf{x}_{j,i}^{0:k}). \quad (2.126)$$

Approximating (2.125) via direct Monte Carlo sampling is quite computationally demanding as we need to sample a large state space to compute the marginalization integral in (2.125) so as to take into account the effect in measurements of the states of the other components of the partition. Therefore, multiple PFs usually make use of some approximations that enable the use of several parallel computationally efficient PFs which share information among themselves to approximate (2.125). We proceed to review two methods in the literature: the first-order multiple PF and C-PF PROP.

2.10.1 Multiple particle filter

The first multiple PF (MPF) was presented in [26], providing the general multiple filtering framework as well as a first-order approximation to (2.125). At time $k-1$, the l -th PF approximates the predicted value of $\mathbf{x}_l^k$. There are different ways of obtaining

these predictions. Here we consider

$$\hat{\mathbf{x}}_l^k \approx \sum_{i=1}^N \omega_{l,i}^{k-1} \cdot \mathbf{x}_{l,i}^{k|k-1} \quad (2.127)$$

where $\mathbf{x}_{l,i}^{k|k-1}$ is obtained using the dynamic model to propagate particle $\mathbf{x}_{l,i}^{k-1}$ as

$$\mathbf{x}_{l,i}^{k|k-1} \sim p(\mathbf{x}_l^k | \mathbf{x}_{l,i}^{k-1}). \quad (2.128)$$

Thus, $\hat{\mathbf{x}}_l^k$ is the predicted state of the l -th component of the state at time k given its particle posterior approximation at time $k-1$. Once these predictions have been computed, they are sent to the rest of the PFs. At time k , the j -th PF receives all these estimates of the states of the other components and can now build

$$\hat{\mathbf{X}}_{-\{j\}}^k = [(\hat{\mathbf{x}}_1^k)^T, \dots, (\hat{\mathbf{x}}_{j-1}^k)^T, (\hat{\mathbf{x}}_{j+1}^k)^T, \dots, (\hat{\mathbf{x}}_t^k)^T]^T. \quad (2.129)$$

The j -th PF then makes use of the following first-order approximation

$$p(\mathbf{X}_{-\{j\}}^k | \mathbf{z}^{1:k-1}) \approx \delta(\mathbf{X}_{-\{j\}}^k - \hat{\mathbf{X}}_{-\{j\}}^k). \quad (2.130)$$

Using (2.130), the marginal in (2.125) can be expressed as

$$p(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{x}_j^k, \hat{\mathbf{X}}_{-\{j\}}^k) p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{0:k-1} | \mathbf{z}^{1:k-1}). \quad (2.131)$$

Now, it should be noted that (2.131) has the form of a particle filter prediction and update with a modified likelihood function $p(\mathbf{z}^k | \mathbf{x}_j^k, \hat{\mathbf{X}}_{-\{j\}}^k)$. Then, particles at time k are sequentially drawn from a proposal or importance function $q_j(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k})$ which is chosen to factorize as

$$q_j(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k}) = p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) q_j(\mathbf{x}_j^{0:k-1} | \mathbf{z}^{1:k-1}). \quad (2.132)$$

As with the SIS method in Section 2.7.2, particles $\mathbf{x}_{j,i}^{0:k-1}$ in (2.126) were drawn at the previous time step from $q_j(\mathbf{x}_j^{0:k-1} | \mathbf{z}^{1:k-1})$, so that a new particle, $\mathbf{x}_{j,i}^{0:k}$, is drawn by sampling from $p(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,i}^{k-1})$ and appending it to the previous i -th particle [7, 26]. Particle weights are computed according to the principle of importance sampling as

$$\omega_{j,i}^k \propto \frac{p(\mathbf{x}_{j,i}^{0:k} | \mathbf{z}^{1:k})}{q_j(\mathbf{x}_{j,i}^{0:k} | \mathbf{z}^{1:k})} \quad (2.133)$$

where substituting (2.131) and (2.132) into (2.133) yields

$$\begin{aligned}\omega_{j,i}^k &\propto \frac{p\left(\mathbf{z}^k | \mathbf{x}_{j,i}^k, \hat{\mathbf{X}}_{-\{j\}}^k\right) p\left(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,i}^{k-1}\right) p\left(\mathbf{x}_{j,i}^{0:k-1} | \mathbf{z}^{1:k-1}\right)}{p\left(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,i}^{k-1}\right) q_j\left(\mathbf{x}_{j,i}^{0:k-1} | \mathbf{z}^{1:k-1}\right)} \\ &= p\left(\mathbf{z}^k | \mathbf{x}_{j,i}^k, \hat{\mathbf{X}}_{-\{j\}}^k\right) \omega_{j,i}^{k-1}.\end{aligned}\quad (2.134)$$

This recursion is repeated for each component of the partition at each time step. Note that, in practice, due to the way samples are drawn through (2.132), we can discard the history of states of the particles up to time $k-2$ to approximate $p\left(\mathbf{x}_j^k | \mathbf{z}^{1:k}\right)$. This allows a memory-efficient implementation of the MPF. Nevertheless, the algorithm is actually dealing with the whole sequence of states in $\mathbf{x}_j^{0:k}$, so that the dimension of the state grows with every time step, which causes sample degeneracy [44] (see Section 2.7.1.1). In order to deal with this problem, MPF considers an adaptive resampling scheme [26] based on the monitoring of the effective sample size of each filter [88, 136].

2.10.2 C-PF-PROP

The first-order approximation MPF reviewed in the previous section can help alleviate the curse of dimensionality compared to a standard particle filter [26, 39, 40]. However, its main limitation comes from the approximation in (2.130), which can make the filter perform poorly if $p\left(\mathbf{X}_{-\{j\}}^k | \mathbf{z}^{1:k-1}\right)$ is not narrowly concentrated around $\hat{\mathbf{X}}_{-\{j\}}^k$.

The C-PF-PROP [15] addresses this limitation by making each particle filter transmit a second-order approximation of its marginal posterior PDFs. Thus, instead of using the approximation in (2.130), the j -th PF in C-PF-PROP uses a Gaussian approximation, taking into account not only the mean of the predicted PDF, but also its covariance matrix. The marginal predicted PDF of $p(\mathbf{X}_{-\{j\}}^k | \mathbf{z}^{1:k-1})$ in (2.125) is thus approximated as

$$p\left(\mathbf{X}_{-\{j\}}^k | \mathbf{z}^{1:k-1}\right) \approx \prod_{l=1, l \neq j}^t \mathcal{N}\left(\mathbf{x}_l^k; \hat{\mathbf{x}}_l^k, \hat{\mathbf{C}}_l^k\right). \quad (2.135)$$

Note that $\hat{\mathbf{x}}_l^k$ is given by (2.127) and the covariance matrix can be computed from (2.126) as

$$\begin{aligned}\hat{\mathbf{C}}_l^k &= \mathbb{E}\left[\left(\mathbf{x}_l^k - \hat{\mathbf{x}}_l^k\right)\left(\mathbf{x}_l^k - \hat{\mathbf{x}}_l^k\right)^T \middle| \mathbf{z}^{1:k-1}\right] \\ &\approx \sum_{i=1}^N \omega_{l,i}^{k-1} \cdot \left(\mathbf{x}_{l,i}^{k|k-1} - \hat{\mathbf{x}}_l^k\right) \cdot \left(\mathbf{x}_{l,i}^{k|k-1} - \hat{\mathbf{x}}_l^k\right)^T.\end{aligned}\quad (2.136)$$

Once $\hat{\mathbf{x}}_l^k$ and $\hat{\mathbf{C}}_l^k$ have been computed¹, they are distributed to the rest of the filters as in the MPF so that they can take into account this information about the rest of the components of the state to compute their marginal densities.

The j -th PF uses all the received information to build $\hat{\mathbf{X}}_{-\{j\}}^k$ and $\hat{\mathbf{C}}_{-\{j\}}^k$, respectively the collection of all the first and second predicted moments of the components of the marginal states, except for the j -th. Using (2.136) in (2.125), C-PF-PROP then approximates the marginal posterior PDFs by

$$p(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k}) \propto p(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k-1}) p(\mathbf{z}^k | \mathbf{x}_j^k, \hat{\mathbf{X}}_{-\{j\}}^k, \hat{\mathbf{C}}_{-\{j\}}^k) \quad (2.137)$$

where we use the notation

$$p(\mathbf{z}^k | \mathbf{x}_j^k, \hat{\mathbf{X}}_{-\{j\}}^k, \hat{\mathbf{C}}_{-\{j\}}^k) \triangleq \int p(\mathbf{z}^k | \mathbf{X}^k) \prod_{l=1, l \neq j}^t [\mathcal{N}(\mathbf{x}_l^k; \hat{\mathbf{x}}_l^k, \hat{\mathbf{C}}_l^k)] d\mathbf{X}_{-\{j\}}^k. \quad (2.138)$$

As in the MPF, particles are drawn from the importance sampling function in (2.132), so that using (2.133), particle weights are computed as

$$\omega_{j,i}^k \propto p(\mathbf{z}^k | \mathbf{x}_{j,i}^k, \hat{\mathbf{X}}_{-\{j\}}^k, \hat{\mathbf{C}}_{-\{j\}}^k) \omega_{j,i}^{k-1}. \quad (2.139)$$

The problem with this recursion is that (2.138) does not generally have an analytical solution. In order to solve this, an analytical approximation, which includes the calculation of Jacobians, was provided in [15] for the received signal strength indicator (RSSI) measurement model. However, it is not indicated how the analytical approximation of [15, -Eq. (29)], would be done for other measurement models, which limits the applicability of C-PF-PROP. Chapter 4 addresses this limitation by proposing two novel multiple particle filter methods, which make use of a second-order approximation to the marginals as C-PF-PROP does, but overcoming its mentioned drawbacks, being applicable to a wide family of measurement models, and outperforming C-PF-PROP.

¹The estimation of $\hat{\mathbf{x}}_l$ and $\hat{\mathbf{C}}_l^k$ is slightly different in [15], where linearity of the dynamic system is assumed. We want to emphasize that the use of this type of idea is not constrained by nonlinearities, so we prefer to use current expressions in (2.127) and (2.136).

Chapter 3

Auxiliary particle filters for high-dimensional filtering

As noted in Section 2.8, the performance of common PF methods like the SIS, SIR, or the APF plummets when dealing with the problem of high-dimensional state estimation due to the curse of dimensionality. Partitioned PFs like IP, PP or the independent joint OID (IJOID) [190] deal with this issue by means of an independence assumption (see Sections 2.9 and 2.10). IJOID is the one with highest performance among these three filters due to the fact that it samples from the optimal importance density once Assumption 1 in Section 2.9.1 is adopted. However, as sampling from the OID is not always possible, IJOID cannot be applied to any measurement model. On the other hand, although PP is a general method with a higher performance than other methods of this type [50], it has two drawbacks. First, it requires evenly distributed weights as indicated by Assumption 2 in Section 2.9.1. Second, if there is only one component in the partition in (2.98) (i.e., $t = 1$), PP reduces to the SIR PF, which is usually outperformed by the APF.

Based on the above mentioned insights, in this chapter, we build on some of the techniques addressed in Chapter 2 to tackle high-dimensional filtering problems, and propose three different particle filter methods: the auxiliary PP (APP), the target-resampling APP (TRAPP), and the adaptive TRAPP (ATRAPP). It is worth noting that the context in which these methods have been developed is the tracking of a fixed and known number of targets (hence the “*target-resampling*” nomenclature of TRAPP and ATRAPP). Nevertheless, these methods can be easily employed in the problem of tracking an unknown and varying number of targets by considering their inclusion in an appropriate framework for this purpose [50]. It should nonetheless also be taken into account that the presented methods are generally applicable in high-dimensional filtering problems (with state-spaces that can be reasonably partitioned) and their

application is thus not restricted to target tracking.

The methods presented in this chapter make use of the *a priori* independence assumption in Section 2.9.1 and inherit the salient features of the PP filter while tackling its drawbacks. The designed filters should therefore:

- Independently sample each component of the partition in (2.98).
- Allow subparticle crossover.
- Take into account in the sampling procedure the effect in measurements of all the components of the partition in (2.98), like PP does using (2.112).
- Be equivalent to the APF when there is only one component in the partition in (2.98).
- Avoid the particle resampling step of IP and PP to improve particle diversity [136].

Taking this last characteristic into consideration, Assumption 2 in Section 2.9.1 can no longer be generally met so, instead of (2.107), the approximation to the posterior of the partitioned filter becomes (see Appendix A.2)

$$p(\mathbf{X}^k | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{X}^k) \prod_{j=1}^t \sum_{i=1}^N \omega_i^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}). \quad (3.1)$$

One can take samples from (3.1) in many different ways, and each one would define a different particle filter method. In the following, we propose three different importance sampling functions designed for this purpose. Using these three importance sampling functions, three different partitioned particle filters are derived which satisfactorily achieve the above desired features and which are shown to outperform both the IP and PP methods in a simulation with an MTT scenario in Section 3.5. Two comparative tables (Tables 3.1 and 3.2) are provided which summarize the importance sampling functions of IP, PP, TRAPP and APP and their salient features, so that the differences in the target sampling procedure for each filter can be appreciated.

3.1 Auxiliary parallel partition particle filter

As stated before, it is of interest to perform subparticle crossover. APP achieves this goal obtaining samples from the posterior in (3.1) by making use of an auxiliary

variable [130] for each target, therefore sampling in a higher dimension from

$$p(\mathbf{X}^k, \mathbf{a} | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{X}^k) \prod_{j=1}^t \omega_{a_j}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}) \quad (3.2)$$

where $\mathbf{a} = [a_1, a_2, \dots, a_t]^T \in \{1, 2, \dots, N\}^t$, and a_j is an index in the mixture of the j -th target in (3.1). The selection of the auxiliary variable is sound because $p(\mathbf{X}^{k+1}, \mathbf{a} | \mathbf{z}^{1:k+1}) \geq 0$ and by integrating (3.2) over $\mathbf{a}$, one gets the posterior. As previously outlined in Section 2.9.1, subparticle crossover is a desirable property in such filters, as it allows the formation of a particle at time k from the propagation of subparticles that belonged to different particles at time $k-1$. We would like to recall from Chapter 2 that in the prediction and update steps from time $k-1$ to time k , we refer to particles and subparticles at time $k-1$ as *parent* particles and subparticles, while particles and subparticles at time k are referred to as *child* particles and subparticles. Thus, making use of an auxiliary variable, APP performs the crossover between *parent* subparticles. This is, particles at time k are formed based on the propagation of the *parent* subparticles that are prone to obtain *child* subparticles with higher likelihoods. The use of an auxiliary variable not only enables subparticle crossover in APP, but also decreases the computational burden (note that the Chapman-Kolmogorov integral in the prediction step is circumvented).

Samples are produced from an importance density function. In the case of APP, the importance density function samples targets independently and is selected as

$$q^A(\mathbf{X}^k, \mathbf{a} | \mathbf{z}^{1:k}) = \prod_{j=1}^t q_j^A(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \quad (3.3)$$

$$q_j^A(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto b_j(\boldsymbol{\mu}_{j,a_j}^k) \omega_{a_j}^k p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}) \quad (3.4)$$

Similarly to the APF in Section 2.7.5, by considering

$$q_j^A(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto q_j(a_j | \mathbf{z}^{1:k}) q_j(\mathbf{x}_j^k | a_j, \mathbf{z}^{1:k}) \quad (3.5)$$

and defining

$$q_j(\mathbf{x}_j^k | a_j, \mathbf{z}^{1:k}) \triangleq p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}), \quad (3.6)$$

it follows from (3.4), (3.5) and (3.6) that

$$\lambda_{j,a_j} \triangleq q_j(a_j | \mathbf{z}^{1:k}) \propto b_j(\boldsymbol{\mu}_{j,a_j}^k) \omega_{a_j}^{k-1} \quad (3.7)$$

where λ_{j,a_j} are the first-stage weights of the APP filter, with $\boldsymbol{\mu}_{j,a_j}^k$ being some

Figure 3.1: APP sampling procedure (I). The scenario depicts the prediction and update steps in a multiple target tracking scenario with 2 targets. The state is therefore $\mathbf{X}^k = \left[(\mathbf{x}_1^k)^T, (\mathbf{x}_2^k)^T \right]^T \in \mathbb{R}^4$, (thus, the state is partitioned into two components, $t = 2$), where $\mathbf{x}_j^k = [x_j^k, y_j^k]^T$ is the position of the j -th target at time k . (a) State of the targets at time $k-1$ (triangle and square) and the particle approximation, $\mathbf{x}_{j,i}^{k-1}$ $i = 1, \dots, N$, to their posterior PDF (in green). (b) At time k the 2 targets have moved to other positions. APP draws for every particle of the posterior PDF at time $k-1$ a characterization of the state at the next time step $\boldsymbol{\mu}_{j,i}^k$ (in blue) which is obtained according to the dynamic model.

characterization of $\mathbf{x}_j^k$ given $\mathbf{x}_{j,a_j}^{k-1}$, such as the predicted mean, $\boldsymbol{\mu}_{j,a_j}^k = \mathbb{E} \left[\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1} \right]$, or a sample, $\boldsymbol{\mu}_{j,a_j}^k \sim p \left(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1} \right)$.

Making use of (3.2) and (3.3), the updated weights are obtained according to the principle of importance sampling as

$$\begin{aligned}
 \omega_i^k &= \frac{p(\mathbf{X}_i^k, \mathbf{a}_i | \mathbf{z}^{1:k})}{q^A(\mathbf{X}_i^k, \mathbf{a}_i | \mathbf{z}^{1:k})} \\
 &\propto \frac{p(\mathbf{z}^k | \mathbf{X}_i^k) \prod_{j=1}^t \omega_{a_j^i}^{k-1} p(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,a_j^i}^{k-1})}{\prod_{j=1}^t b_j(\boldsymbol{\mu}_{j,a_j^i}^k) \omega_{a_j^i}^{k-1} p(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,a_j^i}^{k-1})} \\
 &= \frac{p(\mathbf{z}^k | \mathbf{X}_i^k)}{\prod_{j=1}^t b_j(\boldsymbol{\mu}_{j,a_j^i}^k)}.
 \end{aligned} \tag{3.8}$$

Figure 3.2: APP sampling procedure (II). Sequence follows from Figure 3.1. (a) Measurement $\mathbf{z}^k$ arrives, and $b_j(\boldsymbol{\mu}_{j,i}^k)$ is computed for $i = 1, \dots, N$, taking into account all nearby targets, yielding higher $b_j(\boldsymbol{\mu}_{j,i}^k)$ (with a hotter color in the figure) for those $\boldsymbol{\mu}_{j,i}^k$ that are closer to the state of the target at time k . (b) An index a_j is sampled for every j , according to the distribution of first-stage weights $\lambda_{j,i} = \omega_i^k b_j(\boldsymbol{\mu}_{j,i}^k)$, figure shows in red those $\boldsymbol{\mu}_{j,a_j}^k$ associated with the obtained samples a_j^i . Note that some $\boldsymbol{\mu}_{j,i}^k$ might be repeated in the set of $\boldsymbol{\mu}_{j,a_j}^k$ $i = 1, \dots, N$.

Note that if there is only one component in the partition in (2.98), i.e., $t = 1$, APP is in fact the conventional auxiliary particle filter [130]. Under the conditions stated in [130], i.e., informative measurements, the use of this method to obtain the samples from the importance sampling function allows for the reduction of the variance of the first-stage weights. Thus, APP obtains a better approximation than PP [50] of the posterior PDF for a fixed number of samples. The sequence of Figures 3.1 to 3.3 show the way particles are produced by the APP method. A scheme which illustrates the crossover of subparticles in APP is provided in Figure 3.4. A pseudocode of the APP method is given in Algorithm 3.1.

It should be noted that as a consequence of the definition of $b_j(\cdot)$, as PP, the APP method can also always be applied to approximate the prediction (2.4) and update (2.5) equations with any given dynamic and measurement models, so that it can be satisfactorily applied to any high-dimensional Bayesian filtering problem under the assumption of *a priori* independence.

Figure 3.3: APP sampling procedure (III). Sequence follows from Figure 3.2. (a) The subparticles $\mathbf{x}_{j,a_j^i}^{k-1}$ at time $k-1$ associated to a sample a_j^i $i = 1, \dots, N$ are shown in red. Note that some of the subparticles $\mathbf{x}_{j,a_j^i}^{k-1}$ might be repeated (b) The new set of subparticles $\mathbf{x}_{j,a_j^i}^{k-1}$ is propagated to time k (shown in green) using $p\left(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,a_j^i}^{k-1}\right)$. Note that this time the predicted density is more concentrated around the true state of the targets than in the initial propagation of subparticles $\mathbf{x}_{j,i}^{k-1}$ shown in Figure 3.1-(b).

Figure 3.4: Subparticle crossover in APP for $t = 3$ using two particles. Once $\mathbf{z}^k$ arrives, the first-stage weights $\lambda_{j,i} = b_j(\mu_{j,i}^k) \omega_i^{k-1}$ for $i = 1, \dots, N$ are computed (represented with a hotter color for higher first-stage weight). Those subparticles at time $k-1$ showing higher first-stage weights are selected (using the sampling of the auxiliary variable a_j according to the distribution of first-stage weights). This new set of particles are propagated to time k and weights can be computed according to (3.8). Note that particles at time k can be formed from propagated subparticles that belonged to different particles at time $k-1$, so subparticle crossover is achieved.

Algorithm 3.1 APP procedure

```

- Initialize particles according to  $p(\mathbf{X}^0)$  with uniform weights,  $\omega_i^0 = \frac{1}{N}$ 
for  $k = 1, \dots, k_{end}$  do ▷  $k_{end}$  is the final time step
    - Collect new measurements  $\mathbf{z}^k$  from sensors
    for  $j = 1, \dots, t$  do
        for  $i = 1, \dots, N$  do
            - Draw  $\boldsymbol{\mu}_{j,i}^k$  using  $p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1})$  or  $E[\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}]$ 
        end for
        - Compute  $\hat{\mathbf{x}}_j^k = \sum_{i=1}^N \omega_i^{k-1} \cdot \boldsymbol{\mu}_{j,i}^k$ 
    end for
    for  $j = 1, \dots, t$  do ▷ Auxiliary filtering
        for  $i = 1, \dots, N$  do
            - Compute  $b_j(\boldsymbol{\mu}_{j,i}^k) = p(\mathbf{z}^k | \hat{\mathbf{X}}_{-\{j\}}^k, \boldsymbol{\mu}_{j,i}^k)$ 
            - Compute  $\lambda_{j,i} = b_j(\boldsymbol{\mu}_{j,i}^k) \omega_i^{k-1}$ 
        end for
        for  $i = 1, \dots, N$  do
            - Normalize  $\lambda_{j,i} = \frac{\lambda_{j,i}}{\sum_{i=1}^N \lambda_{j,i}}$ 
        end for
        for  $i = 1, \dots, N$  do
            - Sample an index  $a_j^i$  from the distribution
              defined by  $(\lambda_{j,1}, \lambda_{j,2}, \dots, \lambda_{j,N})$ 
            - Draw a sample  $\mathbf{x}_{j,i}^k$  using  $p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j^i}^{k-1})$ 
        end for
    end for
    for  $i = 1, \dots, N$  do ▷ Weight update
        - Compute  $\omega_i^k = \frac{p(\mathbf{z}^k | \mathbf{X}_i^k)}{\prod_{j=1}^t b_j(\boldsymbol{\mu}_{j,a_j^i}^k)}$ 
    end for
    for  $i = 1, \dots, N$  do
        - Normalize  $\omega_i^k = \frac{\omega_i^k}{\sum_{i=1}^N \omega_i^k}$ 
    end for
end for

```

3.2 Target-resampling auxiliary parallel partition particle filter

In this section we present the TRAPP PF. The main difference with APP is that TRAPP uses a subparticle resampling step for each component of the partition in (2.98), which we have coined as a target-resampling¹ stage which is useful in some situations, as will be analyzed in Section 3.3.

As happened for APP, TRAPP also makes use of the *a priori* independence assumption in Section 2.9.1 and individually samples the state of each component of the partition in (2.98). Subparticle crossover is again desired, so TRAPP also makes use of an auxiliary variable for each target, therefore sampling in a higher dimension from (3.2). The importance sampling function proposed for the TRAPP method, which is denoted as $q^T(\mathbf{X}^k, \mathbf{a}|\mathbf{z}^{1:k})$ takes the form

$$q^T(\mathbf{X}^k, \mathbf{a}|\mathbf{z}^{1:k}) = \prod_{j=1}^t q_j^T(\mathbf{x}_j^k, a_j|\mathbf{z}^{1:k}) \quad (3.9)$$

$$q_j^T(\mathbf{x}_j^k, a_j|\mathbf{z}^{1:k}) \propto b_j(\mathbf{x}_j^k) \omega_{a_j}^{k-1} p(\mathbf{x}_j^k|\mathbf{x}_{j,a_j}^{k-1}), \quad (3.10)$$

where we recall that $b_j(\mathbf{x}_j^k)$ is given by (2.112) to account for nearby targets. Making use of (3.2) and (3.10), the updated weights are obtained according to the principle of importance sampling as

$$\begin{aligned} \omega_i^k &= \frac{p(\mathbf{X}_i^k, \mathbf{a}_i|\mathbf{z}^{1:k})}{q^T(\mathbf{X}_i^k, \mathbf{a}_i|\mathbf{z}^{1:k})} \\ &\propto \frac{p(\mathbf{z}^k|\mathbf{X}_i^k) \prod_{j=1}^t \omega_{a_j^i}^{k-1} p(\mathbf{x}_{j,i}^k|\mathbf{x}_{j,a_j^i}^{k-1})}{\prod_{j=1}^t b_j(\mathbf{x}_{j,i}^k) \omega_{a_j^i}^{k-1} p(\mathbf{x}_{j,i}^k|\mathbf{x}_{j,a_j^i}^{k-1})} \\ &= \frac{p(\mathbf{z}^k|\mathbf{X}_i^k)}{\prod_{j=1}^t b_j(\mathbf{x}_{j,i}^k)}. \end{aligned} \quad (3.11)$$

However, as happened for the IP and PP filters (see [50]), it is not generally feasible to sample directly from (3.10). TRAPP therefore makes use of the sampling/importance

¹Bear in mind that, although in this thesis we wish to emphasize the applicability of these methods to any high-dimensional filtering problem, these filters have been introduced in [164] for the application of multiple target tracking.

Figure 3.5: TRAPP sampling procedure (I). Sequence follows from Figure 3.3-(a). Recall that the first steps of the TRAPP sampling procedure are the same as the APP. (a) A new set of N subparticles $\mathbf{x}_{j,i}^*$ (shown in blue) with associated *parent* index $a_{j,i}^* = a_j^i$ are drawn using $p\left(\mathbf{x}_j^k | \mathbf{x}_{j,a_j^i}^{k-1}\right)$. Note that due to the use of the auxiliary variable a_j , samples $\mathbf{x}_{j,a_j^i}^*$ are distributed according to the first-pass approximation in (3.12). (b) The quotient $r\left(\mathbf{x}_{j,i}^*, a_{j,i}^*\right)$ in (3.13) is computed for each subparticle $\mathbf{x}_{j,i}^*$, those subparticles with a higher quotient are shown with a hotter color.

resampling method in [8] to take samples from (3.10). Samples are thus obtained from a first-pass approximation of (3.10)

$$\begin{aligned} h_j\left(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}\right) &\propto b_j\left(\boldsymbol{\mu}_{j,a_j}^k\right) \omega_{a_j}^{k-1} p\left(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}\right) \\ &= q_j^A\left(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}\right) \end{aligned} \quad (3.12)$$

where, as before, $\boldsymbol{\mu}_{j,a_j}^k$ is some characterization of $\mathbf{x}_j^k$ given $\mathbf{x}_{j,a_j}^{k-1}$ such as the predicted mean or a sample. A set of M samples from $(\mathbf{x}_j^*, a_j^*)$ are drawn from this first-pass approximation (i.e., using the importance sampling density of APP to draw samples) and the following quotient is computed

$$\begin{aligned} r\left(\mathbf{x}_{j,p}^*, a_{j,p}^*\right) &\propto \frac{q_j^T\left(\mathbf{x}_{j,p}^*, a_{j,p}^* | \mathbf{z}^{1:k}\right)}{h_j\left(\mathbf{x}_{j,p}^*, a_{j,p}^* | \mathbf{z}^{1:k}\right)} \\ &= \frac{q_j^T\left(\mathbf{x}_{j,p}^*, a_{j,p}^* | \mathbf{z}^{1:k}\right)}{q_j^A\left(\mathbf{x}_{j,p}^*, a_{j,p}^* | \mathbf{z}^{1:k}\right)} \\ &= \frac{b_j\left(\mathbf{x}_{j,p}^*\right)}{b_j\left(\boldsymbol{\mu}_{j,p}^*\right)} \end{aligned} \quad (3.13)$$

Figure 3.6: Sequence follows from Figure 3.5. (a) For each subparticle i of component j an index p is sampled, according to the distribution defined by $r(\mathbf{x}_{j,l}^*, a_{j,l}^*)$ $l = 1, \dots, N$, where those $\mathbf{x}_{j,p}^*$ associated with sampled index p are shown in purple. (b) The new set of subparticles at time k is finally obtained by setting $\mathbf{x}_{j,i}^k = \mathbf{x}_{j,p}^*$ and the *parent* index $a_j^i = p$. Note that this time the predicted density is more concentrated around the true state of the targets than the one in Figures 3.1-(b) or 3.3-(b). Particle weights would finally be computed using (3.11).

Table 3.1: Feature comparison between IP, PP, APP and TRAPP algorithms

MTT filter	auxiliary filtering in target sampling	target-resampling	accounts for nearby targets	avoids particle resampling
IP	×	✓	×	×
PP	×	✓	✓	×
APP	✓	×	✓	✓
TRAPP	✓	✓	✓	✓

Table 3.2: Importance sampling function of the j -th component of the partitioned state for IP, PP, APP and TRAPP algorithms

Filter	importance sampling function
IP and PP	$q_j(\mathbf{x}_j^{k+1}, a_j \mathbf{z}^{1:k+1}) \propto b_j(\mathbf{x}_j^{k+1}) p(\mathbf{x}_j^{k+1} \mathbf{x}_{j,a_j}^k)$
APP	$q_j(\mathbf{x}_j^{k+1}, a_j \mathbf{z}^{1:k+1}) \propto b_j(\boldsymbol{\mu}_{j,a_j}^{k+1}) \omega_{a_j}^k p(\mathbf{x}_j^{k+1} \mathbf{x}_{j,a_j}^k)$
TRAPP	$q_j(\mathbf{x}_j^{k+1}, a_j \mathbf{z}^{1:k+1}) \propto b_j(\mathbf{x}_j^{k+1}) \omega_{a_j}^k p(\mathbf{x}_j^{k+1} \mathbf{x}_{j,a_j}^k)$

in order to draw N samples of $q_j^T(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k})$ from the distribution defined by $(\mathbf{x}_{j,p}^*, a_{j,p}^*)$, for $p = 1, 2, \dots, M$ (it is common to set $M = N$) with probability $r(\mathbf{x}_{j,p}^*, a_{j,p}^*)$. The drawn samples for each component are then introduced in (3.11) to compute the particle weights.

Hence, the TRAPP method selects for every component of the partitioned state the *parent* subparticles at time $k - 1$ that are prone to generate *child* subparticles with a higher likelihood given the measurement at time k based on the auxiliary filtering step in equation (3.12) [130]. This causes the crossover of *parent* subparticles as in APP. Then, TRAPP performs too a crossover of the *child* subparticles. That is, TRAPP selects for every component of the partitioned state those subparticles $\mathbf{x}_{j,p}^*$ that show a higher likelihood with the measurement at time k , performing a resampling step for each component of the partition using (3.13). Finally the particle weights are computed using (3.11).

Note again the clear resemblance of the described procedure for drawing samples from (3.10) to the APF. The TRAPP PF is a generalization of the APF for multiple target tracking. Note that for $t = 1$ TRAPP is in fact the same filter presented in [130] with a final resampling stage. The sequence of Figures 3.1 to 3.6 shows the way particles are produced by the TRAPP method (note that the first steps of the algorithm are common to those of the APP filter). A scheme which illustrates the crossover of subparticles in TRAPP is provided in Figure 3.7. A pseudocode of the TRAPP method is pointed out in Algorithm 3.2.

With the aim of clarifying the details and differences of TRAPP, APP, PP [50] and IP [92,128] and how they perform subparticle crossover, Table 3.1 indicates the different properties of these algorithms, while Table 3.2 contains their different importance sampling functions.

Figure 3.7: Subparticle crossover in TRAPP for $t = 3$ using two particles. Once $\mathbf{z}^k$ arrives, the first stage-weights $\lambda_{j,i} = b_j(\mu_{j,i}^k) \omega_i^{k-1}$ for $i = 1, \dots, N$ are computed (represented with a hotter color for higher first-stage weight). Those subparticles at time $k-1$ showing higher first-stage weights, are selected (using the sampling of the auxiliary variable a_j according to the distribution of first-stage weights) and propagated to $\mathbf{x}_{j,i}^*$. For these new set of subparticles $\mathbf{x}_{j,i}^*$, the quotient $r(\mathbf{x}_{j,p}^*, a_{j,p}^*)$ in (3.13) (denoted for the sake of brevity in this scheme as $r_{j,p}$) is now computed and particles at time k are formed with those subparticles in $\mathbf{x}_{j,i}^*$ that show a highest quotient $r_{j,p}$ (represented also with a hotter color for higher $r_{j,p}$). Particles at time k can therefore be formed from propagated subparticles that belonged to different particles at time $k-1$, so that subparticle crossover is achieved.

Algorithm 3.2 TRAPP procedure

- Initialize particles according to $p(\mathbf{X}^0)$ with uniform weights, $\omega_i^0 = \frac{1}{N}$

for $k = 1, \dots, k_{end}$ **do** ▷ k_{end} is the final time step

- Collect new measurements $\mathbf{z}^k$ from sensors

for $j = 1, \dots, t$ **do**

for $i = 1, \dots, N$ **do**

- Draw $\boldsymbol{\mu}_{j,i}^k$ using $p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1})$ or $E[\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}]$

end for

- Compute $\hat{\mathbf{x}}_j^k = \sum_{i=1}^N \omega_i^{k-1} \cdot \boldsymbol{\mu}_{j,i}^k$

end for

for $j = 1, \dots, t$ **do** ▷ Auxiliary filtering

for $i = 1, \dots, N$ **do**

- Compute $b_j(\boldsymbol{\mu}_{j,i}^k) = p(\mathbf{z}^k | \hat{\mathbf{X}}_{-\{j\}}^k, \boldsymbol{\mu}_{j,i}^k)$

- Compute $\lambda_{j,i} = b_j(\boldsymbol{\mu}_{j,i}^k) \omega_i^{k-1}$

end for

for $i = 1, \dots, N$ **do**

- Normalize $\lambda_{j,i} = \frac{\lambda_{j,i}}{\sum_{i=1}^N \lambda_{j,i}}$

end for

for $i = 1, \dots, N$ **do**

- Sample an index a_j^i from the distribution defined by $(\lambda_{j,1}, \lambda_{j,2}, \dots, \lambda_{j,N})$

- Draw a sample $\mathbf{x}_{j,a_j^i}^*$ using $p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j^i}^{k-1})$

- Compute $b_j(\mathbf{x}_{j,a_j^i}^*) = p(\mathbf{z}^k | \hat{\mathbf{X}}_{-\{j\}}^k, \mathbf{x}_{j,a_j^i}^*)$

- Compute $r_j(\mathbf{x}_{j,a_j^i}^*) = \frac{b_j(\mathbf{x}_{j,a_j^i}^*)}{b_j(\boldsymbol{\mu}_{j,a_j^i}^k)}$

end for

for $i = 1, \dots, N$ **do**

- Normalize $r_j(\mathbf{x}_{j,a_j^i}^*) = \frac{r_j(\mathbf{x}_{j,a_j^i}^*)}{\sum_{i=1}^N r_j(\mathbf{x}_{j,a_j^i}^*)}$

end for

for $i = 1, \dots, N$ **do** ▷ target-resampling

- Sample an index p from the distribution defined by $(r_j(\mathbf{x}_{j,a_j^1}^*), r_j(\mathbf{x}_{j,a_j^2}^*), \dots, r_j(\mathbf{x}_{j,a_j^N}^*))$

- Set $\mathbf{x}_{j,i}^k = \mathbf{x}_{j,a_j^p}^*$

end for

end for

for $i = 1, \dots, N$ **do** ▷ Weight Update

- Compute $\omega_i^k = \frac{p(\mathbf{z}^k | \mathbf{X}_i^k)}{\prod_{j=1}^t b_j(\mathbf{x}_{j,i}^k)}$

end for

for $i = 1, \dots, N$ **do**

- Normalize $\omega_i^k = \frac{\omega_i^k}{\sum_{i=1}^N \omega_i^k}$

end for

end for

3.3 On target-resampling

In this section, we motivate the possible benefits of using a target-resampling step, as in TRAPP, compared to not using it, as in APP. In general, as explained in Section 2.7.2, resampling is used in particle filters to avoid particle degeneracy that occurs over time [44, 106]. Without resampling, the variance of the particle weights increases until there is only one particle with a non-negligible weight. Clearly, this is not an accurate representation of the posterior PDF. The underlying cause is that the particle filter is in fact approximating the posterior PDF over the state trajectory $\mathbf{X}^{1:k}$, which implies that the dimension of the state grows at each time step [86]. This degeneracy of particle weights is therefore associated to an increase in the dimension of the state under consideration whose effects can be mitigated by resampling.

In partitioned filters, the dimensionality of the state space being sampled also depends on the number of components t in the partition in (2.98) of the state being filtered. In this sense, particle degeneracy arises in the sampling of high-dimensional space-time states, which can be dealt with target-resampling. In this thesis, we use the concept of target-resampling as one technique to overcome degeneracy in partitioned filters, either this being a consequence of the number t of components or of the number of steps in the filtering procedure.

It also should be taken into account that, as explained in Section 2.7.2.1, resampling produces a sample-impoverishment side-effect [7], meaning that there will be a loss of diversity in the resulting resampled approximation to the PDF, as it will contain repeated samples. We find it useful to illustrate these insights with the following example grounded on an MTT application.

Example 1

A different number of targets ranging from $t = 1$ to $t = 40$ are considered. In each simulation, targets are arranged in a straight line so that for $j = 1, \dots, t$, the j -th target position is given by $\mathbf{p}_j = [j \cdot d_t, 0]^T$, with $d_t = 10\text{m}$. A total of $N_s = 40$ sensors are displayed in the scenario at positions $\mathbf{s}_i = [i \cdot d_t, 12]^T$, where $i = 1, \dots, 40$ and measurements are taken according to the nonlinear measurement model described in Section 3.5.2, using a measurement noise variance of $\sigma_n^2 = 2$. Figure 3.8(a) shows the proposed scenario. To simplify the experiment, we only consider an update step; the a priori position density function for each target, $\mathcal{N}(\mathbf{p}_j; \bar{\mathbf{p}}_j \Sigma)$, is a Gaussian PDF with mean $\bar{\mathbf{p}}_j$ corresponding to the real target position, and covariance matrix $\Sigma = \sigma^2 I_2$, where $\mathbf{I}_n$ is the $n \times n$ identity matrix and $\sigma = 1\text{m}$. In Figure 3.8(b), we show

the optimal sub-pattern assignment [145] (OSPA) position error (with parameters $p = 2$, $c = 10\text{m}$) for a different number of targets averaged in a Monte Carlo simulation with $n = 10000$ runs. It can be observed that if the dimension of the state space is high enough (9 targets in this case), the target-resampling stage makes it possible for TRAPP to achieve a lower error than APP. On the other hand, for a low number of targets APP outperforms TRAPP. This is to be expected as the dimensionality of the state is lower so target-resampling is counterproductive.

In order to clarify this, we also show in Figure 3.8(c) the normalized OSPA position errors with respect to the worst performing filter. In this simple example, APP obtains an error up to a 14% lower than TRAPP for a low number of targets, while TRAPP obtains an error of almost an 11% lower than APP does when considering a high enough number of targets. Note that, for the sake of simplicity, this example uses only a one-step update. However, if we use these algorithms in a normal filtering set-up with a recursive approximation of the filtering PDF, errors can propagate over time. This means that even small differences in the errors in one update can result in bigger differences as time goes on, as shown in the simulation examples of Section 3.5.

In this section we have analyzed target-resampling as a possibly good feature for a high-dimensional partitioned PF. However, performing an additional resampling for the whole particles, which we call particle-resampling (see Table 3.1), as done in IP and PP, is generally detrimental as we would be resampling particles whose subparticles have already been resampled. This double resampling generally implies a loss of particle diversity that can be avoided with the algorithms presented in this chapter.

3.4 Adaptive target-resampling auxiliary parallel partition particle filter

In this section, the adaptive TRAPP particle filter is developed. ATRAPP aims to combine the best features of both APP and TRAPP PFs in a single algorithm, where the decision of performing target-resampling is taken independently for each component of the partition.

A common approach to decide whether to resample in PFs is the monitoring of the effective sample size [88,136], which measures the degeneracy of the subparticle weights, in order to perform a resampling stage whenever it falls below a given threshold.

As previously stated, the target-resampling stage in TRAPP is achieved by taking samples from the distribution defined by $(\mathbf{x}_{j,p}^*, a_{j,p}^*)$, for $p = 1, 2, \dots, M$ with the normalized quotient $r(\mathbf{x}_{j,p}^*, a_{j,p}^*)$ in (3.13). ATRAPP measures subparticle degeneracy

(a)

(b)

(c)

Figure 3.8: (a) Proposed scenario for Example 3.3 with up to $t = 40$ targets. (b) OSPA position error as a function of the number of targets for the APP and TRAPP algorithms for one update step. (c) Normalized OSPA position error with respect to the worst performing filter in each case. It is shown how the target-resampling stage of TRAPP is advisable for a high enough number of targets, while it is not for a low number of targets.

using this distribution (with $M = N$), obtaining the effective component sample size at this stage, N_{eff}^j , to decide if target-resampling is really advisable in the component. One cannot generally exactly evaluate N_{eff}^j , however a commonly accepted estimate [44] of the effective component sample size is given by

$$\hat{N}_{eff}^j = \frac{1}{\sum_{p=1}^N r^2(\mathbf{x}_{j,p}^*, a_{j,p}^*)} \quad (3.14)$$

where $\hat{N}_{eff}^j$ takes values from 1 to N , with these extreme values being achieved respectively when total degeneracy or no degeneracy at all occurs. If this estimate of the effective component sample size drops below a threshold, target-resampling is performed as in TRAPP, avoiding the subparticle degeneracy that would lead the whole filter to an undesired global particle degeneracy. On the other hand, if $\hat{N}_{eff}^j$ is high enough, no target-resampling is necessary. Thus, noting that the first-pass approximation in (3.12) is in fact the importance sampling density used for the APP method in (3.3), the absence of the target-resampling stage makes the sampling procedure equivalent to the one in APP, so that in these cases, only a reindexing step is necessary, setting $\mathbf{x}_{j,i}^k = \mathbf{x}_{j,a_j^i}^*$, for $i = 1, 2, \dots, N$.

The importance sampling function for ATRAPP is therefore defined as

$$q^{AT}(\mathbf{X}^k, \mathbf{a} | \mathbf{z}^{1:k}) = \prod_{j=1}^t q_j^{AT}(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \quad (3.15)$$

$$q_j^{AT}(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto \begin{cases} q_j^T(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}); & \hat{N}_{eff}^j < \Gamma_j^k \\ q_j^A(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}); & \hat{N}_{eff}^j \geq \Gamma_j^k \end{cases} \quad (3.16)$$

where Γ_j^k is the target-resampling threshold for the j -th component of the state at time step k . Given the set of indices of the different components of the state at time k , $S_t^k = \{1, 2, \dots, t\}$, we define a partition over S_t^k as

$$S_t^k = S_{APP}^k \cup S_{TRAPP}^k \quad (3.17)$$

$$\hat{N}_{eff}^j \leq \Gamma_j^k \Rightarrow j \in S_{TRAPP}^k \quad (3.18)$$

$$\hat{N}_{eff}^j > \Gamma_j^k \Rightarrow j \in S_{APP}^k. \quad (3.19)$$

This importance sampling function yields the weight update equation for ATRAPP

$$\omega_i^k \propto \frac{p(\mathbf{z}^k | \mathbf{X}_i^k)}{\prod_{j \in S_{TRAPP}^k} b_j(\mathbf{x}_{j,i}^k) \prod_{j \in S_{APP}^k} b_j(\boldsymbol{\mu}_{j,a_j^i}^k)}. \quad (3.20)$$

Algorithm 3.3 ATRAPP procedure

```

- Initialize particles according to  $p(\mathbf{X}^0)$  with uniform weights,  $\omega_i^0 = \frac{1}{N}$ 
for  $k = 1, \dots, k_{end}$   $\triangleright k_{end}$  is the final time step
    for  $i = 1, \dots, N$  do
         $\mathbf{X}_i^{k|k-1} = \mathbb{E}[\mathbf{X}_i^k | \mathbf{X}_i^{k-1}]$ 
    end for

    for  $j = 1, \dots, t$  do
         $\hat{\mathbf{x}}_j^k = \sum_{i=1}^N \omega_i^{k-1} \cdot \mathbf{x}_{j,i}^{k-1}$ 
    end for

    for  $j = 1, \dots, t$  do
         $\{(\mathbf{x}_{j,1}^k, b_{j,1}^k), \dots, (\mathbf{x}_{j,N}^k, b_{j,N}^k)\} =$ 
         $\text{target\_ATRAPP}(\mathbf{z}^k, \hat{\mathbf{X}}_{-\{j\}}^k, \{(\mathbf{x}_{j,1}^{k-1}, \omega_1^{k-1}), \dots, (\mathbf{x}_{j,N}^{k-1}, \omega_N^{k-1})\})$ 
    end for

    for  $i = 1, \dots, N$  do  $\triangleright$  Weight Update
        - Compute  $\omega_i^k = \frac{p(\mathbf{z}^k | \mathbf{X}_i^k)}{\prod_{j=1}^t b_{j,i}^k}$ 
    end for

    for  $i = 1, \dots, N$  do
        - Normalize  $\omega_i^k = \frac{\omega_i^k}{\sum_{i=1}^N \omega_i^k}$ 
    end for
end for
    
```

It should be noted that the importance sampling function (3.15) of the proposed algorithm considers a different resampling threshold for each component of the partition at each time step, Γ_j^k . This flexible definition can serve different purposes (see [164]): Adaptation to different measurement models and dynamic functions, consideration of hard-to sample regions of the state space (e.g. when targets move in close proximity in an MTT problem), or the accommodation of the method to problems in which dimensionality of the state space can vary using a framework like the one introduced in [50].

The use of the target-resampling threshold, although resulting in potential advantages, as explained, also comes with an associated difficulty, i.e., the search for the optimal thresholds for each situation, which should balance the loss in subparticle diversity that comes with target-resampling with the subparticle depletion problem that may arise if target-resampling is not applied. In this thesis, the search for the optimal

Algorithm 3.4 target_ATRAPP subroutine for the j -th target

```

procedure ( $\mathbf{z}^k, \hat{\mathbf{X}}_{-\{j\}}^k, \{(\mathbf{x}_{j,1}^{k-1}, \omega_1^{k-1}), \dots, (\mathbf{x}_{j,N}^{k-1}, \omega_N^{k-1})\}$ )

  for  $i = 1, \dots, N$  do ▷ Auxiliary filtering
    - Draw  $\boldsymbol{\mu}_{j,i}^k$  using:
      
$$p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}) \text{ or } E[\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}]$$

    - Compute  $b_j(\boldsymbol{\mu}_{j,i}^k) = p(\mathbf{z}^k | \hat{\mathbf{X}}_{-\{j\}}^k, \boldsymbol{\mu}_{j,i}^k)$ 
    - Compute  $\lambda_{j,i} = b_j(\boldsymbol{\mu}_{j,i}^k) \omega_i^{k-1}$ 
  end for

  for  $i = 1, \dots, N$  do
    - Normalize  $\lambda_{j,i} = \frac{\lambda_{j,i}}{\sum_{i=1}^N \lambda_{j,i}}$ 
  end for

  for  $i = 1, \dots, N$  do
    - Sample an index  $a_j^i$  from the distribution
      defined by  $(\lambda_{j,1}, \lambda_{j,2}, \dots, \lambda_{j,N})$ 
    - Draw a sample  $\mathbf{x}_{j,a_j^i}^*$  from  $p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j^i}^{k-1})$ 
  end for

  for  $i = 1, \dots, N$  do
    - Compute  $b_j(\mathbf{x}_{j,a_j^i}^*) = p(\mathbf{z}^k | \hat{\mathbf{X}}_{-\{j\}}^k, \mathbf{x}_{j,a_j^i}^*)$ 
    - Compute  $r_j(\mathbf{x}_{j,a_j^i}^*) = \frac{b_j(\mathbf{x}_{j,a_j^i}^*)}{b_j(\boldsymbol{\mu}_{j,a_j^i}^k)}$ 
  end for

  for  $i = 1, \dots, N$  do
    - Normalize  $r_j(\mathbf{x}_{j,a_j^i}^*) = \frac{r_j(\mathbf{x}_{j,a_j^i}^*)}{\sum_{i=1}^N r_j(\mathbf{x}_{j,a_j^i}^*)}$ 
  end for - Compute  $\hat{N}_{eff}^j = \frac{1}{\sum_{i=1}^N (r_j(\mathbf{x}_{j,a_j^i}^*))^2}$ 

  if  $\hat{N}_{eff}^j > \Gamma_j^k$  then
    for  $i = 1, \dots, N$  do
      - Set  $\mathbf{x}_{j,i}^k = \mathbf{x}_{j,a_j^i}^*$ 
      - Set  $b_{j,i} = b_j(\boldsymbol{\mu}_{j,i}^k)$ 
    end for
  else
    for  $i = 1, \dots, N$  do ▷ target-resampling
      - Sample an index  $p$  from the distribution
        defined by  $(r_j(\mathbf{x}_{j,a_j^1}^*), r_j(\mathbf{x}_{j,a_j^2}^*), \dots, r_j(\mathbf{x}_{j,a_j^N}^*))$ 
      - Set  $\mathbf{x}_{j,i}^k = \mathbf{x}_{j,a_j^p}^*$ 
      - Set  $b_{j,i} = b_j(\mathbf{x}_{j,a_j^p}^*)$ 
    end for
  end if

  return  $\{(\mathbf{x}_{j,1}^k, b_{j,1}), \dots, (\mathbf{x}_{j,N}^k, b_{j,N})\}$ 

end procedure

```

Figure 3.9: Eight simulated trajectories, consisting of 100 steps sampled with a period of $\tau = 1s$. The initial point of each trajectory is indicated by a colored circle, while the final point is represented by a colored square. An arrow shows the position of the target at steps 20, 40, 60 and 80 of each trajectory. When simulating a scenario with t targets, target trajectories 1 to t of the above presented are taken into account.

target-resampling threshold policy is not approached, although in Section 3.5.4.2 some useful insights for adequate threshold selection are given and a target-resampling policy is proposed and analyzed. A pseudocode for the ATRAPP method is given through Algorithms 3.3 and 3.4.

3.5 Experimental results

In this section, the performance of APP, TRAPP and ATRAPP is tested via a set of simulations with a demanding MTT scenario. The proposed scenario is presented along with the employed motion and measurement models. In Section 3.5.3 the performance of APP and TRAPP is analyzed, while Section 3.5.4 is devoted to the analysis of the performance of ATRAPP for different resampling threshold policies.

As a baseline reference for comparison, the performance of other filters in the literature is also considered: the IP and PP methods described in Section 2.9.1 and

the jointly auxiliary (JA) PF [118], which is the direct generalization of the APF for MTT. The JA PF jointly samples the whole state space of all the targets, therefore it accounts for nearby targets in the sampling procedure, but does never perform subparticle crossover.

The tracking performance of the different methods under consideration is evaluated by means of the mean optimal sub-pattern assignment (OSPA) position error [145] (with parameters $c = 10$ and $p = 2$), averaged over the runs of a Monte Carlo simulation with $n = 5000$ realizations. It should be noted that the purpose of the OSPA metric is to consider the tracking performance of the algorithms without taking into account the possible flipping of target labels after the crossing of targets, as OSPA performs a search for the minimum distance permutation between the target positions and the predictions (see Appendix A.9). For fixed and known number of targets, the recently proposed generalized OSPA metric [134], which penalizes missed and false targets rather than cardinality mismatch, is proportional to OSPA and therefore ranks the performances of the algorithms in the same way.

In order to assess the performance of the methods, each Monte Carlo run makes use of a different sequence of measurements according to the model described in Section 3.5.2, however, the i -th Monte Carlo run for all methods uses the same measurement sequence. This variance reduction technique [133], known as common random numbers, allows for the use of a lower number of Monte Carlo runs. This procedure has been used, when possible, in all simulations of this thesis.

3.5.1 Dynamic model

The state of the j -th target is represented by the state vector $\mathbf{x}_j^k = [x_j^k, \dot{x}_j^k, y_j^k, \dot{y}_j^k]^T$, with $\mathbf{p}_j^k = [x_j^k, y_j^k]^T$ being the target position and $\mathbf{v}_j^k = [\dot{x}_j^k, \dot{y}_j^k]^T$ its velocity. The motion of each target is independent

$$p(\mathbf{X}^k | \mathbf{X}^{k-1}) = \prod_{j=1}^t p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) \quad (3.21)$$

with a nearly constant velocity model [12]

$$p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) = \mathcal{N}(\mathbf{x}_j^k; \mathbf{F}\mathbf{x}_j^{k-1}, \mathbf{Q}) \quad (3.22)$$

$$\mathbf{F} = \mathbf{I}_2 \otimes \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix} \quad (3.23)$$

Figure 3.10: OSPA position error with respect to the number of particles when 1 target is under track with different filters: JA, IP, PP APP and TRAPP.

$$\mathbf{Q} = \sigma_u^2 \mathbf{I}_2 \otimes \begin{pmatrix} \tau^3/3 & \tau^2/2 \\ \tau^2/2 & \tau \end{pmatrix} \quad (3.24)$$

where $\mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \mathbf{Q})$ is a Gaussian PDF with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{Q}$ evaluated at $\mathbf{x}$, τ is the sampling period, $\mathbf{I}_n$ is the $n \times n$ identity matrix, $\otimes$ stands for the Kronecker product and σ_u^2 is the continuous-time process noise intensity. The PDF at time $k = 0$ of the position of the j -th target $p(\mathbf{x}_j^0)$ is initialized as a Gaussian distribution centered at the true position of the target, with covariance matrix $\mathbf{Q}$.

Simulations have been carried out with up to 8 targets in the scenario, whose trajectories repeatedly cross each other to recreate a demanding MTT problem (the dimension of the state space can therefore reach $n_x = 32$). The trajectories have been generated according to equation (3.21), taking $\tau = 1\text{s}$ and $\sigma_u = 0.1\text{m/s}^{3/2}$. The simulated target trajectories are shown in Figure 3.9.

3.5.2 Measurement model

Sensors collect measurements according to an additive received power measurement model. The nonlinear measurement equation (2.3) for the s -th sensor at time k is

$$z_s^k = h_s(\mathbf{X}^k) + v_s^k \quad (3.25)$$

(a)

(b)

Figure 3.11: OSPA position error with respect to the number of particles when 3 targets are under track with different filters: (a) JA, IP, PP APP and TRAPP. (b) Only PP, APP and TRAPP.

$$h_s(\mathbf{X}^k) = \sum_{j=1}^t h_{s,j}(\mathbf{x}_j^k) \quad (3.26)$$

$$h_{s,j}(\mathbf{x}_j^k) = \begin{cases} Z_0 & \rho_s(\mathbf{x}_j^k) \leq \rho_0 \\ Z_0 \frac{\rho_0^\alpha}{(\rho_s(\mathbf{x}_j^k))^\alpha} & \rho_s(\mathbf{x}_j^k) > \rho_0 \end{cases} \quad (3.27)$$

$$\rho_s(\mathbf{x}_j^k) \equiv \sqrt{(x_j - s_{s,x})^2 + (y_j - s_{s,y})^2} \quad (3.28)$$

where v_i^k is a zero-mean, Gaussian-distributed noise with variance σ_n^2 . Z_0 is the maximum amplitude produced by a target when it is closer to the sensor than the saturation distance, ρ_0 and α is a path-loss coefficient. The coordinates of the s -th sensor are given by its position vector, $\mathbf{s}_s = [s_{s,x}, s_{s,y}]^T$. In the simulated scenario, measurements are taken from a set of sensors displayed in a flat squared regular grid covering $120\text{m} \times 120\text{m}$, with a spacing of 10m between sensors in both axes, yielding a total of 169 sensors, Z_0 has been taken to be 20, $\alpha = 2$, $\rho_0 = 5\text{m}$ and $\sigma_n^2 = 1$.

Sensor measurements do not depend on the velocity of the targets, as they are only affected by their position. As the relationship between the velocity and the position of the targets is linear, Rao-Blackwellization (see Section 2.8.1) is applied to reduce the curse of dimensionality [144] [120], only including the position of the targets in the PF and optimally estimating their velocities by Kalman filtering.

The results of different particle filters in the above described simulation scenario are described in the following sections. In Section 3.5.3 we first consider the performance of APP and TRAPP in comparison with other particle filter algorithms in the literature. Section 3.5.4 analyzes the performance of ATRAPP for different resampling threshold policies.

3.5.3 APP and TRAPP simulation results

In this section, the performance of the APP and TRAPP methods introduced in Sections 3.1 and 3.2 of this chapter is analyzed in the scenario described in Sections 3.5.1 and 3.5.2.

Figures 3.10 to 3.13 show the results obtained in a scenario with respectively 1, 3, 6 and 8 targets in the scenario. It can be observed in Figure 3.10 that JA and APP PFs exhibit the same performance when there is only one target in the scenario, as they are both equivalent to the APF. The IP and PP PFs performance in this case is notoriously worse because they boil down to the SIR PF when dealing with only one target. The TRAPP PF performance is in between, as it takes advantage of auxiliary filtering but its target-resampling stage pulls down its performance in this case.

(a)

(b)

Figure 3.12: OSPA position error with respect to the number of particles when 6 targets are under track with different filters: (a) JA, IP, PP APP and TRAPP. (b) Only PP, APP and TRAPP.

(a)

(b)

Figure 3.13: OSPA position error with respect to the number of particles when 8 targets are under track with different filters:(a) JA, IP, PP APP and TRAPP. (b) Only PP, APP and TRAPP.

As the number of targets in the scenario becomes bigger in Figures 3.11 to 3.13, the dimension of the state space increases, and JA acutely suffers from the curse of dimensionality as it samples in a very high dimension, which is clearly very inefficient. When there are more than one target in the scenario, if target trajectories cross each other, the performance of IP is severely degraded, as it does not take into account neighboring targets in the sampling procedure. On the other hand, PP, APP and TRAPP are satisfactorily able to deal with dimensionality when tracking 3, 6 and 8 targets, although the use of parent subparticle crossover allows APP and TRAPP to clearly outperform PP.

Figure 3.14 points out how the studied filters deal with the growth of dimensionality of the state space. It is shown how partitioned filters like PP, APP and TRAPP can manage the curse of dimensionality in a better way than JA, also confirming that APP and TRAPP have a better performance than the PP filter. It is worth noting that, in accordance with the insights on target-resampling considered in Section 3.3, the inclusion of this technique within the TRAPP filter allows it to outperform APP when the number of targets becomes higher, in this case making its use advisable when there are 4 or more targets in the scenario. It is worth noting that although the error apparently increases for all the algorithms when a higher number of targets is considered, this trend is not applicable when the 8th target is incorporated in the scenario. Note that the trajectory of this target in Figure 3.9 does not get close to targets 1 to 6. This behavior thus indicates that the increasing error in Figure 3.14 is not due to the algorithms having great difficulties with dimensionality on itself but is instead (to a great extent) caused by the difficult situation arising from the repeatedly crossing of targets 1 to 7 in the center of the scenario.

Although the shift in performance of APP and TRAPP when the dimension of the state space becomes larger is a general trend (in accordance with the concepts explained in Section 3.3), the specific dimension from which it occurs is nevertheless dependent on different aspects of the problem at hand, namely the dynamic and measurement models or the trajectories of the targets of interest. Thus, depending on the specifics of the scenario, the use of APP over TRAPP can be preferable even if a large number of targets is considered. In order to illustrate this point, Figure 3.15 shows the performance of the different partitioned PFs when the variance of the measurement noise in (3.25) larger, with $\sigma_n^2 = 2$. In this case, comparing with Figure 3.12, it is shown how the performance of the different methods generally goes down as expected, while in this case the use of the APP filter is still preferred with 6 targets.

It is important to note that the computational requirements of the APP and TRAPP methods grows both linearly with the number of particles and with the number of

(a)

(b)

Figure 3.14: OSPA position error with respect to the number of targets in the scenario using 300 particles with the different filters in this section. (a) OSPA position error of all filters (b) OSPA position error of PP, APP and TRAPP.

(a)

(b)

Figure 3.15: OSPA position error with respect to the number of particles when 6 targets are under track with IP, APP and APP PFs, and larger measurement noise variance, $\sigma_n^2 = 2$. (a) JA, IP, PP APP and TRAPP. (b) Only PP, APP and TRAPP

Figure 3.16: OSPA position error with respect to the execution time (a higher number of particles causes the execution time to rise) in a scenario with 6 targets. The execution time includes the estimation of the complete trajectory of the targets (100 time steps).

targets, as is the case for the IP and PP filters. Nevertheless, the use of auxiliary filtering in APP and TRAPP requires to draw $\mu_{j,a_j^i}^k$ and compute $b_j(\mu_{j,a_j^i}^k)$ for every particle, an step which is not required by PP and IP, so that its computational complexity is slightly higher. Figure 3.16 shows the OSPA position error of the different methods with respect to their execution time in a scenario with 6 targets. It can be observed how the additional computational load of APP and in this case specifically of TRAPP is worth. In addition, although both JA and IP have a very low execution time, their high error clearly does not make their use advisable. The considered execution times correspond to a MATLAB implementation of the algorithms on an Intel Core i7 computer. Nevertheless, it should be noticed that the results in this kind of analysis can appreciably vary depending on the implementation of the algorithms and the computing platform, among other factors.

3.5.4 ATRAPP simulation results

3.5.4.1 Constant threshold

A first approach to the capabilities of the ATRAPP PF is now considered, with the target-resampling threshold Γ_j^k set constant through all targets and all time steps, thus in this context $\Gamma_j^k = \Gamma$. This approach has some advantages, first, it can be used to analyze the equivalence of the ATRAPP PF with respect to APP and TRAPP PFs when $\Gamma = 0$ and $\Gamma = N$, respectively. In addition, the constant thresholding strategy has the advantage of needing the adjustment of only one parameter, which simplifies the filter design. This parameter is set in advance so that no additional computational expense is required.

In the previous section it has been shown how APP and TRAPP respectively obtain a better performance depending on the dynamic and measurement models, and the number of targets in the scenario (see also Figure 3.8 and [163]). Note how the best performing method between APP and TRAPP switches when considering 2 targets in Figure 3.17(a) with respect to the results obtained when considering 8 targets in the scenario in Figure 3.17(b). This is in accordance with the theoretical insights previously explained in Section 3.3, showing that when the dimensionality of the state space grows, the target-resampling stage is advised.

It would be desirable that a unique thresholding policy in ATRAPP served to reach the performances of both APP and TRAPP PFs regardless of the situation. This may not be generally attainable, however, it is shown in the following how a simple thresholding policy for ATRAPP can bring this objective closer. Figure 3.17 illustrates how in the proposed scenario, a target-resampling threshold of $\Gamma = 0.5 \cdot N$ makes ATRAPP to match the performance of APP when there is not a high number of targets in the scenario and match the performance of TRAPP when the number of targets grows. In this case, simulations indicate that ATRAPP can handle the situation of sampling the state space of 2 targets with their effective component sample sizes not falling below the 50% of the number of particles, thus not usually performing target-resampling and matching the performance of APP. On the other hand, results show that the effective component sample size tends to fall at least below the 50% of the number of particles when sampling the state space of 8 targets, thus making ATRAPP prone to target-resampling in this situation and allowing it to match the best performance of TRAPP in such a complex scenario.

It has been shown how a simple policy of a midpoint threshold selection allows ATRAPP to match the performance of APP and TRAPP when they respectively accomplish best results. However, a fine choice of the target threshold parameter

(a)

(b)

Figure 3.17: OSPA position error with respect to the number of particles with 3 different filters: APP, TRAPP and ATRAPP with $\Gamma = 0.5 \cdot N$. Results show how this parametrization of ATRAPP can match the best performance of APP when 2 targets (a) are present in the scenario as well as the best performance of TRAPP when 8 targets (b) are present in the scenario.

(a)

(b)

Figure 3.18: RMS OSPA position error with respect to the number of particles when 6 targets are under track with different filters; (a) JA, IP, PP, APP, TRAPP and ATRAPP with $\Gamma = 0.8 \cdot N$. (b) only shows the results of the 3 latter algorithms.

Figure 3.19: Target-resampling threshold policy dependent on the number of targets being tracked. The threshold is expressed as a relative number of particles with respect to the total number of particles. The upper bound of $\Gamma = 0.8 \cdot N$ is based on the best results obtained in previous simulations.

should make ATRAPP to only apply target-resampling when really necessary, while favoring sample diversity when possible. Figure 3.18 shows how ATRAPP can, not only match the performances of APP and ATRAPP, but also outperform both of them with a fine choice of the thresholding parameter. In the case of Figure 3.18, a constant target threshold of $\Gamma = 0.8 \cdot N$ is adopted, resulting on the mentioned improvement of the performance of ATRAPP with respect to APP and TRAPP PFs. It is also worth mentioning that although PP does not acutely suffer from the curse of dimensionality, it is clearly outperformed by the three methods presented in this chapter, as they benefit from auxiliary filtering and in the case of ATRAPP of the proposed adaptive target-resampling strategy.

3.5.4.2 Variable threshold

As previously stated, the optimal thresholding policy for ATRAPP can result in an improvement in the filter performance, as it can account for the best possible target-resampling strategy, favoring particle diversity. Such a policy should take into account the different dynamics of each target, the measurement model of the sensors surrounding each target and the number of targets to deal with the dimensionality problems arising in MTT. Nevertheless, such a thresholding policy can be difficult to

formulate.

In this section, we define a sample target-resampling threshold selection policy based on the number of targets in the scenario. We consider that the higher the number of targets in the scenario, the more resampling-prone the optimal target-resampling threshold selection policy should be. An example of such a policy is described in Figure 3.19. This strategy would adequately fit the full version of the MTT problem, with the detection of emerging targets and the estimation of the state of the surviving targets at each time step. Frameworks like the one presented in [50] can be used to employ filtering methods that assume a fixed number of targets in the scenario (which is the case for the methods presented in this chapter) in a problem with an unknown and varying number of targets. In this context, the use of ATRAPP with the proposed policy would adaptively respond to the necessities of the filter as the number of targets in the scenario varies.

Figure 3.20 shows the results of the applied policy for the proposed simulation scenario when the number of targets being tracked varies from $t = 2$ to $t = 8$. It is shown how ATRAPP either matches or improves the performance of APP and TRAPP PFs, a clearer view of this improvement is shown in Figure 3.20(b), where the errors between APP, TRAPP and ATRAPP have been normalized with respect to that of the worst performing filter in each simulation, showing how ATRAPP either matches or outperforms APP and ATRAPP.

The origin of the improvement in ATRAPP is explained in Figure 3.21. It is shown how all the filters increase their instantaneous error when target trajectories cross (almost all crossings approximately start after time step 50, see Figure 3.9). Nevertheless, it is shown how ATRAPP is the filter which better reacts to these difficult situations by performing a faster and more successful recovery, achieving this property by favoring subparticle diversity when possible and performing target-resampling only with it is really needed to avoid subparticle degeneracy.

Finally, Figure 3.22 completes the information provided in Figure 3.16 with the analysis of performance of ATRAPP with respect to its computational load. ATRAPP has a slightly smaller execution time than TRAPP, as it avoids the target-resampling stage whenever the effective component sample size is enough to prevent subparticle degeneracy. This also enhances its performance, so that it is shown in Figure 3.22 how for moderate and high affordable computational loads ATRAPP is the best choice among partitioned filters.

(a)

(b)

Figure 3.20: OSPA position error of APP, TRAPP and ATRAPP PFs with $N = 300$ particles when 2 to 8 targets are present in the scenario. ATRAPP uses the target-resampling threshold selection policy in Figure 3.19. (a) shows the OSPA position error and (b) the error normalized with respect to the worst performing filter in each case.

(a)

(b)

Figure 3.21: RMS OSPA position error in each time step of the trajectories for an scenario with 3 targets (a) and 6 targets (b) for PP, APP, TRAPP and ATRAPP PFs with $N = 300$ particles. ATRAPP uses the target-resampling threshold selection policy in Figure 3.19.

Figure 3.22: OSPA position error with respect to the execution time (a higher number of particles causes the execution time to rise) in a scenario with 6 targets for PP, APP, TRAPP and ATRAPP. ATRAPP uses the target-resampling threshold selection policy in Figure 3.19

3.6 Conclusions

Three different efficient, generally applicable algorithms for MTT, have been presented, APP, TRAPP and ATRAPP PFs. These filters all make use of state partitioning and auxiliary filtering. TRAPP additionally incorporates a target-resampling stage which can help to increase performance under difficult to sample state spaces at the cost nonetheless of subparticle diversity. The third presented method, ATRAPP adaptively performs target-resampling, thus favoring subparticle diversity when possible but avoiding subparticle degeneracy when necessary. The important point of the search for the optimum target-resampling threshold policy in ATRAPP remains to be fully explored. However it has been shown how even some simple policies can make ATRAPP achieve good results in a demanding MTT scenario.

Chapter 4

Multiple auxiliary particle filter and second-order multiple particle filters

This chapter continues the effort in tackling the problem of deriving efficient and reliable methods for filtering high-dimensional state spaces. However, contrary to the approach taken in Chapter 3, where the estimation of the complete joint state was considered in the filtering procedure, in this chapter we follow the strategy of multiple filters to overcome the curse of dimensionality. Multiple filters (see Section 2.10.1) make use of a separate filter to estimate each of the different components of a partition of the state, so that the dimension of the state space which each filter has to deal with is kept relatively low.

The background work on multiple filtering has already been covered in Section 2.10.1. In this chapter, we build on these concepts to first give a detailed explanation of the inclusion of auxiliary filtering within the multiple filtering framework. In addition, two additional novel multiple particle filtering methods are also presented which are designed to overcome some limitations related to the approximation to the marginalization procedure of multiple particle filtering methods in the literature. Three different methods are therefore considered in this chapter. The first of these three methods, in Section 4.1, incorporates the concept of auxiliary filtering to multiple particle filters, so that it can be of help in scenarios with informative measurements [130]. The two latter methods are built with the idea of incorporating second-order approximations to the required marginalization procedures in multiple filtering. This idea was introduced with the C-PF-PROP algorithm [15] (see Section 2.10.2), however, the approximations required in C-PF-PROP made the method only usable for a specific measurement model. The methods introduced in Section 4.2 overcome this limitation making use of sigma-point integration methods to achieve the desired second-order approximation to the marginalization procedure. The methods introduced in this

chapter have resulted in two publications [165, 167].

4.1 Multiple auxiliary particle filter

In this section, we consider a method that makes use of multiple particle filtering and auxiliary filtering to circumvent the difficulties of high-dimensional filtering. As explained in Section 2.10.1, the key point of multiple particle filters is to partition the state space as in (2.98) and run an individual particle filter for every component of the partition. Each particle filter shares information with the rest of the filters to account for the influence of the complete state in the observations collected by sensors.

The MPF in Section 2.10.1 can remarkably deal with the curse of dimensionality [26]. However, when $t = 1$, i.e., when the state of the dynamic system in (2.98) is only formed by a single component, the MPF in [26] boils down to the sequential importance resampling (SIR) PF [63], which is known to be usually outperformed by the APF [130], especially, for informative measurements. Given this intuition, in this section the multiple auxiliary particle filter (MAPF) is derived. The performance of the considered algorithm is tested in a multiple target tracking scenario, with fixed and known number of targets, using a sensor network with a nonlinear measurement model.

The use of auxiliary filtering allows taking into account current measurements in the sampling procedure (see Section 2.7.5). Thus, the MAPF approaches the way samples would be drawn from the OID (see Section 2.7.4), which usually reduces the variance of the particle weights, improving tracking performance. The MAPF can be used along with any measurement model, resulting in a generally applicable algorithm.

A previous work in the literature [30] proposed the use of auxiliary filtering within the MPF framework for the specific problem of studying the neural dynamics in magnetoencephalography and electroencephalography data. The main contribution of this section is, therefore, the detailed derivation of this approach and its application to MTT.

With the aim of approximating the marginal posterior PDF of each component of the partition in (2.98), and contrary to MPF in Section 2.10.1, instead of considering the sampling of the sequence of states $\mathbf{x}_j^{0:k}$ as in (2.125), MAPF obtains samples from the marginal posterior PDF of the state at time k

$$\begin{aligned} p(\mathbf{x}_j^k | \mathbf{z}^{1:k}) &= \int p(\mathbf{X}^k | \mathbf{z}^{1:k}) d\mathbf{X}_{-\{j\}}^k \\ &\propto \int p(\mathbf{z}^k | \mathbf{X}^k) p(\mathbf{X}^k | \mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k. \end{aligned} \tag{4.1}$$

Using the posterior independence assumption for the current state along with (2.102), we can write the predicted density as

$$\begin{aligned} p(\mathbf{X}^k | \mathbf{z}^{1:k-1}) &= \prod_{l=1}^t \int p(\mathbf{x}_l^k | \mathbf{x}_l^{k-1}) p(\mathbf{x}_l^{k-1} | \mathbf{z}^{1:k-1}) d\mathbf{x}_l^{k-1} \\ &= \prod_{l=1}^t p(\mathbf{x}_l^k | \mathbf{z}^{1:k-1}). \end{aligned} \quad (4.2)$$

Given a particle representation at time $k-1$ of the marginal posterior PDF of $\mathbf{x}_j^{k-1}$, the j -th PF can approximate $p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1})$ using the Chapman-Kolmogorov equation so that

$$\begin{aligned} p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1}) &= \int p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{k-1} | \mathbf{z}^{1:k-1}) d\mathbf{x}_j^{k-1} \\ &\approx \int p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) \sum_{i=1}^N \omega_{j,i}^{k-1} \delta(\mathbf{x}_j^{k-1} - \mathbf{x}_{j,i}^{k-1}) d\mathbf{x}_j^{k-1} \\ &= \sum_{i=1}^N \omega_{j,i}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}). \end{aligned} \quad (4.3)$$

The posterior marginal PDF of $\mathbf{x}_j^k$ in (4.1) can therefore be approximated as

$$\begin{aligned} p(\mathbf{x}_j^k | \mathbf{z}^{1:k}) &\propto \int p(\mathbf{z}^k | \mathbf{X}^k) p(\mathbf{X}^k | \mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k \\ &= p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1}) \int p(\mathbf{z}^k | \mathbf{X}^k) p(\mathbf{X}_{-\{j\}}^k | \mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k \\ &\simeq \sum_{i=1}^N \omega_{j,i}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}) \int p(\mathbf{z}^k | \mathbf{X}^k) p(\mathbf{X}_{-\{j\}}^k | \mathbf{z}^{1:k-1}) d\mathbf{X}_{-\{j\}}^k. \end{aligned} \quad (4.4)$$

Like MPF, at time k MAPF also collects the predicted state of the other PFs, $\hat{\mathbf{X}}_{-\{j\}}^k$. Then, using the approximations in (2.130) and (4.3), (4.4) can be written as

$$p(\mathbf{x}_j^k | \mathbf{z}^{1:k}) \approx p(\mathbf{z}^k | \mathbf{x}_j^k, \hat{\mathbf{X}}_{-\{j\}}^k) \sum_{i=1}^N \omega_{j,i}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}). \quad (4.5)$$

MAPF indirectly obtains samples from this PDF using an auxiliary variable a_j for each component of the state [130]. Therefore MAPF samples in a higher dimension from

$$p(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{x}_j^k, \hat{\mathbf{X}}_{-\{j\}}^k) p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}) \omega_{j,a_j}^{k-1} \quad (4.6)$$

where $a_j \in \{1, 2, \dots, N\}$ is an index in the mixture in (4.5). The selection of the

auxiliary variable is sound as $p(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \geq 0$ and integrating (4.6) over a_j , one gets the marginal PDF in (4.5).

Particles are obtained based on the proposal of an importance sampling density $q_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k})$ which takes the form

$$q_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto p(\mathbf{z}^k | \mu_{j,a_j}^k, \hat{\mathbf{X}}_{-\{j\}}^k) p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}) \omega_{j,a_j}^{k-1} \quad (4.7)$$

where μ_{j,a_j}^k is some characterization of $\mathbf{x}_j^k$ given $\mathbf{x}_{j,a_j}^{k-1}$, such as the predicted mean, $\mu_{j,a_j}^k = \mathbb{E}[\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}]$, or a sample, $\mu_{j,a_j}^k \sim p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1})$. As usual in auxiliary filtering, we consider that the importance density can factorize as

$$q_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) = q_j(\mathbf{x}_j^k | a_j, \mathbf{z}^{1:k}) q_j(a_j | \mathbf{z}^{1:k}). \quad (4.8)$$

By defining

$$q_j(\mathbf{x}_j^k | a_j, \mathbf{z}^{1:k}) \triangleq p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}), \quad (4.9)$$

it follows from (4.7) and (4.8) that samples from a_j are drawn according to the first-stage weights of the particles, λ_{j,a_j} , which are defined as

$$\begin{aligned} \lambda_{j,a_j} &\triangleq q_j(a_j | \mathbf{z}^{1:k}) \\ &\propto p(\mathbf{z}^k | \mu_{j,a_j}^k, \hat{\mathbf{X}}_{-\{j\}}^k) \omega_{j,a_j}^{k-1}. \end{aligned} \quad (4.10)$$

The weight update is therefore given by [106]

$$\begin{aligned} \omega_{j,i}^k &\propto \frac{p(\mathbf{x}_j^k, a_j^i | \mathbf{z}^{1:k})}{q_j(\mathbf{x}_j^k, a_j^i | \mathbf{z}^{1:k})} \\ &\propto \frac{p(\mathbf{z}^k | \mathbf{x}_{j,a_j^i}^k, \hat{\mathbf{X}}_{-\{j\}}^k) p(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,a_j^i}^{k-1}) \omega_{j,a_j^i}^{k-1}}{p(\mathbf{z}^k | \mu_{j,a_j^i}^k, \hat{\mathbf{X}}_{-\{j\}}^k) p(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,a_j^i}^{k-1}) \omega_{j,a_j^i}^{k-1}} \\ &= \frac{p(\mathbf{z}^k | \mathbf{x}_{j,a_j^i}^k, \hat{\mathbf{X}}_{-\{j\}}^k)}{p(\mathbf{z}^k | \mu_{j,a_j^i}^k, \hat{\mathbf{X}}_{-\{j\}}^k)}. \end{aligned} \quad (4.11)$$

The use of an auxiliary variable allows for the consideration of the current sensor measurements, $\mathbf{z}^k$, in the drawing of samples from the marginal PDF, thus approaching the way samples would be drawn from the OID. The use of auxiliary particle filtering usually outperforms sequential importance sampling. When measurements are informative [130], this approach reduces the variance of the weights, improving the performance of the algorithm, as demonstrated in the simulations in Section 4.4. A

Algorithm 4.1 PF for the j -th component in MAPF

```

- Initialize particles according to  $p(\mathbf{x}_j^0)$ 
- Compute  $\hat{\mathbf{x}}_j^1$  using (2.127)
- Send  $\hat{\mathbf{x}}_j^1$  to the other  $t - 1$  PFs.
for  $k = 1, \dots, k_{end}$  do
  - Collect new measurements  $\mathbf{z}^k$  from sensors

  for  $l = 1, \dots, t \quad l \neq j$  do
    - Receive  $\hat{\mathbf{x}}_l^k$  from the  $l$ -th PF.
  end for

  for  $i = 1, \dots, N$  do ▷ Auxiliary filtering
    - Draw  $\boldsymbol{\mu}_{j,i}^k$  using:
      
$$p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}) \text{ or } E[\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}]$$

    - Compute  $\lambda_{j,i} = p(\mathbf{z}^k | \boldsymbol{\mu}_{j,i}^k, \hat{\mathbf{X}}_{-\{j\}}^k) \omega_{j,i}^{k-1}$ 
  end for

  for  $i = 1, \dots, N$  do
    - Normalize  $\lambda_{j,i} = \frac{\lambda_{j,i}}{\sum_{i=1}^N \lambda_{j,i}}$ 
  end for

  for  $i = 1, \dots, N$  do
    - Sample an index  $a_j^i$  from the distribution
      defined by  $(\lambda_{j,1}, \lambda_{j,2}, \dots, \lambda_{j,N})$ 
    - Draw a sample  $\mathbf{x}_{j,i}^k$  using  $p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j^i}^{k-1})$ 
  end for

  for  $i = 1, \dots, N$  do ▷ Weight update
    - Compute  $\omega_{j,i}^k = \frac{p(\mathbf{z}^k | \mathbf{x}_{j,i}^k, \hat{\mathbf{X}}_{-\{j\}}^k)}{p(\mathbf{z}^k | \boldsymbol{\mu}_{j,a_j^i}^k, \hat{\mathbf{X}}_{-\{j\}}^k)}$ 
  end for

  for  $i = 1, \dots, N$  do
    - Normalize  $\omega_{j,i}^k = \frac{\omega_{j,i}^k}{\sum_{i=1}^N \omega_{j,i}^k}$ 
  end for

  - Compute  $\hat{\mathbf{x}}_j^{k+1}$  using (2.127)
  - Send  $\hat{\mathbf{x}}_j^{k+1}$  to the other  $t - 1$  PFs.
end for

```

pseudo-code describing the MAPF steps for the particle filter of the j -th component of the partition is provided in Algorithm 4.1. Note that each prediction and update steps are done in parallel for each component of the partition in (2.98), and after the update step each particle filter communicates its predicted state to the other particle filters so that they can successfully approximate $p(\mathbf{z}^k | \mathbf{x}_j^k, \hat{\mathbf{X}}_{-\{j\}}^k)$ at the next time step. The performance of the presented MAPF is demonstrated through simulations in a multiple target tracking scenario in Section 4.4.

4.2 Second-order sigma-point multiple particle filters

As previously pointed out, one of the main challenges in designing a multiple filter is the computation of the required marginalization integrals. In order to tackle this potentially difficult problem, both the MPF and MAPF (see Sections 2.10.1 and 4.1) rely on the first-order approximation in (2.130). However, the performance of MPF and MAPF can severely decay if the predicted PDF of the state of the components of the partition is not adequately represented by a single point using this approximation, for instance, when the state uncertainty is relatively high.

The C-PF-PROP filter (see Section 2.10.2), which was presented in [15], goes beyond this limitation by considering a second-order approximation to the involved marginalization integrals, so that each PF in the method receives not only the predicted means of the adjacent components of the partition but also the covariance matrices of the predictions. C-PF-PROP adequately propagates this uncertainty information through the marginalization integrals, outperforming the original MPF. An important drawback of the C-PF-PROP is however that it requires complex analytical derivations, which have only been provided for the received signal strength indicator measurement model [14]. In addition, a general procedure of how to extend C-PF-PROP to other measurement models is not explicitly mentioned in [14] so it is not straightforward to use it with other measurement models.

In this section, we address the shortcomings of the MPF, MAPF and C-PF-PROP algorithms and develop two multiple particle filters that exchange second-order information and can be used for any additive measurement model with Gaussian noise, without the need of analytical derivations. In the proposed methods, we first indicate how we can make use of sigma-point integration methods to efficiently compute the marginalization integrals [142, 143]. Sigma-points have been broadly applied to compute integrals arising in nonlinear estimation problems (see Section 2.5.2.1). For example, they have given rise to the unscented Kalman filter (UKF) [80, 171, 173], the quadrature Kalman filter (QKF) [5], or the cubature Kalman filter (CKF) [4].

The resulting filter, which is outlined in Section 4.2.1, is referred to as the sigma-point MPF (SP-MPF). A second contribution is the inclusion of auxiliary variables within the SP-MPF algorithm to further improve the performance of the algorithm with informative measurements [130]. The second algorithm, presented in Section 4.2.2, is referred to as the sigma-point multiple auxiliary PF (SP-MAPF). We demonstrate via simulations that the two proposed second-order multiple particle filter algorithms outperform MPF, MAPF and C-PF-PROP as well as other PFs in the literature.

4.2.1 Sigma-point MPF: A sigma-point-based second-order approximation

The sigma-point MPF (SP-MPF), as C-PF-PROP, is a second-order approximation to multiple particle filtering. However, unlike the latter, it can be applied to a wider family of measurement models and does not require the derivation of any analytical approximation for (2.138). Instead, it uses sigma-point integration methods, which have been shown to outperform analytical approximations in filtering applications [80].

First, we find it useful to define

$$\begin{aligned} l_j(\mathbf{z}^k | \mathbf{x}_j^k) &\triangleq \int p(\mathbf{z}^k | \mathbf{X}^k) \prod_{l=1, l \neq j}^t [p(\mathbf{x}_l^{0:k} | \mathbf{z}^{1:k-1})] d\mathbf{X}_{-\{j\}}^{0:k} \\ &= \int p(\mathbf{z}^k | \mathbf{X}^k) \prod_{l=1, l \neq j}^t [p(\mathbf{x}_l^k | \mathbf{z}^{1:k-1})] d\mathbf{X}_{-\{j\}}^k, \end{aligned} \quad (4.12)$$

which represents the likelihood for $\mathbf{x}_j^k$ after removing the influence on the measurements of the rest of the components of the state. Using (4.12), the marginal posterior for SP-MPF can be expressed as

$$p(\mathbf{x}_j^{0:k} | \mathbf{z}^{1:k}) \propto l_j(\mathbf{z}^k | \mathbf{x}_j^k) p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{0:k-1} | \mathbf{z}^{1:k-1}). \quad (4.13)$$

The developed method is applicable for additive Gaussian sensor models, which make the assumption

- A2 (2.3) can be expressed as

$$\mathbf{z}^k = \mathbf{h}(\mathbf{X}^k) + \mathbf{v}^k, \quad (4.14)$$

with

$$\mathbf{h}(\mathbf{X}^k) = \sum_{l=1}^t \mathbf{h}_l(\mathbf{x}_l^k) \quad (4.15)$$

where $\mathbf{h}_l(\mathbf{x}_l^k)$ is the contribution to the sensor measurements of the l -th component of the state and $\mathbf{v}^k$ is a zero-mean Gaussian noise with covariance matrix $\mathbf{R}^k$. Note that (4.14) and (4.15) indicate that additivity of the measurements $\mathbf{h}_l(\mathbf{x}_l^k)$, as well as the measurement noise $\mathbf{v}^k$ is required.

Considering a second-order Gaussian approximation to $p(\mathbf{x}_l^k | \mathbf{z}^{1:k-1})$ as in (2.135), (4.12) can be written as

$$\begin{aligned} l_j(\mathbf{z}^k | \mathbf{x}_j^k) &\approx \int p(\mathbf{z}^k | \mathbf{X}^k) \prod_{l=1, l \neq j}^t \left[\mathcal{N}(\mathbf{x}_l^k; \hat{\mathbf{x}}_l^k, \hat{\mathbf{C}}_l^k) \right] d\mathbf{X}_{-\{j\}}^k \\ &= \int \mathcal{N}(\mathbf{z}^k; \mathbf{h}(\mathbf{X}^k), \mathbf{R}^k) \times \prod_{l=1, l \neq j}^t \left[\mathcal{N}(\mathbf{x}_l^k; \hat{\mathbf{x}}_l^k, \hat{\mathbf{C}}_l^k) \right] d\mathbf{X}_{-\{j\}}^k. \end{aligned} \quad (4.16)$$

Note that the marginalization procedure in (4.16) corresponds to the same type of integral as in (2.36), which we can conveniently compute using sigma-points. Thus, taking into account A2 and the fact that the components of $\mathbf{X}_{-\{j\}}^k$ are independent in (4.16), we aim to use sigma-points to compute a second-order Gaussian approximation to (4.16) so that

$$\begin{aligned} l_j(\mathbf{z}^k | \mathbf{x}_j^k) &\approx l_j^{\mathcal{N}}(\mathbf{z}^k | \mathbf{x}_j^k) \\ &= \mathcal{N}\left(\mathbf{z}^k; \mathbf{h}_j(\mathbf{x}_j^k) + \sum_{l=1, l \neq j}^t \bar{\mathbf{z}}_l^k, \mathbf{R}^k + \sum_{l=1, l \neq j}^t \mathbf{S}_l^k\right) \end{aligned} \quad (4.17)$$

where $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ are respectively the contributions of $\mathbf{x}_l^k$ to the mean and covariance of $\mathbf{z}^k$.

Making use of sigma-point methods, we proceed now to estimate $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$. For each component of the partition, a set of m_l weighted sigma-points $\mathcal{X}_l = \{(\mathcal{X}_l^1, w_l^1), \dots, (\mathcal{X}_l^{m_l}, w_l^{m_l})\}$ are chosen so that they match $\hat{\mathbf{x}}_l^k$ and $\hat{\mathbf{C}}_l^k$, the estimation of the first two moments of $p(\mathbf{x}_l^k | \mathbf{z}^{1:k-1})$, which can be computed using (2.127) and (2.136). Then, sigma-points are propagated through $\mathbf{h}_l(\cdot)$, so that $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ can be approximated as

$$\bar{\mathbf{z}}_l^k \approx \sum_{i=1}^{m_l} w_l^i \mathbf{h}_l(\mathcal{X}_l^i) \quad (4.18)$$

$$\mathbf{S}_l^k \approx \sum_{i=1}^{m_l} w_l^i (\mathbf{h}_l(\mathcal{X}_l^i) - \bar{\mathbf{z}}_l^k) \cdot (\mathbf{h}_l(\mathcal{X}_l^i) - \bar{\mathbf{z}}_l^k)^T. \quad (4.19)$$

Thus, at time k , the j -th PF receives this sigma-point estimation of $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ from

the other $t - 1$ PFs, so that it can successfully approximate $l_j^N(\mathbf{z}^k | \mathbf{x}_j^k)$. SP-MPF also draws particles from the importance density (4.7) so that the particle weight update equation in (2.133) becomes

$$\begin{aligned}\omega_{j,i}^k &\propto \frac{l_j^N(\mathbf{z}^k | \mathbf{x}_j^k) p(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,i}^{k-1}) p(\mathbf{x}_{j,i}^{0:k-1} | \mathbf{z}^{1:k-1})}{p(\mathbf{x}_{j,i}^k | \mathbf{x}_{j,i}^{k-1}) q(\mathbf{x}_{j,i}^{0:k-1} | \mathbf{z}^{1:k-1})} \\ &= \omega_{j,i}^{k-1} l_j^N(\mathbf{z}^k | \mathbf{x}_j^k).\end{aligned}\tag{4.20}$$

Algorithm 4.2 PF for the j -th component in SP-MPF

- Initialize particles according to $p(\mathbf{x}_j^0)$
- Compute $\hat{\mathbf{x}}_j^1$ and $\hat{\mathbf{C}}_j^1$ using (2.127) and (2.136)
- Obtain a set of sigma-points and weights $\mathcal{X}_j$, whose first two moments match $\hat{\mathbf{x}}_j^1$ and $\hat{\mathbf{C}}_j^1$
- Compute $\bar{\mathbf{z}}_j^1$ and $\mathbf{S}_j^1$ using (4.18) and (4.19) using a sigma-point method
- Send $\bar{\mathbf{z}}_j^1$ and $\mathbf{S}_j^1$ to the other $t - 1$ PFs.

for $k = 1, \dots, k_{end}$ **do** $\triangleright k_{end}$ is the final time step

- Collect new measurements $\mathbf{z}^k$ from sensors

for $l = 1, \dots, t$ $l \neq j$ **do**

- Receive $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ from the l -th PF.

end for

for $i = 1, \dots, N$ **do**

- Draw a sample $\mathbf{x}_{j,i}^k$ using $p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1})$
- Compute $l_j^N(\mathbf{z}^k | \mathbf{x}_{j,i}^k)$ using (4.17)
- Compute $\omega_{j,i}^k = \omega_{j,i}^{k-1} l_j^N(\mathbf{z}^k | \mathbf{x}_{j,i}^k)$

end for

for $i = 1, \dots, N$ **do**

- Normalize $\omega_{j,i}^k = \frac{\omega_{j,i}^k}{\sum_{i=1}^N \omega_{j,i}^k}$

end for

- Compute $\hat{N}_{eff}^j = \frac{1}{\sum_{i=1}^N (\omega_{j,i}^k)^2}$

if $\hat{N}_{eff}^j < \Gamma$ **then**

- Resample the particle set $\{(\mathbf{x}_{j,1}^k, \omega_{j,1}^k), \dots, (\mathbf{x}_{j,N}^k, \omega_{j,N}^k)\}$ to obtain evenly weighted samples

end if

- Compute $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ using (2.127) and (2.136)
- Obtain a set of sigma-points and weights $\mathcal{X}_j$, whose first two moments match $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ using a sigma-point method
- Compute $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ using (4.18) and (4.19)
- Send $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ to the other $t - 1$ PFs.

end for

The j -th PF finally computes $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$, the first two moments of the predicted PDF of the state at time $k + 1$, using (2.127) and (2.136). Then, it draws its own set of sigma-points $\mathcal{X}_j$ that match $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ and use them to approximate $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ using (4.18) and (4.19) so that they can be transmitted to the other $t - 1$ PFs, which would use them in their own recursion at time $k + 1$.

It should be noted that, given the obtained particle approximation of $p(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k})$, if the interest lies only in approximating the marginal posterior PDF of the current state $p(\mathbf{x}_j^k|\mathbf{z}^{1:k})$, we can just discard the samples corresponding to previous states [7].

Thus, as a result of the way particles are drawn from $q_j(\mathbf{x}_j^{0:k}|\mathbf{z}^{1:k})$, approximating $p(\mathbf{x}_j^k|\mathbf{z}^{1:k})$ only requires to store the states of the particles up to time $k - 1$, so that practical implementations can be memory-efficient. However, the underlying state-space which the filter is dealing with is in fact that of $\mathbf{x}_j^{0:k}$. Note that the dimension, thus, grows with every time step, which provokes the so called particle degeneracy [44], as explained in Section 2.7.1.1. In order to prevent this problem, an adaptive resampling scheme based on the monitoring of the effective sample size N_{eff}^j [88, 136] is considered in each filter (see Section 2.7.1.1). One cannot generally exactly evaluate N_{eff}^j , however we make use of the estimate given by [7]

$$\hat{N}_{eff}^j = \frac{1}{\sum_{i=1}^N (\omega_{j,i}^k)^2}, \quad (4.21)$$

which takes values in the interval $[1, N]$. If $\hat{N}_{eff}^j$ falls below a given threshold, Γ , particles are resampled. This restores the effective sampling size to N and prevents particle degeneracy but results in a loss of diversity among particles. A pseudocode for SP-MPF is given in Algorithm 4.2.

4.2.2 Sigma-point multiple auxiliary particle filter

The objective of this section is to improve SP-MPF by use of auxiliary sampling [130]. The use of an auxiliary variable allows for the consideration of the actual measurements in the drawing of samples from the marginal PDF, approaching the way samples would be drawn from the optimal importance density [7, 44], reducing weight variance (see Section 2.7.5). This technique has been shown to improve particle filters performance under informative sensor measurements.

Contrary to the previously discussed method, instead of considering the posterior PDF of the sequence of states $\mathbf{x}_j^{0:k}$ as in (4.13), SP-MAPF obtains samples from $p(\mathbf{x}_j^k|\mathbf{z}^{1:k})$, the marginal posterior PDF of the state at time k .

Given a particle representation of the marginal posterior PDF at time $k-1$, the j -th PF can approximate $p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1})$ using the Chapman-Kolmogorov equation as

$$\begin{aligned} p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1}) &= \int p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) p(\mathbf{x}_j^{k-1} | \mathbf{z}^{1:k-1}) d\mathbf{x}_j^{k-1} \\ &\approx \int p(\mathbf{x}_j^k | \mathbf{x}_j^{k-1}) \sum_{i=1}^N \omega_{j,i}^{k-1} \delta(\mathbf{x}_j^{k-1} - \mathbf{x}_{j,i}^{k-1}) d\mathbf{x}_j^{k-1} \\ &= \sum_{i=1}^N \omega_{j,i}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}). \end{aligned} \quad (4.22)$$

As in SP-MPF, at time k the j -th PF of SP-MAPF collects the shared estimations $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ from the rest of the components of the partition, so that it can also approximate $l_j^N(\mathbf{z}^k | \mathbf{x}_j^k)$ in (4.17). Using (4.22), the marginal posterior PDF can therefore be written as

$$p(\mathbf{x}_j^k | \mathbf{z}^{1:k}) \propto l_j^N(\mathbf{z}^k | \mathbf{x}_j^k) \sum_{i=1}^N \omega_{j,i}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}). \quad (4.23)$$

SP-MAPF indirectly obtains samples from this PDF using an auxiliary variable, a_j , for each component of the state [130], which removes the sum in (4.23). Thus, SP-MAPF also avoids the increase in computational cost arising from the Chapman-Kolmogorov step integral in (4.22). Therefore, SP-MAPF samples in a higher dimension from

$$p(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto l_j^N(\mathbf{z}^k | \mathbf{x}_j^k) \omega_{j,a_j}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}) \quad (4.24)$$

where $a_j \in \{1, 2, \dots, N\}$ is an index in the mixture in (4.23). The selection of the auxiliary variable is sound as $p(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \geq 0$ and integrating (4.24) over a_j , one gets the marginal PDF in (4.23).

SP-MAPF also produces samples from an importance density in a higher dimension

$$q_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto l_j^N(\mathbf{z}^k | \boldsymbol{\mu}_{j,a_j}^k) \omega_{j,a_j}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}), \quad (4.25)$$

with $\boldsymbol{\mu}_{j,a_j}^k$ being some characterization of $\mathbf{x}_j^k$ given $\mathbf{x}_{j,a_j}^{k-1}$, such as the predicted mean, $\boldsymbol{\mu}_{j,a_j}^k = \mathbb{E}[\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}]$, or a sample, $\boldsymbol{\mu}_{j,a_j}^k \sim p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1})$. We consider that this importance density can be factorized as

$$q_j(\mathbf{x}_j^k, a_j | \mathbf{z}^{1:k}) \propto q_j(\mathbf{x}_j^k | a_j, \mathbf{z}^{1:k}) q_j(a_j | \mathbf{z}^{1:k}). \quad (4.26)$$

As with the other methods using auxiliary filtering in this thesis, we consider that

$$q_j(\mathbf{x}_j^k | a_j, \mathbf{z}^{1:k}) \triangleq p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j}^{k-1}), \quad (4.27)$$

Algorithm 4.3 PF for the j -th component in SP-MAPF

- Initialize particles according to $p(\mathbf{x}_j^0)$
- Compute $\hat{\mathbf{x}}_j^1$ and $\hat{\mathbf{C}}_j^1$ using (2.127) and (2.136)
- Obtain a set of sigma-points and weights $\mathcal{X}_j$, whose first two moments match $\hat{\mathbf{x}}_j^1$ and $\hat{\mathbf{C}}_j^1$ using a sigma-point method
- Compute $\bar{\mathbf{z}}_j^1$ and $\mathbf{S}_j^1$ using (4.18) and (4.19)
- Send $\bar{\mathbf{z}}_j^1$ and $\mathbf{S}_j^1$ to the other $t - 1$ PFs.

for $k = 1, \dots, k_{end}$ $\triangleright k_{end}$ is the final time step

- Collect new measurements $\mathbf{z}^k$ from sensors

for $l = 1, \dots, t \quad l \neq j$ **do**

- Receive $\bar{\mathbf{z}}_l^k$ and $\mathbf{S}_l^k$ from the l -th PF.

end for

for $i = 1, \dots, N$ **do**

- Draw $\boldsymbol{\mu}_{j,i}^k$ using:
 $p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1})$ or $E[\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}]$
- Compute $l_j^N(\mathbf{z}^k | \boldsymbol{\mu}_{j,i}^k)$ using (4.17)
- Compute $\lambda_{j,i}^k = \omega_{j,i}^{k-1} l_j^N(\mathbf{z}^k | \boldsymbol{\mu}_{j,i}^k)$

end for

for $i = 1, \dots, N$ **do**

- Normalize $\lambda_{j,i}^k = \frac{\lambda_{j,i}^k}{\sum_{i=1}^N \lambda_{j,i}^k}$

end for

for $i = 1, \dots, N$ **do**

- Sample an index a_j^i from the distribution
defined by $(\lambda_{j,1}, \lambda_{j,2}, \dots, \lambda_{j,N})$
- Draw a sample $\mathbf{x}_{j,i}^k$ using $p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j^i}^{k-1})$
- Compute $l_j^N(\mathbf{z}^k | \mathbf{x}_{j,a_j^i}^k)$ using (4.17)
- Compute $\omega_{j,i}^k = \frac{l_j^N(\mathbf{z}^k | \mathbf{x}_{j,a_j^i}^k)}{l_j^N(\mathbf{z}^k | \boldsymbol{\mu}_{j,a_j^i}^k)}$

end for

for $i = 1, \dots, N$ **do**

- Normalize $\omega_{j,i}^k = \frac{\omega_{j,i}^k}{\sum_{i=1}^N \omega_{j,i}^k}$

end for

- Compute $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ using (2.127) and (2.136)
- Obtain a set of sigma-points and weights $\mathcal{X}_j$, whose first two moments match $\hat{\mathbf{x}}_j^{k+1}$ and $\hat{\mathbf{C}}_j^{k+1}$ using a sigma-point method.
- Compute $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ using (4.18) and (4.19)
- Send $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ to the other $t - 1$ PFs.

end for

so that samples from a_j are drawn from $q_j(a_j|\mathbf{z}^{1:k})$ according to the distribution of first-stage weights, $\lambda_{j,i}$, which considering (4.25), (4.26) and (4.27) are given by

$$\lambda_{j,i} \propto \omega_i^{k-1} l_j^N \left(\mathbf{z}^k | \boldsymbol{\mu}_{j,a_j^i}^k \right). \quad (4.28)$$

Making use of (4.24) and (4.26), the updated weights are computed according to the principle of importance sampling as

$$\begin{aligned} \omega_{j,i}^k &\propto \frac{p(\mathbf{x}_{j,i}^k, a_j^i | \mathbf{z}^{1:k})}{q_j(\mathbf{x}_{j,i}^k, a_j^i | \mathbf{z}^{1:k})} \\ &\propto \frac{l_j^N \left(\mathbf{z}^k | \mathbf{x}_{j,a_j^i}^k \right) \omega_{j,a_j^i}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j^i}^{k-1})}{l_j^N \left(\mathbf{z}^k | \boldsymbol{\mu}_{j,a_j^i}^k \right) \omega_{j,a_j^i}^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,a_j^i}^{k-1})} \\ &= \frac{l_j^N \left(\mathbf{z}^k | \mathbf{x}_{j,a_j^i}^k \right)}{l_j^N \left(\mathbf{z}^k | \boldsymbol{\mu}_{j,a_j^i}^k \right)}. \end{aligned} \quad (4.29)$$

As in SP-MPF, the j -th particle filter finally estimates the first two moments of the predicted PDF at the next time step using (2.127) and (2.136), and draws a set of sigma-points $\mathcal{X}_j$ to approximate $\bar{\mathbf{z}}_j^{k+1}$ and $\mathbf{S}_j^{k+1}$ using (4.18) and (4.19), which are then sent to the other $t - 1$ PFs.

A pseudocode for SP-MAPF is given in Algorithm 4.3. The main features and differences among the different multiple particle filtering algorithms have been summarized in Table 4.1.

4.3 On the computational complexity of multiple particle filters

In this section, we briefly analyze the computational complexity of the different multiple particle filter methods considered in this thesis in comparison to that of MPF (see Section 2.10.1), as this algorithm is the most basic multiple particle filter. The key aspect is that the computational complexity of the presented methods is $\mathcal{O}(N \cdot t)$, i.e., linear with respect to the number of particles, which is a desirable property for a particle filter algorithm, though not necessary [86, 113], and also linear with respect to t , the number of components of the partition in (2.98).

We first consider the case of the MAPF, presented in Section 4.1. The main difference of this method with respect to MPF is the necessity of computing the first-stage weights

Table 4.1: Principal features of the different multiple particle filtering algorithms

Filter	MPF	MAPF	C-PF-PROP	SP-MPF	SP-MAPF
Approximation to (2.121)	Dirac delta approximation	Dirac delta approximation	Analytical approximation	sigma-point methods	sigma-point methods
Order of the approximation	mean	mean	mean and covariance	mean and covariance	mean and covariance
Auxiliary filtering	×	✓	×	×	✓
Does not require analytical derivations	✓	✓	×	✓	✓

of particles, a step whose computational complexity is $\mathcal{O}(N)$, also depending on the specifics of the measurement model, and which generally makes the computational cost of MAPF higher than that of MPF. It should also be noted that the additional computational cost of drawing samples from the auxiliary variable in MAPF is similar to that of performing the resampling of MPF, so that we do not consider it relevant in this comparison.

The main increase in computational complexity of SP-MPF with respect to MPF lies in the following required additional steps:

1. The computation of the covariance matrix $\hat{\mathbf{C}}_l^k$, which is $\mathcal{O}(N \cdot n_{x_l}^2)$.
2. The drawing of sigma-points, which considering the use of a Cholesky factorization of $\hat{\mathbf{C}}_l^k$, is $\mathcal{O}(n_{x_l}^3)$ [172].
3. The transformation of sigma-points, whose computational complexity depends on the measurement model, and computation of the mean and covariance of the transformed sigma-points, which are $\mathcal{O}(m \cdot n_z)$ and $\mathcal{O}(m \cdot n_z^2)$, respectively.

The important result of this analysis is that, typically, we have that $N \gg m$, $N \gg n_{x_l}$ and $N \gg n_z$, so that the computational complexity is mainly dominated by the number of particles. As the computational complexities of SP-MPF and MPF are both linear with respect to the number of particles, this means that the increase in computational complexity of SP-MPF with respect to MPF will not be severe (as will be shown in Figure 4.17 for the simulation examples of Section 4.4). In addition, if we aim to achieve a required performance goal in our application, which is the case in many engineering problems, SP-MPF allows for the decrease of the number of particles N with respect to MPF. Therefore, the required running time to achieve our performance goal can be effectively lowered with respect to MPF. A discussion considering this aspect based on our simulations can be found in Section 4.4.2.2.

As happened with MAPF, in the case of SP-MAPF the evaluation of the first stage weights is also required, a procedure with a computational complexity $\mathcal{O}(N)$ that also depends on the measurement model. It is important to note that, the computational complexity of SP-MAPF also remains linear with respect to the number of particles.

As usual in auxiliary filtering, the increase in computational complexity of MAPF and SP-MAPF generally pays off under informative measurements [130]. In these cases, the performance improvement using SP-MAPF will in principle enable us to use a considerably smaller number of particles to achieve a performance goal. This results in a lower computational burden than previous methods in the literature, as discussed in Section 4.4.2.2.

Figure 4.1: SP-MPF particles (thin dots) and sigma-points (thick dots) in a scenario with 3 different targets at a given time step.

4.4 Experimental results

In this section, the performance of the methods previously proposed in this chapter is analyzed via simulations of a multiple target tracking scenario with fixed and known number of targets. This problem adequately fits the multiple particle filtering approach, as the marginal posterior PDF of the state of each target can be individually approximated by a different PF of a multiple filter.

The performance of the described methods is compared against other PFs in the literature: the PP [50] method described in Section 2.9.1, the APP, TRAPP and ATRAPP PFs [163, 164] presented in Chapter 3, the MPF [26] in Section 2.10.1, and the C-PF-PROP [15] whose details can also be found in Section 2.10.2. For the sake of clarity, the performance of the JA and IP filters are not taken into account in this section, as they were shown in Chapter 3 to be unable of adequately tackling the curse of dimensionality and target trajectory crossings, resulting in a high tracking error, so that they are not longer of interest for performance comparison purposes. Both MPF and SP-MPF use a resampling threshold of $\Gamma = 0.8 \cdot N$, while ATRAPP considers the target-resampling threshold policy of Figure 3.19. The considered dynamic and measurement models of the simulations are introduced in Section 4.4.1, while the performance in the presented scenario for different PF methods is discussed in Section 4.4.2.

As in the previous chapter, the tracking performance of the different methods under consideration is evaluated by means of the mean optimal sub-pattern assignment

(OSPA) position error [145] (with parameters $c = 10$ and $p = 2$), averaged over the runs of a Monte Carlo simulation with $n = 5000$ realizations. It should be kept in mind that the purpose of the OSPA metric is to consider the tracking performance of the algorithms without taking into account the possible flipping of target labels after the crossing of targets, as OSPA performs a search for the minimum distance permutation between the target positions and the predictions.

At this point, it is worth recalling that multiple filters separately approximate the marginal posterior PDF of each target, while JA, PP, APP, TRAPP and ATRAPP approximate the joint posterior PDF of all targets. Then, when we say that a method uses N particles, this means that MPF, MAPF, C-PF-PROP, SP-MPF and MAPF use t different PFs with N particles taking values in $\mathbb{R}^2$ (as Rao-Blackwellization is considered and particles only contain target position elements, see Section 4.4.1), while PP, APP, TRAPP and ATRAPP use a unique PF with N particles, which take values in $\mathbb{R}^{2t}$.

We have chosen the unscented transform [79, 80] as the sigma-point method used in the implementation of SP-MPF and SP-MAPF. Thus, a total number of $m = 2 \cdot n_x + 1$ sigma-points is used to represent the marginal posterior PDF of each target, where n_x is the dimension of the state space being represented by the sigma-points. In this case $n_x = 2$, as Rao-Blackwellization is also considered (see Section 4.4.1). The weight of the central sigma-point, which is a parameter of the unscented transform, has been set to $w_o = 1/3$. We have also implemented the cubature rule as the sigma-point integration method but the results remain basically unaltered, so it is no longer considered for clarity in the figures.

Before performance considerations are discussed later in this section, we think it is convenient to illustrate how SP-MPF and SP-MAPF work. To this end, Figure 4.1 shows the inner information managed by the different PFs of the proposed methods in a scenario with 3 targets. As previously explained in Section 4.2.1, the marginal posterior PDF of each target is approximated by a PF using a weighted set of particles. In addition, every PF also computes a set of weighted sigma-points (in this case drawn using the unscented transform) that matches the first two moments of its predicted marginal posterior PDF. Figure 4.1 shows in a different color for each target (or each PF) the particles and sigma-points which represent their marginal PDFs at a given time step.

4.4.1 Dynamic and measurement model

The dynamic model used in this simulations is a nearly constant velocity model, already described in Section 3.5.1 with $\tau = 1\text{s}$ and $\sigma_u = 0.1\text{m/s}^{3/2}$. The simulated target trajectories are shown in Figure 3.9. The PDF at time $k = 0$ of the position of the j -th target $p(\mathbf{x}_j^0)$ is initialized as a Gaussian distribution centered at the true position of the target, with covariance matrix $\mathbf{Q}$.

Measurements at time k , $\mathbf{z}^k = [z_i^k, \dots, z_{N_s}^k]^T$, are received through $N_s = 169$ sensors in a sensor network displayed in a 2-D regular grid covering $120\text{m} \times 120\text{m}$, with a spacing of 10m between sensors in both axes.

The presented simulations consider two different measurement models: the first one is the additive received power measurement model already described in Section 3.5.2 (with $Z_0 = 20$, $\rho_0 = 5$, $\alpha = 2$ and $\sigma_n^2 = 1$). However, we wish to also compare the performance of the methods presented in this chapter with the C-PF-PROP method presented in [15] (see Section 2.10.2). Therefore, in our simulations we also make use of the received signal strength indicator (RSSI) measurement model for which the required analytical expressions for C-PF-PROP are available in [14, 15].

In the RSSI measurement model, the nonlinear measurement equation (4.14) for the i -th sensor at time k then is

$$z_s^k = h_s(\mathbf{X}^k) + v_s^k \quad (4.30)$$

$$h_s(\mathbf{X}^k) = \sum_{j=1}^t h_{s,j}(\mathbf{x}_j^k) \quad (4.31)$$

$$h_{s,j}(\mathbf{x}_j^k) = \frac{\Phi}{(\rho_s(\mathbf{x}_j^k))^\alpha + \epsilon} \quad (4.32)$$

where v_s^k is a zero-mean, Gaussian-distributed noise with variance σ_n^2 , Φ determines the signal power received by a sensor, ϵ is a saturation parameter which limits the signal power that can be received from near targets, α is a path-loss coefficient and $\rho_s(\mathbf{x}_j^k)$ is the distance from the j -th target to the s -th sensor, which is given by (3.28). The value of the sensor model parameters have been set to $\sigma_n^2 = 1$, $\Phi = 500$, $\epsilon = 25$ and $\alpha = 2$.

Under the above dynamic and measurement models, nonlinearities arise from the relationship between target positions and measurements in (4.30), while the relationship between the velocities and the positions of the targets is linear and Gaussian. This allows for the optimal computation of the velocity elements of the state by means of Kalman filtering, conditional on the trajectory of the targets. As previously mentioned in Section 2.8.1, this technique, known as Rao-Blackwellization,

Figure 4.2: OSPA position error with respect to the number of particles when 1 target is under track with PP, APP, TRAPP, ATRAPP, MPF, MAPF, SP-MPF and SP-MAPF with the additive received power measurement model in Section 3.5.2.

improves the performance of the PFs for a given sample size [3,50,120], as the dimension of the state space considered by each PF in the MPF is halved. Therefore in this case, particles in our simulations only contain the position elements of the targets, while velocities are estimated using a Kalman filter for each particle, conditional on its position.

4.4.2 Simulation results

Figures 4.2 to 4.5 show the OSPA position error [145] (with $p = 2$ and $c = 10$) of the different considered methods under the additive received power measurement model in Section 3.5.2 for 1, 4, 6 and 8 targets in the scenario. Equivalently, Figures 4.6 to 4.9 shows the OSPA position error under the RSSI measurement model previously described in Section 4.4.1 also for 1, 4, 6 and 8 targets in the scenario. The performance results for all filters are jointly shown in the displayed figures. However, in order to organize the argumentation, we first comment on the performance results of MAPF in Section 4.4.2.1, while the results of SP-MPF and SP-MAPF are analyzed in Section 4.4.2.2.

Figure 4.3: OSPA position error with respect to the number of particles when 4 targets are under track with PP, APP, TRAPP, ATRAPP, MPF, MAPF, SP-MPF and SP-MAPF with the additive received power measurement model in Section 3.5.2.

Figure 4.4: OSPA position error with respect to the number of particles when 6 targets are under track with PP, APP, TRAPP, ATRAPP, MPF, MAPF, SP-MPF and SP-MAPF with the additive received power measurement model in Section 3.5.2.

Figure 4.5: OSPA position error with respect to the number of particles when 8 target are under track with PP, APP, TRAPP, ATRAPP, MPF, MAPF, SP-MPF and SP-MAPF with the additive received power measurement model in Section 3.5.2.

Figure 4.6: OSPA position error with respect to the number of particles when 1 target is under track with PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF with the RSSI measurement model in Section 4.4.1.

Figure 4.7: OSPA position error with respect to the number of particles when 4 targets are under track with PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF with the RSSI measurement model in Section 4.4.1.

Figure 4.8: OSPA position error with respect to the number of particles when 6 targets are under track with PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF with the RSSI measurement model in Section 4.4.1.

Figure 4.9: OSPA position error with respect to the number of particles when 8 targets are under track with PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF with the RSSI measurement model in Section 4.4.1.

4.4.2.1 MAPF

Before characterizing the performance of MAPF, it is first worth recalling that, as MPF, this filter makes use of the first-order approximation in (2.130) in order to approximate the marginalization integrals of multiple filters. Thus, its performance is not expected to overcome that of the second-order multiple filters (i.e., C-PF-PROP, SP-MPF and SP-MAPF), and its natural baseline reference for comparison is therefore the performance of the MPF filter. On the other hand, when considering the performance of filters that approximate the joint posterior PDF of the targets such as the PP, APP, TRAPP and ATRAPP in comparison with that of MAPF, results will generally depend on the scenario at hand, with the best performing filter among these varying depending on different factors, mainly the dynamic and measurement models.

The performance of MAPF in comparison with that of JA is first briefly considered. Performance results for JA with the additive received power measurement model are shown and analyzed in Section 3.5.3. Using this same measurement model, performance results for MAPF are shown in Figures 4.2 to 4.5. Note that contrary to JA, MAPF is able of successfully overcoming the curse of dimensionality when the number of targets in the scenario grows, which severely affects JA (see Figure 3.13a). In addition, it is also shown how the other PFs that approximate the joint posterior PDF (PP, APP, TRAPP and ATRAPP) are able of successfully dealing with dimensionality, but are

nevertheless outperformed by the MAPF, specially when a larger number of targets are present in the scenario (see Figure 4.5). Using this measurement model, MAPF is therefore the overall best performing filter but for the second-order filters, SP-MPF and SP-MAPF.

It is of special interest to note the difference in performance between MPF and MAPF, as they both use the same technique to deal with the curse of dimensionality, i.e., separately approximating the marginal PDFs for each target using a first-order approximation of the marginalization integrals. The MAPF clearly outperforms MPF, thus backing up the previously outlined insights on the use of auxiliary filtering to improve the performance of multiple filters. Another feature worth mentioning of MAPF is the remarkable results obtained when considering a very low number of particles (see Figures 4.3 and 4.5), thus making it specially advisable when very strict computational or memory constraints need to be considered.

The performance improvement of MAPF with respect to MPF is also observable when considering the use of the RSSI measurement model in Figures 4.6 to 4.9, with MAPF always being the best filter among these two. Nevertheless, in this case, MAPF does not outperform those filters which approximate the joint PDF (PP, APP TRAPP and ATRAPP). This effect is not straightforwardly explainable, but it is worth noting that MPF is also clearly outperformed by the filters approximating the joint PDF and that second-order multiple particle filters (C-PF-PROP, SP-MPF and SP-MAPF) have an apparently better behavior than MPF and MAPF. It is therefore reasonable to assume that the first-order approximation in (2.130) used by MPF and MAPF is in this case harming their performance. Interestingly, the filters approximating the joint PDF also make use of a similar approach to this first-order approximation when using (2.112) in order to draw particles. Nevertheless, the availability of the joint posterior PDF, which these filters approximate, allows them to outperform MPF and MAPF.

Figure 4.10 illustrates in a clearer way how the different PFs react to the curse of dimensionality, showing the OSPA position error when tracking from 2 to 8 targets for a fixed number of particles. It can be observed again how MAPF consistently outperforms MPF with both measurement models.

Table 4.2 shows the OSPA position errors in a scenario with 3 targets of the different multiple particle filter algorithms discussed in this section as a function of the measurement noise variance using the RSSI measurement model. It can be observed that for low ($\sigma_n^2 = 0.04$) and mild ($\sigma_n^2 = 1$) measurement noise the auxiliary versions of the filters outperform their counterparts, and in particular, among the first-order multiple filters, MAPF outperforms MPF. However, for high measurement noise ($\sigma_n^2 = 100$) the best performing filters are switched and MPF outperforms MAPF. This

Table 4.2: OSPA position error [m] ($p = 2$, $c = 10\text{m}$) as a function of the measurement noise variance, σ_n^2 in the RSSI measurement model in Section 4.4.1, for the different multiple particle filters considered in this section with 3 targets under track and 300 particles.

Filter	$\sigma_n^2 = 0.04$	$\sigma_n^2 = 1$	$\sigma_n^2 = 100$
MPF	1.20m	1.58m	4.37m
MAPF	0.83m	1.32m	4.57m
C-PF-PROP	0.25m	1.21m	4.36m
SP-MPF	0.22m	0.91m	3.89m
SP-MAPF	0.19m	0.83m	4.17m

is caused by the informative measurements occurring under low and mild measurement noise variance, as it is in these cases where the use of an auxiliary variable in particle filters is best advised [130].

Figures 4.15 and 4.17 illustrate the computational complexity of the different methods based on their execution time with respect to the number of particles. The computational demand of the MAPF method is higher than that of the MPF. This is a consequence of the use of auxiliary filtering, which requires to additionally compute the first-stage weights in (4.10) to draw samples from the auxiliary variable.

Figures 4.16 and 4.18 show the average OSPA position error of the different methods as a function of their execution time respectively shown in Figures 4.15 and 4.17. Note that for an equal amount of required computational resources, MAPF is able of outperforming MPF. The performance results of MAPF along with the additive received power measurement model in Figure 4.16 are once again shown to be quite remarkable, outperforming all the other filters except for SP-MPF and SP-MAPF.

4.4.2.2 SP-MPF and SP-MAPF

Before analyzing the performance of the SP-MPF and SP-MAPF filters, we first seek to characterize the accuracy of the approximation of the proposed filters to the marginalization integral in (2.121). In order to also assess the accuracy of the

(a) Additive received power measurements

(b) RSSI measurements

Figure 4.10: OSPA position error of PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF with $N = 350$ particles when 2 to 8 targets are present in the scenario. ATRAPP uses the target-resampling threshold selection policy in Figure 3.19. (a) Additive received power measurement model (b) RSSI measurement model.

C-PF-PROP approximation, we consider the use of the RSSI measurement model. As a baseline for comparison, we have performed simulations considering the direct marginalization method in [113, Eq. (49)], which computes a particle approximation of the *true* marginalization integrals in (2.121), while C-PF-PROP computes an analytic approximation described in [15] and SP-MPF computes a sigma-point Gaussian approximation. Figure 4.11 shows the OSPA position error in a scenario with 2 and 3 targets. Note that, as expected, the direct marginalization method in [113, Eq. (49)] consistently achieves the lowest error, as it computes a particle approximation of the true marginalization integrals. Nonetheless, it is worth noting that, contrary to C-PF-PROP (which was specifically designed for this measurement model), the SP-MPF method tightly approaches this performance baseline, meaning that the marginalization integrals are being accurately approximated using the sigma-point integration methods. SP-MAPF achieves better results in this case, however, its results are not shown in Figure 4.11 as the method in [113, Eq. (49)] does not use auxiliary filtering and it would thus not serve as a performance baseline for SP-MAPF. It should also be noted, that the computational complexity of SP-MPF is $\mathcal{O}(t \cdot N)$, while the method in [113, Eq. (49)] is $\mathcal{O}(N^t)$. Thus, the use of this direct marginalization method is unaffordable for a higher number of targets, i.e., a higher number of components in the partition in (2.98).

We also analyze the efficiency of the importance sampling schemes of the different multiple particle filters by means of their effective target sample size [88, 136]. Figure 4.12 shows the mean effective target sample size at each time step, with three targets under track, for the different multiple particle filter algorithms when they use 300 particles. In order to have a fair comparison in this analysis, as MPF, C-PF-PROP and SP-MPF perform a resampling stage (with all weights being equal to $\omega_{j,i}^k = \frac{1}{N}$ after resampling), the effective sample size is being computed for these methods before resampling. It can be observed that SP-MAPF has the highest effective sample size at all times, meaning that it accurately approximates the posterior PDFs. Note that at those time instants where the involved PDFs are harder to approximate (i.e., when targets get in close proximity, from time $k = 60$), the difference in the effective sample size from the second-order approximations of SP-MAPF and SP-MPF respectively broadens with respect to the first-order approximations of MAPF and MPF, indicating once again the remarkable performance of the algorithms presented in this section. It is also worth noting that SP-MPF also presents a higher effective sample size than C-PF-PROP in these situations, which is in accordance with the performance results shown in Figure 4.11 and 4.13.

We also provide more details in this comparison for the case where there are 3 targets

(a)

(b)

Figure 4.11: OSPA position error at each time step of MPF, C-PF-PROP, SP-MPF and direct marginalization filter in [113] using 100 particles with (a) 2 targets and (b) 3 targets with the RSSI measurement model in Section 4.4.1.

Figure 4.12: Mean effective sample size for the different multiple particle filters considered in this section in a scenario with 3 targets and 300 particles with the RSSI measurement model in Section 4.4.1 and a measurement noise variance of $\sigma_n^2 = 1$. As there is a particle filter for each target, we show the averaged effective sample size for all targets.

in the scenario. The averaged OSPA error against time for the different algorithms is provided in Figure 4.13. As expected, among the PFs that implement the multiple particle filtering approach, those that use a first-order approximation to the marginal posterior target densities (i.e., MPF and MAPF) are outperformed by the ones using higher-order approximations (C-PF-PROP, SP-MPF and SP-MAPF). Note that, as expected, the error generally increases for all algorithms at those time instants when the targets trajectories cross each other (see Figure 3.9). It is in fact in these difficult situations when the biggest difference in performance is observed when comparing SP-MPF or SP-MAPF with the rest of the filters. Note that, at those time instants when targets are far from each other (in the initial steps of the trajectories), the performance of all filters is similar, as the effect of other targets on the measurements of sensors surrounding an specific target is negligible.

A further analysis of the performance of the filters under the RSSI measurement model can be followed through Figures 4.6 to 4.9. It is worth noting that SP-MPF and SP-MAPF both consistently outperform C-PF-PROP, a feature which speaks particularly well of the algorithms presented in this section, as the design of C-PF-PROP specifically fits the RSSI measurement model. It can also be observed that SP-MAPF is the overall best-performing filter as it gathers all the advantages from the techniques considered in this section, dealing with the dimensionality effects by multiple particle filtering and Rao-Blackwellization, relying on sigma-point methods to

Figure 4.13: Averaged OSPA position error at each time step of the different algorithms with $N = 350$ particles in a scenario with 3 targets with the RSSI measurement model in Section 4.4.1.

achieve a higher-order approach to the marginalization integrals, and using improved importance densities by means of auxiliary filtering.

The performance of those filters that consider the estimation of the joint state (PP APP, TRAPP and ATRAPP) should also not to be overlooked, as they also competently deal with the proposed high-dimensional multiple target tracking problem, especially, for a high number of particles. In addition, some applications can benefit from the availability of joint state estimates [54].

For the sake of completeness, in this case we also provide the averaged root mean square (RMS) position error against the number of particles for 3 and 6 targets present in the scenario in Figure 4.14 (results are provided for the RSSI measurement model to allow for a performance comparison with respect to C-PF-PROP). Note that in this case, the error in both situations increases compared to the OSPA position error of Figure 4.8. This effect is a consequence of the lack of a cutoff parameter as in OSPA, which bounds the error contribution of the outlier estimates. In addition, the RMS estimates do not consider an optimal association from estimates to the real position of targets, which also greatly increases the error. Interestingly, the performance of the MAPF is notoriously degraded when using the RMS position error with respect to the OSPA position error in Figure 4.8. This is caused by the MAPF being prone to switching target estimates with the true position of targets when their trajectories cross. Thus, if target labeling were to be taken into account, MAPF would not achieve a good performance in this case. Nevertheless, it is important to note that, once again, SP-MPF and SP-MAPF are the overall best performing algorithms, also clearly outperforming C-PF-PROP. Given that we do not consider target labeling, in the following we continue to use the OSPA position error to consider the optimal associations from estimates to the real target positions.

In order to illustrate how the different algorithms react to an increase in the dimensionality of the state space, Figure 4.10 shows their averaged OSPA position error using a fixed number of particles with respect to the number of targets in the scenario. It is confirmed through the results in this figure, that those filters using higher-order approximations to the marginal posterior PDFs outperform the first-order approaches, specially for a high number of targets, with SP-MPF and SP-MAPF clearly achieving the lowest error in all cases. It is worth noting that although the error apparently increases for all the algorithms when a higher number of targets is considered, this trend is not applicable when the 8th target is incorporated in the scenario, as the averaged error in this case descends. Note that the trajectory of this target in Figure 3.9 does not get close to targets 1 to 6. This behavior thus indicates that the increasing error in Figure 4.10 is not due to the algorithms having great difficulties with dimensionality

(a)

(b)

Figure 4.14: Averaged RMS position error with respect to the number of particles of the different algorithms in a scenario with 3 targets (a) and with 6 targets (b) using the RSSI measurement model in Section 4.4.1.

Figure 4.15: Mean execution time with respect to the number of particles for the different algorithms when $t = 6$ targets are present at the scenario. Execution times consider the estimation of the 100 steps of the trajectories with the additive received power measurement model in Section 3.5.2.

on itself but is instead (to a great extent) caused by the difficult situation arising from the repeatedly crossing of targets 1 to 7 in the center of the scenario. Once again, this shows that these filters, specially SP-MPF and SP-MAPF, remarkably deal with these difficult scenarios.

As shown in Section 4.3, it is important to take into account that these algorithms do not have the same computational resources requirements for a given number of particles. Figures 4.15 and 4.17 illustrate their computational complexity based on their execution time with respect to the number of particles. Note that as a general rule, those algorithms using auxiliary filtering have a higher demand of computational resources, as they have to additionally compute the first-stage weights to draw samples from the auxiliary variables. It is also worth noting that the execution time of SP-MPF is very similar to that of C-PF-PROP in Figure 4.17. Figures 4.16 and 4.18 show the average OSPA position error of the different methods as a function of their required execution time. Note that for an equal amount of required computational resources, SP-MPF and SP-MAPF are also clearly the best performing methods, except for those cases in which a really small execution time is required, where filters with very low computational requirements are advisable.

Figure 4.16: PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF OSPA error with respect to the execution time with $t = 6$ targets present at the scenario with the additive received power measurement model in Section 3.5.2. The execution time includes the estimation of the complete trajectory of the targets (100 time steps).

Figure 4.17: Mean execution time with respect to the number of particles for the different algorithms when $t = 6$ targets are present at the scenario. Execution times consider the estimation of the 100 steps of the trajectories with the RSSI measurement model in Section 4.4.1.

Figure 4.18: PP, APP, TRAPP, ATRAPP, MPF, MAPF, C-PF-PROP, SP-MPF and SP-MAPF OSPA error with respect to the execution time with $t = 6$ targets present at the scenario using the RSSI measurement model in Section 4.4.1. The execution time includes the estimation of the complete trajectory of the targets (100 time steps).

Regarding the performance of the different algorithms with respect to the measurement noise, Table 4.2 shows the OSPA position errors in a scenario with 3 targets of the different multiple particle filter algorithms using different measurement noise variances with the RSSI measurement model described in Section 4.4.1. It can be observed that for low ($\sigma_n^2 = 0.04$) and mild ($\sigma_n^2 = 1$) measurement noise, SP-MAPF outperforms SP-MPF. However, for high measurement noise ($\sigma_n^2 = 100$) the best performing filter is switched and SP-MPF outperforms SP-MAPF. As previously explained, this is caused by the informative measurements occurring under low and mild measurement noise variance, as it is in these cases where the use of an auxiliary variable in particle filters is best advised [130]. Note that, in any case, either SP-MPF or SP-MAPF are the overall best performing filter for any given measurement noise variance.

Up to this point in this section, all the described results have considered the use of the RSSI measurement model in Section 4.4.1. Nonetheless, observing Figures 4.2 to 4.5 it is shown how SP-MPF and SP-MAPF are also clearly the best performing methods when considering the use of the additive received power measurement model in Section 3.5.2. In this case, it should be noted that the methods presented in this chapter have also a remarkable performance when compared to the other filters. This

is specially noteworthy in the most demanding scenarios when a higher number of targets is present (see Figures 4.4 and 4.5), with a great performance of SP-MPF and SP-MAPF in these situations. This effect can also be clearly observed in Figure 4.10(a). Results for the SP-MAPF in this case are specially outstanding, as the filter achieves quite a low error even with a very low number of particles (again, see Figures 4.4 and 4.5). It is worth noting that the C-PF-PROP method cannot be considered in this case, as the required analytical expressions to compute the marginalization integrals are only available for the RSSI measurement model [15].

4.5 Conclusions

In this chapter, it has been pointed out that the performance of existing multiple particle filtering methods in the literature can be improved by performing a more accurate marginalization procedure. To this end, we have detailed the introduction of auxiliary filtering in the multiple filtering framework in the MAPF method. Two new algorithms have also been introduced, SP-MPF and SP-MAPF, which fit in the multiple particle filtering paradigm and provide a second-order approximation to the involved marginal PDFs. The proposed algorithms are valid for additive Gaussian models, do not require the calculation of Jacobians and make use of sigma-point integration methods for accurate marginalization. This enables them to outperform MPF and C-PF-PROP due to a higher accuracy in the marginalization step. In addition, compared to C-PF-PROP, the presented methods have the advantage of being directly applicable to a wider range of models.

The performance of the presented methods has been evaluated in a high-dimensional multiple target tracking problem, demonstrating that SP-MPF and SP-MAPF widely outperform previously presented algorithms in the literature.

Part II

Sensor management and fully adaptive radar

Chapter 5

Fully adaptive radar: fundamentals and applications

The capacity of modern radar and surveillance systems to dynamically change their operating configuration has recently brought a great interest over the field of sensor management [59, 73, 74]. The core idea is to use the available information from the scenario in which a system is deployed to adaptively change its configuration with the intention of achieving a performance improvement. It is worth noting that the literature often also refers to this problem as a sensor selection problem, assuming the availability of a set of virtual sensors, each one representing a possible configuration of the physical available sensors in the system [73].

The adaptation of a system to its operating situation can be carried out through a wide variety of actions, depending on the versatility of its sensors. Typical applications include choosing from different available modes of operation, like agile selection of the most appropriate waveform to be transmitted [62, 126, 150], resource scheduling [29], or dynamic optimization of the sensor position and orientation [91].

The recent advances in the use of multi-function radars have also enhanced the interest in the field of sensor management [59]. A natural problem in such systems is to determine the best possible system configuration and resource allocation to its different and possibly simultaneous modes of operation, which depends on the variable requirements of the surrounding scenario.

The sensor management problem has been previously tackled in the literature making use of a wide variety of approaches, including, among others, control theory [162], Markov decision processes (MDPs) and partially observed MDPs (POMDPs) [162] or game theory [147]. Inspired by the neuroscience approach to decision making and the neurobiological processes of animal cognition, two novel approaches to the

problem of sensor management have also been recently proposed, namely the cognitive radar [59, 67, 70, 71, 73, 74] and fully adaptive radar (FAR) [17–19]. Before any further discussions are made, given that there exists some lack of clear frontiers in the literature in the definition of the cognitive radar and fully adaptive radar, we find it useful to first go into some depth in these approaches.

Cognition allows to “pay attention to stimuli, whether external or internal, to identify these stimuli, and to plan meaningful responses to them” [87]. In order to catalog a system as *cognitive* [71] the following fundamental building blocks or cognitive functions described in [48] are required:

Perception-action cycle. The propagation of information about the environment is made available to the system in a cyclic basis. The perception-action cycle allows interpreting stimuli and planning responses at each time step. A system implementing the perception-action cycle is therefore able of adapting to its environment to ensure it meets its performance goals.

Memory. This function is narrowly related to the learning capacity of a cognitive system. A cognitive system needs to learn from the consequences of its own past goal-directed actions resulting from interactions of the system with the environment. Learning (and therefore memory) of artificial systems is a widely studied problem in the field of artificial intelligence and can be achieved for example by considering the inclusion in the system of some machine learning strategy like for example, an artificial neural network.

Attention. This is the process of selectively concentrating on a discrete aspect of information, while ignoring other information. A distinction relevant to our field is that made between perceptual attention and executive attention [71, 135, 188]. Perceptual attention refers to the processes of stimulus selection, while on the other hand, executive attention inhibits inappropriate responses and activates appropriate ones.

Intelligence. This function takes into account all previous building blocks in order to apply the best possible goal-directed action, considering all possible uncertainties in the system as well as in the environment.

Language. It determines how a system exchanges information with other systems. This final feature of cognition is often disregarded in the type of problems tackled in this thesis, although it can play an important role in other type of problems.

According to [48], the memory function of cognition is driven by the perception–action cycle; attention is usually memory driven; while intelligence generally builds on the

three previous functions. Literature on cognitive radar systems generally considers systems which implement at least the first 4 cognitive functions described above, while the fully adaptive radar framework presented in [17–19] is mainly concerned with the formalization in a simple but yet complete scheme of the complexity of the perception-action cycle, which as previously argued, is at the core of cognition. Nevertheless, the accommodation of some of the other cognitive functions above described are usually considered by some authors in the FAR framework [153–155].

Throughout the second part of this thesis, we build on the approach of the FAR framework to sensor management. In this chapter, the general formulation of the FAR framework is first reviewed [17–19] in Section 5.1. Various contributions in the literature within this framework, which consider different objectives to be addressed by a sensor network, are reviewed in this chapter. In particular, the use of the FAR framework is first illustrated by its application to a sensor network engaged in single target tracking [17] in Section 5.2. The problem of simultaneous single target detection and tracking in the FAR framework [18, 19] is also reviewed in Section 5.3. After considering these contributions in the literature of the FAR framework, we consider the problem of MTT with fixed and known number of targets within the FAR framework, which is a contribution of this thesis. The contents regarding this latter work have been published in [169].

5.1 The fully adaptive radar framework

As previously exposed, the fully adaptive radar framework [17–19] aims to take advantage of the available information about the environment in which the system is deployed to improve its performance, typically reducing the uncertainty of the results or optimizing the use of the available resources. As in the first part of this thesis, we follow the approach of dynamic systems, where the dynamic and measurement models are described by

$$\mathbf{x}^k = \mathbf{f}^k(\mathbf{x}^{k-1}, \mathbf{w}^{k-1}) \quad (5.1)$$

$$\mathbf{z}^k = \mathbf{h}^k(\mathbf{x}^k, \boldsymbol{\theta}^k, \mathbf{v}^k(\boldsymbol{\theta}^k)) \quad (5.2)$$

where $\mathbf{x}^k$ and $\mathbf{z}^k$ are respectively the state and observation vectors of the dynamic system at time k , $\mathbf{f}(\cdot, \cdot)$ and $\mathbf{h}(\cdot, \cdot, \cdot)$ can be nonlinear functions, $\mathbf{w}^{k-1}$ is the process noise at time $k - 1$, and $\mathbf{v}^k(\boldsymbol{\theta}^k)$ is the measurement noise at time k . Noise vectors at each time step are assumed to be independent. Note in (5.2) that we consider the case of a system in which the measurement model as well as the measurement noise,

Figure 5.1: FAR scheme. The perception-action cycle allows for the incorporation of the environment information in the controller, which decides the sensor parameters that better suit the predicted necessities of the system. A delay of 1 step is considered so that time step references are sound.

and thus, the quality of the measurements at time k , depend on the sensor parameters $\boldsymbol{\theta}^k \in \mathbb{R}^{n_{\boldsymbol{\Theta}}}$, with $n_{\boldsymbol{\Theta}}$ being the dimension of the sensor parameter space. In the FAR framework, $\boldsymbol{\theta}^k$ is a parameter of the sensor system which is selected at each time step k to maximize estimation performance.

As in the rest of this thesis, the Bayesian setting is adopted. Thus, we aim to recursively approximate the state posterior probability density function (PDF) $p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$, where $\boldsymbol{\theta}^{1:k}$ represents the sequence of sensor parameters from time 1 to time k .

Note that the posterior PDF $p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ will generally depend on the selected sensor parameters. Thus, as previously discussed, the objective is to select the best possible configuration of sensor parameters $\boldsymbol{\theta}^k$ at each time step so that the posterior $p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ allows for a better estimation of the state $\mathbf{x}^k$. The FAR framework achieves this objective using the perception-action cycle [17, 71] with actions (changes in the system configuration, modeled by different selections of $\boldsymbol{\theta}^k$) being taken according to the current knowledge of the environment. We now describe the main blocks of the FAR framework, which are shown in Figure 5.1:

Environment. Provides the dynamic model $p(\mathbf{x}^k | \mathbf{x}^{k-1})$ and the initial PDF of the state $p(\mathbf{x}^0)$.

Sensor. Takes measurements according to the measurement model in (5.2). It also provides both the likelihood function and a sensor cost function $R_\Theta(\boldsymbol{\theta}^k)$ which models the cost of using a given sensor configuration $\boldsymbol{\theta}^k$.

Processor. Computes the prior and posterior PDFs and uses a cost function $C(\hat{\mathbf{x}}^k, \mathbf{x}^k)$ in order to produce an estimate $\hat{\mathbf{x}}^k$ of the state of the dynamic system.

Controller. Considers the prior PDF and the sensor cost function in order to select the sensor configuration $\boldsymbol{\theta}^k$ for the next time step. The method described in Section 6.2.1 is executed in this block.

We now proceed to describe how the blocks in Figure 5.1 interact with each other, providing the functionality of the FAR framework. As explained in Chapter 2, making use of Bayesian filtering, given an initial state PDF $p(\mathbf{x}^0)$, the posterior PDF of the state at time k based on the measurements up to time k , can be obtained by sequentially applying a two-step recursion: prediction and update [11]. Then, the predicted and posterior PDFs of the state are respectively computed using the Chapman-Kolmogorov and Bayes equations, as in (2.4) and (2.5), but including the dependencies on the sensor parameters $\boldsymbol{\theta}^k$, as

$$p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = \int p(\mathbf{x}^k | \mathbf{x}^{k-1}) p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^{k-1} \quad (5.3)$$

$$p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}). \quad (5.4)$$

This procedure is executed in the processor block in Figure 5.1, commonly relying on some approximations such as a particle filter, or a Kalman filter. The estimation of the state $\hat{\mathbf{x}}^k$ in the processor is found minimizing the conditional Bayes risk, $\mathcal{R}_C$, which is the expected value of a processor cost function $C(\hat{\mathbf{x}}^k, \mathbf{x}^k)$ with respect to the posterior PDF of the state [176, (Chapter 2.4.1)]

$$\mathcal{R}_C(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) \triangleq \int C(\hat{\mathbf{x}}^k, \mathbf{x}^k) \cdot p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) d\mathbf{x}^k. \quad (5.5)$$

However, sensor parameters $\boldsymbol{\theta}^k$ need to be chosen before the measurement $\mathbf{z}^k$ is available, so the minimization of the conditional Bayes risk cannot be used in the controller. Instead, a prediction needs to be taken into account. For this purpose, we consider the joint PDF of $\mathbf{z}^k$ and $\mathbf{x}^k$, which can be obtained as a function of the prior

density, as

$$p(\mathbf{x}^k, \mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k}) = p(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}). \quad (5.6)$$

The predicted conditional Bayes risk of the estimation $\hat{\mathbf{x}}^k$, $\mathcal{R}_C^\uparrow$, takes the expectation of the processor cost function with respect to the previous joint density as

$$\mathcal{R}_C^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \triangleq \int \int C(\hat{\mathbf{x}}^k, \mathbf{x}^k) p(\mathbf{x}^k, \mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k}) d\mathbf{x}^k d\mathbf{z}^k, \quad (5.7)$$

which is a function of the past measurements $\mathbf{z}^{1:k-1}$, the past sensor parameters $\boldsymbol{\theta}^{1:k-1}$ and the next sensor parameters $\boldsymbol{\theta}^k$ (it is not a function of $\mathbf{x}^k$ and $\mathbf{z}^k$ since they have been averaged in the expectation integral). The selection of the next sensor parameters minimizes a loss function, which considers both the predicted conditional Bayes risk and the sensor cost function

$$L_{C,\Theta}(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \equiv L\left\{\mathcal{R}_C^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}), R_\Theta(\boldsymbol{\theta}^k)\right\}. \quad (5.8)$$

It is important to note that the approach to minimize the loss function in (5.8) will generally depend on the specifics of the problem as well as the sensor cost function $R_\Theta(\boldsymbol{\theta}^k)$. A commonly used and tractable setting [19] is to assume that (5.8) additively combines the predicted conditional Bayes risk and the sensor cost function as

$$L_{C,\Theta}(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = \mathcal{R}_C^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) + R_\Theta(\boldsymbol{\theta}^k), \quad (5.9)$$

with the sensor cost function $R_\Theta(\boldsymbol{\theta}^k)$ being constant, provided that $\boldsymbol{\theta}^k$ lies in an admissible set Θ , and infinite otherwise [19]

$$R_\Theta(\boldsymbol{\theta}^k) = \begin{cases} K & \boldsymbol{\theta}^k \in \Theta \\ \infty & \boldsymbol{\theta}^k \notin \Theta \end{cases}. \quad (5.10)$$

where K is a constant.

In such cases, the selection problem of the best possible sensor parameters at the next time step $\boldsymbol{\theta}^k$ can be expressed as a constrained optimization problem

$$\boldsymbol{\theta}^k = \arg \min_{\boldsymbol{\theta}} \mathcal{R}_C^\uparrow(\boldsymbol{\theta} | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \quad \text{s.t.} \quad \boldsymbol{\theta} \in \Theta \quad (5.11)$$

where s.t. stands for “subject to”. Unfortunately, as detailed in [17–19], it is not generally easy to directly tackle the minimization of (5.11) in a general setting, so that alternative approaches or approximations are required.

The following section briefly indicates how the important problem of resource management in a sensor network can be adequately posed in the FAR framework.

5.1.1 Resource management in FAR

The problem of resource management within the FAR framework using a sensor network with N_s sensors is briefly described in this section. The vector of sensor parameters in this case resembles the amount of resources allocated to each sensor at time k , $\boldsymbol{\theta}^k = [\theta_1^k, \dots, \theta_{N_s}^k]^T$. It is considered that the quality of the measurements received by a sensor depends on the amount of resources assigned to the sensor. The possibly nonlinear measurement equation in (5.2) for the s -th sensor at time k can therefore be expressed as

$$z_s^k = h_s(\mathbf{x}^k) + \frac{v_s^k}{\sqrt{\theta_s^k}} \quad (5.12)$$

where the measurement noise at the s -th sensor, v_s^k , is zero-mean Gaussian-distributed with variance σ_n^2 and $\theta_s^k > 0$ is the amount of resources allocated to the s -th sensor. The likelihood function for each sensor is therefore

$$p(z_s^k | \mathbf{x}^k; \theta_s^k) = \sqrt{\theta_s^k / (2\pi\sigma_n^2)} \cdot \exp \left[-\frac{(z_s^k - h_s(\mathbf{x}^k))^2}{2\sigma_n^2 / \theta_s^k} \right]. \quad (5.13)$$

The amount of total resources is constrained to a limited budget b , which is shared among all the sensors in the sensor network. Thus, following the approach indicated by (5.10), the sensor cost function $R_\Theta(\boldsymbol{\theta}^k)$ is specified by defining an admissible set of sensor parameters Θ as

$$\boldsymbol{\theta}^k \in \Theta \iff \begin{cases} \theta_s^k > 0 & s = 1, \dots, N_s, \\ \sum_{s=1}^{N_s} \theta_s^k \leq b \end{cases}, \quad (5.14)$$

so that the next set of sensor parameters can be obtained by performing the constrained minimization of the predicted conditional Bayes risk in (5.11).

5.2 Single target tracking in fully adaptive radar

The application of the FAR framework to the problem of sensor parameter selection for single target tracking [17] is reviewed in this section. First, a suitable cost function for this problem is specified, and its predicted conditional Bayes Risk in (5.11) is considered. However, it is shown how this approach is not of practical use as it

is not possible to analytically evaluate the obtained expression. Nevertheless, as shown in Section 5.2.1, the predicted conditional Bayes risk can be bounded using the predicted conditional Cramér-Rao lower bound (PC-CRLB), so that sensor parameters can therefore be finally chosen by minimizing the PC-CRLB.

A common cost function in single target tracking is the squared error, which is given by

$$C(\hat{\mathbf{x}}^k, \mathbf{x}^k) = (\hat{\mathbf{x}}^k - \mathbf{x}^k)^T (\hat{\mathbf{x}}^k - \mathbf{x}^k) \quad (5.15)$$

$$= \text{tr} \{ [\hat{\mathbf{x}}^k - \mathbf{x}^k] [\hat{\mathbf{x}}^k - \mathbf{x}^k]^T \}. \quad (5.16)$$

In the following, we will make use of this latter expression in (5.16) of the cost function. The conditional Bayes risk in (5.5) is therefore the mean squared error (MSE)

$$\mathcal{R}_C(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) = \int \text{tr} \{ [\hat{\mathbf{x}}^k - \mathbf{x}^k] [\hat{\mathbf{x}}^k - \mathbf{x}^k]^T \} \cdot p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) d\mathbf{x}^k. \quad (5.17)$$

The optimal estimator $\hat{\mathbf{x}}^k$ that minimizes the conditional Bayes risk in (5.17) is the minimum MSE (MMSE) estimator, which is the conditional mean

$$\hat{\mathbf{x}}^k \triangleq \int \mathbf{x}^k \cdot p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) d\mathbf{x}^k. \quad (5.18)$$

On the other hand, given the cost function in (5.16), the predicted conditional Bayes risk in (5.7) takes the form

$$\mathcal{R}_C^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = \text{tr} \{ \boldsymbol{\Sigma}^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \} \quad (5.19)$$

where the predicted conditional MSE matrix, $\boldsymbol{\Sigma}^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$, is obtained with respect to the joint density in (5.6) as

$$\boldsymbol{\Sigma}^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = \int \int ([\hat{\mathbf{x}}^k - \mathbf{x}^k] [\hat{\mathbf{x}}^k - \mathbf{x}^k]^T) \cdot p(\mathbf{x}^k, \mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k d\mathbf{z}^k. \quad (5.20)$$

5.2.1 The predicted conditional Cramér-Rao lower bound

The evaluation of $\boldsymbol{\Sigma}^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$ in (5.20) is not always analytically tractable, so that instead, the use of its PC-CRLB is proposed in [17]. The PC-CRLB is the

inverse of the predicted conditional Bayesian information matrix (PC-BIM)

$$\mathbf{B}_k^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = -E \left[\Delta_{\mathbf{x}}^{\mathbf{x}} \ln p(\mathbf{x}^k, \mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \right] \quad (5.21)$$

where the expectation is computed with respect to the joint density in (5.6) and with

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_{n_x}} \right]^T \quad (5.22)$$

and

$$\Delta_{\mathbf{x}}^{\mathbf{x}} f(\mathbf{x}) = \nabla_{\mathbf{x}} [\nabla_{\mathbf{x}} f(\mathbf{x})]^T, \quad (5.23)$$

respectively denoting the gradient of a function $f(\mathbf{x})$ with respect to the components of a vector $\mathbf{x} \in \mathbb{R}^{n_x}$ and the Hessian matrix (which contains the second-order partial derivatives with respect to the elements of $\mathbf{x}$) of the function $f(\mathbf{x})$.

Thus, given the cost function in (5.19) and under some mild regularity conditions over the involved functions [85], the PC-CRLB meets¹

$$\mathcal{R}_C^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \geq \text{tr} \left\{ \mathbf{B}_k^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\}, \quad (5.24)$$

so that, based on (5.24), instead of minimizing the predicted conditional Bayes risk as in (5.11), the minimization of the PC-CRLB is proposed to select the optimal $\boldsymbol{\theta}^k$ for the next time step. Note that the joint density in (5.6) is considered and not just the likelihood function, as in common CRLB formulations for parameter estimation. This is due to the fact that in this setting the measurements and, also, the state are modeled as being stochastic [136].

As stated in [176], the PC-BIM in (5.21) can be decomposed as the sum of a prior term and a data term as

$$\mathbf{B}_k^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = \mathbf{B}_k^-(\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) + \mathbf{J}_k^-(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \quad (5.25)$$

where the first term, the predicted information matrix (PIM), does not depend on the next set of sensor parameters $\boldsymbol{\theta}^k$, and is defined as

$$\mathbf{B}_k^-(\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \equiv -E^- \left[\Delta_{\mathbf{x}}^{\mathbf{x}} \ln p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \right], \quad (5.26)$$

with E^- meaning that in this case the expectation is taken with respect to the prior density $p(\mathbf{x}^k | \mathbf{z}^{1:k-1}, \boldsymbol{\theta}^{1:k-1})$. In the linear-Gaussian case, the PIM is equal to the inverse

¹Common formulations of a matrix lower bound are of the form $\mathbf{B} \succeq \mathbf{A}$ (where the use of the symbol $\succeq$ is used to indicate that the matrix $\mathbf{B} - \mathbf{A}$ is positive semidefinite). Equivalently, the bound can be expressed as $\text{tr}(\mathbf{B}) \geq \text{tr}(\mathbf{A})$.

of the predicted covariance matrix. However, in general, approximations have to be made when the PIM cannot be easily computed. If a discrete approximation to the posterior PDF is being used, as with a point mass filter or a particle filter, such that

$$p(\mathbf{x}^k | \mathbf{z}^{1:k-1}, \boldsymbol{\theta}^{1:k-1}) \approx \sum_{i=1}^N \omega_i^{k-1} \delta(\mathbf{x}^k - \mathbf{x}_i^k), \quad (5.27)$$

one can always easily approximate the PIM using the covariance of the samples of the discrete approximation as

$$\mathbf{B}_k^- (\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \approx \boldsymbol{\Sigma} (\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \quad (5.28)$$

where

$$\begin{aligned} \boldsymbol{\Sigma} (\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\equiv E^- \left[(\mathbf{x}_i^k - \bar{\mathbf{x}}^k) (\mathbf{x}_i^k - \bar{\mathbf{x}}^k)^T \right] \\ &\approx \sum_{i=1}^N \omega_i^{k-1} (\mathbf{x}_i^k - \bar{\mathbf{x}}^k) \cdot (\mathbf{x}_i^k - \bar{\mathbf{x}}^k)^T \end{aligned} \quad (5.29)$$

and

$$\bar{\mathbf{x}}^k \approx \sum_{i=1}^N \omega_i^{k-1} \mathbf{x}_i^k. \quad (5.30)$$

On the other hand, the data term in (5.25) is the expected value of the Fisher information matrix (EFIM)

$$\mathbf{J}_k^- (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \equiv E^- \left\{ -E \left[\Delta_{\mathbf{x}}^{\mathbf{x}} \ln p(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) \right] \right\}. \quad (5.31)$$

Assuming that there is an analytical expression of $p(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k)$, the inner expectation (5.31) can be evaluated, while the expectation with respect to the prior can be numerically computed. Once the Bayesian information matrix, $\mathbf{B}_k^- (\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$, and the EFIM, $\mathbf{J}_k^- (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$, are available, the PC-BIM in (5.25) can be computed.

As a final remark, in order to minimize the PC-CRLB with respect to $\boldsymbol{\theta}^k$, it might be useful to use the following expression to compute its gradient at the direction of the s -th sensor parameter θ_s^k as ([19, 175] App. A.7.3)

$$\begin{aligned} g_s(\boldsymbol{\theta}^k) &\equiv \frac{\partial}{\partial \theta_s^k} \left[\text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\} \right] \\ &= -\text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-2} \mathbf{J}_k^- (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \right\}. \end{aligned} \quad (5.32)$$

Note that the approach of minimizing the PC-CRLB in order to select the sensor

parameters $\boldsymbol{\theta}^k$ at the next time step requires to derive some analytical expressions for $\Delta_{\mathbf{x}}^{\mathbf{x}} \ln p(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k)$. In the following, we give details of the PC-CRLB derivation for the problem of resource management in a sensor network in Section 5.1.1.

5.2.1.1 PC-CRLB for resource management in a sensor network

The problem of resource management in a sensor network using the FAR framework, which has been previously introduced in Section 5.1.1, allows for the above expressions for the PC-CRLB to be further developed.

Considering that sensors measurements are independent as in (5.12), the likelihood function can be expressed as

$$p(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) = \prod_{s=1}^{N_s} p(z_s^k | \mathbf{x}^k; \theta_s^k), \quad (5.33)$$

so that, using (5.33) in (5.31) the EFIM becomes

$$\begin{aligned} \mathbf{J}_k^- (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\equiv E^- \left\{ -E \left[\Delta_{\mathbf{x}}^{\mathbf{x}} \ln (p(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k)) \right] \right\} \\ &= E^- \left\{ -E \left[\Delta_{\mathbf{x}}^{\mathbf{x}} \ln \left(\prod_{s=1}^{N_s} p(z_s^k | \mathbf{x}^k; \theta_s^k) \right) \right] \right\} \\ &= E^- \left\{ -E \left[\Delta_{\mathbf{x}}^{\mathbf{x}} \sum_{s=1}^{N_s} \ln (p(z_s^k | \mathbf{x}^k; \theta_s^k)) \right] \right\} \\ &= \sum_{s=1}^{N_s} \mathbf{J}_{k,s}^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \end{aligned} \quad (5.34)$$

where $\mathbf{J}_{k,s}^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$ is the s -th sensor EFIM at time k

$$\mathbf{J}_{k,s}^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \equiv E^- \left\{ -E \left[\Delta_{\mathbf{x}}^{\mathbf{x}} \ln (p(z_s^k | \mathbf{x}^k; \theta_s^k)) \right] \right\}, \quad (5.35)$$

which in this case, following the detailed derivation in Appendix A.4, can be expressed as

$$\mathbf{J}_{k,s}^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = E^- \left\{ \frac{\theta_s^k}{\sigma_n^2} \nabla_{\mathbf{x}} h_s(\mathbf{x}^k) [\nabla_{\mathbf{x}} h_s(\mathbf{x}^k)]^T \right\}. \quad (5.36)$$

It is worth recalling that if a discrete approximation to the prior PDF is available, like the one in (5.27), the remaining expectation in (5.36) can be numerically computed. The detailed derivation of the required expressions for this problem for two commonly used measurement models is covered in Appendices A.5 and A.6. Simulations exemplifying the use of the PC-CRLB in the FAR framework for single target tracking will be posed in Section 5.4.1.

5.3 Simultaneous target detection and tracking in fully adaptive radar

Building on the previous approach to single target tracking, in this section we briefly review the application of the FAR framework to the problem of simultaneous single target detection and tracking, following [18, 19]. The objective is to rapidly detect the presence or absence of a target in the scenario. In this context, achieving a correct decision in a fast manner is done by the maximizing the probability of making a correct decision at each time step [157]. If the target is present in the scenario, its position is also estimated.

It is worth noting that the processor needs to correctly handle three different but related objectives in the simultaneous target detection and tracking problem: the detection of the target, a precise tracking of its position, but also, and usually gone unnoticed, is the problem of track maintenance, i.e., once the presence of the target is confirmed the processor needs to adequately verify track maintenance in the scenario, i.e., to reliably make sure that the target is still present at the scenario during the duration of the target trajectory, and when the target leaves the scenario, to rapidly notice its absence.

To tackle the detection and track maintenance problems, at each time step, two hypotheses are considered: H_0 and H_1 . Hypothesis H_0 considers that there is no target in the scenario, i.e., the state of the target is $\mathbf{x}^k = \emptyset$, with $\emptyset$ being the null (or target absent) state. On the other hand, under hypothesis H_1 , the target is present in the scenario and its state takes values in $\mathbf{x}^k \in \mathbb{R}^{n_x}$. The state of the dynamic system therefore takes values in the augmented space $\mathbf{x}^k \in \mathbb{R}^{n_x} \cup \{\emptyset\}$.

5.3.1 Bayesian recursion for simultaneous target detection and tracking

A distribution of the state of the dynamic system in the augmented state space $\mathbb{R}^{n_x} \cup \{\emptyset\}$ at any given time k will be denoted as $\boldsymbol{\pi}(\mathbf{x}^k)$ and be characterized by two components, namely the probability of existence of the target r^k , and the conditional PDF of the state given that $\mathbf{x}^k \in \mathbb{R}^{n_x}$, $p(\mathbf{x}^k)$. Thus, the posterior PDF can be written as a Bernoulli distribution as [18, 19, 157]

$$\boldsymbol{\pi}(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) = (1 - r^k) \delta(\mathbf{x}^k - \emptyset) + r^k \cdot p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}). \quad (5.37)$$

with the initial PDF at time $k = 0$ being described by

$$\boldsymbol{\pi}(\mathbf{x}^0) = (1 - r^0) \delta(\mathbf{x}^k - \emptyset) + r^0 \cdot p(\mathbf{x}^0). \quad (5.38)$$

The state transition equation distinguishes the case when the target is absent and the case when the target is at the scenario at the previous time step. If the target is absent at time $k - 1$

$$\boldsymbol{\pi}(\mathbf{x}^k | \mathbf{x}^{k-1} = \emptyset) = (1 - p_b^k) \delta(\mathbf{x}^k - \emptyset) + p_b^k \cdot b^k(\mathbf{x}^k) \quad (5.39)$$

where p_b^k is the probability of a target being born at time k and $b^k(\mathbf{x}^k)$ stands for the state PDF of a target born at time k . On the other hand, if the target is present at the scenario at time $k - 1$, the state transition equation takes the form

$$\boldsymbol{\pi}(\mathbf{x}^k | \mathbf{x}^{k-1} \in \mathbb{R}^{n_x}) = (1 - p_S^{k-1}(\mathbf{x}^{k-1})) \delta(\mathbf{x}^k - \emptyset) + p_S^{k-1}(\mathbf{x}^{k-1}) \cdot p(\mathbf{x}^k | \mathbf{x}^{k-1}) \quad (5.40)$$

where $p_S^{k-1}(\mathbf{x}^{k-1})$ is the probability of a target with state $\mathbf{x}^{k-1}$ surviving from time $k - 1$ to time k and $p(\mathbf{x}^k | \mathbf{x}^{k-1})$ is the dynamic model of the surviving target.

Considering (5.39), (5.40) and the posterior at time $k - 1$, $\boldsymbol{\pi}(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$, the predicted PDF of the state at time k can be computed using the Chapman-Kolmogorov equation (5.3), giving rise also to a Bernoulli distribution

$$\boldsymbol{\pi}(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = (1 - r^{k-}) \delta(\mathbf{x}^k - \emptyset) + r^{k-} \cdot p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}), \quad (5.41)$$

with components

$$r^{k-} = (1 - r^{k-1}) p_b^k + r^{k-1} \int p_S^{k-1}(\mathbf{x}^{k-1}) p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^{k-1} \quad (5.42)$$

and

$$p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = \frac{1}{r^{k-}} [(1 - r^{k-1}) p_b^k b^k(\mathbf{x}^k) + r^{k-1} s^k(\mathbf{x}^k)] \quad (5.43)$$

where

$$s^k(\mathbf{x}^k) = \int p(\mathbf{x}^k | \mathbf{x}^{k-1}) p_S^{k-1}(\mathbf{x}^{k-1}) p(\mathbf{x}^{k-1} | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^{k-1}. \quad (5.44)$$

Once the measurement is collected, the update step uses Bayes rule as in (5.4) to compute the posterior PDF. However, we first need to take into account that the likelihood of measurements can be dependent on the existence of the target, thus it is necessary to consider the measurement model $\mathbf{h}(\emptyset)$ and the likelihood of a given

measurement for the situation of no target

$$l_{\emptyset}(\mathbf{z}^k | \emptyset; \boldsymbol{\theta}^k) \quad \mathbf{x}^k = \emptyset \quad (5.45)$$

and its counterpart when the target is in the scenario

$$l_x(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) \quad \mathbf{x}^k \in \mathbb{R}^{n_x}, \quad (5.46)$$

so that we can compute

$$p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k) = (1 - r^{k-}) l_{\emptyset}(\mathbf{z}^k | \emptyset; \boldsymbol{\theta}^k) + r^{k-} \int l_x(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k. \quad (5.47)$$

Thus, the posterior at time k is again a Bernoulli distribution

$$\boldsymbol{\pi}(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) = (1 - r^k) \delta(\mathbf{x}^k - \emptyset) + r^k \cdot p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}), \quad (5.48)$$

with components

$$r^k = \frac{r^{k-} \int l_x(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k}{p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k)}, \quad (5.49)$$

and

$$p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) = \frac{l_x(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})}{\int l_x(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k}. \quad (5.50)$$

The following alternative equivalent expression for the posterior probability of existence can also be useful

$$(1 - r^k) = \frac{(1 - r^{k-}) l_{\emptyset}(\mathbf{z}^k | \emptyset; \boldsymbol{\theta}^k)}{p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k)} \quad (5.51)$$

$$r^k = 1 - \frac{(1 - r^{k-}) l_{\emptyset}(\mathbf{z}^k | \emptyset; \boldsymbol{\theta}^k)}{p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k)}. \quad (5.52)$$

5.3.2 Estimation for target detection and tracking

The detection problem requires to decide which of the two hypothesis H_0 or H_1 is more likely true, given all past measurements $\mathbf{z}^{1:k}$. Thus, the state estimate takes one of two

values,

$$\hat{\mathbf{x}}^k(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) = \begin{cases} \emptyset & H_0 \\ \boldsymbol{\mu}^k \in \mathbb{R}^{n_x} & H_1 \end{cases} \quad (5.53)$$

where $\boldsymbol{\mu}^k$ is the conditional mean in (5.18).

The selected cost function in [18, 19] is the standard binary detection cost function [84, 176]

$$\begin{aligned} C(\hat{\mathbf{x}}^k, \mathbf{x}^k) &= C_{00} \mathbb{I} \{ \hat{\mathbf{x}}^k(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) = \emptyset, \mathbf{x}^k = \emptyset \} \\ &+ C_{10} \mathbb{I} \{ \hat{\mathbf{x}}^k(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) \in \mathbb{R}^{n_x}, \mathbf{x}^k = \emptyset \} \\ &+ C_{01} \mathbb{I} \{ \hat{\mathbf{x}}^k(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) = \emptyset, \mathbf{x}^k \in \mathbb{R}^{n_x} \} \\ &+ C_{11} \mathbb{I} \{ \hat{\mathbf{x}}^k(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) \in \mathbb{R}^{n_x}, \mathbf{x}^k \in \mathbb{R}^{n_x} \} \end{aligned} \quad (5.54)$$

where $\mathbb{I} \{ \cdot \}$ is the indicator function which is equal to 1 when its argument is true and zero otherwise. And C_{ij} is the cost of deciding H_i when H_j is true. In order to build the estimator $\hat{\mathbf{x}}^k(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$, we define the following Bayesian likelihood ratio

$$\begin{aligned} \bar{\Lambda}(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) &\equiv \frac{r^k}{(1 - r^k)} \\ &= \frac{r^{k-} \int l_x(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k}{(1 - r^{k-}) l_{\emptyset}(\mathbf{z}^k | \emptyset; \boldsymbol{\theta}^k)} \\ &= \frac{r^{k-}}{(1 - r^{k-})} \bar{\mathcal{L}}(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k}), \end{aligned} \quad (5.55)$$

with $\bar{\mathcal{L}}(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k})$ standing for the expected likelihood ratio for the received measurement $\mathbf{z}^k$

$$\bar{\mathcal{L}}(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k}) = \frac{\int l_x(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k}{l_{\emptyset}(\mathbf{z}^k | \emptyset; \boldsymbol{\theta}^k)}. \quad (5.56)$$

Under the assumption that $C_{10} > C_{00}$ and $C_{01} > C_{11}$, using the cost function in (5.54), the decision rule that minimizes the conditional Bayes risk in (5.5) is [84, 176]

$$\bar{\Lambda}(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) \underset{\hat{\mathbf{x}}^k = \emptyset}{\overset{\hat{\mathbf{x}}^k \in \mathbb{R}^{n_x}}{\geq}} \gamma, \quad (5.57)$$

with

$$\gamma = \frac{C_{10} - C_{00}}{C_{01} - C_{11}} \quad (5.58)$$

or alternatively, using the likelihood ratio for the received measurement

$$\bar{\mathcal{L}}(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k}) \stackrel{\substack{\hat{\mathbf{x}}^k \in \mathbb{R}^{n_x} \\ \hat{\mathbf{x}}^k = \emptyset}}{\geq} \gamma \cdot \frac{1 - r^{k-}}{r^{k-}}. \quad (5.59)$$

5.3.3 Predicted Bayes risk for target detection and tracking

It is common to assume that taking the correct decision has no associated cost, i.e., $C_{11} = C_{00} = 0$. Under this simplification, we can write the predicted conditional Bayes risk in (5.7) as

$$\begin{aligned} \mathcal{R}_C^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &= \int \int C(\hat{\mathbf{x}}^k, \mathbf{x}^k) p(\mathbf{x}^k, \mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k}) d\mathbf{x}^k d\mathbf{z}^k \\ &= C_{10}(1 - r^{k-}) P_F(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \\ &\quad + C_{01} r^{k-} P_M(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \end{aligned} \quad (5.60)$$

where $P_F(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$ is the probability of producing a false alarm (i.e., deciding H_1 when H_0 is true) using the threshold γ in (5.57)

$$\begin{aligned} P_F(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &= E[\mathbb{I}\{\bar{\Lambda}(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) > \gamma, \mathbf{x}^k = \emptyset\}] \\ &= \Pr(\bar{\Lambda}(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) > \gamma, \mathbf{x}^k = \emptyset) \end{aligned} \quad (5.61)$$

and $P_M(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$ is the probability of having a missed detection (i.e., deciding H_0 when H_1 is true), which is averaged over the predicted density in $\mathbb{R}^{n_x}$

$$\begin{aligned} P_M(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &= E^- \{E[\mathbb{I}\{\bar{\Lambda}(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) < \gamma, \mathbf{x}^k \in \mathbb{R}^{n_x}\}]\} \\ &= E^- \{\Pr(\bar{\Lambda}(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) < \gamma, \mathbf{x}^k \in \mathbb{R}^{n_x})\}. \end{aligned} \quad (5.62)$$

Computation of these probabilities is generally difficult, so that in [18, 19] a surrogate function is instead minimized in order for the controller to choose the next set of sensor parameters $\boldsymbol{\theta}^k$. In particular, it is proposed to employ the minimization of the PC-CRLB as in (5.24), thus maximizing Bayesian information, which is argued to be useful in order to make statistical inferences about $\mathbf{x}^k$ in both the detection and estimation settings. The idea is not only to maximize detection performance with the selection of $\boldsymbol{\theta}^k$, but also to try to concentrate $\boldsymbol{\pi}(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ so that the tracking error is also kept low. The suitability of this approach is analyzed in Section 6.3, showing that it might be ill conditioned to adequately tackle the problem of target detection.

5.4 Multiple target tracking with fixed and known number of targets in fully adaptive radar

This section illustrates the application of the FAR framework to the problem of multiple target tracking (MTT) [11]. The general MTT problem is usually posed considering a varying and unknown number of targets, so that we have to deal with targets appearing and disappearing from the scenario while tracking the state of the targets currently in the scenario, i.e., MTT requires the estimation of the number of targets in the surveillance area as well as their kinematic features. In this general setting, a sensor management strategy therefore would have to deal with target detection, tracking and track maintenance necessities. In order to ease the controller design, in this section we assume that a fixed and known number of targets t is present in the scenario where a sensor network is deployed.

As previously mentioned in the first part of this thesis, MTT is a challenging problem even if a fixed and known number of targets is considered, as the curse of dimensionality naturally arises when the number of targets under track is high, rapidly degrading the accuracy in the estimation of the state of the targets. Ensuring the best possible quality of the provided measurements in a sensor management scenario can therefore be really useful in tackling such a difficult problem. In this section we develop a FAR controller for the problem of resource management to track a fixed and known number of targets in a sensor network. The main results of this section were presented in [169].

The Bayesian recursion for the estimation at time k of a multitarget state $\mathbf{X}^k$ (where $\mathbf{X}^k$ contains the individual state of all t targets in the scenario as in (2.98)) has been extensively covered in the first part of this thesis. Nevertheless, in this case, the dependencies with respect to the sensor parameters needs to be taken into account, so that the predicted and updated PDFs become

$$p(\mathbf{X}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = \int p(\mathbf{X}^k | \mathbf{X}^{k-1}) p(\mathbf{X}^{k-1} | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{X}^{k-1} \quad (5.63)$$

and

$$p(\mathbf{X}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) \propto p(\mathbf{z}^k | \mathbf{X}^k; \boldsymbol{\theta}^k) p(\mathbf{X}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}). \quad (5.64)$$

Definition of a processor cost function is also required. As previously pointed out in the first part of this thesis, association of tracks to estimated target positions is a key aspect of the estimation process in MTT, so that common cost functions for MTT like OSPA [145] usually take this feature into account. In this case, we do not stress target labeling as an estimation objective, so that the allocation of resources does not depend

on the association procedure. Nonetheless, for the sake of simplicity and tractability the processor cost function is chosen to be again the sum of squared errors

$$C(\hat{\mathbf{X}}^k, \mathbf{X}^k) = \text{tr} \left\{ \left[\hat{\mathbf{X}}^k - \mathbf{X}^k \right] \left[\hat{\mathbf{X}}^k - \mathbf{X}^k \right]^T \right\}. \quad (5.65)$$

Although the use of this cost function for the problem of MTT comes with some disadvantages, which will be later discussed, it has the great advantage of enabling the use of the derivations previously shown in Section 5.2. We will therefore rely on the minimization of the PC-CRLB in order to choose the best possible set of sensor parameters at each time step. Following from the PC-BIM decomposition in (5.25), the PIM can be computed as

$$\mathbf{B}_k^- (\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \equiv -E^- \left[\Delta_{\mathbf{X}}^{\mathbf{X}} \ln p(\mathbf{X}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \right], \quad (5.66)$$

while, provided that sensors independently take measurements, the EFIM can be expressed as a sum of sensor EFIMs (see 5.2.1.1) as

$$\mathbf{J}_k^- (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \equiv E^- \left\{ -E \left[\Delta_{\mathbf{X}}^{\mathbf{X}} \ln p(\mathbf{z}^k | \mathbf{X}^k; \boldsymbol{\theta}^k) \right] \right\} \quad (5.67)$$

$$\begin{aligned} &= \sum_{s=1}^{N_s} \mathbf{J}_{k,s}^- (\boldsymbol{\theta}_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \\ \mathbf{J}_{k,s}^- (\boldsymbol{\theta}_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\equiv E^- \left\{ -E \left[\Delta_{\mathbf{X}}^{\mathbf{X}} \ln (p(\mathbf{z}_s^k | \mathbf{X}^k; \boldsymbol{\theta}_s^k)) \right] \right\}. \end{aligned} \quad (5.68)$$

It is worth recalling that if a discrete approximation to the prior PDF is available

$$p(\mathbf{X}^k | \mathbf{z}^{1:k-1}, \boldsymbol{\theta}^{1:k-1}) \approx \sum_{i=1}^N \omega_i^{k-1} \delta(\mathbf{X}^k - \mathbf{X}_i^k), \quad (5.69)$$

the expectation $E^- \{\cdot\}$ can be numerically computed. In the case of the PIM in (5.65), it can be approximated by the predicted discrete covariance matrix as

$$\mathbf{B}_k^- (\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \approx \sum_{i=1}^N \omega_i^{k-1} (\mathbf{X}_i^k - \bar{\mathbf{X}}^k) \cdot (\mathbf{X}_i^k - \bar{\mathbf{X}}^k)^T. \quad (5.70)$$

In a resource management problem with a measurement model of the form (5.12), and following the derivation in Appendix A.4, we can express (5.68) as

$$\mathbf{J}_s^- (\boldsymbol{\theta}_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = E^- \left\{ \frac{\theta_s^k}{\sigma_n^2} \nabla_{\mathbf{X}} h_s(\mathbf{X}^k) [\nabla_{\mathbf{X}} h_s(\mathbf{X}^k)]^T \right\} \quad (5.71)$$

where given the partitioned state in (2.98) the gradient with respect to $\mathbf{X}^k$ will have

the following structure

$$\nabla_{\mathbf{X}} h_s(\mathbf{X}^k) = \left[(\nabla_{\mathbf{x}_1} h_s(\mathbf{X}^k))^T, \dots, (\nabla_{\mathbf{x}_t} h_s(\mathbf{X}^k))^T \right]^T. \quad (5.72)$$

In addition, if an additive measurement model is considered, so that

$$h_s(\mathbf{X}^k) = \sum_{j=1}^t h_{s,j}(\mathbf{x}_j^k), \quad (5.73)$$

the gradient in (5.72) can be further simplified as

$$\nabla_{\mathbf{X}} h_s(\mathbf{X}^k) = \left[(\nabla_{\mathbf{x}_1} h_{s,1}(\mathbf{x}_1^k))^T, \dots, (\nabla_{\mathbf{x}_t} h_{s,t}(\mathbf{x}_t^k))^T \right]^T. \quad (5.74)$$

Considering (5.74), it is useful to define

$$\begin{aligned} \mathbf{G}_s &\equiv \nabla_{\mathbf{X}} h_s(\mathbf{X}^k) [\nabla_{\mathbf{X}} h_s(\mathbf{X}^k)]^T \\ &= \begin{bmatrix} \mathbf{G}_s^{1,1} & \mathbf{G}_s^{1,2} & \dots & \mathbf{G}_s^{1,t} \\ \mathbf{G}_s^{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{G}_s^{t,1} & \dots & \dots & \mathbf{G}_s^{t,t} \end{bmatrix}, \end{aligned} \quad (5.75)$$

with the matrix block

$$\mathbf{G}_s^{j,l} = \nabla_{\mathbf{x}_j} h_{s,j}(\mathbf{x}_j^k) [\nabla_{\mathbf{x}_l} h_{s,l}(\mathbf{x}_l^k)]^T, \quad (5.76)$$

so that the sensor EFIM in (5.71) can be expressed as

$$\mathbf{J}_s^-(\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = \frac{\theta_s^k}{\sigma_n^2} E^- \{ \mathbf{G}_s \} \quad (5.77)$$

where it is worth noting that the expectation with respect to the prior density in (5.77) can be computed independently for each block in (5.76).

As a final remark, it is worth emphasizing that the estimator $\widehat{\mathbf{X}}^k$ resulting from the minimization of the conditional Bayes risk in (5.5) with the cost function in (5.65) may give rise to some problems when applied to the MTT problem. This approach generally works well when targets have always been far from each other, as $\widehat{\mathbf{X}}^k$ can correctly estimate the positions of the targets (giving estimates where the targets actually are) and, if required, it could correctly extract target labeling information.

Estimation problems might arise nevertheless when target trajectories cross, or more problematically, when target trajectories separate after remaining close to each other for a relatively long period of time (i.e., when the posterior PDFs of the targets significantly overlap in all state parameters). In these cases, the posterior PDF of the state of each target $\mathbf{x}_j^k$ involved in the above described difficult situation can become multimodal, so that under the cost function in (5.65), the estimation of the positions of the targets in such cases will lie in the middle of the true positions of the targets [23, 24, 49].

The above issue associated with the cost function selection will definitely be a limitation for the application of the above described framework and a main way for improvement in future work. Nevertheless, we can, at least partially, overcome this limitation by making use of a particle filter to propagate the posterior PDF of $\mathbf{X}^k$.

Particle filters have the advantage of being able to approximate arbitrary PDFs. However the necessity of the resampling stage to overcome particle degeneracy (see Section 2.7.1.1) implies that particle filters cannot maintain multimodalities in the posterior PDF for large periods of time [23, 24, 121, 122, 179]. Thus, one of the modes of the PF will be lost due to particle impoverishment. This feature is generally a disadvantage of PF methods, as the generally desired purpose would be to propagate the true multimodal PDFs of the targets (note that it is in this multimodality where the information about target labeling lies). However, provided that we are not interested in target labeling, we can stand the vanishing of modes of the PDFs, as after the disappearance of some modes of the multimodal PDFs, the PF will generally tend to keep unimodal densities with one mode centered in the true position of each target, from where we can adequately extract its estimated position. See Appendix A.3 for a further discussion on this aspect.

In Appendix A.7, the derivation of the PC-CRLB for an MTT problem with the additive received power measurement model in Section 3.5.2 is briefly analyzed.

5.4.1 Experimental results

This section illustrates the previous concepts of FAR applied to the problem of MTT with a fixed and known number of targets. Section 5.4.1.1 briefly describes the considered dynamic and measurement models, and Section 5.4.1.2 presents the simulation results.

In contrast to the point mass filter used in the application of the FAR framework in [17], the APP PF [163] (see Section 3.1) with $N = 300$ particles is selected as the Bayesian state tracking algorithm. The selection of an algorithm that can handle high

Figure 5.2: Simulated target trajectories and sensors. Trajectories, consist of 100 steps sampled with a period of $\tau = 1s$. The initial and final points of each trajectory are respectively indicated by a colored circle and square. Arrows show the position and direction of the target every 20 time steps. Sensor positions are indicated with a red dot.

dimensional state spaces is a crucial point when considering multiple targets, otherwise the curse of dimensionality would severely affect the performance of the system.

5.4.1.1 Dynamic and measurement model

Simulations consider up to 4 targets simultaneously moving in the scenario with the motion of each target being modeled as independent with a nearly constant velocity [12] as described in Section 3.5.1 with $\tau = 1s$ and $\sigma_u = 0.1m/s^{3/2}$. The PDF at time $k = 0$ of the position of the j -th target $p(\mathbf{x}_j^0)$ is initialized as a Gaussian distribution centered at the true position of the target, with covariance matrix $\mathbf{Q}$ given by (3.24).

A resource management scenario is proposed, so that we assume a sensor cost function as the one outlined using (5.10) and (5.14) with $b = 1$. A sensor network is considered with $N_s = 121$ sensors displayed in a rectangular grid with a separation between sensors of 10m. An additive received power measurement model is considered, so that measurements of the s -th sensor at time k are given by

$$z_s^k = h_s(\mathbf{X}^k) + \frac{v_s^k}{\sqrt{\theta_s^k}} \quad (5.78)$$

where v_s^k is a zero-mean unit-variance ($\sigma_n^2 = 1$) Gaussian-distributed noise, and $h_s(\mathbf{X}^k)$ is given by (3.26) with $Z_0 = 20$, $\alpha = 2$ and $\rho_0 = 5\text{m}$. The simulated scenario is represented in Figure 5.2.

5.4.1.2 Simulation results

Figures 5.3 and 5.4 show the target trajectories and the sensor network layout along with the average resource distribution of the simulated scenario using a Monte Carlo with $n = 1000$ runs. It can be observed how the use of the proposed FAR approach assigns the available resources to the sensors that are nearer to the trajectories of the targets, which is the expected behavior of the algorithm as the resources are allocated to those sensors that are able to obtain better measurements. Note also that when the number of targets grows, the FAR approach distributes the available resources through the sensor network in order to allow for the adequate tracking of all targets.

Figure 5.5 compares the optimal subpattern assignment OSPA [145] position error (with parameters $p = 2$ and $c = 10$) in the proposed MTT scenario using the FAR approach for MTT with an uniform resource distribution, i.e., $\theta_i^k = 1/N_s$ for $i = 1, \dots, N_s$. This is a reasonable approach when no information about the environment is available to the controller, thus being a good baseline to illustrate the benefits of a FAR framework approach to manage limited resource budgets. It can be observed how the FAR approach remarkably improves the performance with respect to the uniform setting, as a better resource distribution allows for better measurements from the relevant sensors at each time step. Note that in this case, the improvement seems to saturate when the number of targets grows. This is an expected behavior, as for a high number of targets the resources need to be shared among more sensors, therefore approaching the performance of the uniform resource distribution.

5.5 Conclusions

In this chapter, we have first introduced the main concepts of the FAR framework and have particularized it for its use in the problems of single target tracking, single target detection and track maintenance and multiple target tracking with a fixed and known number of targets. It is worth mentioning that the latter approach is a contribution of this thesis, and results on this topic have been published in [169].

(a)

(b)

Figure 5.3: Average resource distribution using the FAR resource management strategy when tracking respectively (a) one and (b) two targets. The average amount of resources received by a sensor during the simulation is indicated by its area and color, with a bigger area (and hotter color) being assigned to sensors with higher amount of resources.

(a)

(b)

Figure 5.4: Average resource distribution using the FAR resource management strategy tracking respectively (a) three and (b) four targets. The average amount of resources received by a sensor during the simulation is indicated by its area and color, with a bigger area (and hotter color) being assigned to sensors with higher amount of resources.

Figure 5.5: Average OSPA position error using the APP PF with 300 particles with 1 to 4 targets in the scenario, for the uniform and the FAR resource distributions. Monte Carlo simulations were carried out with $n = 1000$ iterations.

Chapter 6

Robust approach to FAR

In this chapter the performance of the application of the FAR framework, presented in Chapter 5, is analyzed in some scenarios of interest. We show that ill conditioned scenarios may trigger some robustness issues when the FAR framework is employed as depicted in Chapter 5. The root causes and the adverse consequences of these problems are thoroughly studied, and alternative approaches within the FAR framework are proposed to the procedures outlined in Chapter 5. The proposed methods are shown to be able of mitigating the previously identified robustness problems.

A first issue in the application of the FAR framework to the problem of single target tracking using a sensor network that measures received signal strength is thoroughly described in Section 6.1. The cause of the problem is identified to be in the minimization of the PC-CRLB, which results in ill conditioned resource distributions. A robust approach to this problem is proposed in Section 6.2. The proposed method is based on an enabling linear-Gaussian approximation that allows for the replacement of the PC-CRLB minimization in the controller of the FAR system.

The second robustness issue is identified in the application of the FAR framework to the problem of simultaneous single target detection and tracking. The proposed method in the literature to tackle this problem (which has been outlined in Section 5.3) considers the use of a surrogate function which maximizes Bayesian information in order to choose the sensor parameters at the next time step. However, Section 6.3 shows that such an approach does not always suit the objectives of target detection and track maintenance within the simultaneous target detection and tracking problem. An alternative approach to adequately tackle the problem is proposed in Section 6.4.

6.1 Robustness issues and limitations of the minimization of PC-CRLB for single target tracking

In this section we consider the FAR approach in order to select the best possible configuration of a sensor network with resource constraints which attempts to track a single target moving in the environment. We first analyze the performance of the system with respect to the available resource budget, showing that when scarce resource budgets are available the minimization of the PC-CRLB can lead to the emergence of ill conditioned resource distributions.

The considered simulated sensor network consists of $N_s = 16$ sensors deployed in a 2-D regular grid covering $60\text{m} \times 60\text{m}$, with a spacing of 20m between sensors in both axes. The precision of the measurements of a given sensor depends on the amount of resources assigned to it as in (5.12), with a limited available resource budget shared by the whole sensor network as indicated by (5.14). Sensors collect measurements according to the received power model in Section 3.5.2 with parameters $Z_0 = 20$ and $\rho_0 = 6.66\text{m}$, $\alpha = 2$ and $\sigma_n^2 = 1$.

The motion of the target has been modeled as linear with a nearly constant velocity [12] as in (3.22) with $\sigma_u = 0.1\text{m/s}^{3/2}$ and $\tau = 1\text{s}$. The trajectory of the target being tracked and the described sensor network scenario are both shown in Figure 6.1. The state of the target $\mathbf{x}^k$ is estimated using the auxiliary particle filter method in [130] with $N = 300$ particles. The PDF at time $k = 0$ of the position of the target $p(\mathbf{x}^0)$ is initialized as Gaussian distribution centered at the true position of the target, with covariance matrix $\mathbf{Q}$. The PC-CRLB at each time step is computed using the PF discrete approximation of the prior covariance as in (5.28) for the PIM, while the sensor EFIMs are approximated using the expression in (A.43) obtained for the received power measurement model.

In order to quantify the performance with the application of the resource distributions obtained by the minimization of the PC-CRLB as proposed in [17, 19], we also consider two alternative baseline strategies for the controller: the uniform resource distribution and an N -nearest resource distribution. Provided that the uniform resource distribution belongs to the feasible set Θ , this strategy can be used with any measurement model as a baseline which emulates the situation of no available information in the controller, assigning the same amount of resources to all sensors, i.e.,

$$\theta_i^k = \frac{1}{N_s} \quad \forall i, \forall k, \quad (6.1)$$

while the N -nearest strategy is an heuristic approach which first obtains at each time

Figure 6.1: Simulated scenario. The positions of the $N_s = 16$ sensors are indicated with a red dot and the simulated target trajectory is shown in green. The initial and final points of the trajectory are respectively indicated by a colored circle and square.

step an estimation of the position of the target $\bar{\mathbf{x}}^k$ as in (5.30), so that the distance from the predicted position of the target to the s -th sensor $\rho_s(\bar{\mathbf{x}}^k)$ can be computed using (3.28). Once the distance from sensors to the predicted position of the target is available, an active sensor set $S_{N_n}^k$ is formed which contains the N_n sensors (i.e., the cardinality of the active sensor set is $|S_{N_n}^k| = N_n$) which have a smaller distance to the predicted position of the target. The N -nearest method therefore assigns resources to sensors as

$$\theta_i^k = \begin{cases} \frac{1}{N_n} & \mathbf{s}_i \in S_{N_n}^k \\ 0 & \mathbf{s}_i \notin S_{N_n}^k \end{cases} \quad (6.2)$$

where we have set $N_n = 4$ for the current simulations.

The described policies for sensor resource distributions have been tested in the above presented scenario with a varying resource budget b (see (5.14)). The minimization of the PC-CRLB has been carried out using a constrained gradient descent with the *fmincon* command in MATLAB.

6.1.1 Performance of the PC-CRLB minimization with a large resource budget

We first analyze the performance of the sensor network under the different policies when a large budget, with

$$B \equiv \log_{10}(b) = 3, \quad (6.3)$$

is available. In order to understand the general behavior of the controller in the assignment of resources it is necessary to note that, as indicated by the sensor EFIM expression in (A.41), under the considered measurement model, the m -th sensor $\mathbf{s}_m$ only provides information of the radial (with respect to the sensor position) component of the state, $\hat{\boldsymbol{\rho}}_m(\mathbf{x})$. Thus, in order to keep low the overall uncertainty in all the components of the state, the controller usually tends to also collect intelligence from a different sensor $\mathbf{s}_n$ whose measurement with information in the radial component $\hat{\boldsymbol{\rho}}_n(\mathbf{x})$ can complement that gathered by $\mathbf{s}_m$. It is also necessary to take also into account that, with the considered measurement model, the sensors with the most reliable information will be those with a smaller distance to the target, so that these sensors will typically have a higher amount of allocated resources. Nonetheless, it is important to note that this statement does not apply if the target lies in the saturation region of a near sensor.

Figure 6.2 represents the typical result of the application of the PC-CRLB minimization with a large budget. It is worth noting that the controller always makes use of all the available resources, as in principle, in the formulated problem there is no incentive in saving any of the available resources. It is shown how, in the initial time steps, resources are assigned to sensors $\mathbf{s}_8 = [40, 0]^T$ (in blue) and $\mathbf{s}_{12} = [40, 20]^T$ (in red), with the biggest part of resources being given to $\mathbf{s}_8$. Note that $\mathbf{s}_{12}$ is nearer to the current predicted position of the target, so its information even with a smaller portion of assigned resources will be reliable, while $\mathbf{s}_8$ requires more resources to acquire information that can adequately complement that collected by $\mathbf{s}_{12}$. Following such a strategy, the controller is keeping low the overall uncertainty in the target state. As the target moves through the scenario (Figure 6.2(b)), resources are assigned to $\mathbf{s}_{11} = [20, 20]^T$ (in yellow) which can now better complement the information provided by $\mathbf{s}_{12}$ than $\mathbf{s}_8$, while the relative distribution of resources among $\mathbf{s}_{11}$ and $\mathbf{s}_{12}$ is in this case more even, as their distances to the predicted position of the target is almost the same. Once the target approaches the middle of its trajectory at time $k = 55$, the nearest sensors are $\mathbf{s}_{11}$ and $\mathbf{s}_7 = [20, 0]^T$ (in orange), however, given the prior target position PDF at this time, we have that $\hat{\boldsymbol{\rho}}_{11}(\mathbf{x}^{55}) \approx -\hat{\boldsymbol{\rho}}_7(\mathbf{x}^{55})$, so that the measurements of these sensors can only reduce the uncertainty in a single component

Figure 6.2: Simulated scenario with a large available budget ($B = 3$) for the PC-CRLB minimization at 4 different time steps of interest: (a) $k = 2$, (b) $k = 24$, (c) $k = 53$, (d) $k = 100$. The lower part of each figure represents the resources assigned to the each sensor (normalized to 1) up to the current time step. The upper part represents each sensor with a red dot, particles of the APF are represented as green dots, while the real and estimated target positions are respectively depicted as a black cross and circle. Sensors being assigned with any positive amount of resources are surrounded by a colored circle, with the color associated to each sensor being the same as in the lower part of the figure.

of the state. Thus, before the uncertainty in other state components rises beyond an acceptable point, the controller assigns resources to other farther sensors, namely in this case $\mathbf{s}_{12}$, $\mathbf{s}_8$ or $\mathbf{s}_6 = [0, 0]^T$ (in blue). In the final part of the target trajectory, the controller assigns resources simultaneously to $\mathbf{s}_6$ and $\mathbf{s}_7$.

6.1.2 Performance of the PC-CRLB minimization with a scarce resource budget

We now analyze the performance of the sensor network under the controller different policies when a scarce budget, with

$$B \equiv \log_{10}(b) = 1, \quad (6.4)$$

is available. In such conditions, when there are few resources available, the minimization of the PC-CRLB in the controller usually tends to allocate all resources to a single sensor, typically the nearest one (for a detailed explanation for this behavior, see Appendix A.8.2). Figure 6.3(a) illustrates this situation in the simulated scenario. Note that initially all resources are assigned to sensor $\mathbf{s}_{12}$, which minimizes the PC-CRLB. However, if this decision is kept for a relatively large period of time (see Figure 6.3(b)), the posterior PDF tends to spread over the region of high likelihood for this sensor, which depending on the measurement model, can increase the uncertainty in some components of the state. In this case, with the received power measurement model, the posterior PDF tends to spread across a *ring* around $\mathbf{s}_{12}$. As the target moves through the scenario and eventually approaches another sensor (in this case $\mathbf{s}_{11}$), the controller allocates all the available resources to it as in Figure 6.3(c). For the received power measurement model which is being considered in the simulations, this causes the region of high likelihood to be located in a *ring* surrounding sensor $\mathbf{s}_{11}$, which given the prior PDF, makes the posterior to be prone to becoming bimodal (see Figure 6.3(c)). Note that both modes of the posterior PDF are approximately equally likely for the processor, having only received relevant measurements from sensors $\mathbf{s}_{11}$ and $\mathbf{s}_{12}$. At this point, a measurement from an alternative suitable sensor with some allocated resources would be necessary in order to confirm the likelihood of one mode and discard the other mode.

Note that the spread of the posterior PDF, as well as the emergence of a multimodal PDF, resulting from the resource distribution decided by the controller in the minimization of the PC-CRLB, can severely increase the error of the estimation in the processor, so that the performance of the system can be severely degraded. In addition, multimodal posterior PDFs are difficult to handle for some filtering methods,

Figure 6.3: Simulated scenario with a scarce available budget ($B = 1$) for the PC-CRLB minimization at 4 different time steps of interest: (a) $k = 2$, (b) $k = 19$, (c) $k = 29$, (d) $k = 37$. The representation of the scenario follows the same approach as Figure 6.2.

Figure 6.4: RMS position error, in logarithmic scale, with a cutoff parameter (i.e., $\text{RMS}^{(c)} = \min(c, \text{RMS})$, where $c = 10m$) as a function of the available resource budget, B (also in logarithmic scale), for three different controller policies: the PC-CRLB minimization, the 4-Nearest, and the uniform resource distribution.

and in particular for particle filters as a result of the resampling step [136] (needed to avoid sample degeneracy [44], see 2.7.1.1). As shown in Figure 6.3(d), this effect can increase the estimation error in the processor even further, as the filter will eventually lose all but a single mode [24, 121, 122, 179] (not necessarily the one corresponding to the true position of the target), which can result in the divergence of the tracker. A similar procedure to resampling is performed in the the prior PDF in auxiliary particle filters [130], so that this kind of particle filters also have difficulties in managing multimodal PDFs.

6.1.3 Practical limitations of the FAR framework

In this section, the performance of the FAR framework with respect to the available resource budget is analyzed. Figure 6.4 shows the target tracking position error as a function of the available resource budget in the scenario for the PC-CRLB minimization as well as the two previously discussed baseline controller strategies in 5000 Monte Carlo runs. As previously mentioned, the processor makes use of an auxiliary particle filter [130] with $N = 300$ particles. We first compare the behavior of the two heuristic strategies, noting that 4-Nearest strategy achieves a much lower error than that of the

Figure 6.5: Resource distribution through time for the PC-CRLB minimization approach with 2 different resource budgets. In the upper part, scarce resource budget ($\log_{10}(b) = 1$) case. In the lower part, large resource budget ($\log_{10}(b) = 3$) case.

uniform resource distribution, demonstrating that the use of the information available to the system can be beneficial to better fulfill its operation objective. It is also worth noting that the 4-Nearest strategy consistently outperforms the uniform resource distribution in a wide range of available resource budgets. The PC-CRLB minimization achieves good performance when used under high resource budgets, however this strategy suffers from some robustness issues for medium and scarce resource budgets, as previously mentioned in Section 6.1, being outperformed by the 4-Nearest strategy and eventually even by the uniform strategy.

To illustrate the behavior exhibited in Figure 6.4 by the performance of the PC-CRLB minimization, Figure 6.5 comparatively shows the resulting resource distributions for two different budgets. When a sufficiently high budget is available, resources are simultaneously allocated to a combination of 2 or even 3 sensors, which in addition to the information provided by the prior PDF is enough to avoid the formation of multimodal posterior PDFs. In addition, it can be observed that with a large resource budget, the amount of resources each sensor receives is smoothly varied as the target moves in the scenario, with the exception of the time instants in which the controller switches allocated resources from one sensor to another that can provide more reliable information about the same state component. On the other hand, when resources are

Figure 6.6: True trajectory of the tracked target superimposed over 1000 estimations with different measurements of the target trajectory using two different controller policies: the uniform resource distribution in (6.1) and the minimization of the PC-CRLB with a scarce resource budget $B = \log_{10}(b) = 1$. The uniform controller strategy has a better performance than the minimization of the PC-CRLB.

scarce, the optimization procedure in the PC-CRLB minimization tends to allocate all the resources to a single sensor, with abrupt changes in the distribution of resources as the position of the target in the scenario varies.

The consequences in the tracking accuracy of the above disclosed robustness issues in the application of the PC-CRLB with scarce resource budgets are shown in Figure 6.6. It can be observed that in a non-negligible number of Monte Carlo runs, when the posterior becomes bimodal, the eventually surviving mode in the particle filter is not the one corresponding to the true trajectory of the target, so that in those cases the particle filter is prone to divergence, causing the error of the PC-CRLB minimization strategy to rise as shown in Figure 6.4. Nevertheless, Figure 6.6 also shows that if the surviving mode is the correct one, the estimation of the target trajectory using the minimization of the PC-CRLB has a lower error than the uniform resource distribution. This indicates that, were it not for the initial spread in the posterior PDF and the creation of multimodal PDFs (which additionally comes with the multimodality-holding problems of some filtering methods), the use of the strategy of the minimization of the PC-CRLB would of course work better than the baseline uniform resource distribution

strategy, and therefore be advisable.

It is worth noting that the problem of holding multimodal PDFs for long periods of time can be eased by making use of nonlinear filtering methods with deterministic sampling such as the point mass filter used in [19] (see Section 2.6.1). However, for a given required tracking precision, the use of filtering methods with deterministic sampling result in a severe increase in the computational cost with respect to particle filters, specially if the dimension of the state space is high. There are some implementations of particle filters which have been specially designed to allow propagation of multimodal PDFs for relatively long periods of time [121, 122], which can also help to ease the problem. Nevertheless, the spread in the posterior PDF and the emergence of multimodal PDFs also makes the accuracy of the estimation of the processor of the system to decrease, even if the multimodal PDF is held and divergence of the tracker avoided.

In the following section, we present a novel approach that substitutes the minimization of the PC-CRLB in the controller of the FAR framework and successfully avoids the robustness issues previously detailed.

6.2 A robust sigma-point approach to single target tracking in FAR

An alternative for the optimization procedure in the controller of the FAR framework is now presented. The proposed method is based on a linear-Gaussian approximation of the measurement equation that allows for an analytic prediction of the effect of the different sensor configurations of the system. This approximation can be efficiently implemented through sigma-point methods [142], greatly simplifying the optimization procedure and providing a significant decrease in its required computational load with respect to the minimization of the PC-CRLB. The method outlined in this section has been published in [166].

6.2.1 Linear-Gaussian approximation for the FAR optimization procedure

As pointed out in Section 6.1, the minimization of the PC-CRLB in the FAR framework suffers from some robustness problems when applied to difficult or severely constrained sensor management scenarios. In this section, we propose an alternative to the controller optimization procedure which successfully avoids the above mentioned

robustness issues. The presented method is based on the selection of those sensor parameters, $\boldsymbol{\theta}^k$, that minimize some suitable function of the uncertainty of the posterior PDF, which is considered through an approximation of its posterior covariance matrix, $\mathbf{W}_{\boldsymbol{\theta}^k}$. In order to make this possible, $\mathbf{W}_{\boldsymbol{\theta}^k}$ needs to take into account the dependence of the posterior PDF with respect to the sensor parameters. However, as previously mentioned, the main difficulty lies in the fact that the computation of the posterior PDF requires the measurement at the current time step $\mathbf{z}^k$, which is not available until the sensor parameters have been selected. In the proposed approach, we circumvent this problem by means of Gaussian filtering. The method described in this section is executed in the controller, and thus, the required approximations are only used by this block, while other blocks such as the processor do not necessarily require them in their computations.

In Gaussian filtering techniques the prior density is approximated by

$$p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \approx \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{x}}^k, \bar{\mathbf{C}}^k) \quad (6.5)$$

where $\mathcal{N}(\mathbf{x}^k; \bar{\mathbf{x}}^k, \bar{\mathbf{C}}^k)$ is a Gaussian PDF with mean $\bar{\mathbf{x}}^k$ and covariance matrix $\bar{\mathbf{C}}^k$ evaluated at $\mathbf{x}^k$. In this setting, the posterior becomes Gaussian by performing the enabling approximation [53]

$$\mathbf{h}(\mathbf{x}^k) \approx \tilde{\mathbf{h}}(\mathbf{x}^k) = \mathbf{H}^k \mathbf{x}^k + \mathbf{a}^k + \mathbf{e}^k \quad (6.6)$$

where $\tilde{\mathbf{h}}(\mathbf{x}^k)$ is an affine approximation of $\mathbf{h}(\mathbf{x}^k)$, with $\mathbf{H}^k \in \mathbb{R}^{n_z \times n_x}$, $\mathbf{a}^k \in \mathbb{R}^{n_z}$ and $\mathbf{e}^k \in \mathbb{R}^{n_z}$ is a Gaussian distributed random variable with zero mean and covariance matrix $\boldsymbol{\Omega}^k$ and uncorrelated with $\mathbf{x}^k$ and the measurement noise $\mathbf{v}^k(\boldsymbol{\theta}^k)$, which is also Gaussian distributed with zero-mean and covariance matrix $\mathbf{R}_{\boldsymbol{\theta}^k}^k$. The approximation of a Gaussian prior density in (6.5) along with the affine approximation of the measurement model in (6.6) allows for the computation of an analytical approximation of the likelihood $p(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k)$ using the concepts outlined in Appendix A.1 as

$$p(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) \approx \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \mathbf{x}^k + \mathbf{a}^k, \boldsymbol{\Omega}^k + \mathbf{R}_{\boldsymbol{\theta}^k}^k), \quad (6.7)$$

which also allows for a Gaussian approximation of the posterior PDF as

$$p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) \approx \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{u}}_{\boldsymbol{\theta}^k}, \mathbf{W}_{\boldsymbol{\theta}^k}) \quad (6.8)$$

where the mean and covariance matrix are obtained as

$$\bar{\mathbf{u}}_{\boldsymbol{\theta}^k} = \bar{\mathbf{x}}^k + \bar{\mathbf{C}}^k (\mathbf{H}^k)^T (\mathbf{S}_{\boldsymbol{\theta}^k}^k)^{-1} (\mathbf{z}^k - \mathbf{H}^k \bar{\mathbf{x}}^k - \mathbf{a}^k) \quad (6.9)$$

$$\mathbf{W}_{\boldsymbol{\theta}^k} = \bar{\mathbf{C}}^k - \bar{\mathbf{C}}^k (\mathbf{H}^k)^T (\mathbf{S}_{\boldsymbol{\theta}^k}^k)^{-1} \mathbf{H}^k \bar{\mathbf{C}}^k, \quad (6.10)$$

with

$$\mathbf{S}_{\boldsymbol{\theta}^k}^k = \mathbf{H}^k \bar{\mathbf{C}}^k (\mathbf{H}^k)^T + \boldsymbol{\Omega}^k + \mathbf{R}_{\boldsymbol{\theta}^k}^k. \quad (6.11)$$

Note that the approximated covariance matrix (6.10) of the posterior takes into account an explicit dependence with respect to $\boldsymbol{\theta}^k$. As previously mentioned, it is not straightforward to tackle the direct minimization of (5.7) with respect to $\boldsymbol{\theta}^k$ in a general setting and approximations are required. Note however, that the Bayesian risk in (5.5) is usually related to the uncertainty in the knowledge of the state $\mathbf{x}^k$, which is captured by the covariance of its posterior PDF. Thus, considering the approximation of the posterior covariance matrix in (6.10), instead of minimizing the predicted conditional Bayes risk as in (5.11), we aim to minimize a surrogate function of the covariance matrix $L_{\mathcal{N}}(\mathbf{W}_{\boldsymbol{\theta}^k})$ so that $\boldsymbol{\theta}^k$ is chosen by the solution to the minimization problem

$$\arg \min_{\boldsymbol{\theta}^k} L_{\mathcal{N}}(\mathbf{W}_{\boldsymbol{\theta}^k}) \quad s.t. \quad \boldsymbol{\theta}^k \in \Theta \quad (6.12)$$

where $L_{\mathcal{N}}(\mathbf{W}_{\boldsymbol{\theta}^k})$ can be any suitable function that can depend on the application at hand, for example, the trace of the covariance matrix $\mathbf{W}_{\boldsymbol{\theta}^k}$, or its Frobenius norm.

There are different possibilities for selecting the parameters of the approximation in (6.6), which have been frequently used in the literature [53], giving rise to some celebrated methods like the extended Kalman filter, the iterated extended Kalman filter [16] or some sigma-point Kalman filters such as the Gauss-Hermite Kalman filter [5], the unscented Kalman filter [79] or the cubature Kalman filter [4], among others [77, 142]. However, as previously mentioned we are restricted by the unavailability of the current measurement $\mathbf{z}^k$, thus, we are limited to the approximations with respect to the prior PDF mentioned in [53]. In this case, we consider statistical linear regression so that

$$\mathbf{H}^k = (\boldsymbol{\Psi}^k)^T (\bar{\mathbf{C}}^k)^{-1} \quad (6.13)$$

$$\mathbf{a}^k = \bar{\mathbf{z}}^k - \mathbf{H}^k \bar{\mathbf{x}}^k \quad (6.14)$$

$$\boldsymbol{\Omega}^k = \boldsymbol{\Phi}^k - \mathbf{H}^k \bar{\mathbf{C}}^k (\mathbf{H}^k)^T \quad (6.15)$$

where the required quantities of interest can be computed with respect to the prior PDF as

$$\bar{\mathbf{z}}^k = \int \mathbf{h}(\mathbf{x}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k \quad (6.16)$$

$$\boldsymbol{\Psi}^k = \int (\mathbf{x}^k - \bar{\mathbf{x}}^k) (\mathbf{h}(\mathbf{x}^k) - \bar{\mathbf{z}}^k)^T p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k \quad (6.17)$$

Figure 6.7: RMS position error, in logarithmic scale, with a cutoff parameter (i.e., $\text{RMS}^{(c)} = \min(c, \text{RMS})$, where $c = 10m$) as a function of the available resource budget, b (also in logarithmic scale), for four different controller policies: the PC-CRLB minimization, the minimization of the trace of the predicted state covariance, the 4-Nearest, and the uniform resource distribution.

$$\Phi^k = \int (\mathbf{h}(\mathbf{x}^k) - \bar{\mathbf{z}}^k) (\mathbf{h}(\mathbf{x}^k) - \bar{\mathbf{z}}^k)^T p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k. \quad (6.18)$$

The integrals in (6.16)-(6.18) can be approximated using sigma-point methods [142]. As explained in Section 2.5.2.1, these methods use a set of m sigma-points with their corresponding weights, $\mathcal{X} = \{(\mathcal{X}^1, w^1), \dots, (\mathcal{X}^m, w^m)\}$, which match the first two moments of $p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$, and approximate (6.16)-(6.17) as

$$\bar{\mathbf{z}}^k \approx \sum_{i=1}^m w^i \mathbf{h}(\mathcal{X}^i) \quad (6.19)$$

$$\Psi^k \approx \sum_{i=1}^m w^i (\mathcal{X}^i - \bar{\mathbf{x}}^k) \cdot (\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}}^k)^T \quad (6.20)$$

$$\Phi^k \approx \sum_{i=1}^m w^i (\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}}^k) \cdot (\mathbf{h}(\mathcal{X}^i) - \bar{\mathbf{z}}^k)^T. \quad (6.21)$$

It is important to note that the posterior and predicted PDFs computed by the processor in Figure 5.1 can have a more general form than (6.5) and (6.8), and only the controller needs to compute these approximations to select the next sensor parameters

θ^k . Thus, at each time step, a general prior PDF $p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \theta^{1:k-1})$ is provided by the processor (e.g. using a particle filter approximation), and the controller approximates both $\bar{\mathbf{x}}^k$ and $\bar{\mathbf{C}}^k$, the first and second moments of the prior PDF, in order to compute (6.5) and apply the method described in this section.

It is worth mentioning that, in addition, the computational load of the presented method is significantly lower than that of the PC-CRLB minimization. Numerical results on this point as well as on performance of these methods are also considered in Section 6.2.2.

6.2.2 Simulations

In this section, the performance of the presented method is analyzed in the single target tracking scenario used in Section 6.1. Considering through (5.12) the dependence of sensor measurements with respect to the resource distribution, the covariance matrix of sensor measurements, $\mathbf{R}_{\theta^k}^k$ is a diagonal matrix whose diagonal elements are

$$\mathbf{R}_{\theta^k}^k [s, s] = \frac{1}{\theta_s^k}. \quad (6.22)$$

We have considered different surrogate functions $L_{\mathcal{N}}(\mathbf{W}_{\theta^k})$ in (6.12). In particular, the trace and the Frobenius norm of the posterior covariance matrix. However, equivalent performance was observed for both of them, so that for the sake of clarity, we only present results for the trace function. The processor, which performs the state filtering procedure, is based on an auxiliary particle filter with 300 particles. The minimization of the PC-CRLB and $L_{\mathcal{N}}(\mathbf{W}_{\theta^k})$ has been carried out using a constrained gradient descent with the *fmincon* command in MATLAB.

Figure 6.7 shows the averaged target tracking RMS position error as a function of the available resource budget in the scenario for the 4 different resource budget distributions using a total of 5000 Monte Carlo runs. The strategy presented in this section, the minimization of the trace of the predicted state covariance based on a linear-Gaussian approximation, satisfactorily performs under small resource budgets, clearly beating the performance of the PC-CRLB minimization, and even matching the very robust performance of the heuristic approach of the 4-Nearest policy. Medium and large resource budgets allow the strategy presented in this section to outperform the 4-Nearest policy, as well as match the performance of the minimization of the PC-CRLB with large resource budgets. In addition, it is worth noting that the resource management strategy presented in this section has a mean execution time of the scenario presented in this section of 6.52s, while the minimization of the PC-

Figure 6.8: Simulated scenario with a scarce available budget ($B = 1$) for the PC-CRLB minimization at 4 different time steps of interest: (a) $k = 2$, (b) $k = 11$, (c) $k = 29$, (d) $k = 100$. The representation of the scenario follows the same approach as Figure 6.2.

Figure 6.9: True trajectory of the tracked target superimposed over 1000 estimations with different measurements of the target trajectory using two different controller policies with a scarce resource budget $B = \log_{10}(b) = 1$: the uniform resource distribution in (6.1) and the minimization of the trace of the predicted state covariance proposed in this section. This latter controller policy has a better performance.

CRLB takes 12.11s. Thus, the presented method is considerably more computationally efficient than the minimization of the PC-CRLB.

It should also be noted in Figure 6.7 that all the implemented sensor management policies cease increasing performance with very high budgets. This is due to the very narrow likelihoods that appear with such precise measurements, which cause the auxiliary particle filter to have a very low effective sample size, limiting the performance of the filter of the processor. Other filtering algorithms better suited for these narrow likelihoods could be used with such high budgets instead of the auxiliary particle filter.

Figure 6.8 illustrates in comparison with Figure 6.3 how the method presented in this section outperforms the PC-CRLB minimization with a scarce resource budget. Note how both approaches in Figures 6.3(a) and 6.8(a) initially assign all the available resources to sensor $\mathbf{s}_{12}$, however, this causes the uncertainty of the state posterior to rapidly grow in the y direction. The contribution of the sensor EFIM of other sensors is nonetheless much lower than that of $\mathbf{s}_{12}$, so that the minimization of the PC-CRLB still assigns all resources to this sensor as shown in 6.3(b). On the other hand, the method presented in this section takes into account that the uncertainty of the state

Figure 6.10: Simulated scenario with a large available budget ($B = 3$) for the PC-CRLB minimization at 4 different time steps of interest: (a) $k = 2$, (b) $k = 24$, (c) $k = 53$, (d) $k = 100$. The representation of the scenario follows the same approach as Figure 6.2.

Figure 6.11: Resource distribution through time for 2 different resource management policies. In the upper part, the minimization of the PC-CRLB. In the lower part, the minimization of the trace of the predicted state covariance.

in the y direction is much bigger than in the x direction, and assigns resources to the sensor at position $\mathbf{s}_8$, whose measurement can efficiently reduce the uncertainty in the y direction as shown in 6.8(b). It is shown through Figure 6.8(c) that by using this latter strategy, the posterior PDF has remained unimodal, adequately tracking the target in the scenario, while in contrast, in Figure 6.3(c), the minimization of the PC-CRLB has assigned all the resources to another sensor, triggering the formation of a bimodal PDF that increases the estimation error, and can take the filter to divergence [168] as shown in Figure 6.3(d). On the other hand it is shown in Figure 6.8(d) how the controller approach presented in this section can successfully track the state of the target without any further problems.

Figure 6.9 shows the resulting trajectories of the performed simulation using the minimization of the trace of the predicted state covariance matrix along with a scarce budget. It is shown how, in contrast to the results depicted in Figure 6.6 using the minimization of the PC-CRLB, the proposed approach almost never suffers from track divergence and is able of correctly and robustly estimating the position of the target along the trajectory.

Figure 6.10 illustrates how the minimization of the trace of the predicted state covariance can also adequately perform in scenarios with a large available budget. Interestingly, in such cases the proposed approach achieves so by mimicking the

Table 6.1: Principal features of the different sensor management strategies

FAR optimization strategy	Generalizable for any measurement model	Avoids problem specific derivations	Robustness in analyzed difficult scenarios
Uniform distribution	✓	✓	✓
N-nearest	×	✓	✓
PC-CRLB	✓	×	×
Gaussian approximation	✓	✓	✓

decisions made by the controller when using the minimization of the PC-CRLB as shown in Figure 6.11. Thus, the minimization of the trace of the predicted state covariance can approximately achieve the resource distributions of the PC-CRLB minimization when it works at its best, while avoiding its robustness issues in difficult or hardly constrained scenarios. Table 6.1 summarizes the properties of the different tested methods for sensor parameter selection in the FAR framework.

6.3 Suitability of PC-CRLB minimization for simultaneous target detection and tracking

A key element in the application of the FAR framework to a problem of interest is the design of an appropriate cost function [27, 117], as it will directly influence the sensor parameter selection through the minimization of its predicted conditional Bayes risk in (5.7) and thus affect the system performance. A common and difficult case is the problem in which the system attempts to simultaneously achieve two (or more) different objectives. In the following, it is discussed that systems engaged in a simultaneous target detection and tracking operation like the one considered in Section 5.3 [18, 19] can be considered to fall into this category.

Simultaneous target detection and tracking systems recursively update the proba-

bility of existence of a target r^k and its associated posterior density $p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ in (5.37) (see Section 5.3). This kind of systems can be understood as operating in two different modes: When the target is declared absent, the system updates r^k , trying to maximize detection performance, but also to make the region covered by the posterior more concentrated, which can help increase the detection capabilities at the next time step, and in the accurate initialization of the track. When the target is declared present, the system again tries to concentrate the posterior around the true value of $\mathbf{x}^k$ to minimize the tracking error, but is also concerned with track maintenance, trying to accurately update r^k so that when the target disappears from the scenario, the system can promptly terminate the track and minimize the probability of false alarm in such situations. Note that in both cases, the system simultaneously handles one objective related to the update of r^k and another related to the update of $p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$.

As previously stated in Section 5.3, target detection is often tackled by minimizing the conditional Bayes Risk with respect to the cost function in (5.54), while on the contrary, target tracking makes use of the cost function in (5.15). In the FAR framework, the best possible set of sensor parameters in order to yield a good detection performance at the next time step should therefore be obtained by considering the minimization of the predicted conditional Bayes risk in (5.60), while selection of the next sensor parameters for a good tracking performance of the system could be obtained by minimizing the PC-CRLB in (5.24) or preferably for robustness considerations using the approach outlined in Section 6.2.1.

In an attempt to bridge the above described gap and tackle the problem of simultaneous target detection and tracking within the FAR framework, the seminal works in the field [18, 19] proposed the minimization of a surrogate function of the predicted conditional Bayes risk in (5.60), which can yield a sensor configuration able of simultaneously handling both detection and tracking problems. As a matter of fact, this surrogate function is chosen to be PC-CRLB in (5.24). It is advocated that in doing so, the controller maximizes Bayesian information, which is useful for making any statistical inferences about the state of the target in both estimation and detection settings. In this section this point is briefly discussed, also arguing against it. It is shown that the minimization of the PC-CRLB might not correctly serve to the purpose of target detection and track termination, so that to adequately approach simultaneous target detection and tracking in the FAR framework, a cost function better suited to the problem is required.

In order to better understand the difficulties of the proposed problem, it is worth recalling that the minimization of the predicted conditional Bayes risk for target detection in (5.60) is based on maximizing the hypothesis discrimination capabilities of

the likelihood ratio test in (5.59) for all possible target states and sensor measurements. This is usually achieved by giving preference to those sensors which have the chance of obtaining very different measurements when the null hypothesis is true (i.e., no target in the scenario) than when the alternative hypothesis (i.e., a target is present in the scenario) is true. On the other hand, the minimization of the predicted conditional Bayes risk for target tracking in (5.19) is based on the minimization of the uncertainty of the target state (i.e., on concentrating $p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ around the true value of $\mathbf{x}^k$ in the update step). This objective can be achieved by considering the measurement of those sensors that can better discriminate from similar target states, a feature which can be quantified by their sensor EFIM.

The two above objectives are not necessarily aligned and, depending on the measurement model under consideration, the seek for the optimal sensor configuration $\boldsymbol{\theta}^k$ for each of them might yield very different results. We now consider such a case through an illustrative example following Figure 6.12: consider the received power measurement model in Section 3.5.2, and its counterpart when there is no target in the scenario

$$h_s(\emptyset) = 0 \quad (6.23)$$

and assume that for some sensor of interest $\mathbf{s}_i$, $p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ is completely contained in its saturation region $\rho_i(\mathbf{x}^k) < \rho_0$. Note that if the alternative hypothesis were true, those sensors which could obtain measurements with a bigger difference from (6.23) and thus provide better results for hypothesis discrimination in the detection procedure would be precisely those for which $\rho_s(\mathbf{x}^k) < \rho_0$. Thus, assigning resources $\mathbf{s}_i$ would be a fair option for achieving good detection performance. Nevertheless, all possible target states within $p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ would produce the same measurement in sensor $\mathbf{s}_i$ (as $p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ is completely contained in its saturation region) so that the measurement of this sensor would not help in providing additional information about the exact value of the state $\mathbf{x}^k$. This is summarized in the fact that the sensor EFIM for this sensor is zero (see (A.41)), indicating that in order to provide a good tracking performance, resources should be assigned to some other sensor. Strategies for achieving good tracking and detection performance therefore collide and would yield very different sensor configurations. Figure 6.12 illustrates the measurement model, sensor FIM and likelihood ratio test for sensor $\mathbf{s}_i$ of the previous example.

Solving the problem of simultaneous target and tracking within the FAR framework therefore requires a careful design of the associated cost function, which can adequately and concurrently cover both objectives. In the next section, we provide such a solution and illustrate its use with simulations.

Figure 6.12: Assuming a target is present at the scenario, so that the state $\mathbf{x}^k \in \mathbb{R}^4$, and the availability of a resource budget $b = 1$: (a) Measurement $z^k = h(\mathbf{x}^k)$ for a sensor positioned at $\mathbf{s} = [0, 0]^T$ according to the received power measurement model in Section 3.5.2 with $Z_0 = 10$, $\rho_0 = 6.66\text{m}$, $\alpha = 2$ and $\sigma_n^2 = 1$. (b) Frobenius norm of the unitary (i.e., $\theta^k = 1$) i -th sensor FIM as a function of the state $\mathbf{x}^k$. Considering the measurement z^k in (a) along with the measurement for no target in (6.23), the log-likelihood ratio for z^k , $\log\left(\frac{l_x(z^k|\mathbf{x}^k;\theta^k)}{l_O(z^k|O;\theta^k)}\right)$, is shown in (c).

6.4 Bernoulli-Gaussian approach to simultaneous target detection and tracking in the FAR framework

With the objective of overcoming the robustness issues described in Section 6.3, an alternative approach to the one outlined in Section 5.3 for the problem of simultaneous target detection and tracking of a single target in the FAR framework is proposed in this section.

First, an appropriate cost function that suits the Bernoulli formulation of the problem in Section 5.3.1 is proposed. This cost function is able of simultaneously and correctly gathering possible errors in both detection and tracking settings. Then, an approximation to the posterior PDF is obtained based on the ideas developed in Section 6.2 using a linear-Gaussian approximation for the problem of robust sensor parameter selection in single target tracking. This approximation will be managed by the controller in Figure 5.1 prior to the arrival of $\mathbf{z}^k$, so that it can select the next sensor configuration.

In order to account for possible errors in both detection and tracking, the cost function is built based on the generalized OSPA (GOSPA) metric, which has been recently presented in [134] (see Appendix A.9 for a detailed description of the GOSPA metric).

It is considered that the posterior PDF is represented by a Bernoulli density as in (5.37). The GOSPA metric is defined on the space of sets of targets, rather than on the augmented state space $\mathbb{R}^{n_x} \cup \{\emptyset\}$ considered in (5.37) for the problem of simultaneous single target detection and tracking. However, it can be easily adapted to adequately suit this latter formulation. In this problem there can only be either no target ($\mathbf{x}^k = \emptyset$) or a single target present in the scenario ($\mathbf{x}^k \in \mathbb{R}^{n_x}$). Thus, considering for the sake of brevity $\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k \equiv \hat{\mathbf{x}}^k(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ to denote the estimator, the cost of $\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k$ using the squared GOSPA metric with parameters $p = 2$ and $\alpha = 2$ (see Appendix A.9) can only take the values

$$C_G(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k) = \begin{cases} d^{(c)}(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k)^2 & \mathbf{x}^k \in \mathbb{R}^{n_x} \quad \hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k \in \mathbb{R}^{n_x} \\ \frac{c^2}{2} & \mathbf{x}^k = \emptyset \quad \hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k \in \mathbb{R}^{n_x} \\ \frac{c^2}{2} & \mathbf{x}^k \in \mathbb{R}^{n_x} \quad \hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k = \emptyset \\ 0 & \mathbf{x}^k = \emptyset \quad \hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k = \emptyset \end{cases} \quad (6.24)$$

where parameter c is the cutoff distance and $d^{(c)}(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k)$ is the cutoff metric associated to a base distance metric $d(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k)$ (see Appendix A.9, [134]). It is of interest to highlight that the considered value of the GOSPA parameter $\alpha = 2$ is the one recommended by the authors in [134], as it is the only value that enables the

decomposition of the error into localization error for properly detected targets, and costs for missed targets and false targets, as in traditional performance evaluation methods for MTT [22, Sec. 13.6], [45, 47, 109, 138]. Thus, the value of $\alpha = 2$ is adequate to tackle the problem of simultaneous target detection and tracking.

The aim of the controller is therefore to minimize the predicted conditional Bayes risk of the squared GOSPA error, which can be written as

$$\begin{aligned} \mathcal{R}_{C_G}^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &= \int \int C_G(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k) \boldsymbol{\pi}(\mathbf{x}^k, \mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k}) d\mathbf{x}^k d\mathbf{z}^k \quad (6.25) \\ &= \int p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k}) \cdot \\ &\quad \left[\int C_G(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k) \boldsymbol{\pi}(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) d\mathbf{x}^k \right] d\mathbf{z}^k \end{aligned}$$

where making use of the Bernoulli form of the posterior in (5.37) and the expression for $C_G(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k)$ in (6.24), the inner integral can be expressed as

$$\int C_G(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k) \boldsymbol{\pi}(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) d\mathbf{x}^k = I_{\mathbf{z};\boldsymbol{\theta}}^1 + I_{\mathbf{z};\boldsymbol{\theta}}^2, \quad (6.26)$$

with

$$\begin{aligned} I_{\mathbf{z};\boldsymbol{\theta}}^1 &= \mathbb{I}\{\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k = \emptyset\} \int \left[0 \cdot (1 - r^k) \delta(\mathbf{x}^k - \emptyset) + \frac{c^2}{2} r^k p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) \right] d\mathbf{x}^k \\ &= \mathbb{I}\{\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k = \emptyset\} \frac{c^2}{2} r^k \end{aligned} \quad (6.27)$$

$$\begin{aligned} I_{\mathbf{z};\boldsymbol{\theta}}^2 &= \mathbb{I}\{\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k \in \mathbb{R}^{n_x}\} \int \left[\frac{c^2}{2} \delta(\mathbf{x}^k - \emptyset) (1 - r^k) + d^{(c)}(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k)^2 r^k p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) \right] d\mathbf{x}^k \\ &= \mathbb{I}\{\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k \in \mathbb{R}^{n_x}\} \left[\frac{c^2}{2} (1 - r^k) + r^k \int d^{(c)}(\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k, \mathbf{x}^k)^2 p(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) d\mathbf{x}^k \right]. \end{aligned} \quad (6.28)$$

Computing and minimizing (6.25) with respect to $\boldsymbol{\theta}^k$ in the controller can nevertheless be difficult. Thus, in the following, we make use of a linear-Gaussian approximation based on the method described in Section 5.3.1 to approximate (6.25).

6.4.1 Bernoulli-Gaussian approximation

In this section, a Bernoulli-Gaussian approximation is proposed, which enables an approximation of the risk in (6.26) by the controller, so that it can perform a good choice of the sensor parameters $\boldsymbol{\theta}^k$ before the measurements $\mathbf{z}^k$ are received. After considering

the proposed Bernoulli-Gaussian approximation, an estimator to be handled by the controller is specified in Section 6.4.1.1, while Section 6.4.1.2 finally details the policy followed by the controller in the proposed method.

Following (5.41) and (6.5), the prior is approximated as a Bernoulli-Gaussian density as

$$\begin{aligned}\pi(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &= (1 - r^{k-}) \delta(\mathbf{x}^k - \emptyset) + r^{k-} p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \\ &\approx (1 - r^{k-}) \delta(\mathbf{x}^k - \emptyset) + r^{k-} \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{x}}^k, \bar{\mathbf{C}}^k),\end{aligned}\quad (6.29)$$

while the enabling approximation in (6.6) is made with respect to the measurement model when the target is in the scenario. A Gaussian zero-mean measurement noise, whose variance might be dependent on the selected sensor parameters is considered, so that we can approximate the measurement likelihood when the target is present at the scenario as

$$l_x(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) \approx \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \mathbf{x}^k + \mathbf{a}^k, \boldsymbol{\Omega}^k + \mathbf{R}_{\boldsymbol{\theta}^k}^k), \quad (6.30)$$

while

$$l_{\emptyset}(\mathbf{z}^k | \emptyset; \boldsymbol{\theta}^k) = \mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\boldsymbol{\theta}^k}^k). \quad (6.31)$$

Using (6.29), (6.30) and (6.31), the joint PDF in (6.25) can be approximated as

$$\begin{aligned}\pi(\mathbf{x}^k, \mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &= (1 - r^{k-}) l_{\emptyset}(\mathbf{z}^k | \emptyset; \boldsymbol{\theta}^k) \delta(\mathbf{x}^k - \emptyset) \\ &\quad + r^{k-} l_x(\mathbf{z}^k | \mathbf{x}^k; \boldsymbol{\theta}^k) p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \\ &\approx (1 - r^{k-}) \mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\boldsymbol{\theta}^k}^k) \delta(\mathbf{x}^k - \emptyset) \\ &\quad + r^{k-} \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \mathbf{x}^k + \mathbf{a}^k, \boldsymbol{\Omega}^k + \mathbf{R}_{\boldsymbol{\theta}^k}^k) \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{x}}^k, \bar{\mathbf{C}}^k),\end{aligned}\quad (6.32)$$

while the marginal PDF of the next measurement becomes (see (5.47))

$$\begin{aligned}p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k) &= \int \pi(\mathbf{x}^k, \mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) d\mathbf{x}^k \\ &\approx (1 - r^{k-}) \mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\boldsymbol{\theta}^k}^k) \\ &\quad + r^{k-} \int \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \mathbf{x}^k + \mathbf{a}^k, \boldsymbol{\Omega}^k + \mathbf{R}_{\boldsymbol{\theta}^k}^k) \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{x}}^k, \bar{\mathbf{C}}^k) d\mathbf{x}^k \\ &= (1 - r^{k-}) \mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\boldsymbol{\theta}^k}^k) + r^{k-} \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \bar{\mathbf{x}}^k + \mathbf{a}^k, \mathbf{S}_{\boldsymbol{\theta}^k}^k),\end{aligned}\quad (6.33)$$

with $\mathbf{S}_{\boldsymbol{\theta}^k}^k$ being given by (6.11). Finally, it is useful to define

$$r_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}} = \frac{r^{k-} \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \bar{\mathbf{x}}^k + \mathbf{a}^k, \mathbf{S}_{\boldsymbol{\theta}^k}^k)}{p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k)}. \quad (6.34)$$

Using (6.33), it follows that

$$\begin{aligned} 1 - r_{\mathbf{z};\theta}^{\mathcal{N}} &= 1 - \frac{r^{k-} \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \bar{\mathbf{x}}^k + \mathbf{a}^k, \mathbf{S}_{\theta^k}^k)}{p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k)} \\ &= \frac{(1 - r^{k-}) \mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\theta^k}^k)}{p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k)}, \end{aligned} \quad (6.35)$$

so that $\pi(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k})$ can also be approximated with a Bernoulli-Gaussian density as

$$\begin{aligned} \pi(\mathbf{x}^k | \mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) &= \frac{\pi(\mathbf{x}^k, \mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})}{p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k)} \\ &\approx \frac{(1 - r^{k-}) \mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\theta^k}^k)}{p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k)} \delta(\mathbf{x}^k - \emptyset) \\ &\quad + \frac{r^{k-} \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \mathbf{x}^k + \mathbf{a}^k, \boldsymbol{\Omega}^k + \mathbf{R}_{\theta^k}^k)}{p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k)} \\ &\quad + \frac{\mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \bar{\mathbf{x}}^k + \mathbf{a}^k, \mathbf{S}_{\theta^k}^k) \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{x}}^k, \bar{\mathbf{C}}^k)}{\mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \bar{\mathbf{x}}^k + \mathbf{a}^k, \mathbf{S}_{\theta^k}^k)} \\ &= (1 - r_{\mathbf{z};\theta}^{\mathcal{N}}) \delta(\mathbf{x}^k - \emptyset) + r_{\mathbf{z};\theta}^{\mathcal{N}} \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{u}}_{\mathbf{z};\theta}, \mathbf{W}_{\theta}) \end{aligned} \quad (6.36)$$

where $\bar{\mathbf{u}}_{\mathbf{z};\theta}$ and $\mathbf{W}_{\theta}$ are respectively given by (6.9) and (6.10). It is worth remembering that, as pointed out in Section 6.2, $\mathbf{W}_{\theta}$ is not dependent on the actual measurement $\mathbf{z}^k$, and can be efficiently computed using sigma-point integration methods.

The base distance metric in GOSPA that measures the localization error is set to be the Euclidean metric

$$d(\hat{\mathbf{x}}_{\mathbf{z};\theta}^k, \mathbf{x}^k) = \sqrt{\text{tr}\{[\hat{\mathbf{x}}_{\mathbf{z};\theta}^k - \mathbf{x}^k][\hat{\mathbf{x}}_{\mathbf{z};\theta}^k - \mathbf{x}^k]^T\}}. \quad (6.37)$$

In addition, it is assumed that $\mathcal{N}(\mathbf{x}^k; \bar{\mathbf{u}}_{\mathbf{z};\theta}, \mathbf{W}_{\theta})$ is mainly contained within a circle of radius c so that we can approximate

$$d^{(c)}(\hat{\mathbf{x}}_{\mathbf{z};\theta}^k, \mathbf{x}^k) \approx d(\hat{\mathbf{x}}_{\mathbf{z};\theta}^k, \mathbf{x}^k). \quad (6.38)$$

6.4.1.1 Specification of the estimator using the Bernoulli-Gaussian approximation

In order to approximate (6.25) in the controller of the FAR, it is required at this point to specify the estimator under consideration $\hat{\mathbf{x}}_{\mathbf{z};\theta}^k$. To this purpose, the likelihood ratio in (5.56) is considered. Making use of (6.29), (6.30) and (6.31), we can approximate

the likelihood ratio in (5.56) as

$$\begin{aligned}
 \bar{\mathcal{L}}(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k}) &\approx \bar{\mathcal{L}}_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}} \\
 &\equiv \frac{\int \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \mathbf{x}^k + \mathbf{a}^k, \boldsymbol{\Omega}^k + \mathbf{R}_{\boldsymbol{\theta}^k}^k) \mathcal{N}(\mathbf{x}^k; \bar{\mathbf{x}}^k, \bar{\mathbf{C}}^k) d\mathbf{x}^k}{\mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\boldsymbol{\theta}^k}^k)} \\
 &= \frac{\mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \bar{\mathbf{x}}^k + \mathbf{a}^k, \mathbf{S}_{\boldsymbol{\theta}^k}^k)}{\mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\boldsymbol{\theta}^k}^k)},
 \end{aligned} \tag{6.39}$$

while the Bayesian likelihood ratio described in (5.55) can be also approximated as

$$\begin{aligned}
 \bar{\Lambda}(\mathbf{z}^{1:k}; \boldsymbol{\theta}^{1:k}) &\approx \bar{\Lambda}_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}} \\
 &\equiv \frac{r^{k-}}{(1 - r^{k-})} \bar{\mathcal{L}}_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}}.
 \end{aligned} \tag{6.40}$$

Making use of (6.40), we can finally define the estimator considered by the controller as

$$\hat{\mathbf{x}}_{\mathbf{z};\boldsymbol{\theta}}^k = \begin{cases} \emptyset & \bar{\Lambda}_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}} < \gamma \\ \boldsymbol{\mu}^k \in \mathbb{R}^{n_x} & \bar{\Lambda}_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}} \geq \gamma \end{cases} \tag{6.41}$$

where $\boldsymbol{\mu}^k$ stands for the conditional mean as in (5.18), which is the optimal estimate in an RMSE sense.

6.4.1.2 FAR controller policy using the Bernoulli-Gaussian approximation

In this section, we describe the approximation of the predicted risk in (6.25) carried out by the controller using the Bernoulli-Gaussian approximation.

Using (6.36), and considering (6.37) and (6.38), the expressions in (6.27) and (6.28) are first approximated as

$$\begin{aligned}
 I_{\mathbf{z};\boldsymbol{\theta}}^1 &\approx I_{\mathbf{z};\boldsymbol{\theta}}^{1,\mathcal{N}} \\
 &= \mathbb{I}\{\bar{\Lambda}_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}} < \gamma\} \frac{c^2}{2} r_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}}
 \end{aligned} \tag{6.42}$$

and

$$\begin{aligned}
 I_{\mathbf{z};\boldsymbol{\theta}}^2 &\approx I_{\mathbf{z};\boldsymbol{\theta}}^{2,\mathcal{N}} \\
 &= \mathbb{I}\{\bar{\Lambda}_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}} \geq \gamma\} \left[\frac{c^2}{2} (1 - r_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}}) + r_{\mathbf{z};\boldsymbol{\theta}}^{\mathcal{N}} \text{tr}(\mathbf{W}_{\boldsymbol{\theta}^k}) \right].
 \end{aligned} \tag{6.43}$$

The predicted conditional Bayes risk of the GOSPA error in (6.25) can therefore be

approximated as

$$\begin{aligned}
 \mathcal{R}_{C_G}^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx \int \left(I_{\mathbf{z}; \boldsymbol{\theta}}^{1, \mathcal{N}} + I_{\mathbf{z}; \boldsymbol{\theta}}^{2, \mathcal{N}} \right) p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k) d\mathbf{z}^k \\
 &= \frac{c^2}{2} \int \mathbb{I} \{ \bar{\Lambda}_{\mathbf{z}; \boldsymbol{\theta}}^{\mathcal{N}} < \gamma \} r_{\mathbf{z}; \boldsymbol{\theta}}^{\mathcal{N}} p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k) d\mathbf{z}^k \\
 &\quad + \frac{c^2}{2} \int \mathbb{I} \{ \bar{\Lambda}_{\mathbf{z}; \boldsymbol{\theta}}^{\mathcal{N}} \geq \gamma \} (1 - r_{\mathbf{z}; \boldsymbol{\theta}}^{\mathcal{N}}) p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k) d\mathbf{z}^k \\
 &\quad + \int \mathbb{I} \{ \bar{\Lambda}_{\mathbf{z}; \boldsymbol{\theta}}^{\mathcal{N}} \geq \gamma \} r_{\mathbf{z}; \boldsymbol{\theta}}^{\mathcal{N}} \text{tr}(\mathbf{W}_{\boldsymbol{\theta}^k}) p(\mathbf{z}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^k) d\mathbf{z}^k,
 \end{aligned} \tag{6.44}$$

which, making use of (6.34) and (6.35) and considering that, as previously emphasized, $\mathbf{W}_{\boldsymbol{\theta}^k}$ does not depend on the current measurement $\mathbf{z}^k$, can be further simplified as

$$\begin{aligned}
 \mathcal{R}_{C_G}^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx \frac{c^2}{2} r^{k-} \int \mathbb{I} \{ \bar{\Lambda}_{\mathbf{z}; \boldsymbol{\theta}}^{\mathcal{N}} < \gamma \} \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \bar{\mathbf{x}}^k + \mathbf{a}^k, \mathbf{S}_{\boldsymbol{\theta}^k}^k) d\mathbf{z}^k \\
 &\quad + \frac{c^2}{2} (1 - r^{k-}) \int \mathbb{I} \{ \bar{\Lambda}_{\mathbf{z}; \boldsymbol{\theta}}^{\mathcal{N}} \geq \gamma \} \mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\boldsymbol{\theta}^k}^k) d\mathbf{z}^k \\
 &\quad + r^{k-} \text{tr}(\mathbf{W}_{\boldsymbol{\theta}^k}) \int \mathbb{I} \{ \bar{\Lambda}_{\mathbf{z}; \boldsymbol{\theta}}^{\mathcal{N}} \geq \gamma \} \mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \bar{\mathbf{x}}^k + \mathbf{a}^k, \mathbf{S}_{\boldsymbol{\theta}^k}^k) d\mathbf{z}^k.
 \end{aligned} \tag{6.45}$$

The integrals in (6.45) could be approximated by means of MC integration methods. However such a computationally intensive approach would not be useful, as in order to minimize $\mathcal{R}_{C_G}^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$ with respect to $\boldsymbol{\theta}^k$, the controller needs to compute (6.45) for every sensor configuration $\boldsymbol{\theta}^k$ under consideration in the optimization procedure. On the other hand, given that the involved densities in the three different integrands of (6.45) are Gaussian, the use of sigma-point methods is considered. Thus, assuming scalar measurements, two different sets of $m = 2N_s + 1$ sigma-points are drawn, $\mathcal{Z}_\phi = \left\{ \left(\mathcal{Z}_\phi^1, w^1 \right), \dots, \left(\mathcal{Z}_\phi^m, w^m \right) \right\}$ and $\mathcal{Z}_x = \{ (\mathcal{Z}_x^1, w^1), \dots, (\mathcal{Z}_x^m, w^m) \}$, which respectively match the first two moments of $\mathcal{N}(\mathbf{z}^k; \mathbf{h}(\emptyset), \mathbf{R}_{\boldsymbol{\theta}^k}^k)$ and $\mathcal{N}(\mathbf{z}^k; \mathbf{H}^k \bar{\mathbf{x}}^k + \mathbf{a}^k, \mathbf{S}_{\boldsymbol{\theta}^k}^k)$, so that (6.45) is approximated as

$$\begin{aligned}
 \mathcal{R}_{C_G}^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx \frac{c^2}{2} r^{k-} \sum_{i=1}^m w^i \mathbb{I} \{ \bar{\Lambda}_{\mathcal{Z}_x^i; \boldsymbol{\theta}}^{\mathcal{N}} < \gamma \} \\
 &\quad + \frac{c^2}{2} (1 - r^{k-}) \sum_{i=1}^m w^i \mathbb{I} \{ \bar{\Lambda}_{\mathcal{Z}_\phi^i; \boldsymbol{\theta}}^{\mathcal{N}} \geq \gamma \} \\
 &\quad + r^{k-} \text{tr}(\mathbf{W}_{\boldsymbol{\theta}^k}) \sum_{i=1}^m w^i \mathbb{I} \{ \bar{\Lambda}_{\mathcal{Z}_x^i; \boldsymbol{\theta}}^{\mathcal{N}} \geq \gamma \}.
 \end{aligned} \tag{6.46}$$

Finally, it should be noted that the selection of the detection threshold γ is also required.

6.4.2 Note on the accuracy of the Bernoulli-Gaussian approximation

A brief comment on the approximation in (6.29) is due. Note that, conditioned on the existence of the target at time k (i.e., $\mathbf{x}^k \in \mathbb{R}^{n_x}$), the *true* prior PDF of the state in (5.43) is formed by a mixture of two weighted PDFs. The first component is $b^k(\mathbf{x}^k)$, the PDF of a target that enters the scenario at time k (provided it was absent at time $k-1$), and the second component, $s^k(\mathbf{x}^k)$, represents the PDF of the target provided that it was present in the scenario at time $k-1$ and survives to time k . It should be noted that even if both these components are unimodal, the resulting prior PDF in (6.29) will in general not be unimodal.

If the prior PDF in (5.43) is not approximately unimodal, then using (6.29) might not generally be a good approximation. In these cases, instead of using a Gaussian approximation as in (6.29), it would be more adequate to consider a Gaussian-mixture approximation with two (or more, if required for the prior PDF at hand) components. The adaptation of the expressions derived in Section 6.4 to such an approximation is straightforward, and would result in the evaluation of the predicted risk expectation in (6.45) using Gaussian-mixture sigma-point integration methods [89, 97, 148, 158].

Despite the above remarks, in common cases of practical use, the Bernoulli-Gaussian approximation in (6.29) can be enough to represent the prior PDF of the state. For instance, when there is enough evidence to be confident about the presence or absence of a target in the scenario, the weighting of the mixture in (5.43) will be such that one of the components of the PDF will stand out, and the Bernoulli-Gaussian approximation will perform satisfactorily.

The two following additional common situations are also of interest. If the target is not present in the scenario, given the past measurements, $s^k(\mathbf{x}^k)$ will usually be approximately uniform across the state space. Thus if, according to some available prior knowledge, targets are known to enter the scenario at some specific points of interest, so that $b^k(\mathbf{x}^k)$ is unimodal, then the mixture in (5.43) will also be approximately unimodal. Note that the converse case is also usual as, if there is not any prior knowledge about how targets enter the scenario, it is common to assume that $b^k(\mathbf{x}^k)$ is uniform. In this case if, given the past measurements, $s^k(\mathbf{x}^k)$ is unimodal (which is usual if the target is present at the scenario) the mixture in (5.43) will also be again approximately unimodal and can also be adequately approximated using (6.29). If such circumstances are not met and the prior in (5.43) is not unimodal, one would have to rely on the above mentioned approach using Gaussian-mixture sigma-point integration methods [89, 97, 148, 158].

Figure 6.13: Simulated target trajectory and sensors. The target enters the scenario at $k = 3$, and leaves it at $k = 39$ with the trajectory being sampled with a period of $\tau = 1s$. The initial and final points of the trajectory are respectively indicated by a colored circle and square. Arrows show the position and direction of the target every 10 time steps. Sensor positions are indicated with a red dot.

6.4.3 Experimental results

In this section we analyze a case of interest which allows to show the benefits of using the previously described Bernoulli-Gaussian approach. As previously mentioned in Section 5.3, the problem of simultaneous detection and tracking is concerned with 3 different objectives: the detection of the target, an accurate tracking of its position, and the track maintenance, i.e., once the presence of the target in the scenario is confirmed the processor needs to reliably verify that the target is still present in the scenario during the duration of the target trajectory, and when the target leaves the scenario, to rapidly notice its absence.

We consider a simple example in which the capabilities of the Bernoulli-Gaussian approach to precisely track a target in a sensor network with resource constraints and satisfactorily perform track maintenance are shown. Section 6.4.3.1 briefly describes the considered dynamic and measurement models, while Section 6.4.3.2 presents the simulation results.

6.4.3.1 Dynamic and measurement model

A resource management scenario is proposed, thus we consider the availability of a limited budget b of resources, so that the sensor cost function takes the form described using (5.10) and (5.14). Measurements at time k , $\mathbf{z}^k = [z_1^k, \dots, z_{N_s}^k]^T$, are received through $N_s = 16$ sensors in a sensor network displayed in a 2-D regular grid covering $30\text{m} \times 30\text{m}$, with a spacing of 10m between sensors in both axes (see Figure 6.13). A received power measurement model is considered, so that measurements of the s -th sensor at time k , with allocated resources θ_s^k , are given by

$$z_s^k = h_s(\mathbf{x}^k) + \frac{v_s^k}{\sqrt{\theta_s^k}} \quad (6.47)$$

where v_s^k is a zero-mean unit-variance ($\sigma_n^2 = 1$) Gaussian-distributed noise. When a target is present at the scenario $h_s(\mathbf{x}^k)$ is given by (3.27) with $Z_0 = 20$, $\alpha = 4$ and $\rho_0 = 5\text{m}$, otherwise if the target is absent, $h_s(\emptyset) = 0$.

Simulations consider a single target which appears in the scenario at time $k = 3$ and leaves the scenario at time $k = 39$ (see Figure 6.13). The state of a target present in the scenario is represented as $\mathbf{x}^k = [x^k, \dot{x}^k, y^k, \dot{y}^k]^T$ (so that the dimension of the state space is $n_x = 4$), with $\mathbf{p}^k = [x^k, y^k]^T$ being the target position and $\mathbf{v}^k = [\dot{x}^k, \dot{y}^k]^T$ is its velocity.

The dynamics of a surviving target, $p(\mathbf{x}^k | \mathbf{x}^{k-1})$ in (5.40), are modeled using a nearly constant velocity model, already described in Section 3.5.1 with $\tau = 1\text{s}$ and $\sigma_u = 0.1\text{m/s}^{3/2}$.

We consider a simplified approach in which the detection process is handled by the controller in the following way: if there is no declared target in the scenario, resources are assigned to sensors according to the uniform resource distribution policy in (6.1) (i.e., resources are equally distributed among sensors). In this situation, whenever a sensor receives a measurement at time k that exceeds a given threshold, the processor initiates the target track according to the Bernoulli Bayesian recursion in Section 5.3.1, and in the following time steps the controller assigns resources to sensors according to the policy under consideration. If a target is declared in the scenario and its posterior probability of existence r^k is such that the Bayesian likelihood ratio in (5.57) goes below the threshold γ , the target is declared absent and the controller returns to its original uniform resource distribution policy.

In this simplified approach, if the target is declared present in the scenario, the Bayesian recursion in Section 5.3.1 is applied, considering $p_b^k = 0$. Note that in making use of this simplified approach we are not focusing in the detection procedure,

Figure 6.14: Average GOSPA position error at each time step using different controller policies: uniform, N-Nearest, the Gaussian approximation in Section 6.2.1 and the Bernoulli-Gaussian approximation in Section 6.4. The resource budget is $B = \log_{10}(b) = 2$. The target enters the scenario at $k = 3$, and leaves it at $k = 39$ as shown in Figure 6.13.

but rather the FAR is considered so as to correctly perform track maintenance and accurately estimate the target position.

The initialization of the target track is done as follows. A Bernoulli prior distribution $\pi(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$ is initialized when a sensor measurement exceeds the measurement threshold at time k . The initial probability of existence of the target is set to $r^{k-} = p_0$, while the associated PDF $p(\mathbf{x}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})$ is set considering a uniform density for the position of the target over the region of the scenario for which the sensor whose measurement exceeded the threshold is the nearest sensor. The initial distribution of the target velocity is a Gaussian with $\mathcal{N}(\mathbf{v}^k; [0, 0]^T, \mathbf{I}_2)$. If more than one sensor measurements exceeds the threshold, the track is initialized as previously described using the sensor with the highest measurement.

The measurement threshold is set so that, given the measurement model in (6.47), the probability of a false alarm is $p_{fa} = 0.001$ [84]. The initial probability of existence is set to $p_0 = 0.1$. The probability of survival of a target from time $k - 1$ to time k conditioned on $\mathbf{x}^{k-1} \in \mathbb{R}^{n_x}$ is set to be $p_S^{k-1}(\mathbf{x}^{k-1}) = 0.95$ (see (5.40)).

We consider different policies for the controller, namely the uniform resource distribution in (6.1), the N-Nearest resource distribution in (6.2), an information-maximization based approach which is in this case the Gaussian approximation outlined

in Section 6.2.1 and the Bernoulli-Gaussian approximation in Section 6.4, setting the cutoff parameter of the GOSPA metric in the controller to $c = 10$.

The use of an auxiliary particle filter [130] with $N = 500$ particles is considered to track the target position, with Rao-Blackwellization of particles also being used as in previous chapters to estimate target velocity (see Section 2.8.1). The performance of the different controller methods is assessed using the GOSPA error metric [134] (with parameters $p = 2$, $\alpha = 2$ and $c = 10$, see Appendix A.9), averaged in a Monte Carlo simulation with $n = 5000$ runs.

As a previous remark, it is worth noting that the target enters the saturation region of sensor $\mathbf{s}_7 = [10, 20]^T$ approximately at time $k = 30$, with the disappearance of the target taking place at time $k = 39$ and being produced inside this saturation region. Thus, considering the measurement model in (6.47), measurements received by $\mathbf{s}_7$ are the ones that can better discriminate the presence or absence of the target at the time it leaves the scenario. Nonetheless, as previously discussed in Section 6.3, when the target PDF completely enters this region, all elements of the sensor EFIM of sensor $\mathbf{s}_7$ will be zero, so that a PC-CRLB minimization would not assign any resources to this sensor. Equivalently, considering the use of another information maximization policy like the one introduced in Section 6.2.1 using a Gaussian approximation would also fail to allocate any resources to $\mathbf{s}_7$, as its measurements would not reduce the covariance of the posterior PDF conditioned on the target existence.

6.4.3.2 Simulation results

We first consider the performance of the different controller strategies when a high resource budget of $B = \log_{10}(b) = 2$ in (5.14) is available. Under such available resources, it is shown in Figure 6.14 that the controller strategy based on the Gaussian approximation as well as the one using a the Bernoulli-Gaussian approach perform satisfactorily. Note that these two policies clearly outperform the uniform resource distribution as well as the heuristic approach of allocating resources to the 4 sensors which are nearer to the estimated target position. Another worth noting aspect is that given the simplified approach considered in the simulations (which has been detailed in Section 6.4.3.1), the detection procedure is not favored by any controller strategy and their capacity of detecting the target at time $k = 3$ is equivalent. This is in accordance with the previously mentioned objective of characterizing the track maintenance ability of the different controller policies in the simultaneous target detection and tracking problem. Nonetheless, it is interesting to note that the controller policies based on the Gaussian approximation and the Bernoulli Gaussian approximation are the ones which

Figure 6.15: Average estimated probability of existence at each time step using different controller policies: uniform, N-Nearest, the Gaussian approximation in Section 6.2.1 and the Bernoulli-Gaussian approximation in Section 6.4. The resource budget is $B = \log_{10}(b) = 2$. The target enters the scenario at $k = 3$, and leaves it at $k = 39$ as shown in Figure 6.13.

more rapidly and successfully lower the GOSPA position error after the detection has been confirmed.

The track maintenance capacity of the different controller policies is also illustrated through Figure 6.15. Note that with such a high available resource budgets, all controller policies are able of successfully maintaining the track and detecting the leaving of the target at time $k = 39$, with a estimated posterior probability of existence $r^k \simeq 1$ during all the time the target is present at the scenario.

Figure 6.16 shows the normalized amount of resources received by sensor $\mathbf{s}_7$ at each time step for the different policies with the above mentioned high resource budget. Note that, as expected, the controller policy based on the information maximization using a Gaussian approximation does not assign any resources to sensor $\mathbf{s}_7$ during the time instants in which the target enters its saturation region (from $k = 30$ to $k = 39$). Interestingly, the controller policy based on the Bernoulli-Gaussian approximation takes this same decision, as it estimates that, with such a high budget measurements coming from surrounding sensors are precise enough to correctly estimate the posterior probability of existence without the need of allocating any resources to $\mathbf{s}_7$, so that all resources can be assigned to minimizing the trace of the posterior covariance of the target, thus improving tracking accuracy.

Figure 6.16: Average resources assigned to sensor s_7 in the scenario in Figure 6.13 at each time step using different controller policies: uniform, N-Nearest, the Gaussian approximation in Section 6.2.1 and the Bernoulli-Gaussian approximation in Section 6.4. The resource budget is $B = \log_{10}(b) = 2$. The target enters the scenario at $k = 3$, and leaves it at $k = 39$ as shown in Figure 6.13.

Figure 6.17: Average GOSPA position error at each time step using different controller policies: uniform, N-Nearest, the Gaussian approximation in Section 6.2.1 and the Bernoulli-Gaussian approximation in Section 6.4. The resource budget is $B = \log_{10}(b) = 0$. The target enters the scenario at $k = 3$, and leaves it at $k = 39$ as shown in Figure 6.13.

In the following, simulations consider that the available resource budget in (5.14) is scarce, so that $B = \log_{10}(b) = 0$. Figure 6.17 shows the average GOSPA position error of the different controller policies in a scenario with such a low budget. Note in this case how, during the initial part of the trajectory of the target, the controller policy based on the Bernoulli-Gaussian approximation has a slightly higher error than the one considering only the information maximization using a Gaussian approximation. This effect is caused by the Bernoulli-Gaussian policy considering the allocation of at least some small amount of resources to sensors that do not have the ability of improving the tracking accuracy, but nonetheless are good to perform track maintenance, and can better indicate if the target is still at the scenario.

The importance of track maintenance is more relevant when the target enters the saturation region of sensor $\mathbf{s}_7$ at time $k = 30$. In this part of the trajectory the policy based on the information maximization using a Gaussian approximation, deeply increases its GOSPA position error, while the policy using the Bernoulli-Gaussian approach still keeps a good performance. Figure 6.18 shows that in the time instants of interest, the controller policy based on the Gaussian approximation experiments a significant drop in the posterior probability of existence, so that this policy has some difficulties in accurately determining the presence of the target in the scenario. Note that the controller policy based on the Bernoulli-Gaussian approach proposed in Section 6.4 is not affected by significant track maintenance problems.

Figure 6.19 shows that several missed detections up to time $k = 38$ and false alarms from time $k = 39$ occur when using the policy based on the Gaussian approximation, while the policy based on the Bernoulli-Gaussian approach does not see any significant increase in its probability of having any of these cardinality errors. The cause behind this behavior can be clearly observed in Figure 6.20, where it can be observed that contrary to the Gaussian approximation policy, the Bernoulli-Gaussian policy allocates a significant amount of resources to sensor $\mathbf{s}_7$ from time $k = 30$ to $k = 39$, which allows for a correct estimation of the presence of the target in the scenario at these time steps. As a final remark, it is also worth highlighting the robustness and performance of the simple, yet effective, heuristic approach of the N-Nearest controller policy.

6.5 Conclusions

In this chapter, we have identified and characterized two robustness issues that may arise in the use of the FAR framework in the problem of selecting the sensor parameters for the objectives of single target tracking and simultaneous target detection and

Figure 6.18: Average estimated probability of existence at each time step using different controller policies: uniform, N-Nearest, the Gaussian approximation in Section 6.2.1 and the Bernoulli-Gaussian approximation in Section 6.4. The resource budget is $B = \log_{10}(b) = 0$. The target enters the scenario at $k = 3$, and leaves it at $k = 39$ as shown in Figure 6.13.

Figure 6.19: Average probability of occurrence of a cardinality error (false alarm or missed detection) at each time step using different controller policies: uniform, N-Nearest, the Gaussian approximation in Section 6.2.1 and the Bernoulli-Gaussian approximation in Section 6.4. The resource budget is $B = \log_{10}(b) = 0$. The target enters the scenario at $k = 3$, and leaves it at $k = 39$ as shown in Figure 6.13.

Figure 6.20: Average resources assigned to sensor $\mathbf{s}_7$ in the scenario in Figure 6.13 at each time step using different controller policies: uniform, N-Nearest, the Gaussian approximation in Section 6.2.1 and the Bernoulli-Gaussian approximation in Section 6.4. The resource budget is $B = \log_{10}(b) = 0$. The target enters the scenario at $k = 3$, and leaves it at $k = 39$ as shown in Figure 6.13.

tracking. In addition, we have introduced two novel methods that can satisfactorily deal with these robustness problems.

Contrary to the minimization of the PC-CRLB, the first of the methods presented in this chapter makes use of a Gaussian approximation to the posterior PDF of the state of the target, does not require the derivation of problem-specific expressions and can be easily and efficiently implemented using sigma-point integration methods. Its performance has been demonstrated in a single target tracking scenario using a sensor network with resource constraints, showing that it can satisfactorily avoid the robustness issues of the PC-CRLB minimization, clearly outperforming it for scarce resource budgets and matching its performance for higher resource budgets.

On the other hand, the second method presented in this chapter is based on a Bernoulli-Gaussian approximation and allows for a robust approach to the problem of simultaneous target detection and tracking in the FAR framework, circumventing the limitations of information maximization approaches like the PC-CRLB minimization or even the linear-Gaussian approach method presented in this same chapter in Section 6.2. The robustness of the second method presented in this chapter has been demonstrated in a scenario where track-maintenance is quite difficult, nonetheless adequately and satisfactorily tackling the problem.

Chapter 7

Conclusions and future work

This chapter presents the conclusions of this thesis and outlines some interesting directions of future work.

7.1 Conclusions

Two well differentiated problems have been considered in this thesis. Part I is related to the important problem of high-dimensional state filtering from the Bayesian perspective, which requires the availability of a dynamic model that describes the evolution of the state with time, and a measurement model that relates the state of the system with the measurements received by sensors. On the other hand, the sensor management problem applies to the frequent case in which measurements received by sensors depend on some parameters that can be tuned by the system controller in order to achieve some objective of interest. Part II of this thesis is devoted to the problem of sensor management within the fully adaptive radar framework.

Regarding the problem of high-dimensional state filtering in Part I of this thesis, Chapter 2 introduces the mathematical framework and reviews some important contributions in the literature related to this field. The limitations of different methods in the literature are first identified and analyzed. In particular, the focus is put on two promising approaches to beat the curse of dimensionality, namely partitioned filters and multiple filters. They both take advantage of some underlying structure of the state space in order to efficiently and robustly perform state filtering.

Two prominent algorithms within partitioned filters are reviewed in Chapter 2. Originally derived for the problem of multitarget tracking, the IP and the PP methods are both able of competently dealing with high-dimensional states. Nevertheless, these

algorithms have several limitations which are outlined in Section 2.9.1. The main contribution of Chapter 3 is to address these limitations making use of auxiliary particle filtering. As a result, two novel partitioned particle filters have been proposed. They both differ in the use of of a subspace resampling (target-resampling) stage which is shown to be useful in difficult filtering situations at the cost, however, of a loss in sample diversity. A third method has been presented in the final part of Chapter 3, which is able of adaptively deciding if the target-resampling stage is advisable. The performance of the proposed filters has been tested in a multitarget tracking problem using a sensor network. However, it is worth noting that these are general algorithms and can be applied to other high-dimensional state filtering problems.

The main difference of multiple filters with respect to partitioned filters is that they separately consider the estimation of each component of a partition of the state by means of an individual filter, thus requiring of a marginalization procedure. The first contribution of Chapter 4 is to consider the use of auxiliary particle filtering within the multiple filtering framework, providing the detailed derivation of a filter whose performance is illustrated in a multitarget tracking application. In addition, two additional multiple filtering methods have been proposed in this chapter. The rationale behind these methods is to provide an accurate approximation of the marginalization integrals required in multiple filtering, an objective which has been achieved by means of sigma-point integration methods. These two latter methods have been shown to significantly outperform previous filters in the literature in a multitarget tracking application.

Chapter 5 introduces the general framework employed in the second part of this thesis, i.e., the fully adaptive radar approach to sensor management. Various applications within this framework, which consider different objectives to be addressed by a sensor network, have been analyzed. The use of the fully adaptive radar framework has, therefore, been first illustrated by reviewing its application to single target tracking using a sensor network. Then, the problem of simultaneous single target detection and tracking in the fully adaptive radar framework has also been reviewed. Finally, the use of the fully adaptive radar framework to the problem of multitarget tracking with a fixed and known number of targets has been introduced.

Chapter 6 discusses the robustness of the application of the different fully adaptive radar implementations covered in Chapter 5. Some flaws in the robustness of these approaches have been uncovered, which can drive to misperformance of the system. Alternative strategies to overcome these flaws have been proposed for the application of the fully adaptive radar framework to single target tracking and simultaneous single target detection and tracking. The robustness of the proposed alternative methods has

been tested through simulated scenarios, where they were shown to reliably perform under difficult situations.

7.2 Future work

The potential directions of future work of this thesis are plenty. In this section, we outline some interesting options, dividing them in two different groups: high-dimensional filtering, and fully adaptive radar.

7.2.1 High-dimensional filtering

An interesting line of research is to consider more realistic dynamic and measurement models. In this aspect, maneuvering targets are a challenging problem for dynamic state estimation [10, 99–104], while measurement models could also take into account other target features besides kinematics such as their shape [51, 55, 189].

An effort in the above direction would help to bring closer the application of the proposed methods to real scenarios. Some interesting applications, already in development, where the presented methods are expected to be useful are the tracking of multiple targets in imaging radars [9, 56, 57, 64, 65, 127, 139, 140] or the estimation of the phenological state of crops using remote sensing data [36–38, 180–182], with this latter application being of use in order to predict seasonal mobility and migrations of vulnerable populations in response to different events such as increases of food prices [194, 195].

It is worth recalling that the root for the outstanding performance of the SP-MPF and SP-MAPF filters presented in Chapter 4 is in the second-order approximation of the densities of the different components of the state, which allows for an accurate approximation of the marginalization integrals of multiple filters. Note in contrast that, using (2.112) in order to draw subparticles, the different partitioned filters (i.e., PP, APP, TRAPP and ATRAPP) make instead use of a first-order approximation of the state of other components of the partition. In order to enhance the performance of partitioned filters, it would therefore be interesting to consider the use of the sigma-point second-order approximation employed by SP-MPF and SP-MAPF to draw subparticles in partitioned filters.

Although both partitioned and multiple filters make use of a partition of the state space, multiple filters individually approximate the marginal PDF of each component while partitioned filters approximate the joint state of all components

of the partition. The availability of joint state estimates can be beneficial for some applications like smoothing, or labeling of targets after trajectory crossings in multiple target tracking [54]. In order to adequately tackle such situations within the multiple filtering framework, it would be useful to build a joint estimate from the marginal densities. Note that a particle from the joint PDF can be easily built by the stacking of subparticles coming from sampling from the marginals, so that the challenging part of this procedure would be the computation of the joint particles weight. The availability of such a method would also allow the consideration of adaptive estimation techniques, where the filtering algorithm can be dynamically changed depending on the instantaneous difficulties and needs of the state estimation problem.

For the sake of simplicity, the use of the multiple filtering framework in this work has focused on the required marginalization procedure of the update step. This has been made possible by assuming that the dynamic model is such that the different components of the partitioned have independent dynamics. Nevertheless, the use of multiple filters is not restricted to this kind of dynamic models. The application of multiple filtering to a broader kind of dynamic models requires, however, an additional marginalization procedure in the prediction step of each filter, with first-order approximations to this step being already available in the literature [40, 42]. There is, therefore, still room for improvement in the computation of these marginalization integrals, and the use of sigma-point integration methods in a similar approach to the one outlined in Chapter 4 with the update step in SP-MPF and SP-MAPF can help to fill this gap.

Another interesting aspect of multiple filters has not been approached in this work: the communication of information between filters. Note that in order to compute the marginalization integrals in multiple filtering, each of the filters requires to receive information about the other components of the state, but can nonetheless individually perform the required computations in individual processing nodes. Different problems can be considered in such a setting [83, 132], among others, the study of performance with a limited transmission bandwidth between different processing nodes, the management of some missing or lost information or the impact of different connectivity configurations between processing nodes. Given that both the accuracy of the estimates of the state as well as the amount of required exchanged information of SP-MPF and SP-MAPF are significantly different to previous multiple filters in the literature, it is worth studying their performance in the above described scenarios.

7.2.2 Fully adaptive radar

As previously pointed out in Section 6.4.2, the method presented in the final part of Chapter 6 to tackle the problem of simultaneous target detection and tracking within the fully adaptive radar framework might be limited by the emergence of multimodal prior PDFs that cannot be adequately approximated using (6.29). As already pointed out in Section 6.4.2, in these cases instead of using a Gaussian approximation as in (6.29), it would be more adequate to consider a Gaussian-mixture approximation with two (or more, if required for the prior PDF at hand) components. This simple modification would help in further increasing the robustness of the proposed method.

Another interesting line of research in this area is the application of the fully adaptive radar framework to multiple target detection and tracking. The tackling of this challenging problem can benefit from the methods described in Chapter 6. The GOSPA metric [134], already considered in Section 6.4.1 for simultaneous single target detection and tracking in the fully adaptive radar framework, is an adequate cost function to tackle this problem, as it can correctly take into account errors coming from several missed detections and false alarms as well as from the accuracy in the tracking of multiple targets. In this aspect, following from the methods outlined in Chapter 6, it would be interesting to study the development of an approach based on a Multi-Bernoulli-Gaussian-mixture approximation of the multitarget prior PDF.

Versatility of multifunction radars allows for the definition of a plethora of different operating modes within the system besides the detection and tracking of targets, e.g., target recognition, jamming protection, clutter and weather survey, target illumination, among others. The final aim in this aspect is therefore to provide an efficient and robust method within the fully adaptive radar framework that can simultaneously handle the different objectives of such type of systems. An interesting direction of research in the fully adaptive radar framework would, therefore, be to implement a controller that tackles this problem making use of the consolidated field of multi-objective optimization [76]. In addition, such an approach would also help in bringing closer current implementations of resource schedulers in multifunction radars to the body of knowledge of the sensor management field.

Regarding practical applications in development in the research group where this thesis has been carried out, an interesting line of research would be to employ the different methods within the fully adaptive radar framework to enhance the performance of 3D radar imaging systems [9, 56, 57, 64, 65, 127, 139, 140], where the sensor could change its configuration to adapt to the operational requirements of complex scenarios, where multiple targets of interest need to be simultaneously explored.

Appendix A

Additional material and related mathematical derivations

A.1 Multivariate Gaussian distributions and related mathematical derivations

In this appendix, multivariate Gaussian PDFs are reviewed and some mathematical derivations related to Gaussian PDFs, which are useful in this thesis, are also provided.

A.1.1 Multivariate Gaussian PDF

A multivariate Gaussian distributed PDF of a vector valued random variable $\mathbf{x} \in \mathbb{R}^{n_x}$ with mean $\bar{\mathbf{x}}$ and covariance matrix $\Sigma_{\mathbf{xx}}$ takes the form

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mathbf{x}; \bar{\mathbf{x}}, \Sigma_{\mathbf{xx}}) \\ &= \frac{1}{(2\pi)^{\frac{n_x}{2}} \det^{\frac{1}{2}}(\Sigma_{\mathbf{xx}})} \exp \left[-\frac{1}{2} (\mathbf{x} - \bar{\mathbf{x}})^T \Sigma_{\mathbf{xx}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \right]. \end{aligned} \quad (\text{A.1})$$

A.1.2 PDF of a linearly transformed multivariate Gaussian PDF

A.1.2.1 Linear transformation

Consider a vector valued Gaussian multivariate random variable $\mathbf{x} \in \mathbb{R}^{n_x}$, whose PDF is thus described by (A.1). Given a linear transformation of $\mathbf{x}$

$$\mathbf{z} = \mathbf{A}\mathbf{x} \quad (\text{A.2})$$

where $\mathbf{A}$ is a known $n_z \times n_x$ invertible matrix, and $\mathbf{z} \in \mathbb{R}^{n_z}$, the PDF $p(\mathbf{z})$ will also be Gaussian [85]. Thus, it will be described by its two first moments, $\bar{\mathbf{z}}$ and $\Sigma_{\mathbf{zz}}$, which can be computed as

$$\bar{\mathbf{z}} = E[\mathbf{z}] \quad (\text{A.3})$$

$$= E[\mathbf{A}\mathbf{x}]$$

$$= \mathbf{A}E[\mathbf{x}]$$

$$= \mathbf{A}\bar{\mathbf{x}}$$

$$\Sigma_{\mathbf{zz}} = E[(\mathbf{z} - E[\mathbf{z}])(\mathbf{z} - E[\mathbf{z}])^T] \quad (\text{A.4})$$

$$= E[(\mathbf{A}\mathbf{x} - \mathbf{A}\bar{\mathbf{x}})(\mathbf{A}\mathbf{x} - \mathbf{A}\bar{\mathbf{x}})^T]$$

$$= E[\mathbf{A}(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T]$$

$$= \mathbf{A}E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T] \mathbf{A}^T$$

$$= \mathbf{A}\Sigma_{\mathbf{xx}}\mathbf{A}^T.$$

The cross-covariance matrix can also be computed as

$$\Sigma_{\mathbf{xz}} = E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{z} - E[\mathbf{z}])^T] \quad (\text{A.5})$$

$$= E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{A}\mathbf{x} - \mathbf{A}\bar{\mathbf{x}})^T]$$

$$= E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T]$$

$$= E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T] \mathbf{A}^T$$

$$= \Sigma_{\mathbf{xx}}\mathbf{A}^T.$$

A.1.2.2 Linear transformation with additive Gaussian noise

In a similar way, consider now the following transformation of $\mathbf{x}$

$$\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{w} \quad (\text{A.6})$$

where $\mathbf{z} \in \mathbb{R}^{n_z}$, $\mathbf{A}$ is a known $n_z \times n_x$ invertible matrix, and $\mathbf{w}$ is a zero-mean Gaussian random variable with PDF $\mathcal{N}(\mathbf{w}; 0, \Sigma_{\mathbf{ww}})$ and independent from $\mathbf{x}$. This case is of interest as it serves to formalize linear model transformations with white Gaussian additive noise, which are widely used. Given that $\mathbf{z}$ is a linear combination of

independent Gaussian random variables, its PDF $p(\mathbf{z})$ will also be Gaussian [85]. We therefore compute its two first moments, $\bar{\mathbf{z}}$ and $\Sigma_{\mathbf{zz}}$, as

$$\bar{\mathbf{z}} = E[\mathbf{z}] \quad (\text{A.7})$$

$$\begin{aligned} &= E[\mathbf{Ax} + \mathbf{w}] \\ &= \mathbf{A}E[\mathbf{x}] + E[\mathbf{w}] \\ &= \mathbf{A}\bar{\mathbf{x}} \\ \Sigma_{\mathbf{zz}} &= E[(\mathbf{z} - E[\mathbf{z}])(\mathbf{z} - E[\mathbf{z}])^T] \\ &= E[(\mathbf{Ax} + \mathbf{w} - \mathbf{A}\bar{\mathbf{x}})(\mathbf{Ax} + \mathbf{w} - \mathbf{A}\bar{\mathbf{x}})^T] \\ &= E[(\mathbf{A}(\mathbf{x} - \bar{\mathbf{x}}) + \mathbf{w})(\mathbf{w}^T + (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T)] \\ &= E[\mathbf{A}(\mathbf{x} - \bar{\mathbf{x}})\mathbf{w}^T] + E[\mathbf{A}(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T] \\ &\quad + E[\mathbf{w}(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T] + E[\mathbf{w}\mathbf{w}^T] \\ &= \mathbf{A}\Sigma_{\mathbf{xx}}\mathbf{A}^T + \Sigma_{\mathbf{ww}}. \end{aligned} \quad (\text{A.8})$$

The cross-covariance matrix can also be computed as

$$\begin{aligned} \Sigma_{\mathbf{xz}} &= E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{z} - E[\mathbf{z}])^T] \\ &= E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{Ax} + \mathbf{w} - \mathbf{A}\bar{\mathbf{x}})^T] \\ &= E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{w}^T + (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T)] \\ &= E[(\mathbf{x} - \bar{\mathbf{x}})\mathbf{w}^T] + E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T] \\ &= \Sigma_{\mathbf{xx}}\mathbf{A}^T. \end{aligned} \quad (\text{A.9})$$

A.1.2.3 Affine transformation with additive Gaussian noise

In a similar way, consider now the following transformation of $\mathbf{x}$

$$\mathbf{z} = \mathbf{Ax} + \mathbf{a} + \mathbf{w} \quad (\text{A.10})$$

where $\mathbf{z} \in \mathbb{R}^{n_z}$, $\mathbf{A}$ is a known $n_z \times n_x$ invertible matrix, $\mathbf{a} \in \mathbb{R}^{n_z}$ is a known vector and $\mathbf{w}$ is a zero-mean Gaussian random variable with PDF $\mathcal{N}(\mathbf{w}; 0, \Sigma_{\mathbf{ww}})$ and independent from $\mathbf{x}$. As before, we compute its two first moments, $\bar{\mathbf{z}}$ and $\Sigma_{\mathbf{zz}}$, as

$$\begin{aligned} \bar{\mathbf{z}} &= E[\mathbf{z}] \\ &= E[\mathbf{Ax} + \mathbf{a} + \mathbf{w}] \\ &= \mathbf{A}E[\mathbf{x}] + \mathbf{a} + E[\mathbf{w}] \end{aligned} \quad (\text{A.11})$$

$$\begin{aligned}
&= \mathbf{A}\bar{\mathbf{x}} + \mathbf{a} \\
\Sigma_{\mathbf{zz}} &= E \left[(\mathbf{z} - E[\mathbf{z}]) (\mathbf{z} - E[\mathbf{z}])^T \right] \\
&= E \left[(\mathbf{Ax} + \mathbf{a} + \mathbf{w} - (\mathbf{A}\bar{\mathbf{x}} + \mathbf{a})) (\mathbf{Ax} + \mathbf{a} + \mathbf{w} - (\mathbf{A}\bar{\mathbf{x}} + \mathbf{a}))^T \right] \\
&= E \left[(\mathbf{A}(\mathbf{x} - \bar{\mathbf{x}}) + \mathbf{w}) (\mathbf{w}^T + (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T) \right] \\
&= E \left[\mathbf{A}(\mathbf{x} - \bar{\mathbf{x}}) \mathbf{w}^T \right] + E \left[\mathbf{A}(\mathbf{x} - \bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T \right] \\
&\quad + E \left[\mathbf{w} (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T \right] + E \left[\mathbf{w} \mathbf{w}^T \right] \\
&= \mathbf{A} \Sigma_{\mathbf{xx}} \mathbf{A}^T + \Sigma_{\mathbf{ww}}.
\end{aligned} \tag{A.12}$$

The cross-covariance matrix can also be computed as

$$\begin{aligned}
\Sigma_{\mathbf{xz}} &= E \left[(\mathbf{x} - E[\mathbf{x}]) (\mathbf{z} - E[\mathbf{z}])^T \right] \\
&= E \left[(\mathbf{x} - \bar{\mathbf{x}}) (\mathbf{Ax} + \mathbf{a} + \mathbf{w} - (\mathbf{A}\bar{\mathbf{x}} + \mathbf{a}))^T \right] \\
&= E \left[(\mathbf{x} - \bar{\mathbf{x}}) (\mathbf{w}^T + (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T) \right] \\
&= E \left[(\mathbf{x} - \bar{\mathbf{x}}) \mathbf{w}^T \right] + E \left[(\mathbf{x} - \bar{\mathbf{x}}) (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{A}^T \right] \\
&= \Sigma_{\mathbf{xx}} \mathbf{A}^T.
\end{aligned} \tag{A.13}$$

A.1.3 PDF of jointly Gaussian multivariate variables

Given a random vector $[\mathbf{x}, \mathbf{z}]^T$ where $\mathbf{x} \in \mathbb{R}^{n_x}$ and $\mathbf{z} \in \mathbb{R}^{n_z}$ are jointly Gaussian. The joint PDF of $[\mathbf{x}, \mathbf{z}]^T$ is

$$\begin{aligned}
p(\mathbf{x}, \mathbf{z}) &= \mathcal{N} \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}; \begin{bmatrix} \bar{\mathbf{x}} \\ \bar{\mathbf{z}} \end{bmatrix}, \Sigma \right) \\
&= \frac{1}{(2\pi)^{\frac{n_x + n_z}{2}} \det^{\frac{1}{2}}(\Sigma)} \exp \left[-\frac{1}{2} Q(\mathbf{x}, \mathbf{z}) \right]
\end{aligned} \tag{A.14}$$

where

$$Q(\mathbf{x}, \mathbf{z}) = \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{z}} \end{bmatrix}^T \Sigma^{-1} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{z}} \end{bmatrix}, \tag{A.15}$$

$$\begin{aligned}
\tilde{\mathbf{x}} &= \mathbf{x} - \bar{\mathbf{x}} \\
\tilde{\mathbf{z}} &= \mathbf{z} - \bar{\mathbf{z}}
\end{aligned} \tag{A.16}$$

and

$$\Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xz} \\ (\Sigma_{xz})^T & \Sigma_{zz} \end{bmatrix}. \quad (\text{A.17})$$

A.1.3.1 Conditional PDFs

In order to obtain the conditional PDFs, $p(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{z}|\mathbf{x})$, we first rewrite the determinant and the inverse of the covariance matrix Σ . Using the block matrix determinant formula we can write

$$\det(\Sigma) = \det(\Sigma_{xx}) \det(\beta) \quad (\text{A.18})$$

where

$$\beta = \left(\Sigma_{zz} - (\Sigma_{xz})^T \Sigma_{xx}^{-1} \Sigma_{xz} \right). \quad (\text{A.19})$$

On the other hand, using block-matrix inversion formulas, we can write

$$\Sigma^{-1} = \begin{bmatrix} \left(\Sigma_{xx} - \Sigma_{xz} \Sigma_{zz}^{-1} (\Sigma_{xz})^T \right)^{-1} & -\Sigma_{xx}^{-1} \Sigma_{xz} \beta^{-1} \\ -\beta^{-1} (\Sigma_{xz})^T \Sigma_{xx}^{-1} & \beta^{-1} \end{bmatrix}, \quad (\text{A.20})$$

using the matrix inversion lemma

$$\left(\Sigma_{xx} - \Sigma_{xz} \Sigma_{zz}^{-1} (\Sigma_{xz})^T \right)^{-1} = \Sigma_{xx}^{-1} + \Sigma_{xx}^{-1} \Sigma_{xz} \left(\Sigma_{zz} - (\Sigma_{xz})^T \Sigma_{xx}^{-1} \Sigma_{xz} \right)^{-1} (\Sigma_{xz})^T \Sigma_{xx}^{-1}, \quad (\text{A.21})$$

so that

$$\Sigma^{-1} = \begin{bmatrix} \Sigma_{xx}^{-1} + \Sigma_{xx}^{-1} \Sigma_{xz} \beta^{-1} (\Sigma_{xz})^T \Sigma_{xx}^{-1} & -\Sigma_{xx}^{-1} \Sigma_{xz} \beta^{-1} \\ -\beta^{-1} (\Sigma_{xz})^T \Sigma_{xx}^{-1} & \beta^{-1} \end{bmatrix} = \begin{bmatrix} \Gamma_{xx} & \Gamma_{xz} \\ \Gamma_{zx} & \Gamma_{zz} \end{bmatrix}. \quad (\text{A.22})$$

Taking into account that the covariance matrix is symmetric, its inverse will also be symmetric. Thus, $\Gamma_{zx} = (\Gamma_{xz})^T$ and we can write $Q(\mathbf{x}, \mathbf{z})$ in (A.15) as

$$\begin{aligned} Q(\mathbf{x}, \mathbf{z}) &= \tilde{\mathbf{x}}^T \Gamma_{xx} \tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T \Gamma_{xz} \tilde{\mathbf{z}} + \tilde{\mathbf{z}}^T \Gamma_{zx} \tilde{\mathbf{x}} + \tilde{\mathbf{z}}^T \Gamma_{zz} \tilde{\mathbf{z}} \\ &= \tilde{\mathbf{x}}^T \Gamma_{xx} \tilde{\mathbf{x}} + \tilde{\mathbf{x}}^T \Gamma_{xz} \tilde{\mathbf{z}} + \left(\tilde{\mathbf{x}}^T (\Gamma_{zx})^T \tilde{\mathbf{z}} \right)^T + \tilde{\mathbf{z}}^T \Gamma_{zz} \tilde{\mathbf{z}} \\ &= \tilde{\mathbf{x}}^T \Gamma_{xx} \tilde{\mathbf{x}} + 2\tilde{\mathbf{x}}^T \Gamma_{xz} \tilde{\mathbf{z}} + \tilde{\mathbf{z}}^T \Gamma_{zz} \tilde{\mathbf{z}} \end{aligned} \quad (\text{A.23})$$

where substituting with the above expressions for the inverse covariance matrix blocks,

and taking into account that $\Sigma_{\mathbf{x}\mathbf{x}}^{-1}$ and β^{-1} are symmetric (as they are the inverse of symmetric matrices)

$$\begin{aligned} Q(\mathbf{x}, \mathbf{z}) &= \tilde{\mathbf{x}}^T \left[\Sigma_{\mathbf{x}\mathbf{x}}^{-1} + \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \Sigma_{\mathbf{x}\mathbf{z}} \beta^{-1} (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \right] \tilde{\mathbf{x}} \\ &\quad + 2\tilde{\mathbf{x}}^T \left[-\Sigma_{\mathbf{x}\mathbf{x}}^{-1} \Sigma_{\mathbf{x}\mathbf{z}} \beta^{-1} \right] \tilde{\mathbf{z}} + \tilde{\mathbf{z}}^T \beta^{-1} \tilde{\mathbf{z}} \\ &= \tilde{\mathbf{x}}^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \tilde{\mathbf{x}} \\ &\quad + \tilde{\mathbf{x}}^T \left[\Sigma_{\mathbf{x}\mathbf{x}}^{-1} \Sigma_{\mathbf{x}\mathbf{z}} \beta^{-1} (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \right] \tilde{\mathbf{x}} - 2\tilde{\mathbf{x}}^T \left[\Sigma_{\mathbf{x}\mathbf{x}}^{-1} \Sigma_{\mathbf{x}\mathbf{z}} \beta^{-1} \right] \tilde{\mathbf{z}} + \tilde{\mathbf{z}}^T \beta^{-1} \tilde{\mathbf{z}}. \end{aligned} \quad (\text{A.24})$$

Now considering,

$$\hat{\mathbf{y}} = (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \tilde{\mathbf{x}}, \quad (\text{A.25})$$

we can write

$$\begin{aligned} Q(\mathbf{x}, \mathbf{z}) &= \tilde{\mathbf{x}}^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \tilde{\mathbf{x}} + \hat{\mathbf{y}}^T \beta^{-1} \hat{\mathbf{y}} - \hat{\mathbf{y}}^T \beta^{-1} \tilde{\mathbf{z}} - \tilde{\mathbf{z}}^T (\beta^{-1})^T \hat{\mathbf{y}} + \tilde{\mathbf{z}}^T \beta^{-1} \tilde{\mathbf{z}} \quad (\text{A.26}) \\ &= \tilde{\mathbf{x}}^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \tilde{\mathbf{x}} + \hat{\mathbf{y}}^T \beta^{-1} [\hat{\mathbf{y}} - \tilde{\mathbf{z}}] - \tilde{\mathbf{z}}^T \beta^{-1} [\hat{\mathbf{y}} - \tilde{\mathbf{z}}] \\ &= \tilde{\mathbf{x}}^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \tilde{\mathbf{x}} + [\hat{\mathbf{y}} - \tilde{\mathbf{z}}]^T \beta^{-1} [\hat{\mathbf{y}} - \tilde{\mathbf{z}}] \\ &= \tilde{\mathbf{x}}^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \tilde{\mathbf{x}} + [\tilde{\mathbf{z}} - \hat{\mathbf{y}}]^T \beta^{-1} [\tilde{\mathbf{z}} - \hat{\mathbf{y}}] \\ &= Q_1(\mathbf{x}) + Q_2(\mathbf{x}, \mathbf{z}) \end{aligned}$$

where

$$Q_1(\mathbf{x}) = \tilde{\mathbf{x}}^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \tilde{\mathbf{x}} \quad (\text{A.27})$$

$$\begin{aligned} &= [\mathbf{x} - \bar{\mathbf{x}}]^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} [\mathbf{x} - \bar{\mathbf{x}}] \\ Q_2(\mathbf{x}, \mathbf{z}) &= [\tilde{\mathbf{z}} - \hat{\mathbf{y}}]^T \beta^{-1} [\tilde{\mathbf{z}} - \hat{\mathbf{y}}] \quad (\text{A.28}) \\ &= \left[\mathbf{z} - \bar{\mathbf{z}} - (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \right]^T \beta^{-1} \left[\mathbf{z} - \bar{\mathbf{z}} - (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \right] \\ &= \left[\mathbf{z} - \left(\bar{\mathbf{z}} + (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \right) \right]^T \beta^{-1} \left[\mathbf{z} - \left(\bar{\mathbf{z}} + (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \right) \right]. \end{aligned}$$

The joint distribution can therefore be written as

$$\begin{aligned} p(\mathbf{x}, \mathbf{z}) &= \frac{1}{(2\pi)^{\frac{n_x+n_z}{2}} \det^{\frac{1}{2}}(\Sigma)} \exp \left[-\frac{1}{2} Q(\mathbf{x}, \mathbf{z}) \right] \quad (\text{A.29}) \\ &= \frac{1}{(2\pi)^{\frac{n_x+n_z}{2}} \det^{\frac{1}{2}}(\Sigma_{\mathbf{x}\mathbf{x}}) \det^{\frac{1}{2}}(\beta)} \exp \left[-\frac{1}{2} (Q_1(\mathbf{x}) + Q_2(\mathbf{x}, \mathbf{z})) \right]. \end{aligned}$$

The conditional PDF can be obtained using

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} \quad (\text{A.30})$$

$$\begin{aligned}
 &= \frac{\frac{1}{(2\pi)^{\frac{n_x+n_z}{2}} \det^{\frac{1}{2}}(\Sigma_{\mathbf{x}\mathbf{x}}) \det^{\frac{1}{2}}(\beta)} \exp \left[-\frac{1}{2} (Q_1(\mathbf{x}) + Q_2(\mathbf{x}, \mathbf{z})) \right]}{\frac{1}{(2\pi)^{\frac{n_x}{2}} \det^{\frac{1}{2}}(\Sigma_{\mathbf{x}\mathbf{x}})} \exp \left[-\frac{1}{2} Q_1(\mathbf{x}) \right]} \\
 &= \frac{1}{(2\pi)^{\frac{n_z}{2}} \det^{\frac{1}{2}}(\beta)} \exp \left[-\frac{1}{2} Q_2(\mathbf{x}, \mathbf{z}) \right] \\
 &= \mathcal{N} \left(\mathbf{z}; \bar{\mathbf{z}} + (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}), \beta \right) \\
 &= \mathcal{N} \left(\mathbf{z}; \bar{\mathbf{z}} + (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} (\mathbf{x} - \bar{\mathbf{x}}), \Sigma_{\mathbf{z}\mathbf{z}} - (\Sigma_{\mathbf{x}\mathbf{z}})^T \Sigma_{\mathbf{x}\mathbf{x}}^{-1} \Sigma_{\mathbf{x}\mathbf{z}} \right).
 \end{aligned}$$

An equivalent expression for the conditional PDF $p(\mathbf{x}|\mathbf{z})$ can be obtained as

$$\begin{aligned}
 p(\mathbf{x}|\mathbf{z}) &= \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{z})} \\
 &= \mathcal{N} \left(\mathbf{x}; \bar{\mathbf{x}} + \Sigma_{\mathbf{x}\mathbf{z}} \Sigma_{\mathbf{z}\mathbf{z}}^{-1} (\mathbf{z} - \bar{\mathbf{z}}), \Sigma_{\mathbf{x}\mathbf{x}} - \Sigma_{\mathbf{x}\mathbf{z}} \Sigma_{\mathbf{z}\mathbf{z}}^{-1} (\Sigma_{\mathbf{x}\mathbf{z}})^T \right).
 \end{aligned} \tag{A.31}$$

A.2 Partitioned particle filter PDF

In this appendix, we analyze the particle filter PDF of a partitioned particle filter. Following (2.103) we have that the prior PDF of a partitioned particle filter takes the form

$$p(\mathbf{X}^k | \mathbf{z}^{1:k-1}) \simeq \sum_{i=1}^N \omega_i^{k-1} \prod_{j=1}^t p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}). \tag{A.32}$$

The marginal PDF of each component in the partition can therefore be obtained as

$$\begin{aligned}
 p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1}) &= \int p(\mathbf{X}^k | \mathbf{z}^{1:k-1}) d\hat{\mathbf{X}}_{-\{j\}}^k \\
 &\simeq \int \left[\sum_{i=1}^N \omega_i^{k-1} \prod_{l=1}^t p(\mathbf{x}_l^k | \mathbf{x}_{l,i}^{k-1}) \right] d\hat{\mathbf{X}}_{-\{j\}}^k \\
 &= \sum_{i=1}^N \omega_i^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}) \int \int \dots \int \left[\prod_{l=1}^t p(\mathbf{x}_l^k | \mathbf{x}_{l,i}^{k-1}) \right] d\mathbf{x}_1^k \dots d\mathbf{x}_{j-1}^k d\mathbf{x}_{j+1}^k \dots d\mathbf{x}_t^k \\
 &= \sum_{i=1}^N \omega_i^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}).
 \end{aligned} \tag{A.33}$$

We now recall that partitioned particle filters assume that the prior of the different components is independent, so that we can write again the prior PDF of the complete state as the product over the marginals as

$$p(\mathbf{X}^k | \mathbf{z}^{1:k-1}) = \prod_{j=1}^t p(\mathbf{x}_j^k | \mathbf{z}^{1:k-1}) \tag{A.34}$$

Figure A.1: Simulated trajectories with 100 steps of two targets. The initial and final points of each trajectory are respectively indicated by a colored circle and square with arrows showing the position and direction of the target every 20 time steps. The positions of the targets are exactly the same up to $k = 30$ (indicated by a black square). After that, target trajectories separate. Sensor positions are indicated as red dots and receive measurements according to the additive received power measurement model in (3.26) .

$$= \prod_{j=1}^t \sum_{i=1}^N \omega_i^{k-1} p(\mathbf{x}_j^k | \mathbf{x}_{j,i}^{k-1}) .$$

A.3 Minimum mean squared error estimation of a multitarget state using a PF

As discussed in Section 5.4, the use of the squared error cost function (which results in a MMSE estimator) in (5.65) to estimate a multiple target state can pose a limit in the applicability of the approach of the minimization of the PC-CRLB (which bounds the error of the MMSE estimator) in order to select the next set of sensor parameters. In Section 5.4, it is argued that the use of a PF to track the multitarget state can help to alleviate this problem. In this section, we illustrate this point with a simulation of a 2 target scenario.

The root of the problem are the situations when targets get in close proximity, or more problematically, when target trajectories separate after remaining close to each other for a relatively long period of time (i.e., when the posterior PDFs of the targets

significantly overlap in all state parameters). In these situations, the posterior PDF of each target state $\mathbf{x}_j^k$ can become multimodal, and the use of the cost function in (5.65) will make the estimation of the positions of the targets to lie in the middle of the true positions of the targets [23, 24, 49].

The use of a PF to propagate the multitarget posterior PDF can help to lower the impact of these situations. The necessity of a resampling stage to overcome particle degeneracy (see Section 2.7.1.1) in PFs limits their capacity of maintaining multimodalities in the posterior PDF for large periods of time [23, 24, 121, 122, 179], so that one of the modes of the PF will eventually be lost due to particle impoverishment. This fact is usually a disadvantage of PF methods, as the generally desired purpose would be to propagate the true multimodal PDFs of the targets. An advantage of propagating these distributions is that it is in the multimodality where the information about target labels lies.

After the disappearance of modes of the multimodal PDFs, PFs will generally tend to keep unimodal densities with one mode associated to the real position of each target of the targets, from where we can adequately extract its estimated position. Thus, if the interest is in the estimation of target positions and there is no interest in target labeling, we can stand the vanishing of modes of the PDFs, so that the estimation error will only be high while the PF retains multimodalities (which will not be long periods of time).

In order to characterize the behavior of PFs in above described situation we now present the results of a simulation of a 2 target scenario where the simulated trajectories of the 2 targets are identical up to a given time, from which on the trajectories separate from each other (see Figure A.1). This is a worst case scenario in the sense described above, as all the parameters of the state of the targets are identical until the target trajectories separate. The state of each target at time k is given by $\mathbf{x}_j^k = [x_j^k, \dot{x}_j^k, y_j^k, \dot{y}_j^k]^T$, i.e., it contains the target position $\mathbf{p}_j^k = [x_j^k, y_j^k]^T$ and velocity $\mathbf{v}_j^k = [\dot{x}_j^k, \dot{y}_j^k]^T$. Sensors receive measurements according to the additive received power measurement model in (3.26). Simulations are carried out using the PF presented in Section 4.2.1.

Figure A.2 shows the resulting MMSE estimates of the proposed trajectories of $n = 10.000$ runs using a PF. Considering the estimation of one of the targets, for example that in Figure A.2(a), note that the estimation is always approximately correct as long as the two targets remain near. However, the separation of the true target trajectories at time $k = 30$ causes the emergence of a bimodal PDF, with one mode centered in the true position of the target of interest in this case and the other one centered in the position of the other target. The consequences of this behavior are

(a)

(b)

Figure A.2: True trajectory of targets 1 (a) and 2 (b) superimposed over 10000 estimations of the trajectory using a PF. The true position of the target at each time step is represented as a solid dot. The 10.000 estimated trajectories are represented as semi-transparent lines.

observed in Figure A.2 in the time steps following the separation of the trajectories, when the MMSE estimations of the targets are not centered in their true positions, but rather lie in the middle of their true positions. This effect is not held for a long period of time, and by time $k = 50$, when the true target positions are separated by 5.07m, only 1.7% of the MMSE estimations are separated from one of the true position of the targets by more than 1.5m. However, these MMSE estimations would have a very poor labeling performance, as the estimated position of the target of interest lies almost indistinctly in the surroundings of its true position or in the surroundings of the position of the other target. Nevertheless, depending on the application, if target labeling is not of interest and we only seek to estimate target positions regardless of their true labels, the use of the MMSE estimator of the position of the targets along with a PF would not have a bad performance, as the multimodalities of the posterior PDF rapidly disappear due to particle impoverishment.

A.4 Sensor EFIM expression for resource management in a sensor network

The derivatives in the sensor EFIMs in (5.34) are

$$\begin{aligned}
 \Delta_{\mathbf{x}}^{\mathbf{x}} [\ln p(z_s^k | \mathbf{x}^k; \theta_s^k)] &= \nabla_{\mathbf{x}} [\nabla_{\mathbf{x}} [\ln (p(z_s^k | \mathbf{x}^k; \theta_s^k))]]^T \quad (\text{A.35}) \\
 &= \nabla_{\mathbf{x}} \left[\nabla_{\mathbf{x}} \left[\ln \left(\sqrt{\theta_s^k / (2\pi\sigma_n^2)} \right) + \left(-\frac{(z_s^k - h_s(\mathbf{x}^k))^2}{2\sigma_n^2/\theta_s^k} \right) \right] \right]^T \\
 &= \nabla_{\mathbf{x}} \left[\nabla_{\mathbf{x}} \left[-\frac{(z_s^k - h_s(\mathbf{x}^k))^2}{2\sigma_n^2/\theta_s^k} \right] \right]^T \\
 &= \nabla_{\mathbf{x}} \left[\frac{\theta_s^k}{\sigma_n^2} (z_s^k - h_s(\mathbf{x}^k)) \nabla_{\mathbf{x}} h_s(\mathbf{x}^k) \right]^T \\
 &= \frac{\theta_s^k}{\sigma_n^2} (z_s^k - h_s(\mathbf{x}^k)) \Delta_{\mathbf{x}}^{\mathbf{x}} h_s(\mathbf{x}^k) - \frac{\theta_s^k}{\sigma_n^2} \left[\nabla_{\mathbf{x}} h_s(\mathbf{x}^k) [\nabla_{\mathbf{x}} h_s(\mathbf{x}^k)]^T \right].
 \end{aligned}$$

Considering this last derivation and introducing it in the expression of the sensor EFIM

$$\begin{aligned}
 \mathbf{J}_{k,s}^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\equiv E^- \left\{ -E \left\{ \Delta_{\mathbf{x}}^{\mathbf{x}} \ln (p(z_s^k | \mathbf{x}^k; \theta_s^k)) \right\} \right\} \quad (\text{A.36}) \\
 &= E^- \left\{ -E \left\{ \frac{\theta_s^k}{\sigma_n^2} [(z_s^k - h_s(\mathbf{x}^k)) \Delta_{\mathbf{x}}^{\mathbf{x}} h_s(\mathbf{x}^k)] \right\} \right\} \\
 &\quad - E^- \left\{ -E \left\{ \frac{\theta_s^k}{\sigma_n^2} \nabla_{\mathbf{x}} h_s(\mathbf{x}^k) [\nabla_{\mathbf{x}} h_s(\mathbf{x}^k)]^T \right\} \right\}
 \end{aligned}$$

where the first term is canceled out in the inner expectation

$$E \left\{ \frac{\theta_s^k}{\sigma_n^2} [(z_s^k - h_s(\mathbf{x}^k)) \Delta_{\mathbf{x}}^{\mathbf{x}} h_s(\mathbf{x}^k)] \right\} = \mathbf{0} \quad (\text{A.37})$$

as $E[z_s^k] = h_s(\mathbf{x}^k)$. On the other hand, the second term passes the inner expectation unaltered, as it does not depend on z_s^k , so that the sensor EFIM can be written as

$$\mathbf{J}_{k,s}^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = E^- \left\{ \frac{\theta_s^k}{\sigma_n^2} \nabla_{\mathbf{x}} h_s(\mathbf{x}^k) [\nabla_{\mathbf{x}} h_s(\mathbf{x}^k)]^T \right\}. \quad (\text{A.38})$$

A.5 Sensor EFIM for received power measurement model

We consider the received power measurement model so that given the target with position $\mathbf{x}^k = [x^k, y^k]^T$ at time k , the s -th sensor with position vector $\mathbf{s}_s = [s_{s,x}, s_{s,y}]^T$ receives measurements according to (3.27). Thus, the gradient in (A.38) is

$$\nabla_{\mathbf{x}} h_s(\mathbf{x}^k) = \begin{cases} \mathbf{0} & \rho_s(\mathbf{x}^k) < \rho_0 \\ -\alpha Z_0 \rho_0^\alpha (\rho_s(\mathbf{x}^k))^{-\alpha-1} \hat{\boldsymbol{\rho}}_s(\mathbf{x}^k) & \rho_s(\mathbf{x}^k) > \rho_0 \end{cases} \quad (\text{A.39})$$

where $\hat{\boldsymbol{\rho}}_s(\mathbf{x})$ is the unitary vector pointing from the sensor to the target

$$\hat{\boldsymbol{\rho}}_s(\mathbf{x}) \equiv \frac{1}{\rho_s(\mathbf{x})} [(x - s_{s,x}), (y - s_{s,y})]^T, \quad (\text{A.40})$$

so that the sensor EFIM in (A.38) takes the form

$$\mathbf{J}_s^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) = \frac{\theta_s^k}{\sigma_n^2} (\alpha Z_0 \rho_0^\alpha)^2 \cdot E^- \left\{ \mathbb{I}_{\rho_0}^{\rho_s} \{\mathbf{x}^k\} (\rho_s(\mathbf{x}^k))^{-2(\alpha+1)} (\hat{\boldsymbol{\rho}}_s(\mathbf{x}^k) \hat{\boldsymbol{\rho}}_s(\mathbf{x}^k)^T) \right\} \quad (\text{A.41})$$

where the indicator variable (which allows for the incorporation of both cases of the computed gradient in (A.39) in a compact expression) is defined as

$$\mathbb{I}_{\rho_0}^{\rho_s} \{\mathbf{x}\} \equiv \begin{cases} 0 & \rho_s(\mathbf{x}) < \rho_0 \\ 1 & \rho_s(\mathbf{x}) > \rho_0 \end{cases}. \quad (\text{A.42})$$

It is worth recalling that if a discrete approximation to the prior PDF is available, like the one in (5.27), the remaining expectation in (A.41) can be numerically computed

as

$$\mathbf{J}_s^-(\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \approx \frac{\theta_s^k}{\sigma_n^2} (\alpha Z_0 \rho_0^\alpha)^2 \cdot \sum_{i=1}^N \omega_i^{k-1} \mathbb{I}_{\rho_0}^{\rho_s} \{ \mathbf{x}_i^k \} (\rho_s(\mathbf{x}_i^k))^{-2(\alpha+1)} \left(\hat{\boldsymbol{\rho}}_s(\mathbf{x}_i^k) \hat{\boldsymbol{\rho}}_s(\mathbf{x}_i^k)^T \right). \quad (\text{A.43})$$

A.6 Sensor EFIM for angle of arrival measurement models

We consider an angle of arrival measurement model so that the so that given the target with position $\mathbf{x}^k = [x^k, y^k]^T$ at time k , the measurement model of the s -th sensor with position vector $\mathbf{s}_s = [s_{s,x}, s_{s,y}]^T$ can be expressed as

$$\begin{aligned} h_s(\mathbf{x}^k) &= \phi_s(\mathbf{x}^k) \\ &= \arctan2(y^k - s_{s,y}, x^k - s_{s,x}) \end{aligned} \quad (\text{A.44})$$

where we have used a 4-quadrant arctangent function (which is often depicted as atan2 or arctan2 in most programming languages)

$$\arctan2(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{if } x > 0, \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0 \end{cases}, \quad (\text{A.45})$$

Given the measurement model in (A.44), at the regions where partial derivatives exist, the measurement function is equal to that of $\arctan\left(\frac{y^k - s_{s,y}}{x^k - s_{s,x}}\right)$, except for a constant, so that the gradient in (A.38) is

$$\begin{aligned} \nabla_{\mathbf{x}} h_s(\mathbf{x}^k) &= \frac{1}{(\rho_s(\mathbf{x}^k))^2} [-(y^k - s_{s,y}), (x^k - s_{s,x})]^T \\ &= \frac{1}{\rho_s(\mathbf{x}^k)} \hat{\mathbf{u}}_s(\mathbf{x}^k) \end{aligned} \quad (\text{A.46})$$

where $\mathbf{u}_s(\mathbf{x}^k)$ is the unit vector perpendicular to the vector pointing from the sensor

to the target

$$\hat{\mathbf{u}}_s(\mathbf{x}^k) \equiv \frac{1}{\rho_s(\mathbf{x}^k)} \left[- (y^k - s_{s,y}), (x^k - s_{s,x}) \right]^T. \quad (\text{A.47})$$

It is worth recalling that if a discrete approximation to the prior PDF is available, like the one in (5.27), the remaining expectation in (A.38) can be numerically computed as

$$\mathbf{J}_s^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \approx \frac{\theta_s^k}{\sigma_n^2} \left[\sum_{i=1}^N \omega_i^{k-1} \frac{1}{\rho_s(\mathbf{x}_i^k)} \left(\hat{\mathbf{u}}_s(\mathbf{x}_i^k) \hat{\mathbf{u}}_s(\mathbf{x}_i^k)^T \right) \right]. \quad (\text{A.48})$$

A.7 Sensor EFIM for additive received power measurement models

We consider the derivation of the sensor EFIM in (5.71) for a resource management problem. Thus measurements are taken following (5.12). An additive received power measurement model $h_s(\mathbf{X}^k)$ is considered, which is described in Section 3.5.2. Considering the multitarget state, $\mathbf{X}^k$ which can be partitioned as in (2.98), we can form the gradient $\nabla_{\mathbf{X}} h_s(\mathbf{X}^k)$ by computing the gradients $\nabla_{\mathbf{x}_j} h_{s,j}(\mathbf{x}_j^k)$ as in (5.74). Thus, given that in the measurement model under consideration $h_{s,j}(\mathbf{x}_j^k)$ is given by (3.27), the gradient in (A.38) can be calculated as

$$\nabla_{\mathbf{x}_j} h_{s,j}(\mathbf{x}_j^k) = \begin{cases} \mathbf{0} & \rho_s(\mathbf{x}_j^k) < \rho_0 \\ -K_0 (\rho_s(\mathbf{x}_j^k))^{-\alpha-1} \hat{\boldsymbol{\rho}}_s(\mathbf{x}_j^k) & \rho_s(\mathbf{x}_j^k) > \rho_0 \end{cases} \quad (\text{A.49})$$

where $K_0 = \alpha Z_0 \rho_0^\alpha$ and $\hat{\boldsymbol{\rho}}_s(\mathbf{x}_j^k)$ is the unitary vector pointing from the sensor to j -th target

$$\hat{\boldsymbol{\rho}}_s(\mathbf{x}_j) \equiv \frac{1}{\rho_s(\mathbf{x}_j)} [(x_j - s_{s,x}), (y_j - s_{s,y})]^T, \quad (\text{A.50})$$

so that each matrix block in (5.76) of the sensor EFIM matrix is computed as

$$\mathbf{G}_s^{j,l} = \mathbb{I}_{\rho_0}^{\rho_s} \{ \mathbf{x}_j^k \} \mathbb{I}_{\rho_0}^{\rho_s} \{ \mathbf{x}_l^k \} K_0^2 (\rho_s(\mathbf{x}_j^k))^{-\alpha-1} (\rho_s(\mathbf{x}_l^k))^{-\alpha-1} \hat{\boldsymbol{\rho}}_s(\mathbf{x}_j^k) [\hat{\boldsymbol{\rho}}_s(\mathbf{x}_l^k)]^T. \quad (\text{A.51})$$

As previously discussed, in order to compose the sensor EFIM in (5.77), the expectation with respect to the prior can be taken separately for each matrix block, so

that if a discrete approximation to the prior is available we can approximate

$$E^{-} \{ \mathbf{G}_s^{j,l} \} \approx K_0^2 \sum_{i=1}^N \omega_i^{k-1} \mathbb{I}_{\rho_0}^{\rho_s} \{ \mathbf{x}_{j,i}^k \} \mathbb{I}_{\rho_0}^{\rho_s} \{ \mathbf{x}_{l,i}^k \} (\rho_s(\mathbf{x}_{j,i}^k))^{-\alpha-1} (\rho_s(\mathbf{x}_{l,i}^k))^{-\alpha-1} \hat{\boldsymbol{\rho}}_s(\mathbf{x}_{j,i}^k) [\hat{\boldsymbol{\rho}}_s(\mathbf{x}_{l,i}^k)]^T. \quad (\text{A.52})$$

A.8 Analysis of the PC-CRLB minimization as a function of the available resource budget

In this appendix we analyze a simplified problem of resource management using the approach outlined in Section 5.2 and employed in Section 6.1 in order to gain some insight into the application of the PC-CRLB minimization. We consider that a single target with $\mathbf{x}^k = [x, y]^T$ is present in the scenario and assume that its predicted covariance matrix of the prior is diagonal with

$$\text{Cov}(\mathbf{x}) = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \quad (\text{A.53})$$

so that under the above simplification, the inverse covariance approximation to the PIM takes the form

$$\begin{aligned} \mathbf{B}_k^{-}(\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx (\text{Cov}(\mathbf{x}^k))^{-1} \\ &= \begin{bmatrix} (\sigma_x^2)^{-1} & 0 \\ 0 & (\sigma_y^2)^{-1} \end{bmatrix}. \end{aligned} \quad (\text{A.54})$$

In addition, we consider there are only 2 sensors in the scenario with positions

$$\mathbf{s}_1 = \mathbf{x}^k + \mathbf{r}_1 \quad (\text{A.55})$$

$$\mathbf{s}_2 = \mathbf{x}^k + \mathbf{r}_2 \quad (\text{A.56})$$

where

$$\mathbf{r}_1 = [r_1, 0]^T \quad (\text{A.57})$$

$$\mathbf{r}_2 = [0, r_2]^T, \quad (\text{A.58})$$

i.e., the relative positions of the sensors with respect to the target are in the directions of the unitary vectors of the state space.

A.8.1 Angular measurement model

Considering the angular measurement model in Section 5.1.1, and using (A.46), we approximate the sensor EFIMs using the FIM evaluated at the true value of $\mathbf{x}^k$ as

$$\begin{aligned} \mathbf{J}_{k,1}^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx \frac{\theta_1^k}{\sigma_n^2 r_1^2} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \frac{b\tilde{\theta}_1^k}{\sigma_n^2 r_1^2} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (\text{A.59})$$

$$\begin{aligned} \mathbf{J}_{k,2}^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx \frac{\theta_2^k}{\sigma_n^2 r_2^2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\ &= \frac{b\tilde{\theta}_2^k}{\sigma_n^2 r_2^2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \quad (\text{A.60})$$

where we have used the normalization

$$\theta_i^k = b\tilde{\theta}_i^k \quad (\text{A.61})$$

with respect to the available budget b , so that assuming that the whole budget is used by the controller (i.e., $\theta_1^k + \theta_2^k = b$), we can write $\tilde{\theta}_2^k = 1 - \tilde{\theta}_1^k$. The feasible set of the parameters in (5.14) in this case is

$$\tilde{\theta}_1^k \in \Theta \iff \left\{ 0 < \tilde{\theta}_1^k \leq 1 \right\}. \quad (\text{A.62})$$

Using (5.25), the PC-BIM can therefore be expressed as

$$\begin{aligned} \mathbf{B}_k^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx \mathbf{B}_k^-(\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) + \sum_{s=1}^2 \mathbf{J}_{k,s}^-(\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \quad (\text{A.63}) \\ &= \begin{bmatrix} (\sigma_x^2)^{-1} + \frac{\tilde{\theta}_2^k}{\sigma_b^2 r_2^2} & 0 \\ 0 & (\sigma_y^2)^{-1} + \frac{\tilde{\theta}_1^k}{\sigma_b^2 r_1^2} \end{bmatrix} \end{aligned}$$

where we have joined in an unique parameter σ_b the quality of sensor measurements depending on both σ_n and b , with

$$\sigma_b^2 = \frac{\sigma_n^2}{b}. \quad (\text{A.64})$$

Given its diagonal structure, the inverse of the PC-BIM can be expressed as

$$\mathbf{B}_k^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \approx \begin{bmatrix} \frac{1}{(\sigma_x^2)^{-1} + \frac{\tilde{\theta}_2^k}{\sigma_b^2 r_2^2}} & 0 \\ 0 & \frac{1}{(\sigma_y^2)^{-1} + \frac{\tilde{\theta}_1^k}{\sigma_b^2 r_1^2}} \end{bmatrix}. \quad (\text{A.65})$$

The PC-CRLB (see 5.24) of choosing θ_1, θ_2 is therefore

$$\begin{aligned} \text{tr} \left\{ \mathbf{B}_k^\uparrow(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\} &\approx \frac{1}{(\sigma_x^2)^{-1} + \frac{\tilde{\theta}_2^k}{\sigma_b^2 r_2^2}} + \frac{1}{(\sigma_y^2)^{-1} + \frac{\tilde{\theta}_1^k}{\sigma_b^2 r_1^2}} \quad (\text{A.66}) \\ &= \frac{\sigma_x^2 \sigma_b^2 r_2^2}{\sigma_b^2 r_2^2 + \sigma_x^2 \tilde{\theta}_2^k} + \frac{\sigma_y^2 \sigma_b^2 r_1^2}{\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k} \\ &= \frac{\sigma_x^2 \sigma_b^2 r_2^2}{\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k)} + \frac{\sigma_y^2 \sigma_b^2 r_1^2}{\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k}, \end{aligned}$$

which is not defined in the points $\tilde{\theta}_1^{k,A_1}$ and $\tilde{\theta}_1^{k,A_2}$

$$\begin{aligned} \sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^{k,A_1}) &= 0 \quad (\text{A.67}) \\ \tilde{\theta}_1^{k,A_1} &= \frac{\sigma_b^2 r_2^2}{\sigma_x^2} + 1 \end{aligned}$$

and

$$\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^{k,A_2} = 0 \quad (\text{A.68})$$

$$\tilde{\theta}_1^{k,A_2} = \frac{-\sigma_b^2 r_1^2}{\sigma_y^2}$$

where considering that the involved parameters take only non-negative values

$$\tilde{\theta}_1^{k,A_2} \leq 0 < 1 \leq \tilde{\theta}_1^{k,A_1} \quad (\text{A.69})$$

for any possible parameter combination, so that this will not suppose any practical problem as $\tilde{\theta}_1^{k,A_1}$ and $\tilde{\theta}_1^{k,A_2}$ are outside or in the frontier of the feasible set in (A.62).

In order to find the best possible sensor configuration, it is of interest to consider

$$\begin{aligned} \frac{\partial \text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\}}{\partial \tilde{\theta}_1^k} &= \frac{\partial}{\partial \tilde{\theta}_1^k} \left[\frac{\sigma_x^2 \sigma_b^2 r_2^2}{\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k)} + \frac{\sigma_y^2 \sigma_b^2 r_1^2}{\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k} \right] \quad (\text{A.70}) \\ &= \frac{0 - \sigma_x^2 \sigma_b^2 r_2^2 (-\sigma_x^2)}{\left[\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k) \right]^2} + \frac{0 - \sigma_y^2 \sigma_b^2 r_1^2 (\sigma_y^2)}{\left[\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k \right]^2} \\ &= \frac{\sigma_x^4 \sigma_b^2 r_2^2}{\left[\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k) \right]^2} - \frac{\sigma_y^4 \sigma_b^2 r_1^2}{\left[\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k \right]^2} \end{aligned}$$

and

$$\begin{aligned} \frac{\partial^2 \text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\}}{\partial \tilde{\theta}_1^k \partial \tilde{\theta}_1^k} &= \frac{\partial}{\partial \tilde{\theta}_1^k} \left[\frac{\partial \text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\}}{\partial \tilde{\theta}_1^k} \right] \quad (\text{A.71}) \\ &= \frac{\partial}{\partial \tilde{\theta}_1^k} \left[\frac{\sigma_x^4 \sigma_b^2 r_2^2}{\left[\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k) \right]^2} - \frac{\sigma_y^4 \sigma_b^2 r_1^2}{\left[\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k \right]^2} \right] \\ &= \frac{0 - 2 \left[\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k) \right] (-\sigma_x^2) \sigma_x^4 \sigma_b^2 r_2^2}{\left[\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k) \right]^4} \\ &\quad - \frac{0 - 2 \left[\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k \right] (\sigma_y^2) \sigma_y^4 \sigma_b^2 r_1^2}{\left[\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k \right]^4} \\ &= \frac{2\sigma_x^6 \sigma_b^2 r_2^2}{\left[\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k) \right]^3} + \frac{2\sigma_y^6 \sigma_b^2 r_1^2}{\left[\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k \right]^3}. \end{aligned}$$

Note that given that all the involved parameters are non-negative (including $(1 - \tilde{\theta}_1^k)$), the second partial derivative for all possible values of $\tilde{\theta}_1^k$ in the admissible set and more specifically in $\tilde{\theta}_1^{k,A_2} \leq \tilde{\theta}_1^k \leq \tilde{\theta}_1^{k,A_1}$ will always be non-negative, so that

the cost function in (A.66) will be convex in the region of interest, which is of great interest as convergence in the search of the sensor configuration with the minimum cost is guaranteed as long as a proper initialization of the procedure is made.

The sensor configuration with the lowest predicted cost will therefore either lie on the frontier of the restrictions (in this case $\tilde{\theta}_1^k = 0$, or $\tilde{\theta}_1^k = 1$) or have a null derivative

$$\frac{\partial \text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k |_{\mathbf{z}^{1:k-1}}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\}}{\partial \tilde{\theta}_1^k} = 0 \quad (\text{A.72})$$

and thus

$$\begin{aligned} \left(\sigma_x^2 \sigma_b r_2 \left[\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k \right] \right)^2 &= \left(\sigma_y^2 \sigma_b r_1 \left[\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k) \right] \right)^2 \\ \sigma_x^2 \sigma_b r_2 \left[\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^k \right] &= \pm \sigma_y^2 \sigma_b r_1 \left[\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^k) \right], \end{aligned} \quad (\text{A.73})$$

from which we can discard the negative option of the equation, as all of the involved parameters are non-negative by construction (including $\tilde{\theta}_2^k = (1 - \tilde{\theta}_1^k)$), and the solution arising from the negative option of (A.73) would need to violate at least one of these non-negativeness conditions. Thus, considering the positive option of the equation yields

$$\begin{aligned} \frac{\sigma_x^2 r_2}{\sigma_y^2 r_1} \left[\sigma_b^2 r_1^2 + \sigma_y^2 \tilde{\theta}_1^{k,sol} \right] &= \left[\sigma_b^2 r_2^2 + \sigma_x^2 (1 - \tilde{\theta}_1^{k,sol}) \right] \\ \tilde{\theta}_1^{k,sol} \left(\frac{\sigma_x^2 r_2}{\sigma_y^2 r_1} \sigma_y^2 + \sigma_x^2 \right) &= \sigma_b^2 r_2^2 + \sigma_x^2 - \sigma_b^2 \frac{\sigma_x^2}{\sigma_y^2} r_2 r_1 \\ \tilde{\theta}_1^{k,sol} \sigma_x^2 \left(\frac{r_2}{r_1} + 1 \right) &= \sigma_b^2 r_2^2 + \sigma_x^2 - \sigma_b^2 \frac{\sigma_x^2}{\sigma_y^2} r_2 r_1 \\ \tilde{\theta}_1^{k,sol} &= \frac{1}{\left(\frac{r_2}{r_1} + 1 \right)} + \frac{\sigma_b^2 r_2 \left(r_2 - \frac{\sigma_x^2}{\sigma_y^2} r_1 \right)}{\sigma_x^2 \left(\frac{r_2}{r_1} + 1 \right)} \\ \tilde{\theta}_2^{k,sol} &= 1 - \tilde{\theta}_1^{k,sol}. \end{aligned} \quad (\text{A.74})$$

It should be noted that the solution in (A.74) guarantees that $\tilde{\theta}_1^{k,sol} + \tilde{\theta}_2^{k,sol} = 1$. However, it does not guarantee that $\tilde{\theta}_1^{k,sol} > 0$, $\tilde{\theta}_2^{k,sol} > 0$, $\tilde{\theta}_1^{k,sol} \leq 1$, $\tilde{\theta}_2^{k,sol} \leq 1$. Nevertheless, as explained in Section A.8.1.1, it is important to note that we can bound the solution in (A.74) using $\tilde{\theta}_1^{k,A_1}$ in (A.67) and $\tilde{\theta}_1^{k,A_2}$ in (A.68) as

$$\tilde{\theta}_1^{k,A_2} \leq \tilde{\theta}_1^{k,sol} \leq \tilde{\theta}_1^{k,A_1}. \quad (\text{A.75})$$

Thus, considering (A.69) and noting that (A.66) is convex in the domain defined by

this bounds (see (A.71)), if it is the case that $\tilde{\theta}_1^{k,sol} \notin \Theta$, the solution within the feasible set with the minimum cost will lie on the frontier of the restrictions (in this case $\tilde{\theta}_1^k = 0$, or $\tilde{\theta}_1^k = 1$).

In the particular case where $\sigma_b = 0$ (i.e., when the sensor noise variance is $\sigma_n^2 = 0$, or the available budget is $b = \infty$), using (A.74), the sensor configuration with the minimum cost is

$$\tilde{\theta}_1^k = \frac{r_1}{r_2 + r_1} \quad (\text{A.76})$$

$$\tilde{\theta}_2^k = 1 - \tilde{\theta}_1^k, \quad (\text{A.77})$$

which is not a relevant case since in all points the cost will be equal to 0 when this happens. Nevertheless this illustrates the case when $\sigma_b \rightarrow 0$, which is of interest as it corresponds to situations with very good sensor measurements or a very high available resource budget. Note that in these cases, the resource distribution does not depend on the covariance matrix elements and is only determined by the distance to the sensors. Note that this solution always satisfies the restriction conditions.

In the particular case where $\sigma_b \rightarrow \infty$, following (A.74), the optimal sensor configuration can be found in

$$\tilde{\theta}_1^k = \begin{cases} \infty & , r_2 > \frac{\sigma_x^2 r_1}{\sigma_y^2} \\ -\infty & , r_2 < \frac{\sigma_x^2 r_1}{\sigma_y^2} \end{cases} \quad (\text{A.78})$$

$$\tilde{\theta}_2^k = 1 - \tilde{\theta}_1^k, \quad (\text{A.79})$$

however, this solution does not satisfy the constraints over the parameters, and the selected optimal configuration will lie in the frontier of the restrictions (in this case $\tilde{\theta}_1^k = 0$, or $\tilde{\theta}_1^k = 1$).

A.8.1.1 Derivation of solution bounds

In this section, we show that the solution $\tilde{\theta}_1^{k,sol}$ obtained in (A.74) is bounded by $\tilde{\theta}_1^{k,A_1}$ and $\tilde{\theta}_1^{k,A_2}$ in (A.67) and (A.68). Note that considering (A.74) and (A.67), we can write

$$\tilde{\theta}_1^{k,sol} = \frac{1}{\left(\frac{r_2}{r_1} + 1\right)} + \frac{\sigma_b^2 r_2 \left(r_2 - \frac{\sigma_x^2}{\sigma_y^2} r_1\right)}{\sigma_x^2 \left(\frac{r_2}{r_1} + 1\right)} \leq \frac{\sigma_b^2 r_2^2}{\sigma_x^2} + 1 = \tilde{\theta}_1^{k,A_1} \quad (\text{A.80})$$

$$\begin{aligned}
 \frac{\sigma_x^2}{\left(\frac{r_2}{r_1} + 1\right)} + \frac{\sigma_b^2 r_2 \left(r_2 - \frac{\sigma_x^2}{\sigma_y^2} r_1\right)}{\left(\frac{r_2}{r_1} + 1\right)} &\leq \sigma_b^2 r_2^2 + \sigma_x^2 \\
 \sigma_x^2 + \sigma_b^2 r_2^2 - \sigma_b^2 r_2 \frac{\sigma_x^2}{\sigma_y^2} r_1 &\leq \sigma_b^2 r_2^2 \left(\frac{r_2}{r_1} + 1\right) + \sigma_x^2 \left(\frac{r_2}{r_1} + 1\right) \\
 -\sigma_b^2 r_2 \frac{\sigma_x^2}{\sigma_y^2} r_1 &\leq \sigma_b^2 r_2^2 \left(\frac{r_2}{r_1}\right) + \sigma_x^2 \left(\frac{r_2}{r_1}\right) \\
 0 &\leq \sigma_b^2 r_2^2 \left(\frac{r_2}{r_1}\right) + \sigma_x^2 \left(\frac{r_2}{r_1}\right) + \sigma_b^2 r_2 \frac{\sigma_x^2}{\sigma_y^2} r_1,
 \end{aligned}$$

which is true for any possible value of the involved parameters. In addition considering (A.74) and (A.68), we can write

$$\begin{aligned}
 \tilde{\theta}_1^{k,sol} = \frac{1}{\left(\frac{r_2}{r_1} + 1\right)} + \frac{\sigma_b^2 r_2 \left(r_2 - \frac{\sigma_x^2}{\sigma_y^2} r_1\right)}{\sigma_x^2 \left(\frac{r_2}{r_1} + 1\right)} &\geq \frac{-\sigma_b^2 r_1^2}{\sigma_y^2} = \tilde{\theta}_1^{k,A_2} \quad (\text{A.81}) \\
 \sigma_x^2 + \sigma_b^2 r_2 \left(r_2 - \frac{\sigma_x^2}{\sigma_y^2} r_1\right) &\geq \frac{-\sigma_b^2 r_1^2}{\sigma_y^2} \sigma_x^2 \left(\frac{r_2}{r_1} + 1\right) \\
 \sigma_x^2 + \sigma_b^2 r_2^2 &\geq \frac{-\sigma_b^2 r_1^2}{\sigma_y^2} \sigma_x^2 \\
 \sigma_x^2 + \sigma_b^2 r_2^2 + \frac{\sigma_b^2 r_1^2}{\sigma_y^2} \sigma_x^2 &\geq 0,
 \end{aligned}$$

which is also true for any possible value of the involved parameters, so that using (A.80) and (A.81), yields

$$\tilde{\theta}_1^{k,A_2} \leq \tilde{\theta}_1^{k,sol} \leq \tilde{\theta}_1^{k,A_1}. \quad (\text{A.82})$$

A.8.2 Received power measurement model

Considering the received power measurement model in 5.1.1 with $\alpha = 2$, using (A.39), we approximate the sensor EFIMs using the FIM evaluated at the true value of $\mathbf{x}^k$ as

$$\begin{aligned}
 \mathbf{J}_{k,1}^- (\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx \theta_1 \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}{\sigma_n^2 r_1^6} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\
 &= \tilde{\theta}_1^k \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}{\sigma_b^2 r_1^6} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{A.83})
 \end{aligned}$$

$$\begin{aligned} \mathbf{J}_{k,2}^{-}(\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx \theta_2 \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_n^2 r_2^6} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \tilde{\theta}_2^k \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_b^2 r_2^6} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (\text{A.84})$$

where

$$K_0^2 = 4Z_0^2 \rho_0^4. \quad (\text{A.85})$$

Using (5.25), the PC-BIM can therefore be expressed as

$$\begin{aligned} \mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) &\approx \mathbf{B}_k^{-}(\mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) + \sum_{s=1}^2 \mathbf{J}_{k,s}^{-}(\theta_s^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1}) \quad (\text{A.86}) \\ &= \begin{bmatrix} (\sigma_x^2)^{-1} + \tilde{\theta}_1^k \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}{\sigma_b^2 r_1^6} & 0 \\ 0 & (\sigma_y^2)^{-1} + \tilde{\theta}_2^k \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_b^2 r_2^6} \end{bmatrix}. \end{aligned}$$

Given its diagonal structure, the inverse of the PC-BIM can be expressed as

$$\mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \approx \begin{bmatrix} \frac{1}{(\sigma_x^2)^{-1} + \tilde{\theta}_1^k \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}{\sigma_b^2 r_1^6}} & 0 \\ 0 & \frac{1}{(\sigma_y^2)^{-1} + \tilde{\theta}_2^k \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_b^2 r_2^6}} \end{bmatrix}. \quad (\text{A.87})$$

The PC-CRLB (see 5.24) of choosing θ_1, θ_2 is therefore

$$\begin{aligned} \text{tr} \left\{ \mathbf{B}_k^{\uparrow}(\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\} &\approx \frac{1}{(\sigma_x^2)^{-1} + \tilde{\theta}_1^k \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}{\sigma_b^2 r_1^6}} + \frac{1}{(\sigma_y^2)^{-1} + \tilde{\theta}_2^k \frac{K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_b^2 r_2^6}} \quad (\text{A.88}) \\ &= \frac{\sigma_x^2 \sigma_b^2 r_1^6}{\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} + \frac{\sigma_y^2 \sigma_b^2 r_2^6}{\sigma_b^2 r_2^6 + \sigma_y^2 \tilde{\theta}_2^k K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}} \\ &= \frac{\sigma_x^2 \sigma_b^2 r_1^6}{\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} + \frac{\sigma_y^2 \sigma_b^2 r_2^6}{\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k\right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}, \end{aligned}$$

which is not defined in the points $\tilde{\theta}_1^{k,A_1}$ and $\tilde{\theta}_1^{k,A_2}$

$$\begin{aligned} \sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^{k,A_1} K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} &= 0 \quad (\text{A.89}) \\ \tilde{\theta}_1^{k,A_1} &= \frac{-\sigma_b^2 r_1^6}{\sigma_x^2 K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} \end{aligned}$$

and

$$\begin{aligned} \sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^{k,A_2}\right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} &= 0 \\ \tilde{\theta}_1^{k,A_2} &= \frac{\sigma_b^2 r_2^6}{\sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}} + 1 \end{aligned} \quad (\text{A.90})$$

where considering that the involved parameters take only non-negative values

$$\tilde{\theta}_1^{k,A_1} \leq 0 < 1 \leq \tilde{\theta}_1^{k,A_2} \quad (\text{A.91})$$

for any possible parameter combination, so that this will not suppose any practical problem as $\tilde{\theta}_1^{k,A_1}$ and $\tilde{\theta}_1^{k,A_2}$ are outside or in the frontier of the feasible set in (A.62).

In order to find the best possible sensor configuration, it is of interest to consider

$$\begin{aligned} & \frac{\partial \text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\}}{\partial \tilde{\theta}_1^k} \\ &= \frac{\partial}{\partial \tilde{\theta}_1^k} \left[\frac{\sigma_x^2 \sigma_b^2 r_1^6}{\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} + \frac{\sigma_y^2 \sigma_b^2 r_2^6}{\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k\right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}} \right] \\ &= \frac{0 - \sigma_x^2 \sigma_b^2 r_1^6 (\sigma_x^2 K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\})}{\left[\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} \right]^2} + \frac{0 - \sigma_y^2 \sigma_b^2 r_2^6 (-\sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\})}{\left[\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k\right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \right]^2} \\ &= - \frac{\sigma_x^4 \sigma_b^2 r_1^6 (K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\})}{\left[\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} \right]^2} + \frac{\sigma_y^4 \sigma_b^2 r_2^6 (K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\})}{\left[\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k\right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \right]^2} \quad (\text{A.92}) \end{aligned}$$

and

$$\begin{aligned} & \frac{\partial^2 \text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\}}{\partial \tilde{\theta}_1^k \partial \tilde{\theta}_1^k} \\ &= \frac{\partial}{\partial \tilde{\theta}_1^k} \left[\frac{\partial \text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\}}{\partial \tilde{\theta}_1^k} \right] \\ &= \frac{\partial}{\partial \tilde{\theta}_1^k} \left[- \frac{\sigma_x^4 \sigma_b^2 r_1^6 (K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\})}{\left[\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} \right]^2} + \frac{\sigma_y^4 \sigma_b^2 r_2^6 (K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\})}{\left[\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k\right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \right]^2} \right] \\ &= - \frac{0 - 2 \left(\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} \right) (\sigma_x^2 K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}) \sigma_x^4 \sigma_b^2 r_1^6 (K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\})}{\left[\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} \right]^4} \end{aligned}$$

$$\begin{aligned}
& 0 - 2 \left(\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k \right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \} \right) \left(-\sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \} \right) \sigma_y^4 \sigma_b^2 r_2^6 \left(K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \} \right) \\
& + \frac{\left[\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k \right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \} \right]^4}{2 K_0^4 \sigma_x^6 \sigma_b^2 r_1^6 \mathbb{I}_{\rho_0}^{r_1} \{ \mathbf{x}^k \}} + \frac{2 K_0^4 \sigma_y^6 \sigma_b^2 r_2^6 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \}}{\left[\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{ \mathbf{x}^k \} \right]^3} + \frac{2 K_0^4 \sigma_y^6 \sigma_b^2 r_2^6 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \}}{\left[\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k \right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \} \right]^3} \quad (\text{A.93})
\end{aligned}$$

where we have used that, given the definition of the indicator function in (A.42),

$$\mathbb{I}_{\rho_0}^r \{ \mathbf{x}^k \} \mathbb{I}_{\rho_0}^r \{ \mathbf{x}^k \} = \mathbb{I}_{\rho_0}^r \{ \mathbf{x}^k \}. \quad (\text{A.94})$$

Note that given that all the involved parameters are non-negative (including $(1 - \tilde{\theta}_1^k)$), the second partial derivative for all possible values of $\tilde{\theta}_1^k$ in the admissible set and more specifically in $\tilde{\theta}_1^{k,A_1} \leq \tilde{\theta}_1^k \leq \tilde{\theta}_1^{k,A_2}$ will always be non-negative, so that the cost function in (A.88) will be convex in the region of interest, which is of great interest as convergence in the search of the sensor configuration with the minimum cost is guaranteed as long as a proper initialization of the procedure is performed.

The sensor configuration with the lowest predicted cost will either lie on the frontier of the restrictions (in this case $\tilde{\theta}_1^k = 0$, or $\tilde{\theta}_1^k = 1$) or have a null derivative

$$\frac{\partial \text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\}}{\partial \tilde{\theta}_1^k} = 0, \quad (\text{A.95})$$

from where we have that

$$\begin{aligned}
& \left(\sigma_y^2 \sigma_b r_2^3 K_0 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \} \left[\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{ \mathbf{x}^k \} \right] \right)^2 \\
& = \left(\sigma_x^2 \sigma_b r_1^3 K_0 \mathbb{I}_{\rho_0}^{r_1} \{ \mathbf{x}^k \} \left[\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k \right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \} \right] \right)^2 \quad (\text{A.96})
\end{aligned}$$

and thus

$$\begin{aligned}
& \sigma_y^2 r_2^3 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \} \left[\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^k K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{ \mathbf{x}^k \} \right] \\
& = \pm \sigma_x^2 r_1^3 \mathbb{I}_{\rho_0}^{r_1} \{ \mathbf{x}^k \} \left[\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^k \right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{ \mathbf{x}^k \} \right] \quad (\text{A.97})
\end{aligned}$$

from which we can discard the negative option of the equation, as all of the involved parameters are non-negative by construction (including $\tilde{\theta}_2^k = (1 - \tilde{\theta}_1^k)$), and the solution arising from the negative option of (A.96) would need to violate at least one of these non-negativeness conditions. Thus, considering the positive option of the

equation yields

$$\begin{aligned}
 & \frac{\sigma_y^2 r_2^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 r_1^3 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} \left[\sigma_b^2 r_1^6 + \sigma_x^2 \tilde{\theta}_1^{k,sol} K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} \right] = \left[\sigma_b^2 r_2^6 + \sigma_y^2 \left(1 - \tilde{\theta}_1^{k,sol} \right) K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \right] \\
 & \tilde{\theta}_1^{k,sol} \left(\frac{\sigma_y^2 r_2^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \sigma_x^2 K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}{\sigma_x^2 r_1^3 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} + \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \right) \\
 & \quad = \sigma_b^2 r_2^6 + \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} - \frac{\sigma_y^2 r_2^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 r_1^3 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} \left[\sigma_b^2 r_1^6 \right] \\
 & \tilde{\theta}_1^{k,sol} \left(\frac{\sigma_y^2 r_2^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} K_0^2}{r_1^3} + \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \right) \\
 & \quad = \sigma_b^2 r_2^6 + \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} - \frac{\sigma_b^2 \sigma_y^2 r_2^3 r_1^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} \\
 & \tilde{\theta}_1^{k,sol} = \frac{1}{\left(\frac{r_2^3}{r_1^3} + 1 \right)} + \frac{\sigma_b^2 r_2^3 \left(r_2^3 - \frac{\sigma_y^2 r_1^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} \right)}{\sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \left(\frac{r_2^3}{r_1^3} + 1 \right)}
 \end{aligned} \tag{A.98}$$

and

$$\tilde{\theta}_2^{k,sol} = 1 - \tilde{\theta}_1^{k,sol}. \tag{A.99}$$

It should be noted that the solution in (A.98) guarantees that $\tilde{\theta}_1^k + \tilde{\theta}_2^k = 1$. However, it does not guarantee that $\tilde{\theta}_1^{k,sol} > 0$, $\tilde{\theta}_2^{k,sol} > 0$, $\tilde{\theta}_1^{k,sol} \leq 1$, $\tilde{\theta}_2^{k,sol} \leq 1$. Nevertheless, as explained in Section (A.8.2.1), we can bound the solution in (A.98) using $\tilde{\theta}_1^{k,A_1}$ in (A.89) and $\tilde{\theta}_1^{k,A_2}$ in (A.90) as

$$\tilde{\theta}_1^{k,A_1} \leq \tilde{\theta}_1^{k,sol} \leq \tilde{\theta}_1^{k,A_2}, \tag{A.100}$$

thus, considering (A.91), and noting that (A.88) is convex in the domain defined by this bounds (see (A.93)), if it is the case that $\tilde{\theta}_1^{k,sol} \notin \Theta$, the solution within the feasible set with the minimum cost will lie on the frontier of the restrictions (in this case $\tilde{\theta}_1^k = 0$, or $\tilde{\theta}_1^k = 1$).

In the case of the received power measurement model it is also relevant to analyze the situations where the target is within the saturation distance of some sensor. If the target lies within the saturation distance of the sensor $\mathbf{s}_2$, the indicator function takes the value $\mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} = 0$, so that using (A.98) yields $\tilde{\theta}_1^k = \infty$, which indicates that all available resources should be given to sensor $\mathbf{s}_1$, and making the solution to lie in the frontier of the restrictions. Alternatively, if the target lies within the saturation distance of the sensor $\mathbf{s}_1$, the indicator function takes the value $\mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} = 0$, so that

using (A.98) yields $\tilde{\theta}_1^k = -\infty$, which indicates that no resources should be given to sensor $\mathbf{s}_1$, and making the solution to lie in the frontier of the restrictions. If the target lies within the saturation distance of both sensors $\mathbf{s}_1$ and $\mathbf{s}_2$, the indicator functions takes the value $\mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} = \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} = 0$, and (A.88) does not depend on $\tilde{\theta}_1^k$ and $\tilde{\theta}_2^k$ and always takes the value

$$\text{tr} \left\{ \mathbf{B}_k^\uparrow (\boldsymbol{\theta}^k | \mathbf{z}^{1:k-1}; \boldsymbol{\theta}^{1:k-1})^{-1} \right\} \approx \sigma_x^2 + \sigma_y^2. \quad (\text{A.101})$$

In the following, we assume that the target lies outside of the saturation distance of both sensors and thus, $\mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\} = \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} = 1$. In the particular case where $\sigma_b = 0$ (i.e., when the sensor noise variance is $\sigma_n^2 = 0$, or the available budget is $b = \infty$), using (A.98), the sensor configuration with the minimum cost is

$$\tilde{\theta}_1^k = \frac{r_1^3}{r_2^3 + r_1^3} \quad (\text{A.102})$$

$$\tilde{\theta}_2^k = 1 - \tilde{\theta}_1^k, \quad (\text{A.103})$$

which is not a relevant case since in all points the cost will be equal to 0 when this happens. Nevertheless this illustrates the case when $\sigma_b \rightarrow 0$, which is of interest as it corresponds to situations with very good sensor measurements or a very high available resource budget. Note that in these cases, the resource distribution does not depend on the covariance matrix elements and is only determined by the distance to the sensors. Note that this solution always satisfies the restriction conditions.

In the particular case where $\sigma_b \rightarrow \infty$, following (A.98), the optimal sensor configuration can be found in

$$\tilde{\theta}_1^k = \begin{cases} \infty & , r_2^3 > \frac{\sigma_y^2 r_1^3}{\sigma_x^2} \\ -\infty & , r_2^3 < \frac{\sigma_y^2 r_1^3}{\sigma_x^2} \end{cases} \quad (\text{A.104})$$

$$\tilde{\theta}_2^k = 1 - \tilde{\theta}_1^k, \quad (\text{A.105})$$

however this solution does not satisfy the constraints over the parameters, and the selected optimal configuration will lie in the frontier of the restrictions (in this case $\tilde{\theta}_1^k = 0$, or $\tilde{\theta}_1^k = 1$) .

A.8.2.1 Derivation of solution bounds

In this section, we show that the solution $\tilde{\theta}_1^{k, \text{sol}}$ obtained in (A.98) is bounded by $\tilde{\theta}_1^{k, A_1}$ and $\tilde{\theta}_1^{k, A_2}$ in (A.89) and (A.90). Note that considering (A.98) and (A.89), we can write

$$\begin{aligned}
 \tilde{\theta}_1^{k,sol} &= \frac{1}{\left(\frac{r_2^3}{r_1^3} + 1\right)} + \frac{\sigma_b^2 r_2^3 \left(r_2^3 - \frac{\sigma_y^2 r_1^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}\right)}{\sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \left(\frac{r_2^3}{r_1^3} + 1\right)} \geq \frac{-\sigma_b^2 r_1^6}{\sigma_x^2 K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} = \tilde{\theta}_1^{k,A_1} \quad (\text{A.106}) \\
 \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} + \sigma_b^2 r_2^3 \left(r_2^3 - \frac{\sigma_y^2 r_1^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}\right) &\geq \frac{-\sigma_b^2 r_1^6 \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \left(\frac{r_2^3}{r_1^3} + 1\right)}{\sigma_x^2 K_0^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} \\
 \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} + \sigma_b^2 r_2^3 (r_2^3) &\geq \frac{-\sigma_b^2 r_1^6 \sigma_y^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} \\
 \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} + \sigma_b^2 r_2^6 + \frac{\sigma_b^2 r_1^6 \sigma_y^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} &\geq 0,
 \end{aligned}$$

which is true for any possible value of the involved parameters. In addition considering (A.98) and (A.90), we can write

$$\begin{aligned}
 \tilde{\theta}_1^{k,sol} &= \frac{1}{\left(\frac{r_2^3}{r_1^3} + 1\right)} + \frac{\sigma_b^2 r_2^3 \left(r_2^3 - \frac{\sigma_y^2 r_1^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}\right)}{\sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \left(\frac{r_2^3}{r_1^3} + 1\right)} \\
 &\leq \frac{\sigma_b^2 r_2^6}{\sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}} + 1 = \tilde{\theta}_1^{k,A_2} \\
 \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} + \sigma_b^2 r_2^3 \left(r_2^3 - \frac{\sigma_y^2 r_1^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}\right) &\leq \left(\frac{\sigma_b^2 r_2^6}{\sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}} + 1\right) \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \left(\frac{r_2^3}{r_1^3} + 1\right) \\
 - \sigma_b^2 r_2^3 \frac{\sigma_y^2 r_1^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}} &\leq \sigma_b^2 r_2^6 \left(\frac{r_2^3}{r_1^3}\right) + \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \frac{r_2^3}{r_1^3} \\
 0 &\leq \sigma_b^2 r_2^6 \left(\frac{r_2^3}{r_1^3}\right) + \sigma_y^2 K_0^2 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\} \frac{r_2^3}{r_1^3} + \sigma_b^2 r_2^3 \frac{\sigma_y^2 r_1^3 \mathbb{I}_{\rho_0}^{r_2} \{\mathbf{x}^k\}}{\sigma_x^2 \mathbb{I}_{\rho_0}^{r_1} \{\mathbf{x}^k\}}, \quad (\text{A.107})
 \end{aligned}$$

which is also true for any possible value of the involved parameters, so that using (A.106) and (A.107), yields

$$\tilde{\theta}_1^{k,A_1} \leq \tilde{\theta}_1^{k,sol} \leq \tilde{\theta}_1^{k,A_2}. \quad (\text{A.108})$$

A.9 Metrics for multiple target tracking

The availability of metrics for MTT is a key factor for adequate performance evaluation of algorithms and the design of estimators in this field. In this section, two metrics for MTT are described: the OSPA and the generalized OSPA metrics. Both of them are

metrics in the space of sets of targets. Nevertheless, this kind of metrics can still be employed when the multitarget state is expressed using vector notation, as in this thesis.

A.9.1 OSPA metric

Let $d(\cdot, \cdot)$ denote a metric in $\mathbb{R}^N$, such that $d(\mathbf{x}, \mathbf{y})$ is the distance between $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ and let $d^{(c)}(\mathbf{x}, \mathbf{y}) = \min(d(\mathbf{x}, \mathbf{y}), c)$ stand for the cutoff metric associated with $d(\mathbf{x}, \mathbf{y})$. Let also $\prod_n$ be the set of all permutations in $\{1, \dots, n\}$ for any $n \in \mathbb{N}$. An element $\pi \in \prod_n$ is a sequence $(\pi(1), \dots, \pi(n))$.

Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_{|X|}\}$ and $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_{|Y|}\}$ be finite subsets of a bounded observation window $W \subset \mathbb{R}^N$, where $|A|$ stands for the cardinality of set A . For $1 \leq p \leq \infty$ and $|X| \leq |Y|$, the optimal sub-pattern assignment (OSPA) metric [145, 146] can be expressed as

$$d_p^{(c)}(X, Y) \triangleq \left(\frac{1}{|Y|} \left(\min_{\pi \in \prod_{|Y|}} \sum_{i=1}^{|X|} d^{(c)}(\mathbf{x}_i, \mathbf{y}_{\pi(i)})^p + c^p (|Y| - |X|) \right) \right)^{\frac{1}{p}}. \quad (\text{A.109})$$

If $|X| > |Y|$, $d_p^{(c)}(X, Y) = d_p^{(c)}(Y, X)$.

Note that OSPA accounts for a sum of errors for all the elements in the smaller set, with a cutoff, and cardinality error cost. It is worth noting that a large value of p severely penalizes outliers and cardinality mismatches, while the cutoff parameter c determines both the maximum contribution of localization errors as well as cardinality errors.

A.9.2 GOSPA metric

Recent contributions in the literature [134] have pointed out some drawbacks in the use of the OSPA metric applied to multiple target estimation. While the OSPA metric is the most widely used metric in MTT, it does not penalize localization errors for properly detected targets, and costs for missed targets and false alarms, as in performance evaluation methods traditionally used in MTT [22, Sec. 13.6], [45, 47, 109, 138]. The recently proposed generalized OSPA metric (GOSPA) [134] addresses this drawback of OSPA. We proceed to describe the metric and to explain how this metric can penalize the three important types of errors in MTT.

Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_{|X|}\}$ and $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_{|Y|}\}$ be finite subsets of $\mathbb{R}^N$. For $0 < \alpha \leq 2$,

$1 \leq p \leq \infty$ and $|X| \leq |Y|$, the GOSPA metric is defined as

$$d_p^{(c,\alpha)}(X, Y) \triangleq \left(\min_{\pi \in \Pi_{|Y|}} \sum_{i=1}^{|X|} d^{(c)}(\mathbf{x}_i, \mathbf{y}_{\pi(i)})^p + \frac{c^p}{\alpha} (|Y| - |X|) \right)^{\frac{1}{p}}. \quad (\text{A.110})$$

If $|X| > |Y|$, $d_p^{(c,\alpha)}(X, Y) = d_p^{(c,\alpha)}(Y, X)$.

The role of parameters p and c is similar to that in OSPA. Although a cardinality mismatch is penalized with $\frac{c^p}{\alpha}$ in GOSPA instead of c^p in OSPA. Thus, the tuning of α allows for the determination of the cardinality error in relation to the maximum localization error. Nevertheless, the value of the GOSPA parameter $\alpha = 2$ is the one recommended by the authors in [134], as it is the only value that enables the decomposition of the error into a localization error for properly detected targets, and costs for missed targets and false targets. In the following, this concept is further explained.

For the recommended value of $\alpha = 2$, the GOSPA metric can be reformulated using assignment functions instead of permutations. Assignment functions (also usually referred to as matchings) have been extensively used in the tracking field for data association problems [22, Sec 6.5]. Using a set representation, an assignment set γ between the pair of finite index sets $\{1, \dots, |X|\}$ and $\{1, \dots, |Y|\}$ is a set with the following properties

1. $\gamma \subseteq \{1, \dots, |X|\} \times \{1, \dots, |Y|\}$
2. $(i, j), (i, j') \in \gamma \implies j = j'$
3. $(i, j), (i', j) \in \gamma \implies i = i'$

where the second and third properties ensure that every i and j receives at most one assignment. A pair of targets $(\mathbf{x}_i, \mathbf{y}_j)$ is referred to as 'localized' if $d(\mathbf{x}_i, \mathbf{y}_j) < c$. Thus, considering the assignment of localized targets, the GOSPA metric for $\alpha = 2$ for $1 \leq p \leq \infty$ can be rewritten as

$$d_p^{(c,2)}(X, Y) \triangleq \left(\min_{\gamma \in \Gamma(|X|, |Y|)} \left(\sum_{(i,j) \in \gamma} d(\mathbf{x}_i, \mathbf{y}_j)^p + \frac{c^p}{2} (|X| + |Y| - 2|\gamma|) \right) \right)^{\frac{1}{p}} \quad (\text{A.111})$$

where $\Gamma(|X|, |Y|)$ is the set of all possible assignment sets. The first term $\sum_{(i,j) \in \gamma} d(\mathbf{x}_i, \mathbf{y}_j)^p$ is the localization error (note that there is no need for the notion of cutoff metric in this setting) and the second term $\frac{c^p}{2} (|X| + |Y| - 2|\gamma|)$ is the error

due to $|X| - |\gamma|$ missed targets and $|Y| - |\gamma|$ false targets, considering X as the ground truth.

Bibliography

- [1] D. L. Alspach and H. W. Sorenson, “Nonlinear Bayesian Estimation Using Gaussian Sum Approximations,” *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972.
- [2] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Dover Publications, 1979.
- [3] C. Andrieu and A. Doucet, “Particle filtering for partially observed Gaussian state space models,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 64, no. 4, pp. 827–836, oct 2002.
- [4] I. Arasaratnam and S. Haykin, “Cubature Kalman filters,” *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, jun 2009.
- [5] I. Arasaratnam, S. Haykin, and R. J. Elliott, “Discrete-Time nonlinear filtering algorithms using Gauss-Hermite quadrature,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.
- [6] I. Arasaratnam, S. Haykin, and T. R. Hurd, “Cubature Kalman filtering for continuous-discrete systems: theory and simulations,” *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 4977–4993, oct 2010.
- [7] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [8] D. B. Rubin, “A Noniterative Sampling/Importance resampling alternative to data augmentation for creating a few imputations when fractions of missing information are modest: The SIR algorithm,” *Journal of the American Statistical Association*, vol. 82, pp. 544–546, 1987.
- [9] A. Badolato, G. Rubio-Cidre, L. Úbeda-Medina, J. Grajal, B. Mencia-Oliva, A. Garcia-Pino, B. Gonzalez-Valdes, and O. Rubiños, “A 300 GHz imaging radar

- for standoff anomaly detection,” in *2015 9th European Conference on Antennas and Propagation, EuCAP 2015*, 2015.
- [10] Y. Bar-Shalom and K. Birmiwal, “Variable Dimension Filter for Maneuvering Target Tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-18, no. 5, pp. 621–629, 1982.
- [11] Y. Bar-Shalom, *Multitarget-multisensor tracking: advanced applications*. Artech House, 2000.
- [12] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. John Wiley & Sons, Inc., 2001.
- [13] M. Barkat, *Signal detection and estimation*. Artech House, Inc., 2005.
- [14] J. Beaudeau, M. F. Bugallo, and P. M. Djurić, “Analysis of the cross-target measurement fusion likelihood for RSSI-based sensors,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, no. 3, pp. 1856–1860, 2014.
- [15] J. P. Beaudeau, M. F. Bugallo, and P. M. Djurić, “RSSI-Based Multi-Target Tracking by Cooperative Agents Using Fusion of Cross-Target Information,” *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5033–5044, 2015.
- [16] B. M. Bell and F. W. Cathey, “The Iterated Kalman Filter Update as a Gauss-Newton Method,” *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.
- [17] K. L. Bell, C. J. Baker, G. E. Smith, J. T. Johnson, and M. Rangaswamy, “Fully adaptive radar for target tracking part I: Single target tracking,” *2014 IEEE Radar Conference*, pp. 0303–0308, may 2014.
- [18] —, “Fully adaptive radar for target tracking part II: Target detection and track initiation,” *2014 IEEE Radar Conference*, no. 4, pp. 0309–0314, may 2014.
- [19] —, “Cognitive Radar Framework for Target Detection and Tracking,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 8, pp. 1427–1439, 2015.
- [20] K. L. Bell, J. T. Johnson, G. E. Smith, C. J. Baker, and M. Rangaswamy, “Cognitive Radar for Target Tracking Using a Software Defined Radar System,” *Radar Conference (RadarCon), 2015 IEEE*, pp. 1394–1399, 2015.

- [21] V. E. Beneš, “Exact finite-dimensional filters for certain diffusions with nonlinear drift,” *Stochastics*, vol. 5, no. 1-2, pp. 65–92, jun 1981.
- [22] S. Blackman and A. House, *Design and analysis of modern tracking systems*. Boston, MA: Artech House, Inc., 1999.
- [23] Y. Boers and H. Driessen, “The mixed labeling problem in multi target particle filtering,” *FUSION 2007 - 2007 10th International Conference on Information Fusion*, 2007.
- [24] Y. Boers, E. Sviestins, and H. Driessen, “Mixed labelling in multitarget particle filtering,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 792–802, 2010.
- [25] X. Boyen and D. Koller, “Tractable Inference for Complex Stochastic Processes,” in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 2013, pp. 33–42.
- [26] M. F. Bugallo, T. Lu, and P. M. Djurić, “Target Tracking by Multiple Particle Filtering,” *IEEE Aerospace Conference*, pp. 1–7, 2007.
- [27] A. Butterfield, A. E. Mitchell, G. E. Smith, K. L. Bell, and M. Rangaswamy, “Metrics for Quantifying Cognitive Radar Performance,” *CIE International Conference of Radar Proceedings*, pp. 1 – 6, 2016.
- [28] O. Cappe, S. J. Godsill, and E. Moulines, “An Overview of Existing Methods and recent advances in sequential Monte Carlo,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, may 2007.
- [29] P. Chavali and A. Nehorai, “Scheduling and Power Allocation in a Cognitive Radar Network for Multiple-Target Tracking,” *Signal Processing, IEEE Transactions on*, vol. 60, no. 2, pp. 715–729, 2012.
- [30] X. Chen and S. Godsill, “Multiple dipolar sources localization for MEG using Bayesian particle filtering,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013, pp. 949–953.
- [31] P. Closas, C. Fernández-Prades, and J. Vilá-Valls, “Multiple quadrature Kalman filtering,” *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6125–6137, 2012.
- [32] F. Daum and J. Huang, “Curse of dimensionality and particle filters,” *Proceedings of 2003 IEEE Aerospace Conference*, vol. 4, pp. 1979–1993, 2003.

- [33] —, “Particle degeneracy: root cause and solution,” *Proc. SPIE 8050, Signal Processing, Sensor Fusion, and Target Recognition XX*, vol. 8050, no. May 2011, pp. 80 500W–80 500W–11, 2011.
- [34] F. E. Daum, “Nonlinear Filters: Beyond the Kalman Filter,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.
- [35] F. E. Daum, R. Company, and E. Division, “New exact nonlinear filters: theory and applications,” *Proc. SPIE 2235, Signal and Data Processing of Small Targets*, vol. 2235, pp. 636–649, 1994.
- [36] C. De Bernardis, F. Vicente-Guijalba, T. Martinez-Marin, and J. M. Lopez-Sanchez, “Contribution to Real-Time Estimation of Crop Phenological States in a Dynamical Framework Based on NDVI Time Series: Data Fusion with SAR and Temperature,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 8, pp. 3512–3523, 2016.
- [37] —, “Particle filter approach for real-time estimation of crop phenological states using time series of NDVI images,” *Remote Sensing*, vol. 8, no. 7, pp. 5–10, 2016.
- [38] C. G. De Bernardis, F. Vicente-Guijalba, T. Martinez-Marin, and J. M. Lopez-Sanchez, “Estimation of Key Dates and Stages in Rice Crops Using Dual-Polarization SAR Time Series and a Particle Filtering Approach,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 3, pp. 1008–1018, 2015.
- [39] P. M. Djurić and M. F. Bugallo, “Improved target tracking with particle filtering,” *2009 IEEE Aerospace conference*, pp. 1–7, mar 2009.
- [40] —, “Particle filtering for high-dimensional systems,” *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP 2013*, pp. 352–355, 2013.
- [41] P. M. Djurić, J. H. Kotecha, J. Zhang, T. Ghirmai, M. F. Bugallo, and J. Míguez, “Particle filtering,” *Signal Processing Magazine, IEEE*, vol. 20, no. 5, pp. 19–38.
- [42] P. M. Djurić, L. Ting, and M. F. Bugallo, “Multiple particle filtering,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 3, pp. 1181–1184, 2007.
- [43] A. Doucet, N. Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer New York, 2001.

- [44] 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, 2000.
- [45] O. E. Drummond and B. E. Fridling, “Ambiguities in evaluating performance of multiple target tracking algorithms,” in *Proceedings of the SPIE Conference on Signal and data processing of small targets*, 1992, pp. 326–337.
- [46] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, “Sonar tracking of multiple targets using joint probabilistic data association,” *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.
- [47] B. E. Fridling and O. E. Drummond, “Performance evaluation methods for multiple-target-tracking algorithms,” in *Proceedings of the SPIE Conference on Signal and data processing of small targets*, 1991, pp. 371–373.
- [48] J. M. Fuster, *Cortex And Mind: Unifying Cognition*. Oxford University Press, 2005.
- [49] Á. F. García-Fernández, “Detection and tracking of multiple targets using wireless sensor networks,” Ph. D. dissertation, Universidad Politécnica de Madrid, 2011.
- [50] Á. F. García-Fernández, J. Grajal, and M. Morelande, “Two-layer particle filter for multiple target detection and tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 1569–1588, 2013.
- [51] Á. F. García-Fernández, J. Grajal, and O. A. Yeste-Ojeda, “Analysis of ISAR images of a helicopter by a facet model,” *Proceedings of the 2008 International Conference on Radar, Radar 2008*, pp. 32–37, 2008.
- [52] Á. F. García-Fernández, M. R. Morelande, and J. Grajal, “Truncated Unscented Kalman Filtering,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3372–3386, 2012.
- [53] Á. F. García-Fernández, L. Svensson, M. R. Morelande, and S. Särkkä, “Posterior Linearization Filter : Principles and Implementation Using Sigma Points,” *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5561–5573, 2015.
- [54] Á. F. García-Fernández, B.-N. Vo, and B.-T. Vo, “MCMC-based posterior independence approximation for RFS multitarget particle filters,” in *Proceedings of the 17th International Conference on Information Fusion (FUSION)*, 2014.

- [55] Á. F. García-Fernández, O. A. Yeste-Ojeda, and J. Grajal, “Facet model of moving targets for ISAR imaging and radar back-scattering simulation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 3, pp. 1455–1467, 2010.
- [56] F. García-Rial, L. Úbeda-Medina, and J. Grajal, “Real-time GPU-based image processing for a 3-D THz radar,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2953–2964, 2017.
- [57] F. García-Rial, L. Úbeda-Medina, J. Grajal, A. Badolato, G. Rubio-Cidre, A. Garcia-Pino, B. Gonzalez-Valdes, and O. Rubinos, “Optimization of a compact THz imaging radar for real-time operation,” *2016 Global Symposium on Millimeter Waves, GSMM 2016 and ESA Workshop on Millimetre-Wave Technology and Applications*, pp. 4–7, 2016.
- [58] W. R. Gilks and C. Berzuini, “Following a moving target - Monte Carlo inference for dynamic Bayesian models,” *Journal of the Royal Statistical Society Series B-Statistical Methodology*, vol. 63, pp. 127–146, 2001.
- [59] F. Gini and M. Rangaswamy, *Knowledge Based Radar Detection, Tracking and Classification (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. New York, NY, USA: Wiley-Interscience, 2008.
- [60] S. J. Godsill, a. Doucet, and M. West, “Monte Carlo Smoothing for Nonlinear Time Series,” *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, 2004.
- [61] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.
- [62] N. A. Goodman, P. R. Venkata, and M. A. Neifeld, “Adaptive waveform design and sequential hypothesis testing for target recognition with active sensors,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 1, pp. 105–113, 2007.
- [63] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings-F on Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.
- [64] J. Grajal, A. Badolato, G. Rubio-Cidre, L. Úbeda-Medina, B. Mencia-Oliva, A. Garcia-Pino, B. Gonzalez-Valdes, and O. Rubinos, “3-D high-resolution imaging radar at 300 GHz with enhanced FoV,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 3, 2015.

- [65] J. Grajal, G. Rubio-Cidre, A. Badolato, L. Úbeda-Medina, F. Garcia-Rial, A. Garcia-Pino, and O. Rubinos, “Compact Radar Front-End for an Imaging Radar at 300 GHz,” *IEEE Transactions on Terahertz Science and Technology*, vol. 7, no. 3, pp. 268–273, 2017.
- [66] M. S. Grewal and A. P. Andrews, “Applications of Kalman Filtering in Aerospace 1960 to the Present,” *IEEE Control Systems*, vol. 30, no. 3, pp. 69–78, 2010.
- [67] J. R. Guerci, *Cognitive Radar: The Knowledge-Aided Fully Adaptive Approach*. Artech House, 2010.
- [68] F. Gustafsson, “Particle filter theory and practice with positioning applications,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–81, 2010.
- [69] S. Haykin, *Kalman filtering and neural networks*, S. Haykin, Ed. Wiley Online Library, 2001.
- [70] —, “Cognitive Radar,” *IEEE Transactions on Signal Processing Magazine*, vol. 23, no. 1, pp. 30–40, 2010.
- [71] S. Haykin, Y. Xue, and P. Setoodeh, “Cognitive Radar: Step Toward Bridging the Gap Between Neuroscience and Engineering,” *Proceedings of the IEEE*, vol. 100, no. 11, pp. 3102–3130, nov 2012.
- [72] G. Hendeby, R. Karlsson, and F. Gustafsson, “Particle Filtering: The Need for Speed,” *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 181403, 2010.
- [73] A. O. Hero and D. Cochran, “Sensor Management: Past, Present, and Future,” *IEEE Sensors Journal*, vol. 11, no. 12, pp. 3064–3075, dec 2011.
- [74] A. O. Hero, D. Castañón, D. Cochran, and K. Kastella, *Foundations and applications of sensor management*. Springer Science & Business Media, 2007.
- [75] Y. Ho and R. Lee, “A Bayesian approach to problems in stochastic estimation and control,” *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pp. 333–339, 1964.
- [76] C.-L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making - Methods and Applications: A State-of-the-Art Survey*. Springer, 1982, vol. 7, no. 1.
- [77] K. Ito and K. Xiong, “Gaussian filters for nonlinear filtering problems,” *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, may 2000.

- [78] A. M. Johansen and A. Doucet, “A note on auxiliary particle filters,” *Statistics & Probability Letters*, vol. 78, no. 12, pp. 1498–1504, sep 2008.
- [79] S. Julier, J. Uhlmann, and H. F. Durrant-whyte, “A new method for the nonlinear transformation of means and covariances in filters and estimators,” in *IEEE Transactions on Automatic Control*, vol. 45, no. 3, 2000, pp. 477–482.
- [80] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” in *Proceedings of the IEEE*, vol. 92, no. 3, 2004, pp. 401–422.
- [81] R. E. Kalman, “A New approach to linear filtering and prediction problems,” *Transactions of the ASME-Journal of Basic Engineering.*, vol. 82, no. Series D, pp. 35–45, 1960.
- [82] R. E. Kalman, P. L. Falb, and M. A. Arbib, *Topics in mathematical system theory*. McGraw-Hill New York, 1969, vol. 1.
- [83] S. Kar and J. M. Moura, “Gossip and distributed Kalman filtering: Weak consensus under weak detectability,” *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1766–1784, 2011.
- [84] S. M. Kay, *Fundamentals of statistical signal processing: Detection theory*, A. V. Oppenheim, Ed. Prentice Hall signal processing series, 1993.
- [85] ———, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall signal processing series, 1993.
- [86] M. Klaas, N. de Freitas, and A. Doucet, “Toward Practical N^2 Monte Carlo: The Marginal Particle Filter,” *Proceedings of the Twenty-First Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pp. 308–315, 2005.
- [87] B. Kolb and I. Q. Whishaw, *Fundamentals of human neuropsychology*. Macmillan, 2009.
- [88] A. Kong, J. S. Liu, and W. H. Wong, “Sequential imputations and Bayesian missing data problems,” vol. 89, no. 425, pp. 278–288, 1994.
- [89] K. K. Kottakki, M. Bhushan, and S. Bhartiya, “An improved Gaussian Sum Unscented Kalman Filter,” *Proceedings of the Third International Conference on Advances in Control and Optimization of Dynamical Systems*, vol. 3, pp. 355–362, 2014.
- [90] S. C. Kramer and H. W. Sorenson, “Recursive Bayesian estimation using piecewise constant approximations,” *Automatica*, vol. 24, no. 6, pp. 789–801, 1988.

- [91] C. M. Kreucher, A. O. Hero III, K. D. Kastella, and M. R. Morelande, “An Information-Based Approach to Sensor Management in Large Dynamic Networks,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 978–999, may 2007.
- [92] C. M. Kreucher, K. Kastella, and A. O. Hero III, “Multitarget tracking using the joint multitarget probability density,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1396–1414, 2005.
- [93] V. Kučera, “Rudolf E. Kalman: Life and Works,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 631–636, 2017.
- [94] H. J. Kushner, “Nonlinear filtering: the exact dynamical equations satisfied by the conditional mode,” *IEEE Transactions on automatic control.*, vol. 12, no. 3, pp. 262–267, 1967.
- [95] T. Lefebvre, H. Bruyninckx, and J. D. Schutter, “Kalman Filters for nonlinear systems: a comparison of performance.” *International Journal of Control*, vol. 77, no. 7, pp. 639–653.
- [96] —, “Kalman Filters for nonlinear systems: a comparison of performance.” *IEEE Transactions on Automatic Control*, vol. 32, no. October, pp. 1–31, 2001.
- [97] P. H. Leong, S. Arulampalam, T. A. Lamahewa, and T. D. Abhayapala, “A Gaussian-sum based cubature Kalman filter for bearings-only tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1161–1176, 2013.
- [98] T. Li, M. Bolić, and P. M. Djurić, “Resampling Methods for Particle Filtering,” *IEEE Signal Processing Magazine*, no. May, pp. 70–86, 2015.
- [99] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking. Part II: Ballistic target models,” in *Proceedings of the 2001 SPIE Conference on Signal and Data Processing of Small Targets*, no. 4473, 2001, pp. 559–581.
- [100] —, “Survey of maneuvering target tracking. Part III: Measurement models,” in *Proceedings of the 2001 SPIE Conference on Signal and Data Processing of Small Targets*, no. 4473, 2001, pp. 423–446.
- [101] —, “Survey of maneuvering target tracking. Part IV: decision-based methods,” in *Proceedings of the 2001 SPIE Conference on Signal and Data Processing of Small Targets*, no. 4473, 2001, pp. 511–534.

- [102] —, “Survey of maneuvering Target Tracking. Part I: Dynamic Models,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.
- [103] —, “Survey of maneuvering Target Tracking. Part V: Multiple -Model Methods,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.
- [104] —, “Survey of maneuvering Target Tracking. Part II: Motion Models of Ballistic and Space Targets,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 96–119, 2010.
- [105] X. Li and Y. Bar-Shalom, “Design of an interacting multiple model algorithm for air traffic control tracking,” *IEEE Transactions on Control Systems Technology*, vol. 1, no. 3, pp. 186–194, 1993.
- [106] J. S. Liu, *Monte Carlo strategies in scientific computing*. New York: Springer, 2002.
- [107] J. S. Liu and R. Chen, “Blind deconvolution via sequential imputations,” *Journal of the american statistical association*, vol. 90, no. 430, pp. 567–576, 1995.
- [108] D. G. Luenberger, *Linear and nonlinear programming*. Addison-Wesley publishing company, 1984.
- [109] S. Mabbs, “A performance assessment environment for radar signal processing and tracking algorithms,” *proceedings of the Pacific Rim Conference on Computers, Communications and Signal Processing*, pp. 9–12, 1993.
- [110] L. Martino, V. Elvira, and F. Louzada, “Alternative effective sample size measures for importance sampling,” in *proceedings of the 2016 Statistical Signal Processing Workshop*, 2016.
- [111] —, “Weighting a resampled particle in sequential Monte Carlo,” in *proceedings of the 2016 Statistical Signal Processing Workshop*, 2016.
- [112] —, “Effective sample size for importance sampling based on discrepancy measures,” *Signal Processing*, vol. 131, pp. 386–401, 2017.
- [113] S. Maskell, M. Rollason, N. Gordon, and D. Salmond, “Efficient particle filtering for multiple target tracking with application to tracking in structured images,” *Image and Vision Computing*, vol. 21, no. 10, pp. 931–939, 2003.

- [114] S. B. McGrayne, *The theory that would not die: how Bayes' rule cracked the enigma code, hunted down Russian submarines, & emerged triumphant from two centuries of controversy*. Yale University Press, 2011.
- [115] D. Mehrholz, L. Leushacke, W. Flury, R. Jehn, H. Klinkrad, and M. Landgraf, "Detecting, tracking and imaging space debris," *ESA Bulletin(0376-4265)*, vol. 109, no. 109, pp. 128–134, 2002.
- [116] R. V. D. Merwe and E. A. Wan, "The Unscented Kalman Filter for Nonlinear Estimation," in *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.
- [117] A. E. Mitchell, G. E. Smith, K. L. Bell, A. Duly, and M. Rangaswamy, "Fully Adaptive Radar Cost Function Design," *IET Radar, Sonar & Navigation*, no. Special Issue on Cognitive Radar, pp. 1301–1306, 2018.
- [118] M. R. Morelande, "Tracking multiple targets with a sensor network," in *Proceedings of the 9th International Conference on Information Fusion (FUSION)*, 2006.
- [119] M. R. Morelande and Á. F. García-Fernández, "Analysis of Kalman filter approximations for nonlinear measurements," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5477–5484, 2013.
- [120] M. R. Morelande, C. M. Kreucher, and K. Kastella, "A Bayesian approach to multiple target detection and tracking," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1589–1604, 2007.
- [121] M. R. Morelande and A. Zhang, "Uniform sampling for multiple target tracking," *Proceedings of the 14th International Conference Information Fusion (FUSION)*, pp. 283–289, 2011.
- [122] M. R. Morelande and A. M. Zhang, "A mode preserving particle filter," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 3984–3987, 2011.
- [123] A. I. Mourikis and S. I. Roumeliotis, "A multi-state Kalman filter for vision-aided inertial navigation," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, no. April, pp. 3565–3572, 2007.
- [124] C. Musso, N. Oudjane, and F. Gland, "Improving Regularised Particle Filters," *Sequential Monte Carlo Methods in Practice*, no. Silverman 1986, pp. 247–271, 2001.

- [125] B. Ng, L. Peshkin, and A. Pfeffer, “Factored particles for scalable monitoring,” *Proceedings of the 18th conference on Uncertainty in artificial intelligence*, pp. 370–377, 2002.
- [126] N. Nguyen, K. Dogancay, and L. Davis, “Adaptive waveform selection for multistatic target tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 688–701, 2015.
- [127] G. Ortiz-Jiménez, F. García-Rial, L. Úbeda-Medina, R. Pagés, N. García, and J. Grajal, “Simulation Framework for a 3-D High-Resolution Imaging Radar at 300 GHz with a Scattering Model Based on Rendering Techniques,” *IEEE Transactions on Terahertz Science and Technology*, vol. 7, no. 4, pp. 404–414, 2017.
- [128] M. Orton and W. Fitzgerald, “A Bayesian approach to tracking multiple targets using sensor arrays and particle filters,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 216–223, 2002.
- [129] A. Petrovskaya and S. Thrun, “Model based vehicle detection and tracking for autonomous urban driving,” *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.
- [130] M. K. Pitt, N. Shephard, and Q. Gate, “Filtering via simulation: Auxiliary particle filters,” *Journal of the American Statistical Association*, pp. 590–599, 1999.
- [131] R. Prasad and M. Ruggieri, *Applied satellite navigation using GPS, GALILEO and augmentation systems*. Artech House, Inc., 2005.
- [132] J. Predd, S. Kulkarni, and H. Poor, “Distributed learning in wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.
- [133] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method.*, I. John Wiley & Sons, Ed., 2008.
- [134] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, “Generalized optimal sub-pattern assignment metric,” *20th International Conference on Information Fusion, Fusion 2017 - Proceedings*, 2017.
- [135] X. Ren, M. Altmeyer, S. Reiss, and K. Schweizer, “Process-based account for the effects of perceptual attention and executive attention on fluid intelligence: An integrative approach,” *Acta Psychologica*, vol. 142, no. 2, pp. 195–202, 2013.

- [136] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter, particle filters for tracking applications*. Artech House, 2004.
- [137] M. Roth and F. Gustafsson, “An efficient implementation of the second order extended Kalman filter,” *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pp. 1–6, 2011.
- [138] R. L. Rothrock and O. E. Drummond, “Performance metrics for multisensor multiple-target tracking,” in *Proceedings of the SPIE Conference on Signal and data processing of small targets*, 2000, pp. 521–531.
- [139] G. Rubio-Cidre, A. Badolato, L. Úbeda-Medina, J. Grajal, B. Mencia-Oliva, and B.-P. Dorta-Naranjo, “DDS-Based Signal-Generation Architecture Comparison for an Imaging Radar at 300 GHz,” *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 11, 2015.
- [140] G. Rubio-Cidre, A. Badolato, L. Úbeda-Medina, B. Mencia-Oliva, J. Grajal, A. Garcia-Pino, B. Gonzalez-Valdes, and O. Rubinos Lopez, “Characterization of a 300 GHz imaging radar for standoff detection,” *2014 International Radar Conference, Radar 2014*, pp. 1–6, 2014.
- [141] R. Sameni, M. B. Shamsollahi, C. Jutten, and G. D. Clifford, “A nonlinear Bayesian filtering framework for ECG denoising,” *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 12, pp. 2172–2185, 2007.
- [142] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- [143] S. Särkkä, J. Hartikainen, L. Svensson, and F. Sandblom, “On the relation between Gaussian process quadratures and sigma-point methods,” *arXiv*, pp. 1–13, 2015.
- [144] T. Schön, F. Gustafsson, and P.-J. Nordlund, “Marginalized particle filters for mixed linear/nonlinear state-space models,” *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, jul 2005.
- [145] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, “A consistent metric for performance evaluation of multi-object filters,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, aug 2008.
- [146] D. Schuhmacher and A. Xia, “A new metric between distributions of point processes,” *Advances in Applied Probability*, vol. 40, no. 3, pp. 651–672, 2008.

- [147] X. Shen, S. Liu, and P. K. Varshney, "Sensor Selection for Nonlinear Systems in Large Sensor Networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 2664–2678, 2014.
- [148] M. Šimandl and J. Duník, "Sigma point Gaussian sum filter design using square root unscented filters," *Proceedings of the 16th IFAC world congress*, vol. 16, pp. 1000–1005, 2005.
- [149] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.
- [150] S. P. Sira, A. Papandreou-Suppappola, and D. Morrell, "Dynamic configuration of time-varying waveforms for agile sensing and tracking in clutter," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3207–3217, 2007.
- [151] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice-Hall, Inc., 1991.
- [152] V. Smidl and A. Quinn, "Variational Bayesian filtering," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5020–5030, oct 2008.
- [153] G. E. Smith, "Cognitive radar experiments at the Ohio State University," *2017 Cognitive Communications for Aerospace Applications Workshop, CCAA 2017*, 2017.
- [154] G. E. Smith, Z. Cammenga, A. Mitchell, K. L. Bell, J. Johnson, M. Rangaswamy, and C. Baker, "Experiments with cognitive radar," *IEEE Aerospace and Electronic Systems Magazine*, vol. 31, no. 12, pp. 34–46, 2016.
- [155] G. E. Smith, Z. Cammenga, A. Mitchell, K. L. Bell, M. Rangaswamy, J. T. Johnson, and C. J. Baker, "Experiments with cognitive radar," *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP 2015*, pp. 293–296, 2016.
- [156] P. J. Smith, M. Shafi, and H. Gao, "Quick simulation: A review of importance sampling techniques in communications systems," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 4, pp. 597–613, 1997.
- [157] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell, *Bayesian multiple target tracking*, 2nd ed. Artech House, 2014.
- [158] O. Straka, J. Duník, and M. Šimandl, "Gaussian sum unscented Kalman filter with adaptive scaling parameters," *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pp. 9–16, 2011.

- [159] S. H. Strogatz, “Exploring complex networks,” *Nature*, vol. 410, pp. 268–277, 2001.
- [160] S. H. Strogatz and D. S. Shafer, “Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering,” p. 512, 1995.
- [161] H. Tanizaki and R. S. Mariano, “Nonlinear Filters: Estimation and Applications,” *Communications in Statistics - Theory and Methods*, vol. 25, no. 6, pp. 1261–1282, 1996.
- [162] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
- [163] L. Úbeda-Medina, Á. F. García-Fernández, and J. Grajal, “Generalizations of the auxiliary particle filter for multiple target tracking,” in *Proceedings of the 17th International Conference on Information Fusion (FUSION)*, 2014.
- [164] —, “Adaptive Auxiliary Particle Filter for Track-Before-Detect With Multiple Targets,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 5, pp. 2317–2330, 2017.
- [165] —, “Target tracking using multiple auxiliary particle filtering,” *Proceedings of the 20th International Conference on Information Fusion, FUSION. Xi’an, China.*, pp. 1–8, 2017.
- [166] —, “Robust sensor parameter selection in fully adaptive radar using a sigma-point Gaussian approximation,” in *Proceedings of the IEEE Radar Conference (RadarCon)*, Oklahoma, 2018.
- [167] —, “Sigma-point multiple particle filtering,” *submitted to Automatica*, 2018.
- [168] L. Úbeda-Medina and J. Grajal, “Implementation of the fully adaptive radar framework: practical limitations,” *Proceedings of the 2017 Radar Conference*.
- [169] —, “Multiple target tracking in the fully adaptive radar framework,” *IEEE Workshop on Statistical Signal Processing Proceedings*, 2016.
- [170] L. Úbeda-Medina, C. Herrera-Yagüe, I. Barriales-Valbuena, P. J. Zufria, and M. Congosto-Martínez, “A combined algorithm for analyzing structural controllability and observability of complex networks,” *10th International Conference on Scientific Computing (CSC’13)*, pp. 172–177, 2013.

- [171] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan, “The Unscented Particle Filter,” *Advances in Neural Information Processing Systems*, vol. 96, no. 6080, pp. 584–590, 2001.
- [172] R. Van Der Merwe and E. A. Wan, “The square-root unscented Kalman filter for state and parameter-estimation,” *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP -01). 2001 IEEE International Conference on*, vol. 6, pp. 3461–3464, 2001.
- [173] R. van der Merwe, E. Wan, and S. J. Julier, “Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion: Applications to Integrated Navigation,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 1–30, 2004.
- [174] P. J. van Leeuwen, “Particle Filtering in Geophysical Systems,” *Monthly Weather Review*, vol. 137, no. 12, pp. 4089–4114, 2009.
- [175] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, 1st ed. New York, NY, USA: Wiley & Sons, 2002.
- [176] —, *Detection, estimation, and modulation theory*. John Wiley & Sons, 2004.
- [177] C. Vergé, C. Dubarry, P. Del Moral, and E. Moulines, “On parallel implementation of sequential Monte Carlo methods: the island particle model,” *Statistics and Computing*, vol. 25, no. 2, pp. 243–260, 2013.
- [178] J. Vermaak, S. Godsill, and P. Perez, “Monte Carlo filtering for multi-target tracking and data association,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 309–332, jan 2005.
- [179] J. Vermaak and A. Doucet, “Maintaining Multi-Modality through Mixture Tracking,” *IEEE International Conference on Computer Vision*, vol. 2, 2003.
- [180] F. Vicente-Guijalba, J. M. López-Sánchez, and T. Martínez-Marín, “A Dynamical System Approach for Agricultural Crop Phenology Tracking using PolSAR Time-Series: Cereals Case Study at C-Band,” *EUSAR 2014; 10th European Conference on Synthetic Aperture Radar; Proceedings of*, pp. 1–4, 2014.
- [181] F. Vicente-Guijalba, T. Martínez-Marín, and J. M. López-Sánchez, “Crop phenology estimation using a multitemporal model and a kalman filtering strategy,” *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 6, pp. 1081–1085, 2014.

- [182] —, “Dynamical Approach for Real-Time Monitoring of Agricultural Crops,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3278–3293, 2015.
- [183] M. Vihola, “Rao-Blackwellised Particle Filtering in Random Set Multitarget Tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 2, pp. 689–705, 2007.
- [184] J. Vilà-Valls, P. Closas, and Á. F. García-Fernández, “Uncertainty Exchange Through Multiple Quadrature Kalman Filtering,” *Signal Processing Letters*, vol. 23, no. 12, pp. 1825–1829, 2016.
- [185] J. Vilà-Valls, P. Closas, Á. F. García-Fernández, and C. Fernández-Prades, “Multiple Sigma-point Kalman Smoothers for High-dimensional State-Space Models,” in *Proceedings of the IEEE 7th International Workshop on computational advances in multi-sensor adaptive processing (CAMSAP)*, 2017.
- [186] S. Wiggins, *Introduction to applied nonlinear dynamical systems and chaos*. Springer Science & Business Media, 2003.
- [187] Y. Wu, D. Hu, M. Wu, and X. Hu, “Unscented Kalman filtering for additive noise case: augmented vs. non-augmented,” *Proceedings of the 2005, American Control Conference, 2005.*, pp. 4051–4055, 2005.
- [188] M. Z. Yanmei Hu, Graham J. Hitch, Alan D. Baddeley and R. J. Allen, “Executive and perceptual attention play different roles in visual working memory: Evidence from suffix and strategy effects,” *Journal of Experimental psychology: Human perception and performance*, vol. 40, no. 4, pp. 1665–1678, 2014.
- [189] O. A. Yeste-Ojeda and J. Grajal, “Atomic Decomposition for ISAR imaging,” *IEEE National Radar Conference - Proceedings*, no. 4, pp. 504–509, 2006.
- [190] W. Yi, M. R. Morelande, L. Kong, and J. Yang, “A computationally efficient particle filter for multitarget tracking using an independence approximation,” *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 843–856, feb 2013.
- [191] V. S. Zaritskii, V. B. Svetnik, and S. L.I., “Monte Carlo technique in problems of optimal data processing,” *Automation and Remote Control*, vol. 12, pp. 95–103, 1975.
- [192] P. J. Zufiria, L. Úbeda-Medina, C. Herrera-Yagüe, and I. Barriales-Valbuena, “Mathematical foundations for efficient structural controllability and observabil-

- ity analysis of complex systems,” *Mathematical Problems in Engineering*, vol. 2014, 2014.
- [193] P. J. Zufiria and J. A. Álvarez-Cubero, “Generalized lexicographic multiobjective combinatorial optimization. Application to cryptography,” *Journal on optimization of the society for industrial and applied mathematics (SIAM)*, vol. 27, no. 4, pp. 2182–2201, 2017.
- [194] P. J. Zufiria, D. Pastor-Escuredo, L. Úbeda-Medina, M. A. Hernández-Medina, I. Barriales-Valbuena, A. J. Morales, D. C. Jacques, W. Nkwambi, M. Bamba Diop, J. Quinn, P. Hidalgo-Sanchís, and M. Luengo-Oroz, “Identifying seasonal mobility profiles from anonymized and aggregated mobile phone data. Application in food security,” *PLoS ONE*, vol. 13, no. 4, pp. 1–20, 2018.
- [195] P. J. Zufiria, D. Pastor-Escuredo, L. Úbeda-Medina, M. A. Hernández-Medina, I. Barriales-Valbuena, A. J. Morales, and W. Nkwambi, “Mobility profiles and calendars for food security and livelihoods analysis,” in *Actes du Challenge D4D, Netmob*, no. April, 2015.

Madrid, a de de 2018

Fdo: Luis Antonio Úbeda Medina