State estimation and trajectory planning using box particle kernels

Nicolas Merlinge

To cite this version:

Nicolas Merlinge. State estimation and trajectory planning using box particle kernels. Automatic. Université Paris Saclay (COmUE); Coventry University, 2018. English. NNT : 2018SACLS425 . tel-01933440

HAL Id: tel-01933440

https://theses.hal.science/tel-01933440v1

Submitted on 23 Nov 2018

université
PARIS-SACLAY

UNIVERSITÉ
PARIS
SUD
Comprendre le monde,
construire l'avenir

Coventry
University

NNT : 2018SACLS425

Thèse de doctorat

State estimation and trajectory planning using box particle kernels

Thèse de doctorat de Coventry University et de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

Ecole doctorale n°580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Automatique

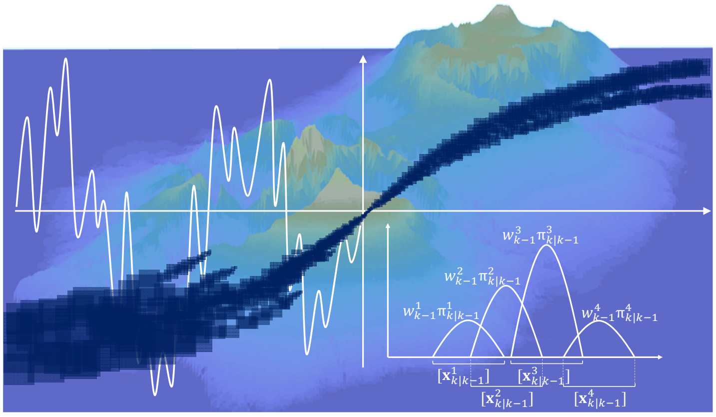
Thèse présentée et soutenue à Palaiseau, le 29/10/2018, par

NICOLAS MERLINGE

Composition du Jury :

François Le Gland Directeur de recherche, INRIA Rennes (IRMAR)	Président
Lyudmila Mihaylova Professeur, Université de Sheffield	Rapporteur
Philippe Bonnifait Professeur, Université de Technologie de Compiègne (Heudiasyc)	Rapporteur
Cristina Stoica-Maniu Professeur adjoint, Université Paris-Saclay (L2S)	Examinateur
Luc Jaulin Professeur, ENSTA Bretagne (STICC)	Examinateur
Hélène Piet-Lahanier Chercheur, ONERA (DTIS)	Directeur de thèse
James Brusey Reader, Université de Coventry (FMFM)	Directeur de thèse
Karim Dahia Chercheur, ONERA (DTIS)	Encadrant
Nadjim Horri Senior Lecturer, Université de Coventry (SMAAE)	Encadrant (invité)
Benoit Frapard Ingénieur, Airbus Defence and Space, Department of Technology and Innovation	Invité

STATE ESTIMATION AND TRAJECTORY PLANNING USING BOX PARTICLE KERNELS



BY

NICOLAS JONATHAN ADRIEN MERLINGE

A thesis submitted in partial fulfilment of the University's requirements for the Degree of Doctor of Philosophy

September 2018

Coventry University

ONERA
THE FRENCH AEROSPACE LAB

université
PARIS-SACLAY

ABSTRACT

State estimation and trajectory planning are two crucial functions for autonomous systems, and in particular for aerospace systems. Particle filters and sample-based trajectory planning approaches have been widely considered to tackle non-linear models and non-Gaussian noises. However, these approaches may produce erratic results due to the sampled approximation of the state density. In addition, they have a high computational cost which limits their practical interest.

This thesis investigates the use of box kernel mixtures to describe multimodal probability density functions. A box kernel mixture is a weighted sum of basic functions (e.g. uniform kernels) that integrate to unity, and whose supports are bounded by boxes, i.e. vectors of intervals. This modelling approach yields a more extensive description of the state density function while requiring a lower computational load. New algorithms are developed, based on a derivation of the Box Particle Filter (BPF) for state estimation, and of a particle based Chance Constrained optimisation (equivalently, failure probability constraint) for trajectory planning under uncertainty.

In order to tackle ambiguous state estimation problems, a Box Regularised Particle Filter (BRPF) is introduced. The BRPF consists of an improved BPF with a guaranteed resampling step, and a smoothing strategy based on kernel regularisation. The proposed strategy is theoretically proved to outperform the original BPF in terms of Mean Integrated Square Error (MISE), and empirically shown to reduce the Root Mean Square Error (RMSE) of estimation. For Terrain Aided Navigation (TAN) BRPF reduces computation load by 75% (4-fold reduction) compared to BPF, and by 97% (33-fold) compared to Particle Filter for a similar performance budget. BRPF is shown to be robust to measurement ambiguity and unknown-but-bounded measurement densities. BRPF is also integrated to federated and distributed architectures to demonstrate its efficiency in multi-sensor and multi-agent systems.

In order to tackle constrained trajectory planning under non-Gaussian uncertainty, a Box Particle Control (BPC) is introduced. BPC relies on an interval bounded kernel mixture state density description, and consists of propagating the state density along a state trajectory at a given horizon. It yields a more accurate description of the state uncertainty than previous particle based algorithms. A chance constrained optimisation is performed, which consists of finding the sequence of future control inputs that minimises a cost function, while ensuring that the probability of constraint violation (failure probability) remains below a given threshold. For similar performance, BPC yields a computation load reduction of 30% (1.4-fold reduction) with respect to Particle Control.

The use of box kernel mixtures in estimation algorithms (e.g. BRPF) and state control methods (e.g. BPC), makes it possible to run complex estimation and control operations in real time on computationally limited devices. The results are quite general and apply where (a) measurement is ambiguous, uncertain but bounded, (b) computational load is constrained, (c) either state estimation (BRPF) or control (BPC) are needed. These characteristics are of particular interest in the aerospace field.

ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude to my supervision team for their continuous support throughout my doctoral research. My directors of studies, James Brusey and Hélène Piet-Lahanier, taught me a lot about research methodology; my two supervisors, Nadjim Horri and Karim Dahia, made me benefit from their exceptional expertise.

I would also like to thank Lyudmila Mihaylova and Philippe Bonnifait for having accepted to be my examiners, Stratis Kanarachos for being my Coventry internal examiner, and François Le Gland as the Paris-Saclay's jury president. Thanks also to Cristina Stoica-Maniu, Luc Jaulin, and Benoit Frapard for having joined the jury of Paris-Saclay.

I am grateful to ONERA and Coventry University for having welcomed me during my thesis. I had the opportunity to work and exchange with many experts in different fields, which made me view my research from new perspectives. In particular, a great thanks to Jean-Michel Allard, Pascal Fély, Sébastien Reynaud, Julien Marzat, Olivier Haas, Elena Gaura, and Dobrila Petrovic for the various discussions we had. Many thanks to the PhD students and colleagues from DTIS, CEVA, FMFM and COGENT for providing a very nice working environment.

Last but not least, I would like to thank my family and my friends for the great support they have provided me. A great thanks to the Li212, the fam's 148, my friends from high-school, my parents and my wife.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

NOMENCLATURE

Sets

$\mathcal{S} = \{\bullet \mid \bullet\}$	conditionally defined set
$\cap, \cup$	intersection, union
$\mathcal{S}_1 \setminus \mathcal{S}_2$	$\mathcal{S}_1$ without $\mathcal{S}_2$
$\times$	Cartesian product
$[\underline{a}, \overline{a}]$, $[\underline{a}, \overline{a}[$, $]\underline{a}, \overline{a}[$	closed, semi-open, and open interval
$\mathbb{N}, \mathbb{N}^*$	set of natural integers and non-zero natural integers
$\mathbb{R}, \mathbb{R}^+, \mathbb{R}^*$	set of real numbers, positive real numbers, and non-zero real numbers
$\mathbb{IR}$	set of real intervals
$\lambda(\mathcal{S})$	Lebesgue measure of set $\mathcal{S}$

Algebra

$x \in \mathbb{R}$	scalar value
$\mathbf{x} \in \mathbb{R}^d$	vector of $d \in \mathbb{N}^*$ dimensions
$\mathbf{M} \in \mathbb{R}^{n \times m}$	matrix of n lines and m columns
$\lVert\mathbf{x}\rVert$	Euclidean norm of $\mathbf{x}$
$\lVert\mathbf{M}\rVert_F$	Frobenius norm of $\mathbf{M}$
$\mathbf{I}_d$	identity matrix in $\mathbb{R}^{d \times d}$
$\mathbf{0}_d$, $\mathbf{0}_{n \times m}$	zero value vector in $\mathbb{R}^d$, zero value matrix in $\mathbb{R}^{n \times m}$
$\otimes$	Kronecker product
$\mathrm{Chol}(\mathbf{M})$	Cholesky decomposition of matrix $\mathbf{M}$

Kernels and functions

$\mathbf{1}_{\mathcal{S}}$	indicator function on set $\mathcal{S}$
$\mathcal{U}_{\mathcal{S}}$	uniform kernel on domain $\mathcal{S}$
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with expectation μ and standard deviation σ
$*$, $\circledast_n$	convolution product, n times convolution
$\nabla^n f(x)$	n times differentiation of function f with respect to x

Probabilities

$\mathbf{x} \sim p(\mathbf{x})$	random vector $\mathbf{x}$ is associated to probability density function p
$p(\mathbf{x} \mid \mathbf{y})$	conditional density of $\mathbf{x}$ given $\mathbf{y}$
$E\,[]$	expectation
$\mathrm{Var}\,[]\,, \mathrm{Cov}\,[]$	variance, covariance

Units

°, rad	degree, radiant

ABBREVIATIONS

BPF	Box Particle Filter
BRPF	Box Regularised Particle Filter
DEM	Digital Elevation Model
EKF	Extended Kalman Filter
GR	Guaranteed Resampling
GS	Geometrical Subdivision
IMU	Inertial Measurement Unit
KF	Kalman Filter
LS	Least Square
MAP	Maximum A Posteriori
MISE	Mean Integrated Square Error
ML	Maximum Likelihood
MLCS	Maximum Likelihood Covariance Subdivision
MPF	Marginalised Particle Filter
MR	Multinomial Resampling
RMSE	Root Mean Square Error
RPF	Regularised Particle Filter
SIR-PF	Sequential Importance Resampling Particle Filter
st.d.	Standard Deviation
TAN	Terrain Aided Navigation

1

INTRODUCTION

State estimation and control are two key subsystems of autonomous aerospace vehicles. *State estimation* aims to retrieve the vehicle's current state (e.g. position, velocity) from available measurements (e.g. GNSS, altimeter), provided there is some knowledge of its dynamics. It is also referred to as the navigation system. The sensor and dynamical model knowledge is often not perfect, which introduces some uncertainty in the estimation process. A state estimation algorithm generally outputs a point-wise estimate of the vehicle's state, associated with a confidence region that is likely to contain the actual state. *State control* aims to bring the vehicle's current state to a desired state (e.g. a desired position, or a desired trajectory). State estimation uncertainty is often considered negligible, which makes the state control problem deterministic.

In the aerospace field, there is an increasing demand for autonomy, particularly for the next generations of aerial and space vehicles. The on-board estimation and control systems will therefore have to perform increasingly complex tasks to meet the requirements of current and future missions (e.g. remote controlled drones, unmanned space exploration). The demand for autonomy is also constrained by embedded hardware constraints. This can be either due to sensor cost reduction or some device robustness requirements. In some applications, accurate sensors cannot be used and must be replaced with other sensors providing less informative data. For example, GNSS measurements may be unavailable due to the environment characteristics (indoor navigation, jamming, underwater navigation, space exploration), and must be replaced with more ambiguous measurements (e.g. vision sensors, or Terrain Aided Navigation methods). This has a strong impact on the estimation and control performance, that must compensate for the lack on information by more complex embedded algorithms.

Measurement ambiguity refers to cases where one measurement occurrence may correspond to several possible states (e.g. a range measurement may correspond to a sphere of states around the sensor, and a terrain sensing measurement may correspond to several geographic areas of similar terrain profiles). Formally, ambiguities occur when the function relating the state to the measurements is not injective. The presence of measurement ambiguities increases the complexity of state estimation, and may prevent estimation algorithms from converging on a unique estimated state with a sufficiently narrow confidence region.

When the vehicle's future trajectory is constrained, measurement ambiguities become a critical issue. Indeed, the uncertainty of the current state knowledge cannot be neglected and may yield further constraint violations, for example the collision of obstacles, an unintentional intrusion in a restricted area or the unforeseen pulling-out of an aircraft flight envelope. The embedded state control must account for uncertainty at the cost of increased complexity to tackle such risks in an autonomous way.

The required increased complexity of embedded algorithms conflicts with the embedded computers computational ability, which is often severely limited for hardware robustness and energy consumption purposes. Therefore, the state estimation and state control have to meet a trade-off between performance (accuracy, robustness to ambiguities) and their required computational complexity.

1.1 MOTIVATION

This work aims to introduce new state estimation and trajectory planning algorithms that ensure robustness to measurement ambiguities, accuracy enhancement with respect to existing methods and computational load reduction. The algorithms developed in this thesis are designed to handle a large class of problems, including non-linear dynamics and non-linear measurements with non-Gaussian additive uncertainties. Some general properties are formally proved, while simulations illustrate their performance in practical cases. The main application is Terrain Aided Navigation (TAN) which consists of retrieving an aerial vehicle's state (position, velocity, and potentially attitude) from its relative elevation measurements with respect to the ground, given a Digital Elevation Model map (DEM) of the flown-over area. This application involves severe measurements ambiguities leading to the failure of many existing estimation methods. The trajectory planning algorithms will be evaluated for a number of applications, including trajectory propagation from TAN estimation.

Stochastic state estimation algorithms rebuild a conditional state density given the past measurements to determine the most probable current state of the vehicle. In the linear-Gaussian case, the state density remains Gaussian, and the optimal estimation solution is provided by the Kalman Filter (KF, [Kal+60]). However, in the case of non-Gaussian uncertainty, non-linear measurements, and severe ambiguities, the KF and its derivatives fail to estimate the state. Some non-parametric estimation approaches have therefore been introduced, such as the Particle Filter (PF, [GSS93]), which consists of approximating the state density by a weighted sum of Dirac functions. A more accurate PF derivation was proposed, called the Regularised Particle Filter (RPF [MOLG01]), which consists of smoothing the state density approximation by a mixture of bounded stochastic kernels. Another approach called the Box Particle Filter (BPF, [AGB07]) consists of approaching the state density by a mixture of weighted uniform kernels bounded by box particles (vectors of intervals). At each time-step, box particles are dynamically propagated and contracted with respect to the available measurements, which yields a weights update. When triggered, a box resampling step replaces low-weighted box particles with subdivisions of high-weighted ones. The estimation solution is shown to be robust to measurements ambiguities and requires a 10 times lower computational load for typical problems than the conventional PF. However, the robustness of BPF to ambiguity comes with low estimation accuracy and conservatism (unnecessarily large confidence margins). Those drawbacks prevent it from being used for critical applications such as aerospace navigation.

The BPF has proven to be robust to non-linear and ambiguous problems, but it may be conservative and inaccurate. When the problem consists of a collection of various measurements of different types, it does not appear suitable to process them all together in a BPF, since doing so may result in an increased computational load and an inaccurate estimation. For example, it is common to have, in addition to the TAN measurements (e.g. laser pointing toward the ground or radar altimeter), several Gaussian sensors (e.g. magnetometer, anemometer, relative sensing toward other physical bodies or vehicles). An allocation of the measurement inputs to dedicated local filters, including one or several BPF, can then be considered. Local filters estimates can then be fused by a Gaussian filter (Federated architectures [Car88]). However, in order to be integrated to a Gaussian federated architecture, the BPF output must be Gaussian, or at least unimodal (i.e. with a single local maximum), which is not guaranteed. A study of the BPF output density characteristics is therefore of interest.

A constrained trajectory planning algorithm consists of finding the future trajectory (defined by a sequence of control inputs) that minimises a cost function (e.g. energy consumption, distance toward objective state) while satisfying some constraints (e.g. collision avoidance or remaining in a flight envelope). When the state uncertainty is not negligible, due for example to measurements ambiguities, the trajectory planning must account for it and deterministic optimisation cannot be used. A way of making the problem deterministic is considering the worst case scenario by ensuring that all possible states satisfy the constraints (e.g. H-infinity methods, or set-based predictive control). However, worst case approaches may be too conservative and are likely to end with no solution. Another approach consists of relaxing the constraints while keeping the problem deterministic by defining a new state constraint that ensures that the probability of constraint violation remains below a desired threshold. This approach, called Chance Constrained optimisation (Charnes [CCS58]), makes it possible to explicitly estimate and manage the risk of constraint violation. This is referred to as *failure probability estimation* in what follows. To tackle non-Gaussian uncertainties, sample-based Chance Constraint optimisation was introduced (Pagnoncelli [PAS09], as well as Particle Control [Bla+10]), which consists of propagating a set of weighted point-wise trajectories to evaluate the failure probability, given a potentially non-convex feasible set (set of allowed trajectories). However, those approaches are only guaranteed to be efficient for a large number of sample-trajectories (theoretically infinite). Furthermore, the sample-based failure probability estimation is not differentiable with respect to the control sequence. This restricts Particle Control to linear programming optimisation techniques, which limits its use to linear dynamics.

1.2 RESEARCH QUESTIONS

In the light of the above problems, this thesis answers the following research questions:

1. Can BPF be formalised for any box kernel and is it of practical use to do so?
2. Can the original BPF be modified to ensure that the actual state belongs to at least one box particle? Uncertainties are assumed to be bounded and the dynamical and observation models are continuous.
3. What is the impact of the choice of box resampling subdivision dimension on the filter's performance? In particular, does accounting for observability enhance performance?
4. Does regularisation improve the BPF's accuracy?
5. Can BPF-like algorithms be integrated to larger estimation architectures such as Gaussian federated filters?
6. Does a box kernel mixture based failure probability estimation require a lower computational load than a Dirac mixture method for similar performance?
7. Is the box kernel failure probability estimation approach compatible with the differentiable optimisation scheme for trajectory planning?

1.3 CONTRIBUTION TO KNOWLEDGE

The work described in this thesis contributes to current knowledge the following:

- A state estimation filter based on a combination of the Box Particle Filter (BPF, [AGB07]) and the Regularised Particle Filter (RPF, [MOLG01]), called the Box Regularised Particle Filter (BRPF);
- Significant improvements in the box resampling step for BPF and BRPF, leading to a guaranteed state estimation consistency and a reduced conservatism;
- A generalisation of the BPF scheme leading to a box kernel mixture formulation (i.e. weighted sum of kernel functions bounded by box supports);

- A study of the integration of BRPF in federated filter architectures and their application to multi-sensor filtering and collaborative navigation;
- A failure probability estimator based on uncertain trajectory prediction using mixtures of box kernels, and its integration in a differentiable Chance Constrained optimisation scheme for constrained trajectory planning under state uncertainty, called the Box Particle Control (BPC).

The use of box kernels mixtures in estimation algorithms (BPF, BRPF) and state control methods (BPC) makes it possible to reduce the computational load of embedded algorithms. Complex autonomous estimation and control operations can therefore be carried out in real time, while running on computationally limited devices. Furthermore, the future progress in embedded chips (e.g. Field Programmable Gate Arrays) should allow the box particle based approaches to be used in an increasing number of applications. They may present benefits to many industries, such as the aerospace field, automotive engineering, autonomous robotics, or medical applications.

1.4 PUBLICATIONS

The work on which this thesis is based has led the following publications:

Journal article

- **N. Merlinge**, K. Dahia, H. Piet-Lahanier, J. Brusey, N. Horri. 'A Box Regularized Particle Filter for state estimation with severely ambiguous and non-linear measurements'. Submitted to Automatica (2018), decision of updated version submission by October 2018

Conference and workshop proceedings

- **N. Merlinge**, K. Dahia, H. Piet-Lahanier. 'A Box Regularized Particle Filter for terrain navigation with highly non-linear measurements'. In the 20th IFAC Symposium on Automatic Control in Aerospace, ACA 2016, Sherbrooke, Quebec, Canada, IFAC-PapersOnLine 49.17, pp. 361-366.
- **N. Merlinge**, J. Marzat, L. Reboul. 'Optimal guidance and observer design for target tracking using bearing-only measurements'. In the ONERA-DLR Aerospace symposium, ODAS 2018, Oberpfaffenhofen, Germany, `http://w3.onera.fr/copernic/node/20`
- **N. Merlinge**, H. Piet-Lahanier, K. Dahia. 'Event-Triggered Model Predictive Control for target tracking with angles-only measurements'. In the 20th World Congress of the International Federation of Automatic Control, Toulouse, France, IFAC 2017, IFAC-PapersOnLine, 50(1), pp. 15959-15964.
- **N. Merlinge**, K. Dahia, H. Piet-Lahanier, J. Brusey, N. Horri. 'The Box Regularized Particle Filter: A probabilistic set-membership observer'. In the 10th Summer Workshop on Interval Methods, and 3rd International Symposium on Set Membership, Applications, Reliability and Theory, Manchester, United Kingdom, SWIM-SMART 2017, `http://www.aerospace.manchester.ac.uk/our-research/ukim/news-and-events/swim-smart-2017/`
- **N. Merlinge**, J. Brusey, N. Horri, K. Dahia, H. Piet-Lahanier. 'Enhanced cooperative navigation by data fusion from IMU, ambiguous terrain navigation, and coarse relative states'. In the 56th Decision and Control Annual Conference, CDC 2017, Melbourne, Australia, IEEE, pp. 375-380.
- **N. Merlinge**, N. Horri, K. Dahia, H. Piet-Lahanier, J. Brusey. 'Box Particle Control for aerospace vehicles guidance under failure probability constraints', 12th UKACC International Conference on Control, Control 2018, Sheffield, United Kingdom

1.5 THESIS STRUCTURE

The rest of the thesis is arranged as follows:

Chapter 2 presents a literature review on state estimation and state control, and in particular the existing Box Particle Filter (BPF) approach for non-linear state estimation, and the Particle Control approach (PC) for non-Gaussian Chance Constrained trajectory planning (Blackmore [Bla+10]).

Chapter 3 introduces the methodology used to evaluate the algorithm's behaviour. Considered dynamical and observation models are described, with a focus on the TAN application. Finally, some evaluation criteria are defined.

Chapter 4 introduces a new approach to BPF called the Box Regularised Particle Filter (BRPF). To begin, a general theoretical framework to describe the BPF scheme is described. An enhanced box resampling approach is introduced. It guarantees that the estimated confidence region of the filter always contains the actual state. Two box resampling subdivision techniques are subsequently introduced. They are empirically shown to significantly improve the estimation accuracy. Finally, a stochastic kernel smoothing approach is introduced. It is theoretically and empirically shown to improve the estimation accuracy.

Chapter 5 studies the BRPF integration in federated architectures. The asymptotic unimodality of the BRPF output density is discussed and BRPF is integrated in several federated architectures. Simulations illustrate the federated BRPF performance for multi-sensor estimation and collaborative navigation for formation flying. The interest of integrating BRPF in federated architectures is shown in terms of computational load reduction and estimation accuracy.

Chapter 6 introduces a predicted failure probability estimator for Chance Constrained trajectory planning. The approach is based on kernel mixtures bounded by box particles, in a similar framework as in Chapter 4. The box kernel mixture density is propagated along a finite horizon trajectory, which allows the failure probability to be estimated. The approach is shown to outperform the particle-based approach in terms of probability estimation accuracy. Finally, the box failure probability estimator is integrated in a differentiable optimisation scheme for Chance Constrained trajectory planning, leading to the Box Particle Control (BPC) method. BPC is shown to be more robust and more computationally efficient than the original Particle Control (PC). It can also be applied to non-linear dynamics, which is not the case for PC.

Chapter 7 concludes the thesis and discusses possible directions for future work.

2

LITERATURE REVIEW ON STATE ESTIMATION AND STATE CONTROL

This chapter provides the background for the work presented in the subsequent chapters of this thesis. Section 2.1 provides a detailed statement on estimation theory and associated embedded algorithms. The main points are particle filters with kernel estimation (Section 2.1.4) and the Box Particle Filter (Section 2.1.6). Section 2.2 states a framework for collaborative estimation and federated architectures. Section 2.3 provides an overview on constrained optimal control under state uncertainty with a focus on chance constrained Particle Control.

2.1 ESTIMATION THEORY

Consider a system (e.g. an autonomous aircraft) whose evolution is described by a set of functions, explicitly defined or derived from differential equations. These functions correspond to the evolution model which provides a time description of the system's variables of interest (e.g. position, velocity, attitude etc.). These variables constitute the system *state vector*. The system does not usually dispose of the full knowledge of its state vector. Its decision process is only derived from the information available on the whole vector or some of its components. Since the dynamical model function is most often a simplification of reality, the value of the system state vector obtained as the dynamical model output can rapidly drift far from the actual system state vector. In order to limit this derivation, one can use some measurements provided by sensors which are functions of the state vector components. However, these measurements are obtained via imperfect sensors and thus do not give an exact output. The estimation problem then consists of retrieving the system's state vector description, given an approximate dynamical model and noisy measurements. A specific aspect of the estimation problem addressed here is the search for recursive state estimation. At each time-step, the state vector value is derived from the previous state value, through an update performed either by the model or by the measurement model or both. In this section, the problem is first presented in 2.1.1. Then, two different formalisms to model uncertainty are introduced, beginning with the *probabilistic framework* in Section 2.1.2. Under Gaussian assumptions, this framework yields Kalman-like filters (Section 2.1.3). For non-Gaussian or non-linear problems, the Particle Filter (PF [GSS93] was introduced and is presented in Section 2.1.4). However, the probabilistic density functions for involved uncertainties may be unknown, with only some boundaries assumption on their support. This hypothesis yields the *set-based framework* (Section 2.1.5). A focus will be made on intervals and boxes (Interval Analysis framework [Jau09]). Although these two frameworks are not usually mixed, an approach named Box Particle Filter (BPF) was introduced by Gning [AGB07] to take advantage of the robustness of Interval Analysis and the accuracy of the Probabilistic framework. This approach is presented in Section 2.1.6. Finally, Section 2.1.7 gives an overview of estimation methods.

2.1.1 State estimation: problem statement

The state vector $\mathbf{x} \in \mathbb{R}^d$ defines the system at any time. It contains every value useful to describe the system for a given application (e.g. position, velocity etc.). In addition, it is possible to act on the system via a control input $\mathbf{u} \in \mathbb{R}^{d_c}$. The state evolution can be represented by a differential equation:

$$\dot{\mathbf{x}} = f_c(\mathbf{x}, \mathbf{u}) \tag{2.1}$$

where $f_c : \mathbb{R}^d \times \mathbb{R}^{d_c} \rightarrow \mathbb{R}^d$ is the dynamical model.

However, this dynamical model f_c is not an accurate description of the system evolution in the general case (see Jazwinski [Jaz07]). This can be due to unmodeled dynamics, for example the complex aerodynamic effects on an aircraft or structure elasticity, or unknown disturbances such as air pressure variation or wind. These uncertainties can be modeled by an additive process noise $\mathbf{w}$ which is integrated in equation 2.1. Therefore, the state $\mathbf{x}$ becomes a stochastic process.

In addition, a discrete-time state evolution is considered. At each time-step $k \in \mathbb{N}$, variables and vectors are noted with index k. Two consecutive time-steps are separated by a time $dt > 0$.

This manuscript deals with the following state evolution equation:

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \tag{2.2}$$

where $f_k : \mathbb{R}^d \times \mathbb{R}^{d_c} \rightarrow \mathbb{R}^d$ is the discrete dynamical model at time-step k.

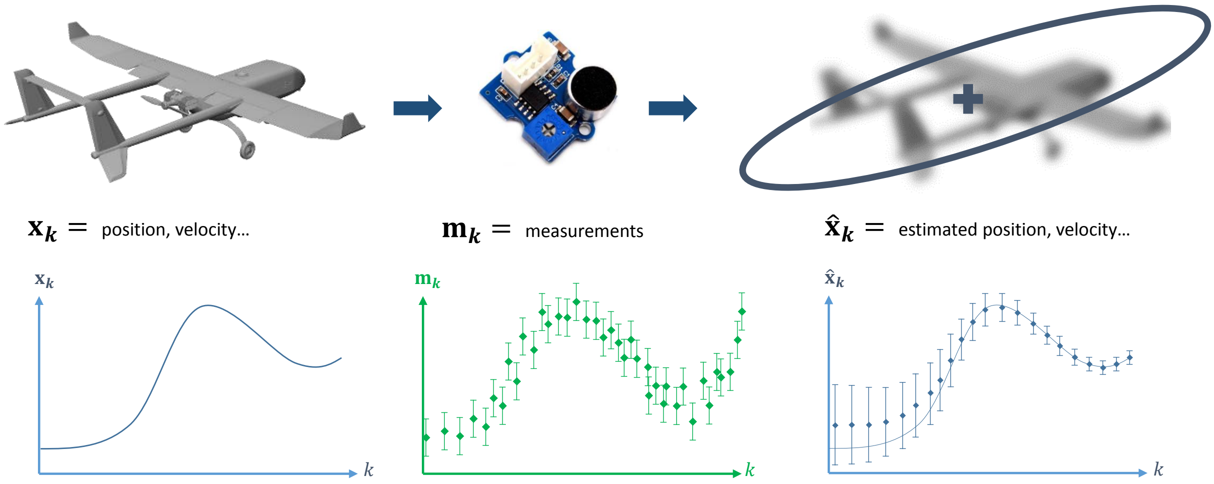
The final objective of the control loop is bringing the state $\mathbf{x}_k$ to a desired state. However, the state is unknown and cannot be directly retrieved due to its initial uncertainty and the process noise. The only available information about $\mathbf{x}_k$ is the measurement vector $\mathbf{m}_k \in \mathbb{R}^{d_m}$ obtained via some sensors. Measurements are linked to the state by the observation equation:

$$\mathbf{m}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k \tag{2.3}$$

where $h_k : \mathbb{R}^d \rightarrow \mathbb{R}^{d_m}$ is the *observation model*. It can be non-linear and may not explicitly involve all the state variables. As sensors are imperfect, measurements must integrate a representation of sensor noise and become stochastic vectors. In this work, sensor noise is modelled by an additive noise $\mathbf{v}_k$. For the sake of brevity, it is assumed that $f_k = f \ \forall k$ and $h_k = h \ \forall k$.

The goal of recursive State Estimation is determining the state at each time-step, given the dynamical model and the measurements. The expected outputs are: the current estimate $\widehat{\mathbf{x}}_k$, and a confidence indicator which quantifies the estimate uncertainty. Figure 2.1 illustrates the State Estimation process.

An iterative estimation algorithm often consists of two steps: the *prediction step*, which accounts for the dynamical transition from time-step $k-1$ to time-step k, and the *correction step* which accounts for the measurements if available, as described by Algorithm 1 (e.g. see Thrun [TBF05]). The state and its associated uncertainty can be described with several frameworks. The probabilistic framework models the noises as random variables following *a priori* known statistical laws, and is introduced in Sections 2.1.2 to 2.1.4. The set-based framework considers uncertainties as *unknown-but-bounded* and is described in Section 2.1.5.



(a) Actual state (b) Measurements (c) Estimate

Figure 2.1: Estimation scheme: an unknown state **x** (a) is estimated via some uncertain measurements **m** (b). The estimate $\widehat{\mathbf{x}}$ is associated with an estimation confidence area which must contain the actual state (c).

Algorithm 1 A general recursive estimator, made of two steps: The prediction step accounts for the dynamics, and the correction step accounts for the measurements.

```
Inputs: initial state density $p(\mathbf{x}_0)$, process noise characteristics, measurement noise characteristics.
Outputs: state estimate $\widehat{\mathbf{x}}_k$ $\forall k$, and an associated uncertainty.
Initialisation: Initialise the state description using the state's initial uncertainty.
for each time-step $k$ do
    Prediction step: propagate the state description from time-step $k-1$ to time-step $k$ using the dynamical model $f$ and the process uncertainty $\mathbf{w}_k$, as described in (2.2).
    Correction step: update the state description using the knowledge of the measurement $\mathbf{m}_k$ and accounting for its uncertainty $\mathbf{v}_k$ and its relationship with the state (2.3).
end for
Return estimate $\widehat{\mathbf{x}}_k$ and an associated uncertainty.
```

2.1.2 Probabilistic framework

The probabilistic framework aims to estimate the most probable current state $\mathbf{x}_k$ given all the past measurements $\mathbf{M}_k = \{\mathbf{m}_1, ..., \mathbf{m}_k\}$. Mathematically, the *most probable state* corresponds to the highest peak of the conditional density $p(\mathbf{x}_k|\mathbf{M}_k)$ of the state given the measurements' trajectory. This idea is illustrated in Figure 2.2.

The conditional state density can be managed in an iterative way by the Chapman-Kolmogorov prediction equation (2.4), and the Bayes' rule correction (2.5). These equations form a theoretical filter named Optimal Filter. It represents the most accurate way of describing the conditional density evolution, given a trajectory of measurements and their associated uncertainties.

In order to determine the conditional state density at current time-step, three other densities must be known:

- The current measurement density $p(\mathbf{m}_k|\mathbf{x}_k)$, which quantifies the distribution of the measurement, conditionally to the actual state,
- The state transition $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ which accounts for non-modeled dynamics,
- The initial state density is denoted $p(\mathbf{x}_0)$, which quantifies the initial state uncertainty.

However, the Optimal Filter is not directly implementable in practice. In order to produce tractable algorithms, several assumptions can be made, leading to a variety of estimation methods, as illustrated in Figure 2.2.

The Optimal Filter framework is first introduced in what follows (page 10). Several estimation methods derived from Optimal Filter are then introduced, such as the Kalman Filter (Linear Gaussian hypothesis, page 13), the Extended Kalman Filter (Non-linear Gaussian hypothesis, page 15), with a focus on the Particle Filter (Non-linear and non-Gaussian hypothesis, page 17). More advanced state density estimation methods, such as kernel estimation (page 21) can be used to improve Particle Filter, leading to the Regularised Particle Filter (RPF, page 23).

The theoretical optimal filter

This section introduces the theoretical iterative formulation for a state density prediction and correction. A *filter* will refer to an algorithm that estimates the state density for the current time-step. A filter is said to be *optimal* if it is theoretically equivalent to the Optimal Filter, under given hypotheses. Filters are often coupled to an *estimator* that derive a point-wise estimate from the estimated density. At each time-step k, the *prior state density* $p(\mathbf{x}_{k-1}|\mathbf{M}_{k-1})$ is convoluted with the process noise density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ by the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k|\mathbf{M}_{k-1}) = \int_{\mathbb{R}^d} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{M}_{k-1}) \mathrm{d}\mathbf{x}_{k-1} \tag{2.4}$$

This yields the *predicted state density* $p(\mathbf{x}_k|\mathbf{M}_{k-1})$, illustrated in Figure 2.3 (a). When a new measurement $\mathbf{m}_k$ is available, it is associated with a sensor noise density $p(\mathbf{m}_k|\mathbf{x}_k)$ conditionally to the current state, as illustrated in Figure 2.3 (b). The hypothesis is made that all measurements are independent and that the state evolution process is Markovian, i.e. the state $\mathbf{x}_k$ only depends on the previous state $\mathbf{x}_{k-1}$ (Norris [Nor98]). Then, the *posterior state density* is obtained by the Bayes' rule:

$$p(\mathbf{x}_k|\mathbf{M}_k) = \frac{1}{q_k} p(\mathbf{m}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{M}_{k-1}) \tag{2.5}$$

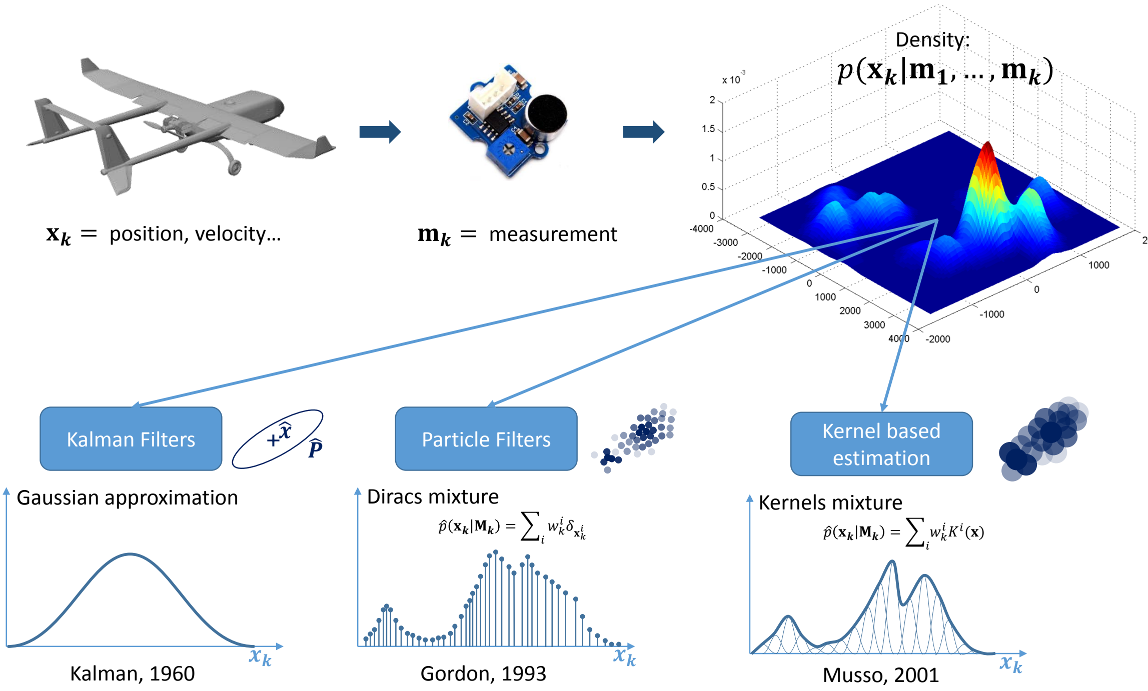


Figure 2.2: Probabilistic estimation scheme and three different state density approximations: Gaussian filtering (e.g. Kalman Filter), Dirac mixture (e.g. Particle Filter), and Kernel mixture (e.g. Regularised Particle Filter)

With $q_k = p(\mathbf{m}_k|\mathbf{M}_{k-1}) = \int p(\mathbf{m}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{M}_{k-1})\mathrm{d}\mathbf{x}_k$. The updating step (or correction step) is illustrated in Figure 2.3 (c).

The estimate is the blue cross in Figure 2.3. The estimated covariance is the blue ellipse around the estimate. The next section introduces three common approaches to determine a point-wise estimate from the involved densities: the Maximum A Posteriori, the Maximum Likelihood, and the Lest Squares estimators.

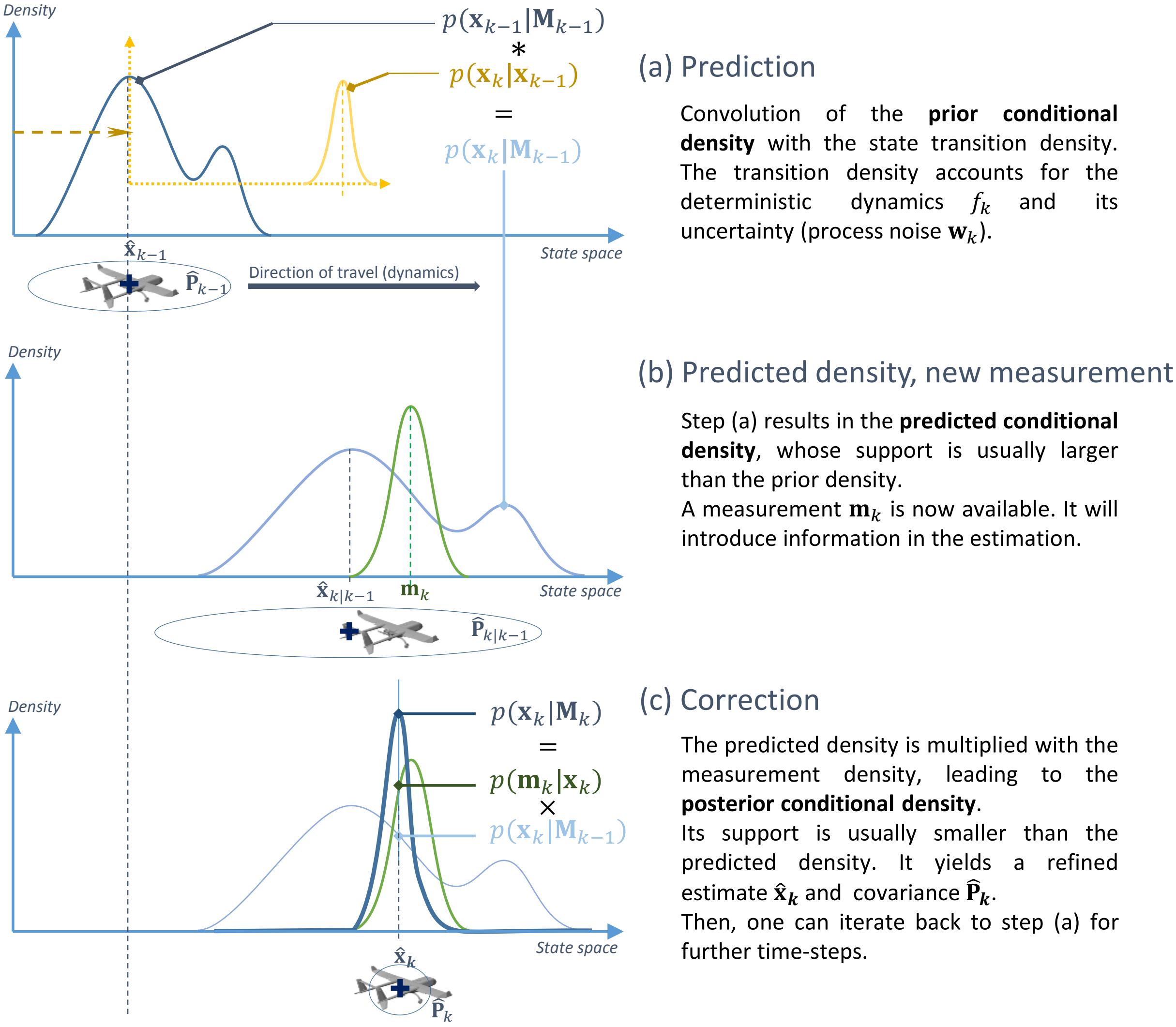
State estimators

The Optimal Filter, introduced in the previous section, is the exact formulation for the iterative estimation of a conditional state density, given all past measurements. However, for many applications, a point-wise state estimate is needed. In what follows, *estimator* refers to a method that derives a point-wise estimate $\widehat{\mathbf{x}}_k \in \mathbb{R}^d$ from a density. A confidence region can also be derived, usually defined as the variance of the estimator. Three estimators are commonly used:

- The **Maximum A Posteriori (MAP)** estimator (Gauvain [GL94]), defined by the state that maximises the posterior conditional density $p(\mathbf{x}_k|\mathbf{M}_k)$, illustrated in Figure 2.4 (a):

$$\widehat{\mathbf{x}}_k = \underset{\mathbf{x}_k}{\operatorname{argmax}}\,(p(\mathbf{x}_k|\mathbf{M}_k)) \tag{2.6}$$

  If the posterior conditional density is unimodal, i.e. it has a unique global maximum, the MAP can be considered as the optimal estimator in the sens that it provides the most probable state from the Optimal Filter density. If there is no prior knowledge, the predicted density $p(\mathbf{x}_k|\mathbf{M}_{k-1})$ is said to be *diffuse* and is associated with an unbounded uniform density on $\mathbb{R}^d$. The posterior density



(a) Prediction

Convolution of the **prior conditional density** with the state transition density. The transition density accounts for the deterministic dynamics f_k and its uncertainty (process noise $\mathbf{w}_k$).

(b) Predicted density, new measurement

Step (a) results in the **predicted conditional density**, whose support is usually larger than the prior density.
A measurement $\mathbf{m}_k$ is now available. It will introduce information in the estimation.

(c) Correction

The predicted density is multiplied with the measurement density, leading to the **posterior conditional density**.
Its support is usually smaller than the predicted density. It yields a refined estimate $\hat{\mathbf{x}}_k$ and covariance $\hat{\mathbf{P}}_k$.
Then, one can iterate back to step (a) for further time-steps.

Figure 2.3: Probabilistic iterative prediction-correction scheme. The prediction step (a) yields an inaccurate estimate associated with a larger covariance (b). The measurement introduces some information, which refines the estimate and the covariance (c). The actual state is illustrated by an aircraft. The estimate (e.g. the aircraft's center of mass) is the blue cross. The estimated covariance is the blue ellipse around the estimate.

only depends on the likelihood, $p(\mathbf{x}_k|\mathbf{M}_k) \propto p(\mathbf{m}_k|\mathbf{x}_k)$, and the MAP is equivalent to maximising the likelihood.

- The **Maximum Likelihood (ML)** estimator (e.g. see Wasserman [Was13]), defined by the state that maximises the measurement density $p(\mathbf{m}_k|\mathbf{x}_k)$.

$$\widehat{\mathbf{x}}_k = \underset{\mathbf{x}_k}{\operatorname{argmax}} \left(p(\mathbf{m}_k|\mathbf{x}_k) \right) \tag{2.7}$$

illustrated in Figure 2.4 (b). In practice, the MAP is often difficult to derive and can be approached by the ML estimator. If the likelihood is Gaussian, a simpler estimator can be derived from the ML estimator: the Least Squares estimator.

- The **Least Squares (LS)** estimator (e.g. see Lee [Lee05]) consists of maximising a Gaussian Likelihood $p(\mathbf{m}_k|\mathbf{x}_k) \propto \exp\left(-\frac{1}{2}\mathbf{\Delta x}^T\mathbf{R}^{-1}\mathbf{\Delta x}\right)$. It is equivalent to minimising the quantity $\Delta\mathbf{x}^T\mathbf{R}^{-1}\mathbf{\Delta x}$, where $\mathbf{\Delta x} \triangleq \mathbf{m}_k - h(\mathbf{x}_k)$ and $\mathbf{R} > 0$, as illustrated in Figure 2.4 (c). Then, the Least Squares estimator can be written as: $\widehat{\mathbf{x}}_k = \underset{\mathbf{x}_k}{\operatorname{argmin}} \left(\Delta\mathbf{x}^T\mathbf{R}^{-1}\mathbf{\Delta x}\right)$. It can be generalized to N independent measurements $\mathbf{m}_k^i \in \mathbb{R}^{d_m}$ of covariance $\mathbf{R}^i > 0$, which yields then the solution:

$$\widehat{\mathbf{x}}_k = \left(\sum_{i=1}^{N} \mathbf{H}^T {\mathbf{R}^i}^{-1} \mathbf{H}\right)^{-1} \left(\sum_{i=1}^{N} \mathbf{H}^T {\mathbf{R}^i}^{-1} \mathbf{m}_k^i\right) \tag{2.8}$$

It corresponds to the average of all measurements, weighted by the inverse of their covariance, in other words, their information (see Appendix A). Its variance is about $\mathrm{Var}_{LS} = \frac{1}{N}$, which is greater than the MAP variance ($\frac{1}{N^2}$). Nevertheless, LS estimator is of strong practical interest due to its low computational cost and is often used to approach the MAP and ML estimators.

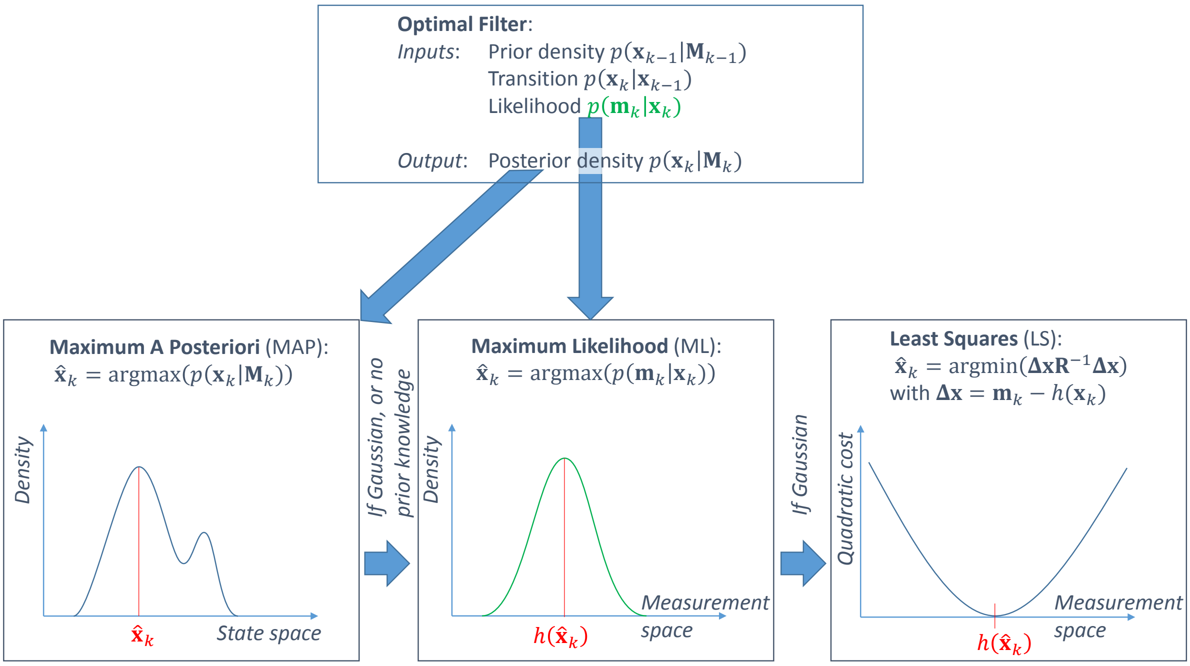
The Optimal Filter and associated estimators can be derived given some assumptions on uncertainties and models, yielding a variety of estimation algorithms (e.g. Kalman Filter, Particle Filter). In what follows, $\widehat{p}(\mathbf{x}|\mathbf{M}))$ is denoted as the estimated conditional density, approximating the Optimal Filter's density $p(\mathbf{x}_k|\mathbf{M}_k)$. The next sections start with the linear Gaussian case, leading to the Kalman Filter. These hypotheses are then gradually relaxed to introduce more general models and the corresponding estimation methods. A focus on the Particle Filter is finally done.

2.1.3 Gaussian filters

This section presents some derivation of the Optimal Filter in the Gaussian case. This yields the Kalman filters.

Linear Gaussian Hypothesis: the Kalman Filter

If the problem can be formulated with a linear dynamics and a linear dependency of the measurement with respect to the state, and if all involved density are Gaussian, the Optimal Filter yields the Kalman Filter (KF) equations [Kal+60]. Under these hypotheses, the Kalman Filter is the optimal solution to the estimation problem. This section recalls the KF equations and its main properties.



(a) Maximum A Posteriori (b) Maximum Likelihood (c) Least Squares

Figure 2.4: Maximum A Posteriori, Maximum Likelihood, and Least Squares estimators. These methods are used to derive a point-wise state estimate $\hat{\mathbf{x}}_k$ from the Optimal Filter's estimated state density. MAP, ML, and LS estimators are equivalent in the Gaussian case, and LS is derived from ML.

The state and measurements' Linear Gaussian models are defined by:

$$\begin{cases} \mathbf{x}_k & = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_k + \mathbf{w}_k \\ \mathbf{m}_k & = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \end{cases} \tag{2.9}$$

where $\mathbf{F}_{k-1}$, $\mathbf{B}_{k-1}$, $\mathbf{H}_k$ are known matrices, and:

$$\begin{cases} \mathbf{w}_k \sim \mathcal{N}\left(\mathbf{0}_d, \mathbf{Q}_k\right) \\ \mathbf{v}_k \sim \mathcal{N}\left(\mathbf{0}_{d_m}, \mathbf{R}_k\right) \\ \mathbf{x}_0 \sim \mathcal{N}\left(\mathbf{s}_0, \mathbf{P}_0\right) \end{cases} \tag{2.10}$$

where $\mathbf{Q}_k \in \mathbb{R}^{d\times d}$ is the process covariance, $\mathbf{R}_k \in \mathbb{R}^{d_m\times d_m}$ the observation covariance, $\mathbf{s}_0 \in \mathbb{R}^d$ the initial guess for the state value, and $\mathbf{P}_0 \in \mathbb{R}^{d\times d}$ the initial state covariance.

Under these hypotheses, it can be shown that the theoretical conditional density $p(\mathbf{x}_k|\mathbf{M}_k)$ remains Gaussian and can be fully described by its expectancy $\mathbb{E}\left[\mathbf{x} \sim p(\mathbf{x}_k|\mathbf{M}_k)\right]$ and its covariance $\mathrm{Cov}\left[\mathbf{x} \sim p(\mathbf{x}_k|\mathbf{M}_k)\right]$. As a result:

- The estimated density $\widehat{p}(\mathbf{x}_k|\mathbf{M}_k) = \mathcal{N}(\widehat{\mathbf{x}}_k, \widehat{P}_k)$ is always equal to the theoretical density $p(\mathbf{x}_k|\mathbf{M}_k)$ derived from the optimal filter equations,
- Its expectancy $\widehat{\mathbf{x}}_k = \mathbb{E}\left[\mathbf{x} \sim \widehat{p}(\mathbf{x}_k|\mathbf{M}_k)\right]$ is equal to the theoretical Maximum A Posteriori (MAP) estimator (2.6), which is equivalent to the Maximum Likelihood (ML) estimator (2.7) and the Least Squares estimator (2.8).
- The KF is said *minimum variance*, i.e. its covariance $\widehat{P}_k = \mathrm{Cov}\left[\mathbf{x} \sim p(\mathbf{x}_k|\mathbf{M}_k)\right]$ is equal to the variance of the Optimal Filter's posterior density. It is also equal to the MAP variance, the ML variance, and the inverse of Fisher Information Matrix (FIM, see Appendix A). It is interesting to note that this minimum variance property is kept if the process and/or measurements noises are non-Gaussian (with linear dynamics and measurements).

The resulting Kalman Filter's equations are presented in Algorithm 2. A detailed description of the derivation of the KF's equations from the optimal filter can be found in Jazwinski [Jaz70] or in Dahia [Dah05, Appendix B].

Non-linear Gaussian Hypothesis: Extended Kalman Filter (EKF) and beyond

The Kalman filter has been extended to non-linear dynamical models and measurement equations:

$$\begin{cases} \mathbf{x}_k & = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \\ \mathbf{m}_k & = h(\mathbf{x}_k) + \mathbf{v}_k \end{cases} \tag{2.14}$$

with the same Gaussian uncertainties densities as previously (2.10).

Functions $f : \mathbb{R}^d \times \mathbb{R}^{d_c} \to \mathbb{R}^d$ and $h : \mathbb{R}^d \to \mathbb{R}^{d_m}$ can be locally linearized at each time-step k, which yields the gradient matrices:

$$\begin{aligned} \mathbf{F}_k &= \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\widehat{\mathbf{x}}_k} \in \mathbb{R}^{d\times d} \\ \mathbf{H}_k &= \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\widehat{\mathbf{x}}_k} \in \mathbb{R}^{d_m\times d} \end{aligned} \tag{2.15}$$

Algorithm 2 Kalman Filter

```
    Inputs: initial state density $\mathcal{N}(\mathbf{s}_0, \mathbf{P}_0)$, process noise density $\mathcal{N}(\mathbf{0}_{d\times 1}, \mathbf{Q}_k)$, measurement density $\mathcal{N}(\mathbf{0}_{d_m\times 1}, \mathbf{R}_k)$.
    Outputs: state estimate $\widehat{\mathbf{x}}_k \ \forall k$, estimated covariance $\widehat{\mathbf{P}}_k \ \forall k$.
Initialisation: the initial estimate is $\widehat{\mathbf{x}}_0 = \mathbf{s}_0$, the initial covariance is $\mathbf{P}_0$.
for each time-step $k$ do
    Prediction step:
        $\widehat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\widehat{\mathbf{x}}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_k$
        $\widehat{\mathbf{P}}_{k|k-1} = \mathbf{F}_{k-1}\widehat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_k$        (2.11)
    Correction step:
        $\widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{m}_k - \mathbf{H}_k\widehat{\mathbf{x}}_{k|k-1}\right)$
        $\widehat{\mathbf{P}}_k = \left(\mathbf{I}_d - \mathbf{K}_k\mathbf{H}_k\right)\widehat{\mathbf{P}}_{k|k-1}$        (2.12)
    where $\mathbf{K}_k$ is the Kalman gain defined by:
        $\mathbf{K}_k = \widehat{\mathbf{P}}_{k|k-1}\mathbf{H}_k^T \left(\mathbf{R}_k + \mathbf{H}_k\widehat{\mathbf{P}}_{k|k-1}\mathbf{H}_k^T\right)^{-1}$        (2.13)
end for
Return $\widehat{\mathbf{x}}_k, \widehat{\mathbf{P}}_k \ \forall k$.
```

This results in the Extended Kalman Filter (EKF). However, the filter's convergence is conditioned by the manual tuning (usually enlarging) of the process and measurements covariance matrices. Some theoretical works have been done to quantify the impact of initial uncertainty and tuning on the filter's convergence. It results that the EKF can converge for tiny initial errors only (Karvonen [Kar+14]). In practice, EKF remains of great interest for many applications where the dynamical model and the measurements equation are only slightly non-linear, for example for an inertial sensor update via GNSS navigation.

However, EKF may fails to converge when non-linearities are more severe or when the initial error is too important. The Unscented Kalman Filter (UKF [WVDM00]) has been proposed to make it more robust. UKF consists of describing the Gaussian state density approximation by a set of deterministically chosen sigma-points in the state space (in practice, between four and ten points). Each sigma point is independently propagated through the non-linear dynamics, and corrected through the non-linear observation equation. Finally, the Gaussian state density is computed by a weighted sum of the transformed sigma-points, which makes it possible to compute the state density using several local linearisations. It results in a more robust algorithm to non-linearity, eventhough its convergence has not been proved in the general case (Karvonen [Kar+14]). Linearisations in Gaussian filters introduce errors that cannot be formalized in a general way. Furthermore, non-linear transformations of Gaussian densities do not preserve the Gaussian quality. Two main approaches have been proposed in order to account for these statements.

The first approach consists of bringing back non-linearities into a linear scheme by performing a mapping from the state and measurement euclidean spaces to a new non-euclidean space in which the equations become linear, e.g. Lie groups [Olv00]. This idea led to several designs of estimators (e.g. on the Special Euclidean Group [Hua+11]). Adaptations of Kalman filters to Lie groups have thus been proposed for the EKF in Markley [Mar03] or the UKF in Crassidis [CM03]. This yields to new convergence and stability results, which is an active subject of research, e.g. with the Invariant EKF [BB17]. These non-euclidean filters are useful for estimation problems that can be formulated in Lie-Groups (e.g. attitude estimation, on the Lie groups of rotation matrices SO(3) (Special Orthogonal group), or the quaternion group). However, they are not universal in terms of model. For example, they may fail to tackle non-analytical non-linear measurements (e.g. for the terrain navigation problem, see 3.2). Furthermore, they are restricted to a known parametric family of densities (e.g. Gaussian densities in the aforementioned algorithms).

Another approach is the Particle Filter, which consists of stochastically approximating the whole state density with a representative sample of possible states. It is able to tackle model non-linearities and is not limited to Gaussian densities. The Particle Filter is introduced in the next section.

2.1.4 Non-linear non-Gaussian filters: particle filters and kernel estimation

This section presents a way of deriving the Optimal Filter in non-linear and non-Gaussian cases by using mixtures of kernels. The Particle Filter scheme is first introduced (Gordon [GSS93]), which approximates the state density with a mixture of weighted Dirac deltas. The Regularised Particle Filter (RPF, [MOLG01]) is then introduced. It is based on kernel estimation approaches (Silverman, [Sil86]) which brings more accuracy by considering mixtures of weighted bounded kernels.

The Sequential Importance Resampling Particle Filter

The Sequential Importance Resampling Particle Filter approach (SIR-PF [GSS93]) was introduced to tackle non-linear dynamics and/or severely non-linear measurements. Uncertainties may also be non-Gaussian. PF consists of empirically rebuilding the conditional state density by a set of weighted points

$\{\mathbf{x}_k^i \in \mathbb{R}^d, w_k^i \in \mathbb{R}^+\}_{i\in[1,N]}$ called particles. They are associated with positive weights w_k^i whose sum is equal to 1. The estimated conditional density at time-step $k-1$ is thus defined by a mixture of weighted Dirac deltas, as illustrated in Figure 2.5.

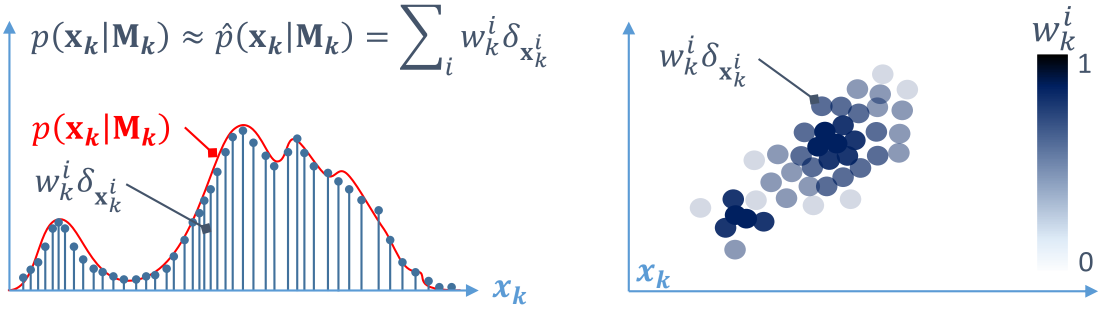


Figure 2.5: Particle Filter scheme: the conditional density is approximated with a mixture of diracs corresponding to a cloud of weighted particles.

The estimated prior conditional density is defined by:

$$\widehat{p}(\mathbf{x}_{k-1}|\mathbf{M}_{k-1}) \triangleq \sum_{i=1}^{N} w_{k-1}^i \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^i) \tag{2.16}$$

where δ is the Dirac delta functions on $\mathbb{R}^d$ equal to 0 on $\mathbb{R}^d \setminus \{\mathbf{0}_d\}$ and to infinity on $\mathbf{0}_d$. By the application of the Chapman-Kolmogorov equation (2.4), one can approximate the predicted density:

$$\widehat{p}(\mathbf{x}_k|\mathbf{M}_{k-1}) = \sum_{i=1}^{N} w_{k-1}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{2.17}$$

where $\{\mathbf{x}_k^i\}$ is the set of predicted particles. Each propagated particle $\mathbf{x}_k^i$ is obtained by a deterministic dynamical propagation using model f, plus a random sample $\mathbf{w}_k^i$ of the process noise:

$$\mathbf{x}_k^i = f(\mathbf{x}_{k-1}^i, \mathbf{u}_k) + \mathbf{w}_k^i \tag{2.18}$$

The process noise density is assumed to be known. By the application of the Bayes' rule (2.5), the updated conditional density is approximated by a Dirac mixture with the same centers $\{\mathbf{x}_k^i\}$, but with updated weights:

$$\widehat{p}(\mathbf{x}_k|\mathbf{M}_k) = \sum_{i=1}^{N} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{2.19}$$

with

$$w_k^i = w_{k-1}^i \frac{1}{q_k} \pi^m(\mathbf{m}_k - h(\mathbf{x}_k^i)) \tag{2.20}$$

where $\pi^m : \mathbb{R}^{d_m} \rightarrow \mathbb{R}^+$ is the analytic measurement noise density $p(\mathbf{m}_k|\mathbf{x}_k)$. The normalisation term q_k is:

$$q_k = \sum_{i=1}^{N} w_{k-1}^i \pi^m(\mathbf{m}_k - h(\mathbf{x}_k^i)) \tag{2.21}$$

and is equal to the quotient term in (2.5).

Formulation (2.19) provides an estimation of the posterior conditional state density. For practical use, one can then derive a point-wise state estimate, associated with a confidence (e.g. the covariance of the random vector associated with the estimated density). The Maximum A Posteriori (MAP, see Section 2.1.2) can be numerically computed, either by weight selection (Dore [DRM07]), or by clustering computation (Cheng [Che95], Murangira [MMD16]). Nevertheless, MAP computation often yields a high computational load. In practice, the estimate is often computed via a Least Squares approximation (LS, see Section 2.1.2), which corresponds to the barycentre of the weighted particle cloud, under the assumption that the actual state density tends to a Gaussian function. The state estimate is then computed by (2.23) and its covariance by (2.24).

The iterative updating process (2.20) asymptotically leads to disparate values of the weights. Since the weights are normalised, one weight will tend to one while all other weight tend to zero. This is called the *weighting degeneracy phenomenon*. In order to keep a representative set of particles, a *resampling step* was introduced (Gordon [GSS93]). It consists of replacing the current set of particles with a new one which describes the state density in a more accurate way in the neighbourhood of currently high-weighted particles. The number of particles usually remains the same, although some adaptive resampling methods have been proposed to make the number of particles decrease over time, based on some convergence criteria, for example, the Kullback-Leibler Distance sampling (KLD-sampling [Fox02]).

A large variety of sampling and resampling algorithms can be used in the context of PF. The most commonly used is *Multinomial Resampling* (see Douc [DC05] and Li [LBD15] for a review on resampling methods). Generally speaking, a resampling algorithm provides an integer value indicating how many duplications of each particle will occur in the new sample: 0 if the particle is killed, 1 if it is kept, and n if it is duplicated into n instances.

A brief description of Multinomial Resampling is provided here. This algorithm defines the number of duplications of each particle with regard to the value of the weights. It is achieved by drawing N independent values $\{u^i\}_{i\in[1,N]}$ following a uniform law on $[0,1]$ and comparing them to the cumulative sum of the weights (corresponding to their cumulative density). A duplication number n^i can be computed for each particle, corresponding to the number of uniform values lying into their cumulative interval. Large weights will induce a greater number of duplications for their particles than for low-weighted ones. Multinomial Resampling is summarized in Algorithm 3. After resampling, weights are usually reset to $1/N$.

Algorithm 3 Multinomial resampling

```
Input: particle weights $\{w^i\}_{i\in[1,N]}$
Output: number of new instances per particles $\{n^i\}_{i\in[1,N]}$
Initialise the duplication counters to $n^i = 0 \ \forall i \in [1,N]$
for $i = 1$ to $N$ do
    Draw $u^i \sim \mathcal{U}_{[0,1]}$
    Find $j \in [1,N]$ such that $u^i \in \left]\sum_{l=1}^{j-1} w^l, \sum_{l=1}^{j} w^l\right]$
    Count $n^j = n^j + 1$
end for
Return $n^i \ \forall i \in [1,N]$
```

It is generally not necessary to resample at each time-step. Indeed, since the resampling aims to limit degeneracy, it should only be performed when this phenomenon arises. Several triggering criteria have been proposed to detect degeneracy. The most commonly used is Kong and Liu criterion [KLW94], based on the variance of the weights:

$$N_{eff} = \frac{1}{\sum_i \left(w_k^i\right)^2} \tag{2.22}$$

The resampling is triggered if $N_{eff} < \theta_{eff} N$ with $\theta_{eff} \in [0,1]$ being tuning parameter. In practice, it is often set between 0.3 and 0.7 for conventional PFs. Other criteria can also be used, for example Pham's criterion [Pha01], based on the weights' entropy.

Whatever be the the triggering criterion, particle filters using a resampling step belong to the *Sequential Importance Resampling Particle Filters* category (SIR-PF). The generic SIR-PF algorithm is summarized in Algorithm 4 and illustrated in Figure 2.6. Under the hypothesis of an observable system, some proof of the almost sure convergence of the estimated density $\widehat{p}$ toward the optimal filter density p has been provided (Crisan [CD02]).

Algorithm 4 Sequential Importance Resampling Particle Filter

```
    *Inputs*: initial state density $p(\mathbf{x}_0)$, process noise density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, measurement density $p(\mathbf{m}_k|\mathbf{x}_k)$.
    *Outputs*: state estimate $\widehat{\mathbf{x}}_k\ \forall k$, estimated covariance $\widehat{\mathbf{P}}_k\ \forall k$.
1:  **Initialisation**: The initial particle set (or particle cloud) is drawn as $\{\mathbf{x}_0^i, w_0^i\}_{i\in[1,N]}$ such that the initial estimated density (2.19) approaches $p(\mathbf{x}_0)$.
2:  **for** each time-step $k$ **do**
3:      **Prediction step**: Propagate particles using the dynamical model $\mathbf{x}_k^i = f(\mathbf{x}_{k-1}^i, \mathbf{u}_k) + \mathbf{w}_k^i$ and the process noise density $\mathbf{x}_k^i \sim p(\mathbf{x}_k|\mathbf{x}_{k-1})$.
4:      **Correction step**: Update weights using the measurement noise density $w_k^i = \frac{1}{q_k} w_{k-1}^i \pi^m(\mathbf{m}_k - h(\mathbf{x}_k^i))$ where $\pi^m = p(\mathbf{m}_k - h(\mathbf{x}_k)|\mathbf{x}_k)$, and $q_k = \sum_{i=1}^N w_{k-1}^i \pi^m(\mathbf{m}_k - h(\mathbf{x}_k^i))$.
5:      Compute estimate $\mathbb{E}\left[\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)\right]$. It can be approximated by an empirical Least Squares estimator (i.e. a barycentre) :
            $\widehat{\mathbf{x}}_k = \sum_i w_k^i \mathbf{x}_k^i$        (2.23)
6:      Compute a confidence indicator, e.g. the covariance $\mathrm{Cov}\left[\mathbf{x}_k \sim \widehat{p}(\mathbf{x}_k|\mathbf{M}_k)\right]$. It can be approximated by:
            $\widehat{\mathbf{P}}_k = \sum_i w_k^i (\mathbf{x}_k^i - \widehat{\mathbf{x}}_k)(\mathbf{x}_k^i - \widehat{\mathbf{x}}_k)^T$        (2.24)
7:      **if** a resampling criterion is satisfied, e.g. $N_{eff} < \theta_{eff} N$, see (2.22) **then**
8:          Draw a new set of particles $\{\mathbf{x}_k^i, w_k^i\}_{i\in[1,N]}$ using a resampling method, e.g. the multinomial resampling (Algorithm 3).
9:          Reset all weights to $1/N$.
10:     **end if**
11: **end for**
12: Return $\widehat{\mathbf{x}}_k, \widehat{\mathbf{P}}_k\ \forall k$.
```

However, although SIR-PF algorithms are able to tackle a larger variety of problems than parametric filters, they are often limited in practice by the discrete characteristics of the density approximation.

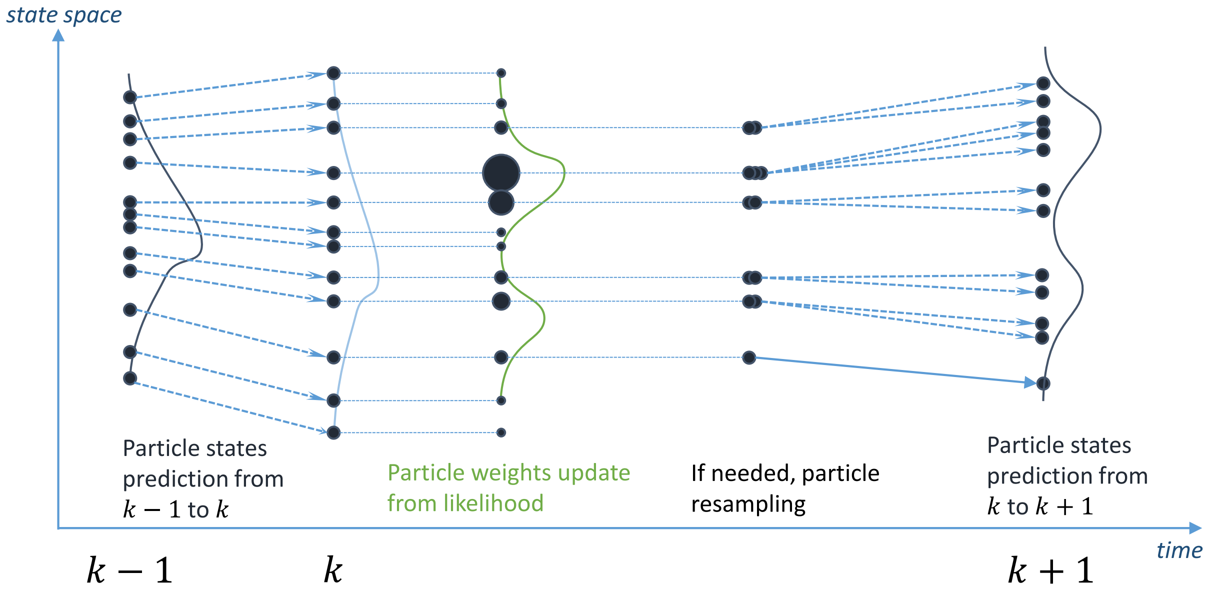


Figure 2.6: Sequential Importance Resampling Particle Filter scheme

Indeed, if the process noise is too low, the resampling will tend to select only a few particles and will asymptotically lead to N copies of a single particle. The estimate is thus not guaranteed to remain close to the actual state, which often results in a filter non-convergence (see Section 3.5).

Furthermore, if the measurement noise is too low, the actual conditional density may concentrate in an area lacking of particles, which leads to ill estimated weights and further non-convergence. Some derivations of the SIR-PF have been introduced to make it more stable. The Marginalised Particle Filter (MPF [CR96]), also referred to as Rao-Blackwellized Particle Filter, tackles a class of estimation problems where the state vector can be split in two part, namely the linear state and the non-linear state. The assumption is made that the linear state conditionally depends on the non-linear state in a linear-Gaussian way. The linear state can then be estimated via a Kalman Filter coupled to a Particle Filter which only tackles the non-linear part of the state. It results in a higher robustness, since particle diracs are maginalized on the non-linear dimensions only, which limits the *curse of dimensionality* (see Giraud [Gir14]). However, marginalisation is not always possible in the general case. To tackle a larger class of problems, the Regularised Particle Filter (RPF [MOLG01]) was introduced, based on the *kernel estimation* theory of Silverman [Sil86]. The idea of *kernel estimation* is associating a non-zero volume support to each particle so that the algorithm can use the particle to explore a complete neighbourhood of the state space. The next sections introduce the kernel estimation concept and its application to SIR-PF.

Kernel density estimation

This section introduces the kernel estimation scheme, which aims to estimate a density $p(\mathbf{x})$ from a sample of data $\{\mathbf{x}^i\}_{i\in[1,N]}$ (e.g. particles). A common way of approximating a density is the use of a histogram, which consists of paving the state space in a number of intervals (bins) associated with the frequency of samples occurrence in their domain, as illustrated in Figure 2.7 (a).

Silverman [Sil86] introduced a more accurate method called *kernel estimation* to tackle this issue. A *probabilistic kernel* is a function from $\mathbb{R}^d$ to $\mathbb{R}$ which integrates to unity on $\mathbb{R}^d$. The main idea is associating a probabilistic kernel function to each sample $\mathbf{x}^i$. The density to estimate is approached by a weighted sum of probabilistic kernels, i.e. a mixture of kernels:

$$\widehat{p}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} K_h(\mathbf{x} - \mathbf{x}^i) \tag{2.25}$$

where K_h is a kernel defined by

$$K_h(\mathbf{x}) = \frac{1}{h^d} K\left(\frac{1}{h}\mathbf{x}\right) \tag{2.26}$$

with K a reference kernel from $\mathbb{R}^d$ to $\mathbb{R}$ such that $\int_{\mathbb{R}} K(x)dx = 1$ and $h > 0$ a bandwidth.

A first version of the kernel estimator was introduced with uniform kernels, referred to the *naive estimator* (Silverman [Sil86], Figure 2.7 (b)). A continuous approximation of the density can be obtained by setting K_h to a smooth kernel, as illustrated in Figure 2.7 (c). The free parameters of kernel estimation are the definition of kernel function K and the value of the bandwidth h.

The Mean Integrated Square Error (MISE) metric can be used to quantify the kernel estimator estimated density $\widehat{p}$ fitting with the actual density p:

$$\text{MISE}(\widehat{p}, p) = \mathbb{E}\left[\int_{\mathbb{R}^d} (\widehat{p}(\mathbf{x}) - p(\mathbf{x}))^2 \, \mathrm{d}\mathbf{x}_k\right] \tag{2.27}$$

An analytic approximation of the MISE can be derived for given kernel K of bandwidth h, in $\mathbb{R}^d$ (Silverman [Sil86]):

$$\text{MISE}(\widehat{p}, p) = \frac{h^4}{4}\alpha^2 \int_{\mathbb{R}^d} (\nabla^2 p(\mathbf{x}))^2 \mathrm{d}\mathbf{x} + \frac{\beta}{Nh^d} \tag{2.28}$$

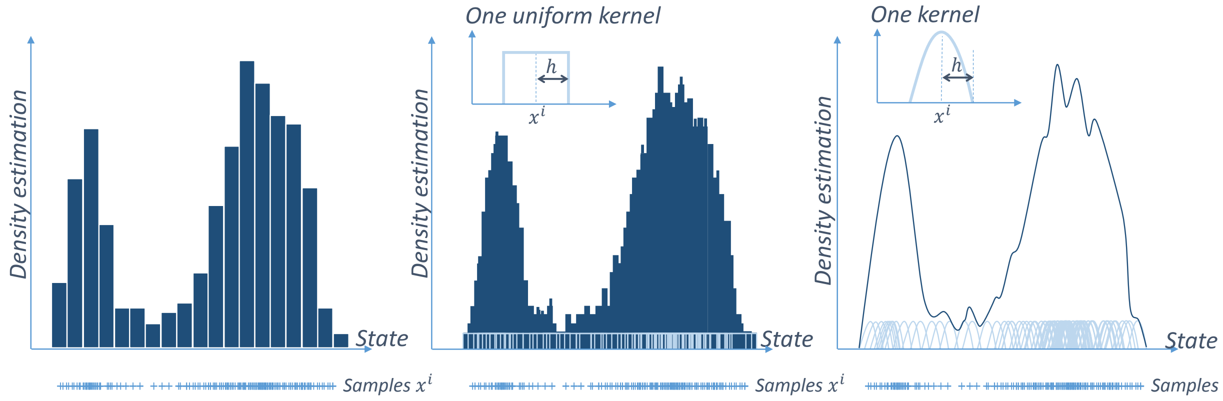
where $\nabla^2 p = \frac{\mathrm{d}^2 p}{\mathrm{d}\mathbf{x}^2}$ and

$$\begin{aligned} \alpha &= \int_{\mathbb{R}^d} x_1^2 K(\mathbf{x}) \mathrm{d}\mathbf{x} \\ \beta &= \int_{\mathbb{R}^d} K(\mathbf{x})^2 \mathrm{d}\mathbf{x} \end{aligned} \tag{2.29}$$

with $x_1 \in \mathbb{R}$ the first component of $\mathbf{x}$. For a given kernel K, the optimal bandwidth h in terms of MISE is:

$$h_{opt} = \left(d\beta\alpha^{-2}\left(\int_{\mathbb{R}^d} (\nabla^2 p(\mathbf{x}))^2 \mathrm{d}\mathbf{x}\right)^{-1} N^{-1}\right)^{\frac{1}{d+4}} \tag{2.30}$$

In practice, kernel estimation can be used in two drastically different ways. The first is an explicit probability density function approximation based on (2.25), which can further be generalized as a weighted mixture of kernels: $\widehat{p}(\mathbf{x}) = \sum_{i=1}^{N} w^i K_h(\mathbf{x} - \mathbf{x}^i)$ with $\sum_i w^i = 1$. This kind of methods can be linked to the Box Particle Filter (Gning [AGB07]), introduced in section 2.1.6. Some approaches have also introduced Gaussian mixture (e.g, Kotecha [KD03], Nemeth [NFM16]), or ball bounded supports mixture (Luo [LQ18b]). The second way of using kernel estimation is a stochastic smoothing of point-wise sampled data. This approach has been introduced in the context of particle filters by Musso [MOLG01], on the basis of Silverman's theory [Sil86]. It consists of smoothing the particle states' empirical distribution $\mathbf{x}_k^i$ by adding random samples from a noise distribution defined by a specific kernel. The kernel and its bandwidth are optimally determined in terms of MISE. This results in a more regular description of the density to be estimated. The algorithm is called Regularised Particle Filter (RPF) and is introduced in the next section.



(a) Histogram (b) Naive estimator (c) Kernel estimator

Figure 2.7: Histogram (a), naive estimator (b), and kernel estimator (c), used to estimate a density from a set of samples $\mathbf{x}^i$. The kernel estimator (Silverman [Sil86]) yields a continuous density.

The Regularised Particle Filter (RPF)

The Regularised Particle Filter (RPF) was introduced in Musso, Oudjane, and Le Gland [Oud00; MOLG01] on the basis of kernel estimation (Silverman [Sil86]). The posterior density defined by a mixture of Dirac functions (2.19) can be rewritten as a mixture of weighted kernels centered on each particle $\mathbf{x}_k^i$, as in (2.25):

$$\widehat{p}(\mathbf{x}_k|\mathbf{M}_k) = \sum_{i=1}^{N} w_k^i K_h(\mathbf{x}_k - \mathbf{x}_k^i) \tag{2.31}$$

The kernel is assumed to be symmetric such that $K_h(-\mathbf{x}) = K_h(\mathbf{x})$. In the special case of all the box particles having the same weight, which is the case just after the resampling step, the kernel that minimises the MISE (2.28) is the Epanechnikov kernel (see Appendix E):

$$K_{opt}(\mathbf{x}) = \begin{cases} \frac{d+2}{2c_d}\left(1 - \|\mathbf{x}\|^2\right) & \text{if } \|\mathbf{x}\| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

where c_d is the volume of the unit hypersphere in $\mathbb{R}^d$. The associated optimal bandwidth is:

$$\begin{aligned} h_{opt} &= \mu A(K) N^{-\frac{1}{d+4}} \\ A(K) &= [8c_d^{-1}(d+4)(2\sqrt{\pi})^d]^{\frac{1}{d+4}} \end{aligned} \tag{2.33}$$

In case of a multimodal posterior state density (see Appendix B), the optimal bandwidth h may be overevaluated. A tuning coefficient $0 < \mu < 1$ has been introduced in (2.33) to limit the impact of regularisation on each individual mode. A robust version of RPF has been proposed where each mode is estimated via clusters and is independently regularised (Murangira [Mur+11]). However, this process has a high computational cost, which limits its use for embedded applications.

The optimal bandwidth (2.33) is expressed for a normalised density whose covariance is a d dimensional identity matrix. In practice, the kernel bandwidth must be adapted to the density's covariance. This can be done by computing the Cholesky decomposition of the empirical covariance of the resampled set of particles (with all weights reset to $1/N$) $\mathbf{A}_k = \text{Chol}\left(\frac{1}{N}\sum_i(\mathbf{x}_k^i - \widehat{\mathbf{x}}_k)^T(\mathbf{x}_k^i - \widehat{\mathbf{x}}_k)\right)$. Regularisation can then be applied by adding an additive noise to each particle as a sample of $\mathbf{A}_k K_h$. The regularisation step is illustrated in Figure 2.8.

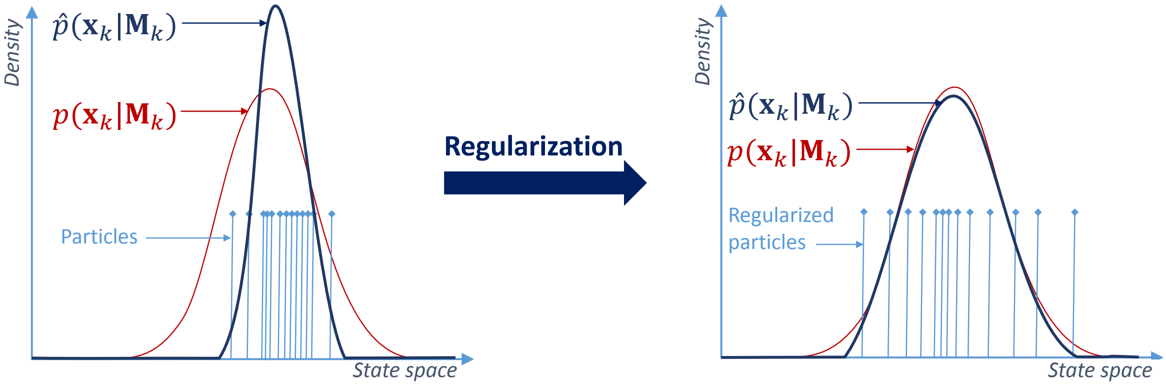


Figure 2.8: Regularisation of a sample of particles after resampling. The Mean Integrated Square Error (MISE) quantifies the fitting between the theoretical density p in red and the empirical density $\widehat{p}$ in blue.

The regularisation step helps the particle cloud to explore the state space in a more regular way by adjusting the empirical density on the theoretical one. Performing this step after each resampling brings more robustness and accuracy, since particles are less likely to concentrate in the actual density's distribution tail. However, the hypothesis is made that the actual density is unimodal, which is not always the case in practice.

The RPF opened the way to a variety of kernel-based particle filters. Among them, one can cite the Convolution Particle Filter (CPF [CR09]), which extends the kernel smoothing concept to the measurements space. More generally, the Kernel Estimation scheme led to a variety of kernel-based particle filters, e.g. the Kalman-Particle Kernel Filter (KPKF [Dah05], [PDM03]), or the Mixture Regularised Rao-Blackwellized Particle Filter (MRPF [MMD16]).

2.1.5 Set-based framework and Interval Analysis

The set-based framework is dedicated to cases where densities are unknown but bounded, i.e. where the only available hypotheses on them are their supports. It aims to estimate the smallest bounded set $\{X_k\}$ which contains the state $\mathbf{x}_k$, given the measurement sets $\{m_1\}, ..., \{m_k\}$, as illustrated in Figure 2.9. The following section briefly recalls the set-based estimation principle. Then, a focus on the interval analysis framework is done.

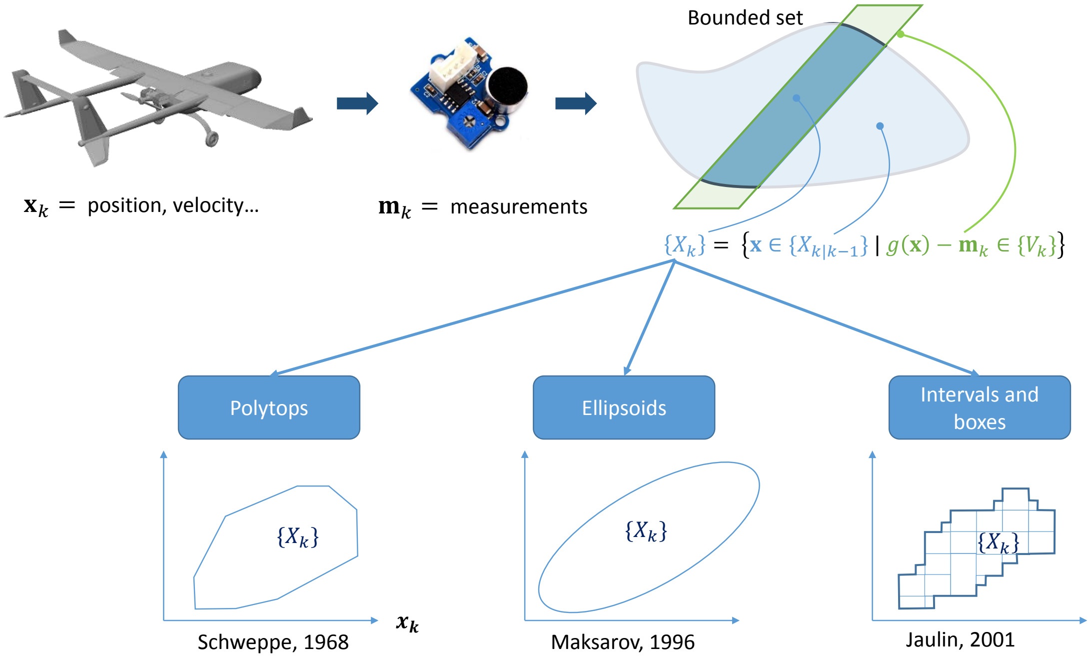


Figure 2.9: Set-based estimation scheme and three different state set approaches: Polytopes, ellipsoids, and paving of intervals, or boxes in the multidimensional case

Set estimation

In the set-based estimation scheme, the process and measurement noises are modelled by bounded sets $\mathbf{w}_k \in \{W_k\}$ and $\mathbf{v}_k \in \{V_k\}$. Therefore, let the measurement set be defined by:

$$\{m_k\} = \left\{\mathbf{y} \in \mathbb{R}^{d_m} | \mathbf{y} - \mathbf{m}_k \in \{V_k\}\right\} \tag{2.34}$$

Let $\{X_{k-1}\}$ be the previous set of states set, and $\{X_k\}$ the current set. The prediction step corresponds to the propagation of all elements of the previous state set, plus the process uncertainty set:

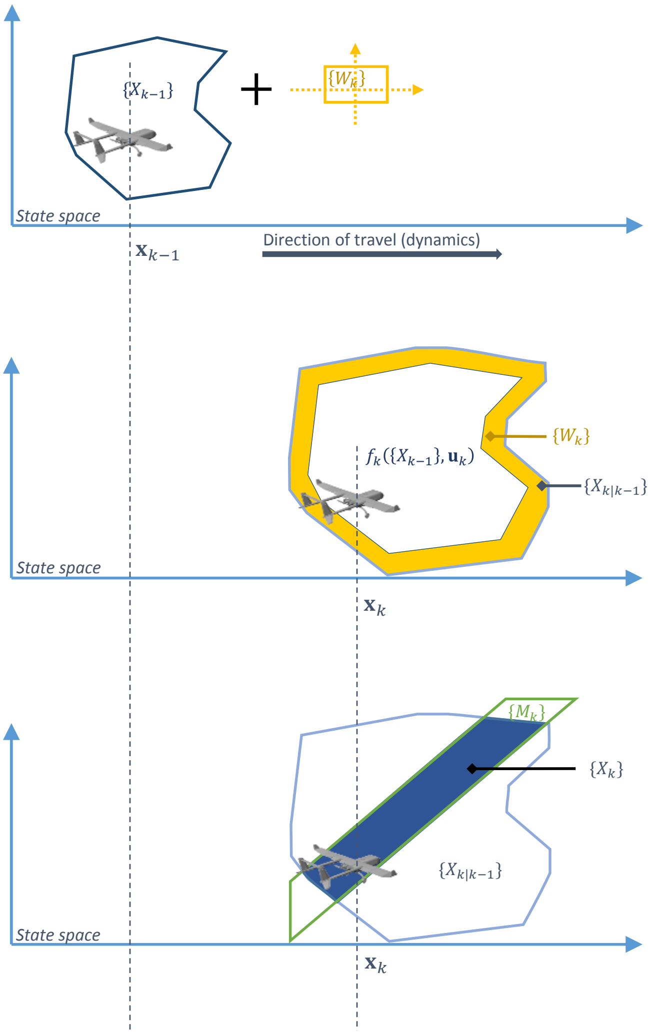
$$\{X_{k|k-1}\} = \left\{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{x} - f(\mathbf{x}', \mathbf{u}_k) \in \{W_k\}, \forall \mathbf{x}' \in \{X_{k-1}\}\right\} \tag{2.35}$$

The correction step corresponds to the action of keeping only the subset of the predicted set which is consistent with the measurement set:

$$\{X_k\} = \left\{\mathbf{x} \in \{X_{k|k-1}\} \mid h(\mathbf{x}) \in \{m_k\}\right\} \tag{2.36}$$

as introduced by Schweppe [Sch68]. Several methods have been proposed to describe these sets and their evolution, as illustrated in Figure 2.9, including: intervals and boxes [Jau09], ellipsoids [MN96], and polytopes [PLW94]. The output of these algorithms is therefore not seen as a point estimate associated with an uncertainty, but as a bounded set guaranteed to contain the actual state. Figure 2.10 illustrates the set-based prediction-correction scheme.

This thesis will focus on the interval scheme, which is described hereafter.



(a) Prediction

The **prior state set** $\{X_{k-1}\}$ is propagated via the deterministic dynamics f_k . The dynamics uncertainty (process noise $\mathbf{w}_k$) is accounted by a set addition of the prior set $\{X_{k-1}\}$ with the process noise set $\{W_k\}$.

(b) Predicted set

Step (a) results in the **predicted state set** $\{X_{k|k-1}\}$, which is usually larger than the prior set.
A measurement set $\{M_k\}$ is now available (often built from a sensor measurement $\mathbf{m}_k$ and the measurement noise set $\{V_k\}$). It will introduce information in the estimation. The measurement set is illustrated in step (c).

(c) Correction

The predicted state set is intersected with the measurement set, leading to the **posterior state set** $\{X_k\}$.
It is usually smaller than the predicted set, which consists of a refined estimation.
Then, one can iterate back to step (a) for further time-steps.

Figure 2.10: Set-based prediction-correction scheme

Interval analysis

This section describes the interval analysis framework on $\mathbb{R}^d$, based on Jaulin [Jau01].

A closed interval on $\mathbb{R} \cup [-\infty, \infty]$ is a set defined by:

$$[x] = [\underline{x}, \overline{x}] \triangleq \{x \in \mathbb{R} \cup \{-\infty, \infty\} \mid \underline{x} \leq x \leq \overline{x}\} \tag{2.37}$$

with $\underline{x} \in \mathbb{R} \cup \{-\infty\}$ the lower bound and $\overline{x} \in \mathbb{R} \cup \{\infty\}$ the upper bound of the interval. By definition, $\underline{x} \leq \overline{x}$. By convention, $\emptyset$ is considered as an interval to ensure that the set of intervals is closed with respect to intersection. Likewise, an open interval is defined as follows:

$$]\underline{x}, \overline{x}[\triangleq \{x \in \mathbb{R} \cup \{-\infty, \infty\} \mid \underline{x} < x < \overline{x}\} \tag{2.38}$$

Let us note $\mathbb{IR}$ the set containing all closed intervals on $\mathbb{R}$.

Classic operations of real arithmetic, namely addition $(+)$, subtraction $(-)$, multiplication $(*)$ and division $(/)$ can be extended to intervals. For the sake of brevity, the multiplication of two intervals can be denoted as $[a][b]$ instead of $[a] * [b]$. Let $\odot$ be one of these operators, then its interval extension is:

$$[x] \odot [y] = [\{x \odot y \mid x \in [x], y \in [y]\}] \tag{2.39}$$

where $[\{\cdot\}]$ is the *wrapping operator* which yields a box containing the set $\{\cdot\}$. In what follows, *wrapping* will refer to every inclusion of real sets into interval sets. If operator $\odot$ is continuous, then the set $\{x \odot y \mid x \in [x], y \in [y]\}$ is an interval and the wrapping operator can be omitted. This formulation remains valid for open-ended and unbounded intervals (see Jaulin [Jau01, Section 2.3.2]). Explicit formulations of the four laws of arithmetic are provided in Table 2.1 for closed intervals. In the case of interval division, the denominator interval must not contain zero.

An interval can be defined by its upper an lower bounds. An equivalent description is the definition by its center and diameter. Their definitions are provided in Table 2.1. The diameter corresponds to the Lebesgue Measure $\lambda([x])$.

A set of vectors of intervals in $d \in \mathbb{N}^*$ dimensions can be defined by the Cartesian product of d interval spaces, i.e. $\mathbb{IR}^d = \mathbb{IR} \times ... \times \mathbb{IR}$. A vector of d intervals is named a *box* and noted $[\mathbf{x}] \in \mathbb{IR}^d$. Box extension of lower bound, upper bound, center and diameter are defined as a vector of d terms that correspond to each interval term's bounds, center, or diameter. However, the Lebesgue measure of a box is not equivalent to the *box diameter*, but to the *volume of the box*. Box diameter and box volume are defined in Table 2.1. Arithmetic operations can be extended to boxes of same size by calculating the resulting intervals term by term.

It is important to note that the properties of basic operators in $\mathbb{IR}$ differ from their properties in $\mathbb{R}$. The following list provides some practical properties:

- Addition and multiplication remain associative and commutative.
- Multiplication is no longer distributive with respect to addition. Instead, it only verifies the *subdistributivity* property:

  $$[x]([y] + [z]) \subset [x][y] + [x][z] \tag{2.40}$$

- If each element of $(\mathbb{R}, \odot)$ admits a symmetric element, it is no longer true for $(\mathbb{IR}, \odot)$. In terms of algebraic structures, if $(\mathbb{R}, \odot)$ is a group, then $(\mathbb{IR}, \odot)$ is only a monoid. An example is given for subtraction. In general, $[x] - [x] \neq \{0\}$. Indeed, it should be interpreted as:

  $$[x] - [x] = \{x - y \mid x \in [x], y \in [x]\} \tag{2.41}$$

Table 2.1: Interval Analysis operations

Operation	Definition
Addition	$[a] + [b] = [\underline{a} + \underline{b}, \overline{a} + \overline{b}]$
Subtraction	$[a] - [b] = [\underline{a} - \overline{b}, \overline{a} - \underline{b}]$
Multiplication	$[a][b] = [\min(\underline{a}\overline{b}, \overline{a}\overline{b}, \underline{a}\underline{b}, \overline{a}\underline{b}), \max(\underline{a}\overline{b}, \overline{a}\overline{b}, \underline{a}\underline{b}, \overline{a}\underline{b})]$
Division	$[a]/[b] = [\underline{a}, \overline{a}][1/\overline{b}, 1/\underline{b}]$ if $0 \notin [b]$
Intersection	$[a] \cap [b] = [\max(\underline{a}, \underline{b}), \min(\overline{a}, \overline{b})]$
Diameter	$\|[a]\| = \overline{a} - \underline{a} = \lambda([a]) \qquad (\in \mathbb{R})$
Center	$c_{[a]} = \frac{1}{2}(\underline{a} + \overline{a})$ By convention, $\|\emptyset\| = 0$
Volume of a box	$\|[\mathbf{a}]\| = \prod_{j=1}^{d} \|[a_j]\| = \lambda([\mathbf{a}]) \qquad (\in \mathbb{R})$
Diameter of a box	$\delta_{[\mathbf{a}]} = [\|[a_1]\|, ..., \|[a_d]\|]^T \; (\in \mathbb{R}^d)$
Center of a box	$\mathbf{c}_{[\mathbf{a}]} = \frac{1}{2}(\underline{\mathbf{a}} + \overline{\mathbf{a}}) \qquad (\in \mathbb{R}^d)$

and not as $[x] - [x] = \{x - x \mid x \in [x]\}$.

In other words, interval operators do not account for possible dependencies between input intervals. The loss of properties expressed by (2.40) and (2.41) are direct consequences of the *dependency effect* that will be described in next section.

Two key concepts in interval analysis applied to state estimation are described in what follows: the inclusion function and the interval contraction.

Interval propagation via inclusion functions

In order to apply interval analysis to state estimation, it is necessary to describe the prediction step (2.35). This is linked to a more general issue, which is the transformation of an interval or a box by a function defined on $\mathbb{R}^d$. Let $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ be a function. The transformation of an interval $[x]$ by f is:

$$f([\mathbf{x}]) = \{f(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\} \tag{2.42}$$

which is not an interval in the general case, as illustrated in Figure 2.11. In order to define a function from $\mathbb{IR}^{d_1}$ to $\mathbb{IR}^{d_2}$, the *inclusion function* concept is defined by:

$$[f]([\mathbf{x}]) = [\{f(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\}] \tag{2.43}$$

which wraps $f([\mathbf{x}])$ to a inclusive box. The analytic expression of $[f]$ can be determined from f using the interval framework introduce in the previous section. If $[f]$ provides the smallest box containing $f([\mathbf{x}])$ for all $[\mathbf{x}] \in \mathbb{IR}^{d_1}$, then $[f]$ is said *minimal*. Concepts of inclusion function and minimal inclusion function are illustrated in Figure 2.11 for a real function f, an arbitrary inclusion function $[f]$ and the minimal inclusion function $[f_{min}]$.

However, some conservatism may be introduced by the inclusion function formulation. It is the consequence of two effects, known as *wrapping effect* and *dependency effect*. In the following chapters, *conservatism* will refer to the evaluation of uncertainty with unnecessarily large margins (quantified by the *pessimism* criterion, see Section 3.5).

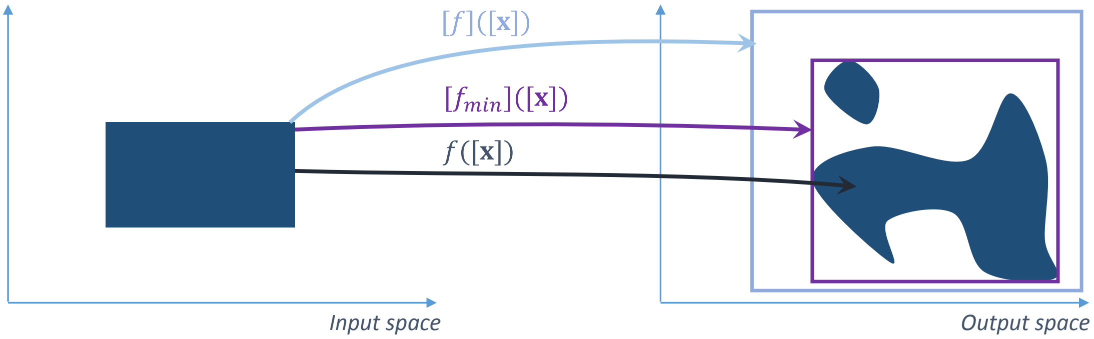


Figure 2.11: Concept of set transformation, inclusion function, and minimal inclusion function

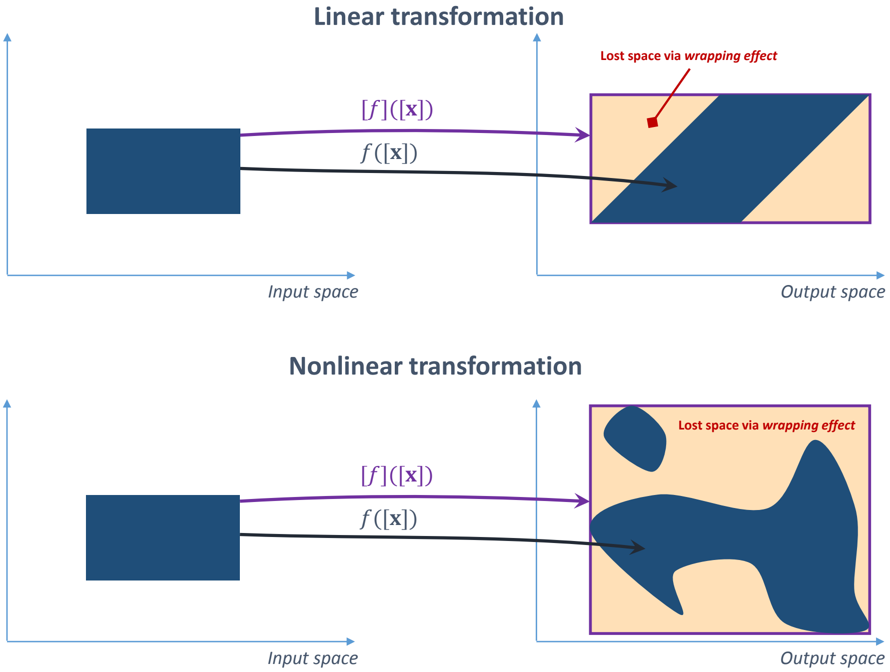


Figure 2.12: Illustration of the concept of wrapping effect for linear and non-linear transformations

The wrapping effect qualifies the natural conservatism of intervals and boxes, whose edges are colinear to the space base axis, causing lost space between the set $f([\mathbf{x}])$ and the box $[f]([\mathbf{x}])$. The wrapping effect can also be linked to discontinuities in function f. It can be intuitively understood with an illustration provided in Figure 2.12.

The dependency effect qualifies the dependencies between interval variables, as described in previous section with the properties of subdistributivity (2.40) and the loss of symmetry (2.41). In general, the dependency effect arises when more than one instance of the same input variable are combined in the same output variable of function f. For example, on $\mathbb{R}$, the function $f_1 : x \mapsto x^2$ is equivalent to the function $f_2 : x \mapsto x \times x$. However, once wrapped into the interval space, the two functions are not equivalent anymore. Indeed, the resulting inclusion functions are respectively $[f_1]([x]) = \{x^2 \mid x \in [x]\} \subset \mathbb{R}^+$ and $[f_2]([x]) = \{x \times y \mid x \in [x], y \in [x]\} \subset \mathbb{R}$. For example, with $[x] = [-2, 2]$, the first inclusion function gives $[f_1]([x]) = [0, 4]$ while the other one gives $[f_2]([x]) = [-4, 4]$.

Sufficient conditions for an inclusion function to be minimal are provided by the following theorem:

Theorem 1. *([Jau01, theorem 2.2]) Consider a function defined by:*

$$\begin{aligned} f :\ & \mathbb{R}^d \to \mathbb{R} \\ & \boldsymbol{x} \mapsto f(\boldsymbol{x}) \end{aligned} \tag{2.44}$$

expressed as a finite composition of arithmetic operators and elementary functions. Let be $\boldsymbol{x} \triangleq [x_1, ... x_d]^T$. If:

1. *All arithmetic operators and elementary functions involved are continuous,*
2. *Each input variable x_i occurs at most once in the formal expression of f,*

then, the inclusion function $[f] : \mathbb{IR}^d \to \mathbb{IR}$ defined by $[f]([\boldsymbol{x}]) = \left[\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in [\boldsymbol{x}]\}\right]$ is minimal.

The first condition accounts for the wrapping effect, while the second condition deals with the dependency effect, which is then removed. This theorem can be applied to any function of $\mathbb{R}^{d_1}$ to $\mathbb{R}^{d_2}$ since the function can be expressed as a vector of d_2 functions from $\mathbb{R}^{d_1}$ to $\mathbb{R}$.

The wrapping effect may also arise from the composition of functions. Consider the composition of two functions $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ and $g : \mathbb{R}^{d_2} \to \mathbb{R}^{d_3}$. Let $h = g \circ f$ be their composition. Then, once wrapped into the interval space, the relation becomes:

$$[h]([x]) \subset [g]([f]([x])) \tag{2.45}$$

As a result, it is of interest to carefully design inclusion functions by choosing the best way of writing real functions before wrapping them into interval spaces. A detailed analysis of the problem will help to avoid discontinuities, multiple occurrences of input variables, and ill-writing of compositions, to satisfy as much as possible Theorem 1.

In what follows, the *natural inclusion function* of a real function f will be called the naive inclusion function obtained by directly wrapping f to the interval framework. Natural inclusion functions are minimal in the case of linear applications, e.g. $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ defined by $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ with $\mathbf{A} \in \mathbb{R}^{d_2 \times d_1}$, by direct application of Theorem 1. For nonlinear cases, several methods have been proposed to reduce the concervatism of inclusion functions, namely (Jaulin [Jau01]): *Centered inclusion functions*, *Mixed* *centered inclusion functions*, and *Taylor inclusion functions*. Unfortunately, there is no general rule about the minimal inclusion characteristics of such methods.

The inclusion function concept can be applied to interval state prediction. Let $[\mathbf{x}_{k-1}] \in \mathbb{IR}^d$ be the box containing the state $\mathbf{x}_{k-1}$ and $f : \mathbb{R}^d \times \mathbb{R}^{d_c} \to \mathbb{R}^d$ the discrete dynamical model. Then, by applying (2.35), one can define the one-step-ahead prediction equation of an interval-based estimation algorithm:

$$\left[\mathbf{x}_{k|k-1}\right] = [f]\left([\mathbf{x}_{k-1}], \mathbf{u}_k\right) + [\mathbf{w}_k] \tag{2.46}$$

with

$$[f]\left([\mathbf{x}_{k-1}], \mathbf{u}_k\right) = \left[\left\{f(\mathbf{x}_{k-1}, \mathbf{u}_k) \mid \mathbf{x}_{k-1} \in [\mathbf{x}_{k-1}]\right\}\right] \tag{2.47}$$

Interval correction via set contraction

The state set has been propagated thanks to an inclusion defined in the previous section. In order to update it using the measurements, as described by (2.36), the concept of *Constraint Satisfaction Problem* (CSP) and of *contractor* must be defined. A CSP $\mathcal{H}$ is defined as follows (e.g. see Jaulin [Jau09]):

$$\mathcal{H} : (G(\mathbf{x}) = 0, \mathbf{x} \in \{\mathbf{x}\}) \tag{2.48}$$

where $\{\mathbf{x}\} \subset \mathbb{R}^d$ is the support of $\mathbf{x} = [x_1, ..., x_d]^T$ and $G : \mathbb{R}^d \to \mathbb{R}$ a function that links variables x_i with n_G relations (called constraints). The solution set to $\mathcal{H}$ is:

$$\mathcal{S} = \{\mathbf{x} \in \{\mathbf{x}\} \mid G(\mathbf{x}) = 0\} \tag{2.49}$$

For intervals, a special case of (2.48) is:

$$\mathcal{H} : (G(\mathbf{x}) = 0, \mathbf{x} \in [\mathbf{x}]) \tag{2.50}$$

Contracting $\mathcal{H}$ means replacing $[\mathbf{x}]$ by a smaller box $[\mathbf{x}']$ such that $\mathcal{S} \subset [\mathbf{x}'] \subset [\mathbf{x}]$. A *contractor* for $\mathcal{H}$ refers to any operator that can be used to contract it. A contractor is said to be *optimal* if it provides the smallest box $[\mathbf{x}']$ which contains $\mathcal{S}$. A wide variety of contractors have been proposed in the literature, namely: *Gauss elimination*, *Gauss-Seidel algorithm*, *Krawczyk method*, *linear programming*, *Newton method*, *parallel linearization*, and *forward-backward propagation*. A review can be found in Jaulin [Jau01].

The set-correction step defined by (2.36) is a special case of CSP where G represents the deterministic observation equation:

$$\left(\mathbf{m} - h(\mathbf{x}) = 0, \mathbf{x} \in \{X_{k|k-1}\}, \mathbf{m} \in \{m_k\}\right) \tag{2.51}$$

In the interval case:

$$\left(\mathbf{m} - h(\mathbf{x}) = 0, \mathbf{x} \in [\mathbf{x}_{k|k-1}], \mathbf{m} \in [\mathbf{m}_k]\right) \tag{2.52}$$

where $[\mathbf{m}_k] = \mathbf{m}_k + [\mathbf{v}_k] \in \mathbb{IR}^{d_m}$ is the measurement box, and $[\mathbf{v}_k]$ the measurement noise density support. The solution is:

$$[\mathbf{x}_k] \triangleq \left\{\left[\mathbf{x} \in [\mathbf{x}_{k|k-1}] \mid h(\mathbf{x}) \in [\mathbf{m}_k]\right\}\right] \tag{2.53}$$

This section introduced set-based estimation, with a specific focus on interval-based set estimation. These approaches are of interest for their strong robustness to ambiguities and their ability to address densities where the only available knowledge is their bounds. However, for non-linear cases (especially non-linear measurements), the number of operations may grow rapidly, which limits their practical use. Furthermore, they can be seen as *worst-case* approaches, which makes them often conservative compared to probabilistic approaches. In order to take advantage of both probabilistic framework (see Section 2.1.2) and set-based framework, several hybrid approaches have been proposed. In the state estimation field, the Interval Kalman Filter [CWS97; JZZ16] can be cited, which keeps the KF properties while being more robust to measurement outliers. In the context of non-linear state estimation, the Box Particle Filter [AGB07] has been introduced and is described in next section.

2.1.6 The Box Particle Filter

The Box Particle Filter (BPF) was initially proposed by Gning [AGB07] as a bridge between Monte Carlo methods and set-based approaches. The reader will find a synthesis of the BPF formalism in Gning [Gni+13]. The prior conditional density is approximated by a weighted sum of $N \in \mathbb{N}^*$ uniform kernels, corresponding to a box particle cloud, illustrated in Figure 2.13.

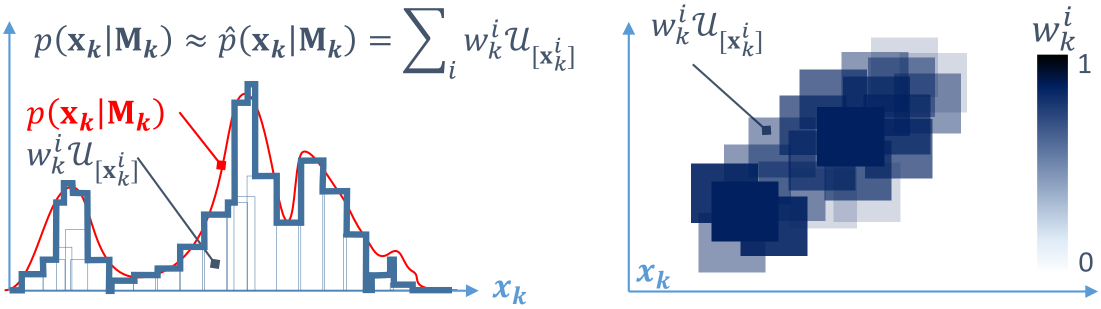


Figure 2.13: Box Particle Filter scheme: the conditional density is approximated with a mixture of uniforms corresponding to a cloud of weighted box particles.

The conditional density is defined by:

$$\hat{p}(\mathbf{x}_{k-1}|\mathbf{M}_{k-1}) = \sum_{i=1}^{N} w_{k-1}^i \mathcal{U}_{[\mathbf{x}_{k-1}^i]} \tag{2.54}$$

In the prediction step, the prior conditional density is propagated through the Chapman-Kolmogorov equation (2.4).

The assumption can be made that the dynamical inclusion function $[f]$ only modifies the support of box $[\mathbf{x}_{k-1}^i]$ without changing the box density nature:

$$\int_{\mathbb{R}^d} \mathcal{U}_{[\mathbf{x}_{k-1}^i]} p(\mathbf{x}_k|\mathbf{x}_{k-1}) \mathrm{d}\mathbf{x}_{k-1} \approx \mathcal{U}_{[f]([\mathbf{x}_{k-1}^i]) + [\mathbf{w}_k]} \tag{2.55}$$

Although Gning [GMA10] proposed an alternative approach to obtain a more precise prediction, approximation (2.55) is often chosen for computational reasons. Then, the resulting estimated conditional density is approached by

$$\hat{p}\left(\mathbf{x}_k|\mathbf{M}_{k-1}\right) = \sum_{i=1}^{N} w_{k-1}^i \mathcal{U}_{[\mathbf{x}_{k|k-1}^i]} \tag{2.56}$$

with

$$[\mathbf{x}^i_{k|k-1}] = [f]\left([\mathbf{x}^i_{k-1}], \mathbf{u}_k\right) + [\mathbf{w}_k] \tag{2.57}$$

In the correction step, the posterior conditional density is obtained by Bayes' rule. By deriving (2.5), it appears that each box $[\mathbf{x}^i_{k|k-1}]$ has to be contracted with respect to the measurement box:

$$[\mathbf{x}^i_k] = \left[\left\{\mathbf{x}_k \in [\mathbf{x}^i_{k|k-1}] \mid h(\mathbf{x}_k) \in [\mathbf{m}_k]\right\}\right] \tag{2.58}$$

The contraction step (2.58) guarantees that each contracted box will contain every possible state which is consistent with the current measurements. This makes BPF robust to measurement ambiguities. Thus, the resulting posterior conditional density is:

$$\widehat{p}(\mathbf{x}_k|\mathbf{M}_k) = \frac{1}{q_k}\sum_{i=1}^{N}\frac{|[\mathbf{x}^i_k]|}{|[\mathbf{x}^i_{k|k-1}]|}w^i_{k-1}\mathcal{U}_{[\mathbf{x}^i_k]} \tag{2.59}$$

As a result, the updated weights are

$$w^i_k = \frac{1}{q_k}\frac{|[\mathbf{x}^i_k]|}{|[\mathbf{x}^i_{k|k-1}]|}w^i_{k-1} \tag{2.60}$$

The ratio between the volume of the contracted box and the volume of the propagated box can be interpreted as an *innovation*. Finally, weights have to be normalised by

$$q_k = \sum_{i=1}^{N}\frac{|[\mathbf{x}^i_k]|}{|[\mathbf{x}^i_{k|k-1}]|}w^i_{k-1} \tag{2.61}$$

A state estimate $\widehat{\mathbf{x}}_k$ can be deduced from the box particle cloud such that:

$$\widehat{\mathbf{x}}_k \triangleq \mathbb{E}\left[\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{M}_k)\right] \approx \sum_{i=1}^{N} w^i_{k-1}\mathbf{c}^i_k \tag{2.62}$$

For the sake of brevity, the box particles centers are denoted $\mathbf{c}^i_k \triangleq \mathbf{c}_{[\mathbf{x}^i_k]}$ (see Table 2.1). A confidence on that estimation can be defined such that:

$$\widehat{\mathbf{P}}_k \triangleq \operatorname{Cov}\left[\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{M}_k)\right] \approx \sum_i w^i_k(\mathbf{c}^i_k - \widehat{\mathbf{x}}_k)(\mathbf{c}^i_k - \widehat{\mathbf{x}}_k)^T \tag{2.63}$$

As in a conventional particle filter, a resampling step is added to avoid the degeneracy problem, when only a small number of box particles are consistent with the measurements and all others have a near-zero weight. Published papers about BPF often use the Multinomial Resampling described in Algorithm 3 to determine the number n^i of new instances of each box particle. Nevertheless, since box particles have a non-zero volume, Gning [AGB07] proposed to *subdivide* them instead of simply duplicating them. This helps to increase the resolution of the filter's exploration in the state space, even for non-observed dimensions. For the sake of simplicity, each box is subdivided into n^i sub-boxes along one dimension $d^{cut,i}_k \in [1, d]$. BPF is summarised in Algorithm 5.

However, there are few general and efficient method to select the $d^{cut,i}_k$ dimension. In Abdallah [AGB07], box particles are subdivided along a fixed dimension, for a specific problem. In Gning [GRM12], $d^{cut,i}_k$ is chosen randomly. Although this method is independent from the model, it is not

Algorithm 5 Box Particle Filter

```
  Inputs: initial state density $p(\mathbf{x}_0)$, process noise box $[\mathbf{w}_k]\ \forall k$, measurement noise box $[\mathbf{v}_k]\ \forall k$.
  Outputs: state estimate $\widehat{\mathbf{x}}_k\ \forall k$, estimated covariance $\widehat{\mathbf{P}}_k\ \forall k$.
Initialisation: The initial box particle set (or box particle cloud) $\{[\mathbf{x}_0^i] \in \mathbb{IR}^d, w_0^i \in \mathbb{R}^{+*}\}_{i\in[1,N]}$ is drawn such that the initial estimated density (2.56) approaches the initial actual density $p(\mathbf{x}_0)$. In practice, it can be initialised with a regular paving on the state density's support, which makes it similar to an histogram.
for each time-step $k$ do
    Prediction step: Propagate box particles using the dynamics $[\mathbf{x}_{k|k-1}^i] = [f]\left([\mathbf{x}_{k-1}^i]\right) + [\mathbf{w}_k]$
    Correction step: Contract each particle with respect to the measurement box $[\mathbf{m}_k]$ (see contractors in Section 2.1.5), such that $[\mathbf{x}_k^i] = \left[\left\{\mathbf{x}_k \in [\mathbf{x}_{k|k-1}^i] \quad | \quad h(\mathbf{x}_k) \in [\mathbf{m}_k]\right\}\right]$
    Update weights by the ratio of the volume before and after contraction $w_k^i = \frac{1}{q_k}\frac{|[\mathbf{x}_k^i]|}{|[\mathbf{x}_{k|k-1}^i]|} w_{k-1}^i$, with $q_k = \sum_{i=1}^N w_k^i$.
    Compute the state estimate $\mathbb{E}\left[\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)\right]$. It can be approximated by a least-square estimator: $\widehat{\mathbf{x}}_k = \sum_i w_k^i \mathbf{c}_k^i$, where $\mathbf{c}_k^i$ is the center of each particle (see Table 2.1).
    Compute a confidence indicator, e.g. the covariance $\widehat{\mathbf{P}}_k = \sum_i w_k^i (\mathbf{c}_k^i - \widehat{\mathbf{x}}_k)(\mathbf{c}_k^i - \widehat{\mathbf{x}}_k)^T$.
    if a resampling criterion is satisfied, e.g. $N_{eff} < \theta_{eff} N$, see (2.22) then
        Draw a new set of box particles $\{[\mathbf{x}_k^i], w_k^i\}_{i\in[1,N]}$ using a resampling method, e.g. Multinomial Resampling (Algorithm 3) to determine the number of new instances per box particles $n^i$.
        Replace low-weighted box particles with $n^i$ subdivisions of high-weighted particles along a randomly picked dimension.
        Reset all weights to $1/N$.
    end if
end for
Return $\widehat{\mathbf{x}}_k, \widehat{\mathbf{P}}_k\ \forall k$.
```

optimal. Indeed, picking the dimension to cut randomly may lead to box shape degeneracy after several resampling executions. This results in non-homogeneity in the boxes' proportions, which may cause a significant lack of accuracy in the case of partial or ambiguous measurements. De Freitas [DF+16] proposes to subdivide each box particle along its larger side. This aims to avoid the box shape degeneracy encountered with the random-subdivision strategy. However, side length comparison is only possible if state parameters have similar order of magnitude. This results in a lack of generality. In Luo [LQ18a], a more general method is introduced, relying on the gradient of observation function. However, this formulation is limited to problems where all state variables are involved in the observation equation, which is not the case, e.g. for Terrain Aided Navigation (see Section 3.2).

Furthermore, the Box Particle Filter assumes from the start that the box particles and the box measurements are supports of uniform kernels. Although Gning [GMA12] has theoretically justified this assumption, it may not be suitable for all descriptions, especially when some distribution hypothesis is known about the measurements.

Finally, in spite of its high robustness, BPF turns often pessimistic and inaccurate when measurements are ambiguous or do not bring enough information.

2.1.7 An overview of state estimation

Two main estimation schemes have been presented in the above sections, leading to a selection of several emblematic approaches. Figure 2.14 illustrates the different approaches on a non-linear non-Gaussian probability density function estimation.

To summarize, Probabilistic Estimation (Section 2.1.2) makes it possible to iteratively estimate a conditional state density knowing a trajectory of measurements. The Kalman Filter (KF, see 2.1.3 on page 13) is optimal for the Gaussian linear case and has minimum-variance for non-Gaussian linear cases. It has been extended to non-linear cases (EKF, UKF, see 2.1.3 on page 15) but these approaches are sub-optimal and suffer from their lack of guarantees. To cope with non-linear cases, recent approaches have been introduced either to map the problem to a non-euclidean manifold (e.g, the Lie groups mentioned in Section 2.1.3 on page 15), or empirically describe the non-linear propagation and update of the density to be estimated, as in particle filters (see 2.1.4 on page 17). More advanced and robust particle filters have been introduced by using Kernel Regularisation (see 2.1.4 on page 23). When uncertainties are analytically unknown, but assumed bounded, set-based approaches make it possible to perform robust and guaranteed estimation (see Section 2.1.5). However, for non-linear cases, these approaches may suffer form high computational costs and are often conservative. In order to tackle robust estimation to ambiguities and non-linearities, a set-based particle filter, called Box Particle Filter was introduced (see Section 2.1.6). Although it is robust to non-linearities and ambiguities, it suffers from conservatism and inaccuracy.

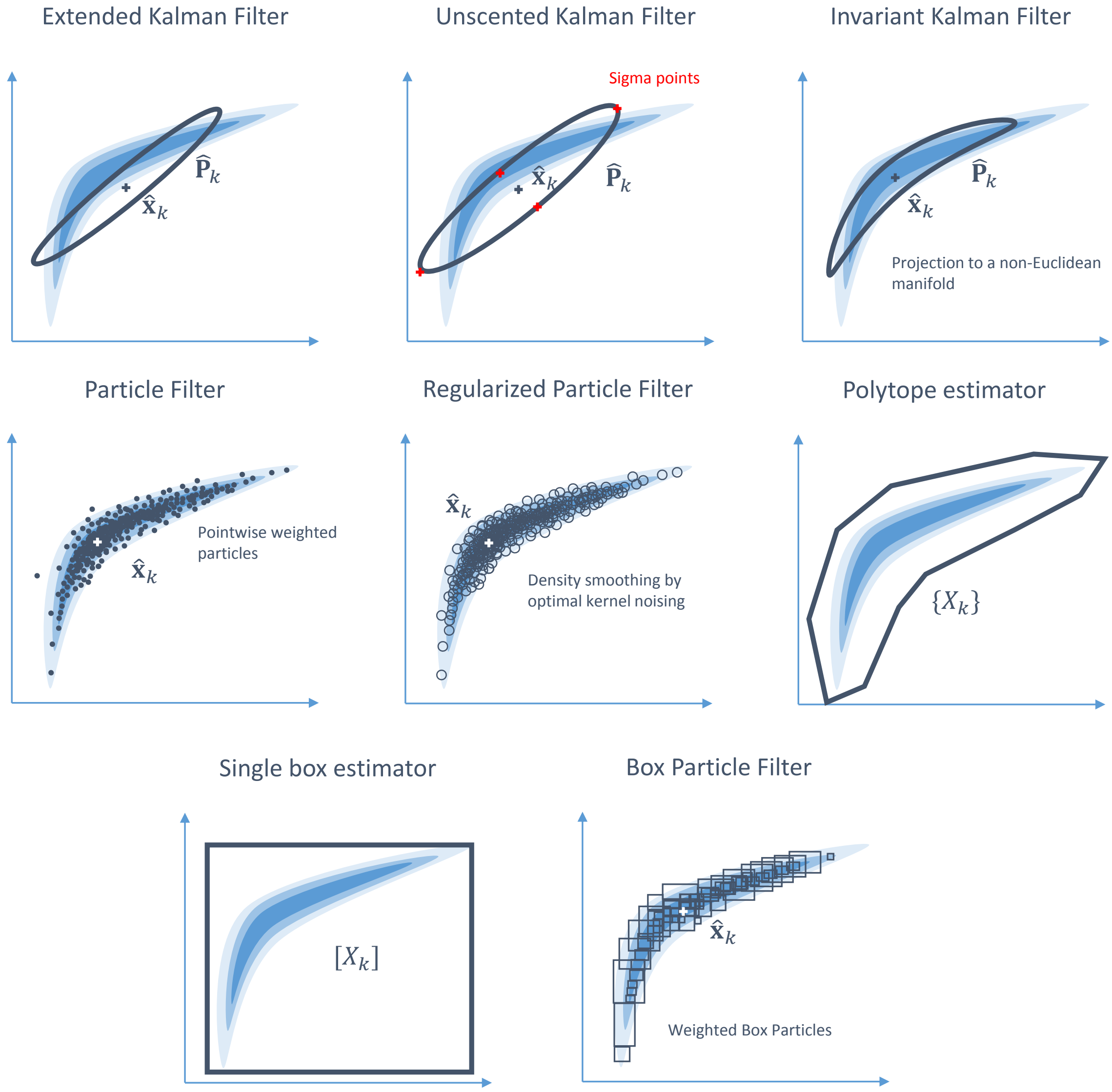


Figure 2.14: Overview of different non-linear state estimation approaches

2.2 COLLABORATIVE ESTIMATION

The previous section introduced the iterative state estimation problem. It can be formalized either in the Bayesian framework 2.1.2 or in the set-based framework 2.1.5. State estimation can be performed by a single estimator on the basis of one or several sensor measurements (e.g. GNSS position measurement combined with baro-altimeter measurement).

However, the sensors may be in different locations, for example embedded in different vehicles. They thus need to communicate and exchange information. State estimation from such a sensor network can be centralised (e.g. several vehicles sending information to a ground station) or distributed (e.g. each vehicle performing its own estimation, given the available measured and exchanged data). It can also be of interest to distribute the state estimation task to several algorithms collaborating together, even if all sensors are localized in the same vehicle. This can bring robustness and reduce the computational load.

Section 2.2.1 introduces a graph-based formalism to describe a collaborative scheme. Section 2.2.2 then introduces the Bayesian framework for collaborative estimation. Section 2.2.3 focuses on the federated architecture which allows each measurement to be dealt with by a dedicated filter, suitable for their specific characteristics (non-linearities, ambiguities, probabilistic hypotheses).

2.2.1 Graph representation

Consider a set of N_a systems called *agents*, for example autonomous aircrafts, that can communicate together. Their interactions can be modeled with the graph representation (Ren [RB08]). Each agent represents one node $i \in [1, N_a]$ of the graph, as illustrated in Figure 2.15. The nodes are linked together with weighted relations called *arcs* representing the ability of two agents to communicate.

The whole system's state $\boldsymbol{\xi}_k$ can be either a common state from which each node seeks measurements (e.g. several sensors measuring the position of a single aircraft), or a concatenation of several local states forming a global state (e.g. concatenation of individual states of several vehicles flying in formation). In the last case, the system's state can be written:

$$\boldsymbol{\xi}_k = \left[\mathbf{x}_k^{1\,T}, ..., \mathbf{x}_k^{N_a\,T} \right]^T \in \mathbb{R}^{dN_a} \tag{2.64}$$

where $\mathbf{x}_k^i$ is the individual state of each node.

In the field of graph theory, the global description of relations between nodes is called *graph topology*. A graph topology can be described by the *adjacency matrix* $\mathbf{A} = \{a_{ij}\} \in \mathbb{R}^{N \times N}$ which shows the weights of the arcs from node j to node i. if j sends an information to i, then $a_{ij} > 0$. Else, $a_{ij} = 0$.

This framework can be extended to varying topologies, where the adjacency matrix is time-dependent. The graph can be directed, as in Figure 2.15. The direction of an arc (j sends information to i) is represented by an arrow. If the graph is not directed, then the arcs are equivalent to double-headed arrows (j sends information to i and i sends information to j). Then, matrix $\mathbf{A}$ is symmetric.

Each agent communicates with a subset of agents called *neighbours*. The neighbours of agent i correspond to the N_i non-zero adjacency coefficient of line i of matrix $\mathbf{A}$: $\{j \in [1, N_a] \mid a_{ij} \neq 0\}$. Intuitively, neighbours of i are the agents j giving information to agent i.

Three kinds of architectures can be considered: *centralised*, *decentralised*, or *distributed* architectures.

- A centralised architecture (Figure 2.16) refers to a fleet in which decision is performed in a computation center i^* (e.g, node $i^* = 1$). The computation center can be an agent or a ground station. It receives measurements from all agents $j \neq i^*$, processes them, and broadcast them back the output (e.g. state estimation). Therefore, the adjacency coefficients satisfy $a_{i^*j} > 0$.

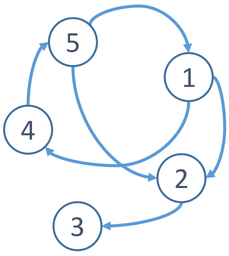


Figure 2.15: Graph example

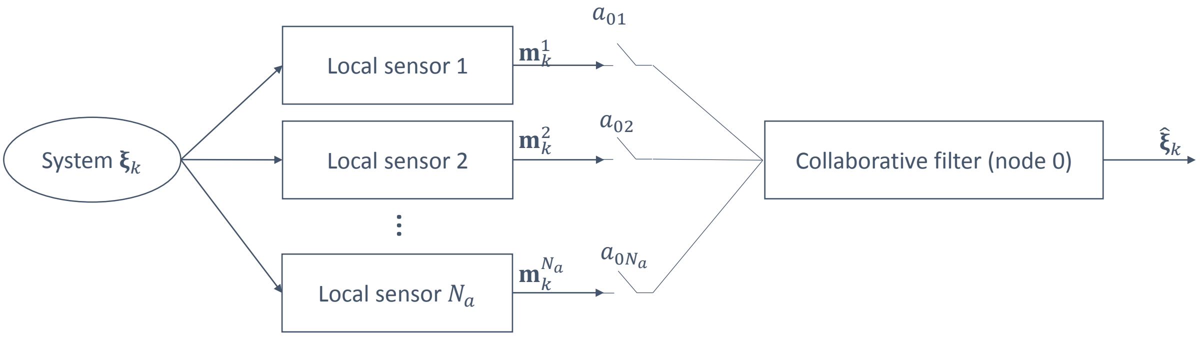


Figure 2.16: Collaborative estimation with a centralised architecture. Each node is a local sensor (e.g. embedded on different vehicles) that sends its measurements to a central node (by convention, $i = 0$) which processes all measurements in a collaborative filter. The communication graph is defined by the adjacency coefficients a_{0i}, represented by switches for a case where $a_{0i} \in \{0, 1\}$.

- In a decentralised architecture, the estimation is performed by all the agents with respect to their own representation of the whole system.
- In a distributed architecture (Figure 2.17), the estimation is performed by all the agents with respect to their local measurements and additional communications. If all agents can emit and receive information, the graph is said to be *fully connected* and the non-diagonal adjacency coefficients satisfy $a_{ij} > 0 \; \forall i \neq j$.

2.2.2 Collaborative fusion filters

The previous section presented a collaborative framework to formalize interactions between several agents. Agents can represent various subsystems belonging to a larger system, e.g. different vehicles belonging to a float, or different sensors embedded in a single vehicle. This section introduces the Bayesian framework for collaborative state estimation. It will be referred to as *fusion*.

Denote f_k the system's dynamical model:

$$\boldsymbol{\xi}_k = f_k(\boldsymbol{\xi}_{k-1}) + \mathbf{w}_k \tag{2.65}$$

where $\boldsymbol{\xi}_k \in \mathbb{R}^d$ is the system's state and $\mathbf{w}_k \in \mathbb{R}^d$ the process noise. If the state is a collection of several individual states (e.g. several vehicles flying in formation), then f is defined as:

$$f_k(\boldsymbol{\xi}_k) \triangleq \left[f_k^i(\mathbf{x}_k^1)^T, ..., f_k^i(\mathbf{x}_k^{N_a})^T \right]^T \tag{2.66}$$

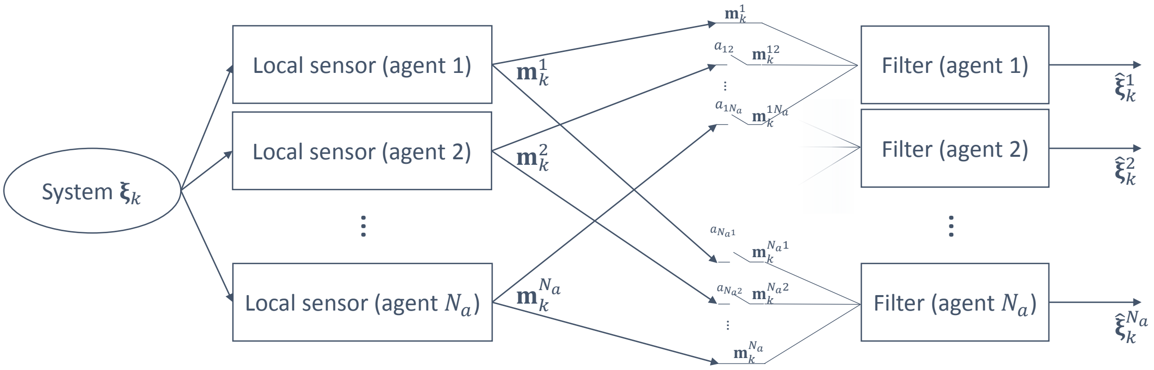


Figure 2.17: Collaborative estimation with a distributed architecture. Each node consists of a local sensor and a filter. The node broadcasts its measurements to its neighbours and fuses its local measurements $\mathbf{m}_k^i$ with received measurements $\mathbf{m}_k^{ij}$. The communication graph is defined by the adjacency coefficients a_{ij}.

Let us consider a node i which aims to estimate the system's state ξ_k. In addition to its local measurements $\mathbf{m}_k^i \in \mathbb{R}^{d_m^i}$, it also receives communicated measurements $\mathbf{m}_k^{ij} \in \mathbb{R}^{d_m^j}$ from its N_i neighbours. Local measurements are linked to the system's state with the local observation equation:

$$\mathbf{m}_k^i = h_k^i(\xi_k) + \mathbf{v}_k^i \tag{2.67}$$

where h_k^i is the local observation equation and $\mathbf{v}_k^i$ the local measurements noise. The local measurement density is noted $p(\mathbf{m}_k^i|\xi_k)$. Communicated measurements are modelled as follows:

$$\mathbf{m}_k^{ij} = h_k^{ij}(\xi_k) + \mathbf{v}_k^{ij} \tag{2.68}$$

where h_k^{ij} is the collaborative observation equation and $\mathbf{v}_k^{ij}$ the local measurements noise. Exponent ij indicates that data is received by agent i from agent j. The collaborative measurement density is $p(\mathbf{m}_k^{ij}|\xi_k)$. For the sake of brevity, it is assumed that $h_k^i = h^i \ \forall k$, $h_k^{ij} = h^{ij} \ \forall k$, $f_k^i = f^i \ \forall k$, and $f_k = f \ \forall k$. In what follows, the joint measurements vector containing both local measurements $\mathbf{m}_k^i$ and communicated measurements will be considered: $\mathbf{z}_k^i \triangleq \left[\mathbf{m}_k^{i\,T}, \mathbf{m}_k^{i1\,T}, ..., \mathbf{m}_k^{ij\,T}, ..., \mathbf{m}_k^{iN_i\,T}\right]^T$. If the measurements are statistically independent, the collaborative likelihood is:

$$p(\mathbf{z}_k^i|\xi_k) = p(\mathbf{m}_k^i|\xi_k) \prod_{j=1}^{N_i} p(\mathbf{m}_k^{ij}|\xi_k) \tag{2.69}$$

Denote $\mathbf{Z}_k^i$ the vector containing all the node measurements' trajectory: $\mathbf{Z}_k^i \triangleq \left[\mathbf{z}_k^{i\,T}, ..., \mathbf{z}_k^{i\,T}\right]^T$ Then, the Optimal Filter (see Section 2.1.2) can be applied to this scheme.

Figure 2.16 illustrates a centralised collaborative architecture. As introduced in last section, each node consists of a local sensor (e.g. embedded on different vehicles). It sends its measurements to the central node ($i = 0$) which processes all measurements in a collaborative filter. The communication graph is defined by the adjacency coefficients a_{0i}.

Figure 2.17 illustrates a distributed collaborative architecture. Each node consists of both a local sensor and a filter. It broadcasts its measurements to its neighbours and fuses its local measurements $\mathbf{m}_k^i$ with received measurements $\mathbf{m}_k^{ij}$. This results in N_a estimates ξ_k^i. If the graph is fully connected, i.e,

$a_{ij} > 0 \forall i, j$, then every estimate tend to the centralised solution, i.e. the estimate that would be obtained with a centralised filter. In what follows, the collaborative estimation scheme will be formalized in a distributed way. To retrieve a centralised formulation, set i to 0.

Each agent has its own estimation $\widehat{\xi}_k^i \in \mathbb{R}^d$ of the state vector ξ_k. In the linear Gaussian case, one can consider to fuse all measurements contained in $\mathbf{z}_k^i$ with a Kalman Filter. As stated by Verhaegen [VVD86], the number of floating point operations per iteration for a Kalman Filter depends on d_m^3 (see Appendix C for more details). For a collaborative Kalman Filter, the measurement's contribution to the computational load depends on N_a^3, which may limit its use. As stated in Grocholsky [Gro02], the Kalman Filter's updating step cannot be simplified in a linear summation of all received innovations:

$$\widehat{\xi}_k^i \neq \widehat{\xi}_{k|k-1}^i + \mathbf{K}_k^i \left(\mathbf{m}_k^i - \mathbf{H}_k^i \widehat{\xi}_{k|k-1}^i \right) + \sum_{j \neq i} \mathbf{K}_k^{ij} \left(\mathbf{m}_k^{ij} - \mathbf{H}_k^{ij} \widehat{\xi}_{k|k-1}^i \right) \tag{2.70}$$

since innovations $\mathbf{m}_k^{ij} - \mathbf{H}_k^{ij} \widehat{\xi}_{k|k-1}^i$ depend on the predicted state $\widehat{\xi}_{k|k-1}^i$ and thus are not independent.

The KF equations also can be written in terms of information (see Appendix A). In the Gaussian case, the information matrix is equivalent to the inverse of the covariance. The information form of the Kalman Filter has been introduced by Maybeck [May82], and derived by Manyika [Man93]. The Information Filter (IF) equations are obtained by introducing new state variables:

$$\begin{aligned} \widehat{\mathbf{y}}^i &\triangleq {\mathbf{P}^i}^{-1} \widehat{\xi}^i \\ \widehat{\mathbf{Y}}^i &\triangleq {\mathbf{P}^i}^{-1} \end{aligned} \tag{2.71}$$

Information from local measurements are defined as follows:

$$\begin{aligned} \mathbf{i}_k^i &\triangleq {\mathbf{H}_k^i}^T {\mathbf{R}_k^i}^{-1} \mathbf{m}_k^i \\ \mathbf{I}_k^i &\triangleq {\mathbf{H}_k^i}^T {\mathbf{R}_k^i}^{-1} \mathbf{H}_k^i \end{aligned} \tag{2.72}$$

Likewise, information from communicated measurements are:

$$\begin{aligned} \mathbf{i}_k^{ij} &\triangleq {\mathbf{H}_k^{ij}}^T {\mathbf{R}_k^{ij}}^{-1} \mathbf{m}_k^{ij} \\ \mathbf{I}_k^{ij} &\triangleq {\mathbf{H}_k^{ij}}^T {\mathbf{R}_k^{ij}}^{-1} \mathbf{H}_k^{ij} \end{aligned} \tag{2.73}$$

This leads to a linear additive formulation of the correction step:

$$\begin{aligned} \widehat{\mathbf{y}}_k^i &= \widehat{\mathbf{y}}_{k|k-1} + \mathbf{i}_k^i + \textstyle\sum_{j \neq i} a_{ij} \mathbf{i}_k^{ij} \\ \widehat{\mathbf{Y}}_k^i &= \widehat{\mathbf{Y}}_{k|k-1} + \mathbf{I}_k^i + \textstyle\sum_{j \neq i} a_{ij} \mathbf{I}_k^{ij} \end{aligned} \tag{2.74}$$

where information contribution are by definition independent. Factors $a_{ij} \in \mathbb{R}$ are the adjacency coefficients of the communication graph. Note that the information form (2.74) is equivalent to a Least-Squares solution (2.8). Then, the impact of measurements on the computational load only depends on N_a instead of N_a^3. It can even further be reduced to the number of neighbours for agent i $N_i \leq N_a$ by only performing the correction for non-zero adjacency coefficients $a_{ij} \neq 0$.

Collaborative estimation for non-linear and/or non-Gaussian problems remains an active topic of research. Collaborative information filters or KF have been extended to non-linear cases (e.g. Sharma [ST08]). Although these approaches give good results with slightly non-linear measurements, they suffer the same drawbacks as conventional EKF or non-linear information filters. Distributed Particle Filters are also developped (Vazquez [VM17]). However, they often require a higher communication load than deterministic Bayesian filter (Nerurkar [NRM09]). For unknown but bounded uncertainties, collaborative

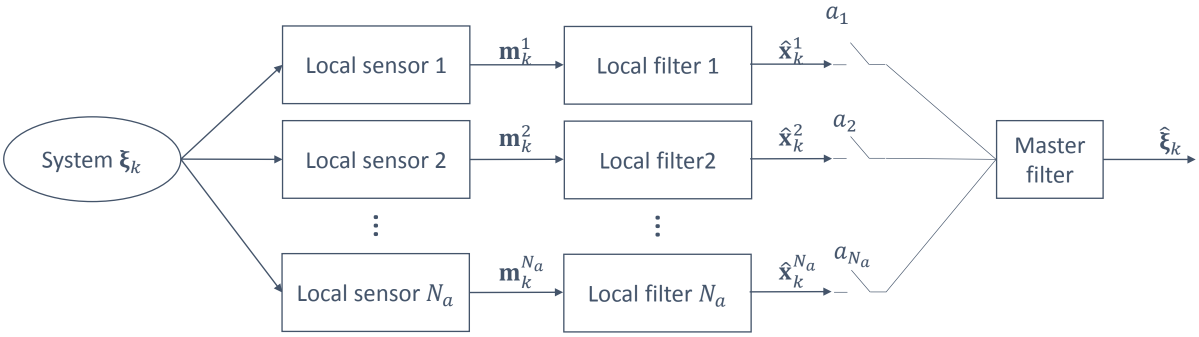


Figure 2.18: Federated filter architecture. Each node consists of a local sensor and a local filter. It broadcasts its measurements to the Master Filter which fuses all received measurements together. In this figure, the Federated Filter is illustrated for a centralised architecture. It can also be extended to distributed schemes.

set-based approaches remain in the same framework than the one introduced in Section 2.1.5. A Constraints Satisfaction Problem can be formulated to estimate a set containing the system's state $\boldsymbol{\xi}$ (Bethencourt [BJ13]).

2.2.3 Federated filters

The centralised and distributed architectures were defined in Section 2.2.1. For both architectures, the formalism introduced in Section 2.2 can be applied. Several ways of organizing the fusion of gathered measurements from the point of view of one agent i will be introduced.

A first filter implementation consists of gathering all measurements and performing the state estimation in a single filter (e.g. Sharma [ST08], Li [LN13]). This strategy can be applied for centralised and distributed architectures, as illustrated in Figure 2.16 and Figure 2.17. However, this single layer fusion may not be robust to measurement ambiguities and measurements outliers.

In order to increase the robustness of fusion architectures, a two-layers architecture was introduced, called *Federated Filter* (Carlson [Car88]). A federated filter architecture consists of a first layer of several *sensor-dedicated local filters* running in parallel. Each local filter deals with measurement $\mathbf{m}^i$ to provide a first estimate $\widehat{\mathbf{x}}_k^i$ of the system's state. A second layer, which consist of one *Master Filter* (MF) fuses the local estimates to obtain a high level estimation $\widehat{\boldsymbol{\xi}}_k$. This architecture is illustrated in Figure 2.18. Under the hypothesis that all uncertainties involved in the second layer are Gaussian, the Master Filter may consists of a Kalman Filter, which can be formulated as an information filter (2.74) (Manyika [Man93]). The Federated Filter architecture can also be used in a distributed way. In this case, each agent has its own Master Filter and receives either local measurements $\mathbf{m}_k^{ij}$ from agents $j \neq i$ and processes itself them into N_a local filters. The Master Filter is shown to be globally optimal, in the sense that it tends to the collaborative Optimal Filter solution (Carlson [Car88]). In practice, if the Master Filter update frequency is lower than that of the local filters, the estimate is conservatively sub-optimal, which means that it remains consistent with the measurements while being slightly pessimistic. This design yields a lower computation load than single layer approaches, and opens the way to more efficient sensor fault detection, failure isolation, and recovery capability. A faulty measurement or local estimate will not deteriorate other local filters' outputs and will be smoothed by them in the Master Filter's fusion. Federated filter architecture also allows non-linearities to be tackled by local dedicated filters in the first layer, while the fusion itself is performed in a linear way by the Master Filter in the second layer.

Carlson's work [Car88] led to various applications, in particular cooperative navigation from non-linear measurements (e.g. Sun [Sun04], Wang [WCG13]).

2.3 STATE CONTROL UNDER STATE UNCERTAINTY AND CONSTRAINTS

Section 2.1 introduced the state estimation problem and some existing solution to retrieve the state $\mathbf{x}_k$ given the past measurements $\mathbf{M}_k$ and the associated uncertainties. The objective of the control loop is bringing the state to a desired state, or following a desired state trajectory. In most cases, the estimation uncertainty (e.g. the estimate covariance) can be neglected after a reasonable number of measurements (or at least asymptotically), which allows the control scheme to be deterministic. The state control block consists of determining the control input $\mathbf{u}_k \in \mathbb{R}^{d_c}$ to bring the next propagated state as close as possible to the desired state, which is equivalent to minimising a cost function expressed at time-step k. If the control problem is not constrained, the control solution can be *reactive*, i.e. only accounting for the current state and desired state, for example pure pursuit or proportional navigation for interception problems (Shneydor [Shn98]). Optimal analytic solutions can even be derived for particular dynamics and cost functions (e.g. aircraft trajectory optimisation, Heymann [HBA96]).

If the state trajectory is constrained, reactive solutions are not sufficient to ensure the constraints satisfaction. To tackle such problems, *predictive* approaches have been introduced (e.g. see Model Predictive Control, MPC [ML99], and Optimal Control [Ber+05]). They rely on a constrained optimisation of a future trajectory, providing a sequence of optimal future control inputs. Periodically repeating the optimisation process while updating the starting point with the state estimate provides a feedback control solution, given the past measurements.

However, in the presence of non-vanishing state uncertainty, the problem becomes more complex. Early works on uncertain state control tackled unconstrained Gaussian state trajectories (e.g. Linear Quadratic Gaussian control, LQG [Ath71]). In order to account for constrained problems, *robust control* approaches consider the worst case trajectory, often subject to bounded uncertainties (e.g. robust MPC [CM87b], tube-based MPC [Lim+10]). However, for some applications, robust constrained control methods may end with conservative solutions. The conservatism can be moderated by relaxing the constraints while still controlling the risk of their violation. To do so, chance constrained optimisation introduces a failure probability constraint which is forced to remain below a desired level of error.

Figure 2.19 illustrates four emblematic control solutions. In this example, the coloured areas represent the forbidden states, and the circles around the successive points of the trajectories are the state uncertainties. The LQG provides the optimal unconstrained solution to a quadratic cost function under linear dynamics and Gaussian uncertainty. The MPC provides an optimal (or near optimal) constrained solution for a deterministic state. In the presence of uncertainty, constraints can be satisfied for the trajectory expectation, but not for all possible trajectory realisations. The robust worst case control provides a solution ensuring that all possible trajectories satisfy the constraints. However, the achieved cost value (e.g. the amount of energy spent on the trajectory) is likely to be high. The chance constrained MPC constitutes a trade-off between the unconstrained solution and the robust solution, by redefining the constraints as a maximum admissible probability of constraint violation.

Section 2.3.1 introduces the framework of constrained trajectory planning and its application to feedback control under uncertainty. Section 2.3.2 introduces the chance constrained optimisation framework and its applications to feedback control.

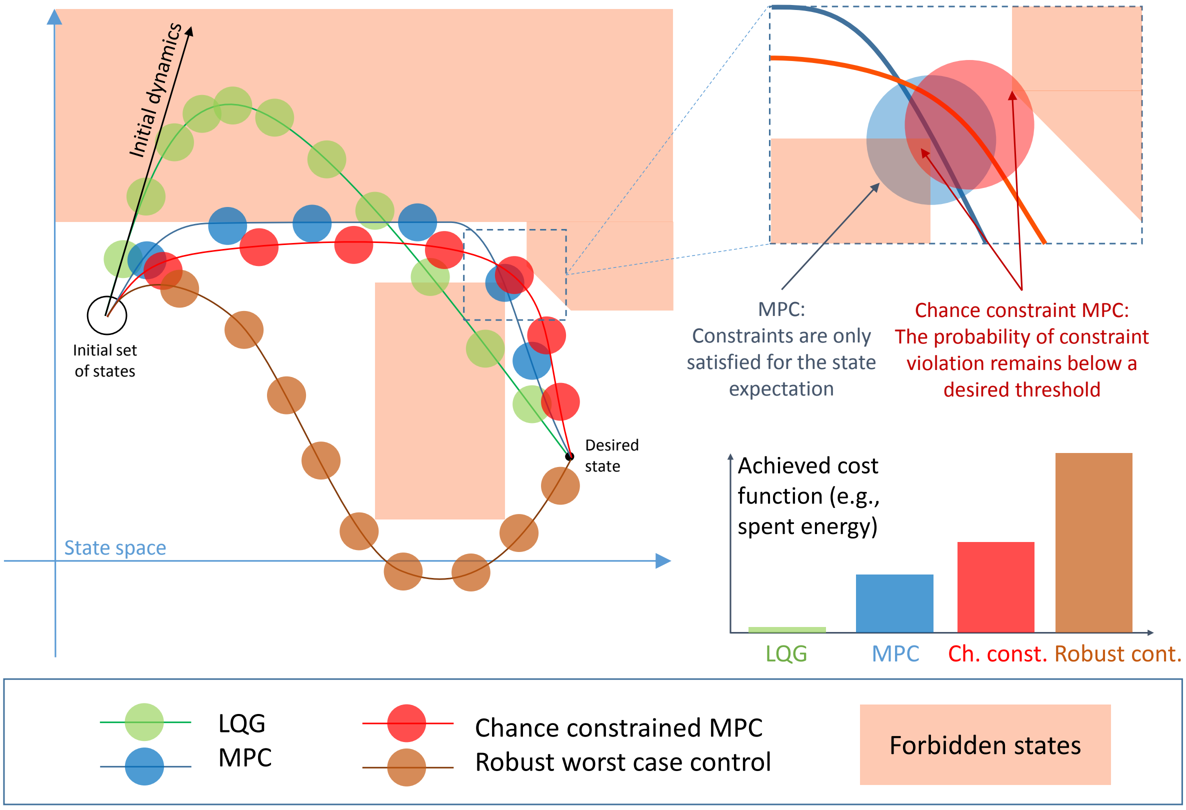


Figure 2.19: Four emblematic optimal control approaches under state uncertainty. Solid lines represent the solution trajectories (expectancies) while filled circles represent the state uncertainties (e.g. density supports for bounded densities). The Linear Quadratic Gaussian (LQG, green trajectory) consists of the optimal unconstrained control solution under Gaussian state uncertainty. The Model Predictive Control (MPC, blue trajectory) consists of a constrained optimisation which does not account for uncertainty. In other words, MPC satisfies the constraints for the state expectation only. The Chance Constrained MPC (red trajectory) consists of keeping the probability of failure under a desired threshold. The robust control approaches (brown trajectory) ensure that every possible state does not violate the constraints. Although robust worst case control approaches yield the safer solution, they are often more conservative than other approaches and may end with no solution.

2.3.1 Trajectory planning

Trajectory optimisation

Consider the following deterministic estimation problem: finding the control sequence $\mathbf{U} = [\mathbf{u}_1^T, ..., \mathbf{u}_K^T]^T \in \mathbb{R}^{d_c K}$ which minimises a cost function $J : \mathbb{R}^{d_c K} \times \mathbb{R}^{d(K+1)} \to \mathbb{R}$ and satisfies the state constraints $G : \mathbb{R}^{d(K+1)} \to \mathbb{R}$ and the control constraints $G_c : \mathbb{R}^{d_c K} \to \mathbb{R}$ along the state trajectory $\mathbf{X} = [\mathbf{x}_0^T, ..., \mathbf{x}_K^T]^T \in \mathbb{R}^{d(K+1)}$:

$$\min_{\mathbf{U}} J(\mathbf{X}, \mathbf{U}), \text{s.t.} \begin{cases} G(\mathbf{X}) \leq 0 \\ \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) \;\; \forall k \in [1, K] \\ G_c(\mathbf{U}) \leq 0 \end{cases} \tag{2.75}$$

A commonly used formulation of the cost function $J(\mathbf{X}, \mathbf{U})$ is:

$$J(\mathbf{X}, \mathbf{U}) = \mathbf{U}^T \mathbf{R} \mathbf{U} + (\mathbf{X} - \mathbf{X}^d)^T \mathbf{Q} (\mathbf{X} - \mathbf{X}^d) + \sum_{k=1}^{K-1} (\mathbf{u}_{k+1} - \mathbf{u}_k)^T \mathbf{S} (\mathbf{u}_{k+1} - \mathbf{u}_k) \tag{2.76}$$

where $\mathbf{R}$, $\mathbf{Q}$, and $\mathbf{S}$ are weighting positive definite matrices respectively quantifying the importance of minimising the following: the control norm (often linked to energy expense), the desired state achievement (which can either be a trajectory to track or a desired terminal state), and the control smoothness (variation between two consecutive control inputs).

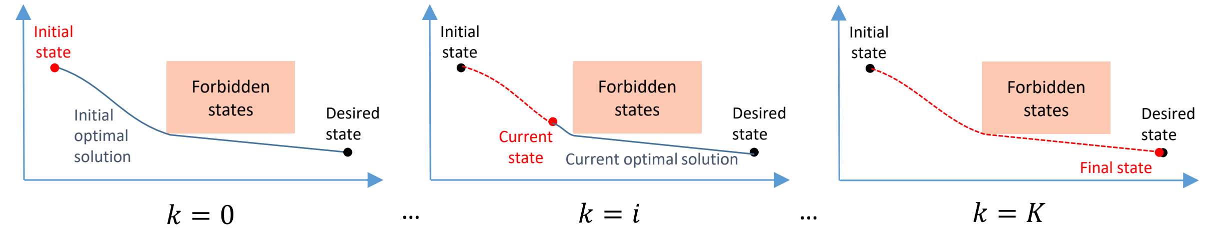
In what follows, the *feasible set* will refer to the set containing all admissible trajectories with respect to the state constraints G:

$$\mathcal{F} \triangleq \left\{ \mathbf{X} \in \mathbb{R}^{d(K+1)} \mid G(\mathbf{X}) \leq 0 \right\} \tag{2.77}$$

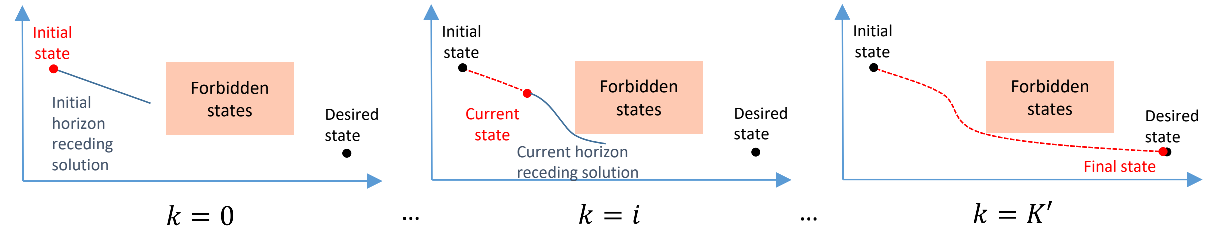
Trajectory optimisation can be applied to feedback control, which consists of solving a new problem of the form (2.75) at each time-step by propagating the trajectory from current time-step k to a given horizon $k + K$. In this context, the initial state of the predicted trajectory $\widehat{\mathbf{x}}_k$ is provided by a state estimator (sometimes referred as an *observer*) given a trajectory of past measurements. Two main ways exist for defining the prediction horizon. The first one consists of a final time horizon K which corresponds to the time-step when the system reaches the terminal desired state. At each time-step k, the optimisation is then performed on the basis of a prediction from k to K, as illustrated in Figure 2.20 (a). The second one is the Receding Horizon Control scheme (RHC, [MM89]), also referred as Model Predictive Control (MPC [ML99]). The RHC consists of solving problem (2.75) from current time-step k to horizon $k + K$, with K a fixed horizon length, as illustrated in Figure 2.20 (b). Both approaches result for each time-step k in a future control sequence $\mathbf{U} = [\mathbf{u}_{k+1}^T, ..., \mathbf{u}_{k+K}^T]^T$ whose only the first control input $\mathbf{u}_{k+1}$ is performed to propagate the actual state $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_{k+1}) + \mathbf{w}_{k+1}$. At the next time-step $k + 1$, the initial predicted state is updated via the current state estimation.

Differentiable optimisation

The most common optimisation methods to solve problem (2.75) belong to the differentiable optimisation scheme. The term differential refers to the assumption that the cost function J and the constraints G, $\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k)$, and G_c are differentiable with respect to the optimisation variable $\mathbf{U} = [\mathbf{u}_1^T, ..., \mathbf{u}_K^T]^T$. Note that since $\mathbf{X} = [\mathbf{x}_0^T, ..., \mathbf{x}_K^T]^T$ depends on $\mathbf{U}$, $G(\mathbf{X})$ also depends on $\mathbf{U}$. Non-linear differentiable



(a) Final state horizon control



(b) Receding Horizon Control (RHC)

Figure 2.20: Two ways of integrating constrained optimisation into a control loop: The final state horizon control (a) consists of regularly updating the optimal trajectory on a non-fixed time horizon (the trajectory lasts until it reaches the desired state). The Receding Horizon Control (RHC) consists of regularly updating the optimal trajectory on the basis of a fixed receding horizon. Although it yields sub-optimal solutions, it has a lower computational load than optimal control approaches, which is of interest for embedded applications.

optimisation solvers under inequality constraints rely on the Karush–Kuhn–Tucker conditions (KKT, [KT14]) which state necessary conditions for admissible points.

Early approaches to satisfy KKT condeitions considered affine constraint satisfaction into gradient descent methods (e.g. Calamai1 [CM87a]). Another way to deal with constraints is using cost penalisation, i.e. writing an equivalent unconstrained problem (or a series of problems) by adding penalisation terms in the cost function (e.g. Conn [CGT91]). If not carefully done, this may result in an unsolvable problem. Nevertheless, advanced methods use this idea to tackle linear and non-linear convex problems (e.g. interior-point method [PW00]). Other methods consider the resolution of a series of simpler quadratic sub-problems to approach non-linear constrained problems, for example, the Sequential Quadratic Programming methods (SQP, see Boggs [BT95], Gill [GMS05]). SQP can be extended to non-convex problems (e.g. Curtis [CO12]).

In what follows, it is assumed that:

- The cost function J is differentiable with respect to $\mathbf{U}$;
- The dynamical model f is differentiable with respect to the control input;
- The constraint G on the trajectory is differentiable with respect to $\mathbf{U}$;
- The constraint G_c on the control sequence is differentiable.

Trajectory planning under uncertainty and application to control

When the initial state and the dynamics become uncertain, the trajectory turns to a random variable. Problem (2.75) becomes a stochastic problem and can no more be solved by deterministic optimisation methods anymore:

$$\min_{\mathbf{U}} J(\mathbf{X}, \mathbf{U}), \text{s.t.} \begin{cases} G(\mathbf{X}) \leq 0 \\ \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \ \forall k \in [1, K] \\ \mathbf{x}_0 \sim p(\mathbf{x}_0) \\ \mathbf{w}_k \sim \pi^x_{k|k-1}(\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k)) \ \forall k \in [1, K] \\ G_c(\mathbf{U}) \leq 0 \end{cases} \tag{2.78}$$

where $p(\mathbf{x}_0)$ is the initial state density and $\pi_{k|k-1}$ the process noise density at time-step k.

A variety of approaches have been introduced to solve problem (2.78) whose cost function and constraints depending on $\mathbf{X}$ are stochastic. In the large majority of them, the cost function is replaced with its expectation $E_{\mathbf{X}}[J(\mathbf{X}, \mathbf{U})]$. A first solution to tackle uncertain constraints is replacing them with penalisation terms which consist of their expectation in the cost function (see *penalisation methods* in the previous section). However, this only guarantees that the constraints are satisfied for the expected value of the trajectory.

In order to ensure that constraints are satisfied for all possible states, robust approaches consist of deterministically solving the problem for the worst case scenario (e.g. see Bemporad [BM99]). Several methods have been introduced, for example min-max formulations (Campo [CM87b]), or set-based Model Predictive Control (e.g. with parallelotopes [BG00], ellipsoids [KRS00], or intervals [LP03]), that rely on the same scheme as that described in Section 2.1.5.

However, these approaches are often conservative and may end with no solution. To reduce the conservatism of such solutions while controlling the risk of constraint violation, the chance constraint concept has been introduced (Charnes [CCS58]), and is described hereafter.

2.3.2 Stochastic optimisation under chance constraint

Chance constrained optimisation

It is possible to relax the worst case constraints while keeping the problem deterministic. In the Chance Constrained optimisation (Charnes [CCS58]), the constraints $G(\mathbf{X}) \leq 0$ are re-written in terms of probability of being violated $P(G(\mathbf{X}) > 0)$, or *failure probability*. The failure probability can be forced to remain below a desired threshold $\delta \in [0, 1]$ (equivalently, one can define the probability $P(G(\mathbf{X}) \leq 0)$ of satisfying the constraints and force it to remain greater than $1 - \delta$). This makes it possible to explicitly estimate and master the risk of constraint violation. The cost function J must also be re-written in a deterministic way. It is usually replaced with its expectation, which yields the following problem:

$$\min_{\mathbf{U}} E_{\mathbf{X}}[J(\mathbf{X}, \mathbf{U})], \text{s.t.} \begin{cases} P(G(\mathbf{X}) > 0) \leq \delta \\ G_c(\mathbf{U}) \leq 0 \end{cases} \tag{2.79}$$

provided that the initial state and the process noise are characterised by known density: $\mathbf{x}_0 \sim p(\mathbf{x}_0)$, and $\mathbf{w}_k \sim \pi^x_{k|k-1}(\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k)) \ \forall k \in [1, K]$.

The failure probability is expressed by:

$$P(G(\mathbf{X}) > 0) = \int_{X \notin \mathcal{F}} p(\mathbf{X}) \mathrm{d}\mathbf{X} \tag{2.80}$$

where $\mathcal{F}$ is the feasible set (2.77) and where $p(\mathbf{X})$ represents the joint states densities at every predicted time-steps:

$$p(\mathbf{X}) \triangleq p(\mathbf{x}_0) \prod_{i=1}^{K} p(\mathbf{x}_i | \mathbf{x}_0) \tag{2.81}$$

where $p(\mathbf{x}_0)$ is the initial state density and $p(\mathbf{x}_i|\mathbf{x}_0) \triangleq \int ... \int p(\mathbf{x}_0) p(\mathbf{x}_1|\mathbf{x}_0) \mathrm{d}\mathbf{x}_0 ... p(\mathbf{x}_i|\mathbf{x}_{i-1}) \mathrm{d}\mathbf{x}_{i-1}$ the i^{th} predicted state density (i times Chapman-Kolmogorov convolution (2.4)) The failure probability can also be denoted $P(X \notin \mathcal{F})$. Some authors have tackled the normal law case [BLW06] and the uniform law case [CEG06]. Nevertheless, these approaches are limited to analytical state densities and convex feasible sets.

Note that the chance constraint formulation (2.80) quantifies the failure probability on the whole trajectory $\mathbf{X} = [\mathbf{x}_0^T, ..., \mathbf{x}_K^T]^T$, given a trajectory density p. Instantaneous formulations have also been introduced (e.g. Heirung [Hei+18], Calafiore [CF13], Mesbah [Mes+14]) and consist of the probability of constraint violations on the state at a given predicted time-step. Such formulations are not equivalent to (2.80) in the general case. Consider an example where the universe of $\mathbf{X}$ is a finite set of 10 possible trajectories i, made of 10 time-steps k: $\mathbf{X}^i = [\mathbf{x}_0^i, ..., \mathbf{x}_9^i]^T$. Assume that, for every control sequence k, the state $\mathbf{x}_k^{i=k+1}$ violates the constraints at time-step k. As a result, the instantaneous constraint violation formulation yields a value of 0.1 at each time-step. However, the global failure probability formulation (2.80) yields a value of 1, since all possible trajectories violate the constraints at lest once.

Sample methods for chance constrained optimisation

To tackle non-analytic densities and non-convex feasible sets, the *sampled based* chance constraint optimisation consists of propagating a set of weighted point-wise trajectories to evaluate the failure probability (Pagnoncelli [PAS09]). It has been applied to trajectory planning for control problems (Particle Control [Bla+10; BOW11]). The initial state density $p(\mathbf{x}_0)$ consists of a mixture of N Dirac deltas of centers $\mathbf{x}_0^i$ and weights w_0^i, likewise in the Particle Filter (2.19). Each particle $\mathbf{x}_0^i$ can be dynamically propagated along the whole trajectory, given a control sequence $\mathbf{U} = [\mathbf{u}_1^T, ..., \mathbf{u}_K^T]^T$ and K realisations of the process noise $\mathbf{w}_k^i$:

$$\begin{aligned} \mathbf{x}_1^i &= f(\mathbf{x}_0^i, \mathbf{u}_1) + \mathbf{w}_1^i \\ &\vdots \\ \mathbf{x}_K^i &= f(\mathbf{x}_{K-1}^i, \mathbf{u}_K) + \mathbf{w}_K^i \end{aligned} \tag{2.82}$$

The failure probability can then be approached by:

$$\widehat{P}(\mathbf{X} \notin \mathcal{F}) = \sum_{i=1}^{N} w_k^i \mathbf{1}_{\mathbf{X}^i \notin \mathcal{F}} \tag{2.83}$$

where $\mathbf{X}^i = [{\mathbf{x}_0^i}^T, ..., \mathbf{x}_K {i^T}]^T$. This approach is guaranteed efficient for a large number of sample-trajectories (theoretically infinite). However, the number of particle is often constrained in practice due to computational limitations, and cannot guarantee the accuracy of the method. Furthermore, the sample-based

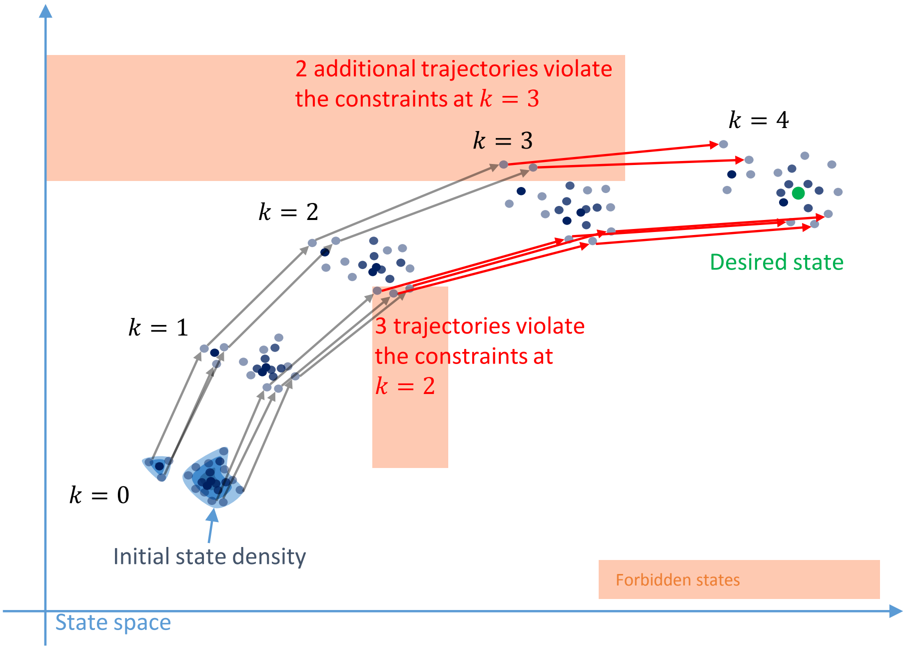


Figure 2.21: Particle Control scheme (Blackmore [Bla+10]): the estimated failure probability consists of the sum of the weights of the predicted trajectories violating the constraints only once

failure probability estimation is not differentiable with respect to the control sequence which constitutes the optimisation variables. This restricts Particle Control to linear programming optimisation techniques, which limits its use to linear dynamics (Blackmore [Bla+10]). Nevertheless, this kind of approach opens the way to non-parametric densities and non-convex feasible sets problems.

Figure 2.21 illustrates the Particle Control scheme for a case where five trajectories violate the constraints. If all weights are equal to $w_0^i = \frac{1}{N}$ and $N = 20$, then $P(\mathbf{X} \notin \mathcal{F}) = 0.25$.

Application to guidance and control

Chance constrained trajectory planning can be applied to feedback guidance and control problems when used in conjunction with a state estimation algorithm. Its integration in Receding Horizon Control (RHC) and Model Predictive Control (MPC) has been widely considered (Schwarm [SN99], Oldewurtel [OJM08], Primatesta [PB17], Heirung [Hei+18]). The sample-based formulation (Blackmore [Bla+10]) yielded several works on sample-based MPC and convergence studies (e.g. Calafiore [CF13]).

The chance constrained RHC scheme consists of solving problem (2.79) at each time-step k with an initial state density $p(\mathbf{x}_k|\mathbf{M}_k)$ (provided, for example, by a probabilistic estimator), along the trajectory from current time-step k to horizon $k+K$. It results in a future control sequence $\mathbf{U} = [\mathbf{u}_{k+1}^T, ..., \mathbf{u}_{k+K}^T]^T$ whose only the first control input $\mathbf{u}_{k+1}$ is performed to propagate the actual state $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_{k+1}) + \mathbf{w}_{k+1}$. Note that if measurements stop to be available during a period of time starting at time-step k, the determined control input ensures that the chance constraint is satisfied until the prediction horizon $k+K$, since the optimisation problem does not account for possible measurements in the uncertainty propagation.

3

METHODOLOGY

This chapter provides elements on the work's methodology, in particular concerning the simulations and their interpretation. The proposed estimation and trajectory planning algorithms are provided with a general formulation. Simulations are presented in order to illustrate the theoretical results and to evaluate the practical impact of the work.

Section 3.1 introduces the dynamical models used to evaluate the state estimation algorithms (Chapters 4 and 5). Linear or time-varying linear dynamical models associated with negligible process noise were chosen in order to focus on the impact of measurements ambiguities and non-linearities. The considered models are the linear *double integrator* dynamics, and the *Inertial Measurement Unit drift dynamics* (IMU).

Section 3.2 introduces the studied observation models. The work focuses on the Terrain Aided Navigation (TAN) application. TAN constitutes a severely ambiguous problem and allows estimation algorithms to be compared in a significant context. Furthermore, it belongs to a larger class of problems that consist of state estimation from vector field measurements. Additional observation models such as *Cartesian measurements* and *polar measurements* may be used in addition to the TAN problem, but do not constitute the focus of the work.

Section 3.3 presents five TAN scenarios covering a large variety of measurements ambiguities. They are featured and classified in terms of two ambiguity criteria: the lack of measurement information and the level of posterior multimodality they yield.

Section 3.4 introduces the dynamical models used to evaluate the trajectory planning algorithms. Two linear models are introduced: the *longitudinal aircraft dynamics* and the *Clohessy-Wiltshire orbital relative dynamics*. A non-linear model will be also considered: the fixed-wing aircraft *unicycle dynamics*.

Section 3.5 defines the numerical criteria used to interpret the simulations: the Root Mean Square Error (RMSE); the average confidence of an estimator; the non-convergence rate; the pessimism rate; and the RMSE rate.

3.1 DYNAMICAL MODELS FOR STATE ESTIMATION

Double integrator

In the following chapters, Double Integrator will refer to the dynamical model defined by:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{I}_3 \end{bmatrix} \mathbf{u} + \mathbf{w} \tag{3.1}$$

where the state $\mathbf{x} = \left[\mathbf{p}^T, \mathbf{V}^T\right]^T = \left[x, y, z, V_x, V_y, V_z\right]^T$ consists of three Cartesian positions and three velocities, the control $\mathbf{u} \in \mathbb{R}^3$ consists of three accelerations, and $\mathbf{w} \in \mathbb{R}^6$ is the process noise. In practice, it can be approximated by the Euler method, for a time-step $dt \in \mathbb{R}^+$:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{I}_3 & dt\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} \mathbf{0}_3 \\ dt\mathbf{I}_3 \end{bmatrix} \mathbf{u}_k + \mathbf{w}_k \tag{3.2}$$

In this thesis, the time-step is set to $dt = 0.1$ s for the simulations.

IMU model

Consider a vehicle whose state vector consists of its geographical position, velocity, and attitude in Earth frame:

$$\mathbf{x}_k = \left[\mathbf{p}^T, \mathbf{V}^T, \boldsymbol{\psi}^T\right]^T \in \mathbb{R}^9 \tag{3.3}$$

where $\mathbf{p} = \left[p_\lambda, p_\phi, p_h\right]^T$ is the geographical position (respectively latitude (*rad*), longitude (*rad*), altitude (*m*)), $\mathbf{V} \in \mathbb{R}^3$ is the velocity vector (ms^{-1}), and $\boldsymbol{\psi} = [\psi, \theta, \varphi]^T$ is the attitude (euler angles in *rad*, respectively heading, pitch, roll). These values are unknown and have to be estimated by the navigation system. The most commonplace navigation approach in the aerospace field relies on an Inertial Measurement Unit (IMU).

The IMU measures the vehicle's acceleration $\mathbf{m}_k^a$ and the vehicle angular rate $\mathbf{m}_k^\omega$ in the inertial frame, via respectively three accelerometers and three gyrometers. The inertial frame is defined as a Galilean reference frame. The initial vehicle's state $\mathbf{x}_0$ is assumed to be known, as well as a gravity model providing the gravitational accelerations for one given state. The combination of $\mathbf{m}_k^a$ and the gravity model yields the non-gravitational accelerations (specific accelerations), i.e. the vehicle acceleration in Earth frame. The measured angular rate $\mathbf{m}_k^\omega$ is also compensated by the Earth rotation rate, which yields the specific angular rate. The IMU is coupled to a double integrator which integrates the specific accelerations and angular rates to provide an estimate of the vehicle's state, called *IMU state* $\widehat{\mathbf{x}}_k^{IMU}$

IMU accelerometers and gyrometers measurements are uncertain and can be modelled as follows:

$$\begin{aligned} \mathbf{m}_k^a &= \mathbf{a}_k + \mathbf{b}_{a,k} + \mathbf{v}_k^a \\ \mathbf{m}_k^\omega &= \boldsymbol{\omega}_k + \mathbf{b}_{\omega,k} + \mathbf{v}_k^\omega \end{aligned} \tag{3.4}$$

where $\mathbf{a}_k \in \mathbb{R}^3$ and $\boldsymbol{\omega}_k \in \mathbb{R}^3$ are the actual vehicle acceleration and the actual angular rate in the inertial frame. Noises $\mathbf{v}_k^a \in \mathbb{R}^3$ and $\mathbf{v}_k^\omega \in \mathbb{R}^3$ are associated with normally distributed random walk process characterised by covariances $\mathbf{R}_a \in \mathbb{R}^{3\times3}$ and $\mathbf{R}_\omega \in \mathbb{R}^{3\times3}$. Vectors $\mathbf{b}_a \in \mathbb{R}^3$ and $\mathbf{b}_\omega \in \mathbb{R}^3$ are called *IMU bias*. By iterative integration of IMU measurements, the IMU state drifts from the actual state and may rapidly provide poor navigation information.

The IMU errors can be modeled as a state vector $\delta\mathbf{x}_k$ which consists of the gap between the actual state and the IMU state (9 state variables) and the IMU bias (6 state variables):

$$\delta\mathbf{x}_k = \left[\delta\mathbf{p}_k^T, \delta\mathbf{V}_k^T, \delta\boldsymbol{\psi}_k^T, \mathbf{b}_{a,k}^T, \mathbf{b}_{\omega,k}^T\right]^T \in \mathbb{R}^{15} \tag{3.5}$$

where $\delta\mathbf{p}_k \in \mathbb{R}^3$ is the metric position error, $\delta\mathbf{V}_k$ the velocity error, $\delta\boldsymbol{\psi}_k$ the attitude angles errors, and $\mathbf{b}_{a,k}$, and $\mathbf{b}_{\omega,k}$ the sensor bias. The inertial errors dynamics can be described by a linearised model:

$$\delta\mathbf{x}_k = \mathbf{A}_k\delta\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{w}_k \tag{3.6}$$

where matrices $\mathbf{A}_k \in \mathbb{R}^{15\times15}$ and $\mathbf{B}_k \in \mathbb{R}^{15\times12}$ depend on the state, the gravity model, the measured specific acceleration $\mathbf{m}_k^a$, and the IMU characteristics. The process noise $\mathbf{w}_k \in \mathbb{R}^{12}$ is a Gaussian random process. Note that model (3.6) is independent from the carrier vehicle dynamical model, since the dynamics is directly measured and integrated by the IMU and integrator system. Also, no explicit control input is involved in (3.6). A complete derivation of inertial equations and inertial errors model can be found in Britting [Bri71] and Dahia [Dah05]. Note that some more precise models can be used to refine the IMU drift dynamics by involving additional variable such as sensors misalignments, or the temperature of the system.

The inertial state has to be corrected using additional measurements (e.g. provided by GNSS, radio navigation, or optical sensors). Knowledge of the IMU state drift allows the problem to be formalised as a state estimation scheme. The use of an estimator (Section 2.1.1) makes it possible to estimate the IMU errors (3.6) and to retrieve the actual state $\mathbf{x}_k$ as a corrected IMU state. This estimation process is called *IMU hybridisation*. The estimator is referred to as *hybridisation filter* and to the additional sensors as *hybridisation sensors*. Figure 3.1 illustrates the IMU hybridisation scheme.

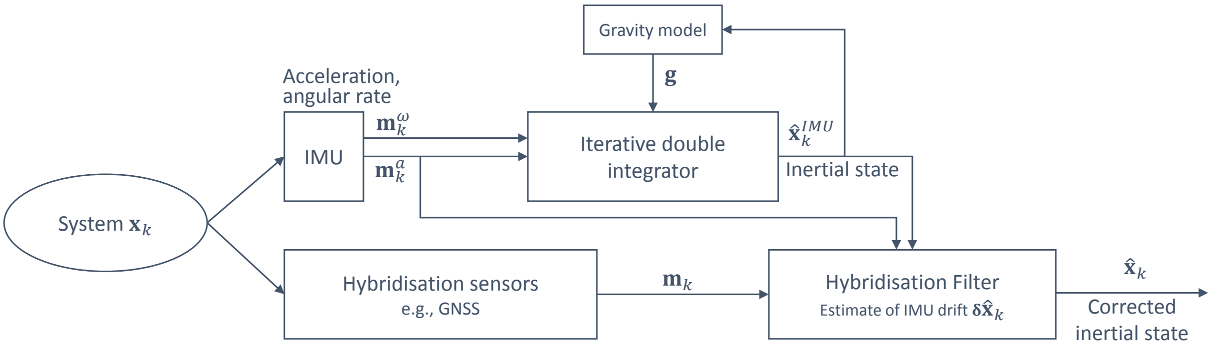


Figure 3.1: IMU hybridization scheme: the IMU measures the vehicle accelerations and angular rates that are integrated to produce an IMU state estimate. Since this estimate drifts, an hybridisation of the IMU with additional measurements is needed, which provides a corrected state estimate.

3.2 OBSERVATION MODELS FOR STATE ESTIMATION

Cartesian measurements

Cartesian measurements refer to the following observation model:

$$\mathbf{m}_k = \begin{bmatrix}\mathbf{I}_3 & 0_3\end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \in \mathbb{R}^3 \tag{3.7}$$

where the state consists of three positions $\mathbf{p}_k^T \in \mathbb{R}^3$ and three velocity variables $\mathbf{v}_k^T \in \mathbb{R}^3$: $\mathbf{x} = [\mathbf{p}_k^T, \mathbf{v}_k^T]^T \in \mathbb{R}^6$. This scheme can be extended to larger states (including for example attitude angles) and to other measurements (e.g. velocity measurements).

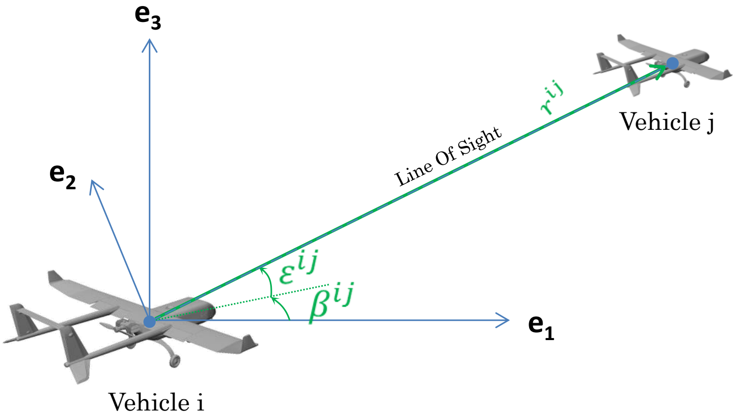


Figure 3.2: Relative measurements $(\beta^{ij}, \varepsilon^{ij}, r^{ij})$

Polar tracking

In the following, the polar tracking measurements are defined in the context of relative sensing from a *chaser* vehicle i toward a *target* vehicle j. Their relative Line Of Sight (LOS) can be defined by a Cartesian vector $\mathbf{r}^{ij} \triangleq [x^{ij}, y^{ij}, z^{ij}]^T \in \mathbb{R}^3$. Measurements consist of the bearing angle $\beta^{ij} \in \mathbb{R}$, the elevation angle $\varepsilon^{ij} \in \mathbb{R}$, and the range $r^{ij} > 0$. As illustrated in Figure 3.2, their relationship with $\mathbf{r}^{ij}$ is defined by:

$$\begin{bmatrix} \beta^{ij} \\ \varepsilon^{ij} \\ r^{ij} \end{bmatrix} = \begin{bmatrix} \operatorname{atan}\left(\frac{y^{ij}}{x^{ij}}\right) \\ \operatorname{atan}\left(\frac{z^{ij}}{\sqrt{x^{ij^2}+y^{ij^2}}}\right) \\ \sqrt{x^{ij^2} + y^{ij^2} + z^{ij^2}} \end{bmatrix} + \mathbf{v}_k \in \mathbb{R}^3 \tag{3.8}$$

Terrain Aided Navigation (TAN)

The Terrain Aided Navigation problem will be referred to as TAN. It consists of estimating the state of a mobile system (e.g. an aircraft) by only measuring the relative altitude with respect to the flown-over terrain and an embedded terrain altitude map. This measurement can be performed by a radar-altimeter or a LASER pointing toward the ground. TAN is of strong interest when exteroceptive sources of navigation are unavailable (e.g. GNSS or radio-navigation). Such situations may occur when GNSS is jammed or when the signal is unreachable. In this work, the TAN sensor is assumed to provide the distance between the system and the ground along the vertical direction, i.e. the line between the system's center of mass and the center of the Earth (or the flown-over body in general). Figure 3.3 illustrates the following observation equation:

$$m_k = z_k - DEM(\lambda_k, \phi_k) + v_k \tag{3.9}$$

where (λ_k, ϕ_k, z_k) are the system's latitude, longitude, and altitude, $DEM : \mathbb{R}^2 \to \mathbb{R}$ is the embedded terrain map and $v_k \in \mathbb{R}$ is the measurement noise. Unless a different measurement noise is specified, the default density is a truncated Gaussian function defined by:

$$p(\mathbf{m}_k|\mathbf{x}_k) \propto \exp\left(-\frac{1}{2}(\mathbf{m}_k - h(\mathbf{x}_k))^T \mathbf{R}^{-1}(\mathbf{m}_k - h(\mathbf{x}_k))\right) \mathbf{1}_{[\mathbf{v}_k]} \tag{3.10}$$

Where $\mathbf{R}$ is a scalar covariance of 15^2 m^2, and the density support is $[\mathbf{v}_k] = [-3 \times 15, +3 \times 15]$ m (e.g. see Murangira [Mur+11]).

There is no analytic description of *DEM*, which is assumed to be obtained from an embedded terrain map. The system trajectory is assumed to be rectilinear uniform, and the measurement rate is 10Hz.

3.3 TERRAIN AIDED NAVIGATION SCENARIOS AND THEIR AMBIGUITIES

The estimation algorithms are tested on several terrain scenarios, corresponding to different levels of ambiguities and information. Appendix A provides a definition of information, which quantifies the informativeness of the measurements with regard to the state. Appendix B provides a definition of measurement ambiguity in terms of posterior multimodalities, i.e. the number of maxima of the posterior theoretical conditional density (2.5). Several conditions can cause multimodalities, such as the non-injectivity of the observation model (2.3). The studied scenarios consist of various terrain types:

1. Scenario TAN Alps: a somewhat ambiguous terrain, plotted in Figure 3.4. This terrain is a typical alpine topography taken from [Mur14, Scenario 1]. The resolution is 100 m. The trajectory duration is 90 s (900 time-steps). The vehicle velocity is about 230 ms^{-1}. An initial state uncertainty standard deviation of 1 km is assumed in position and 4.2 ms^{-1} in velocity. Note that the geographic coordinates of this scenario are fictitious.

2. Scenario TAN Canyon: a severely ambiguous terrain, plotted in Figure 3.5. This terrain is also an alpine topography taken from [Mur14, Scenario 3]. The resolution is 100 m. The trajectory duration is 100 s (1000 time-steps). The vehicle velocity is about 230 ms^{-1}. An initial state uncertainty standard deviation of 1 km is assumed in position and 4.2 ms^{-1} in velocity. Note that the geographic coordinates of this scenario are fictitious.

3. Scenario TAN La Reunion: La Reunion island, plotted in Figure 3.6. This scenario consists of a flyover of the sea during the first part, which only provides information on altitude (it is equivalent to a flat terrain). During the second part, the volcanic island is flown-over, which finally provides information on latitude and longitude. This terrain is available on the IGN website (French *Institut Géographique National*) [IGN18]. The resolution is 250 m. The trajectory duration is 90 s (900 time-steps). The vehicle velocity is about 230 ms^{-1}. An initial state uncertainty standard deviation of 1 km is assumed in position and 4.2 ms^{-1} in velocity.

4. Scenario TAN Moon flyby: a lunar flyby scenario, plotted in Figure 3.7, obtained from Kaguya mission [Kat+08], and available for example on the *Map a planet* website [USG18]. The map resolution is 1 km. A circular equatorial orbit portion of 250 s (2500 time-steps) at an altitude of 100 km and a velocity of 1.6×10^3 ms^{-1} is simulated. An initial state uncertainty standard deviation of 40 km is assumed in position and 6.7 ms^{-1} in velocity. This scenario is a realistic case of severe ambiguities due to lunar craters. Furthermore, planetary exploration in general is a good example of where GNSS-denied navigation applies.

5. Scenario TAN Sinusoidal terrain: a severely ambiguous theoretical terrain, plotted in Figure 3.8, defined by:

$$z = \begin{cases} 10^4 \times \sin(a\lambda) & \text{if } \lambda > 0.28^\circ \\ 10^4 \times \sin(a\phi) & \text{else} \end{cases} \tag{3.11}$$

with $a = \frac{2\pi}{0.05}$. The resolution is 100 m. The trajectory duration is 120 s (1200 time-steps). The vehicle velocity is about 200 ms^{-1}. An initial state uncertainty standard deviation of 1 km is assumed in position and 4.2 ms^{-1} in velocity.

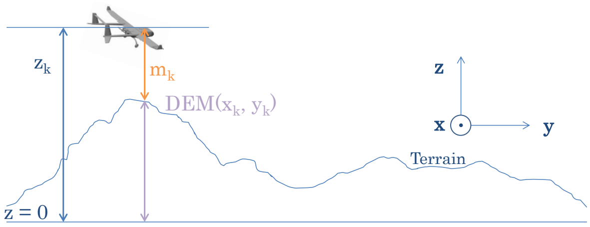


Figure 3.3: Elevation measurement m_k^i in terrain navigation

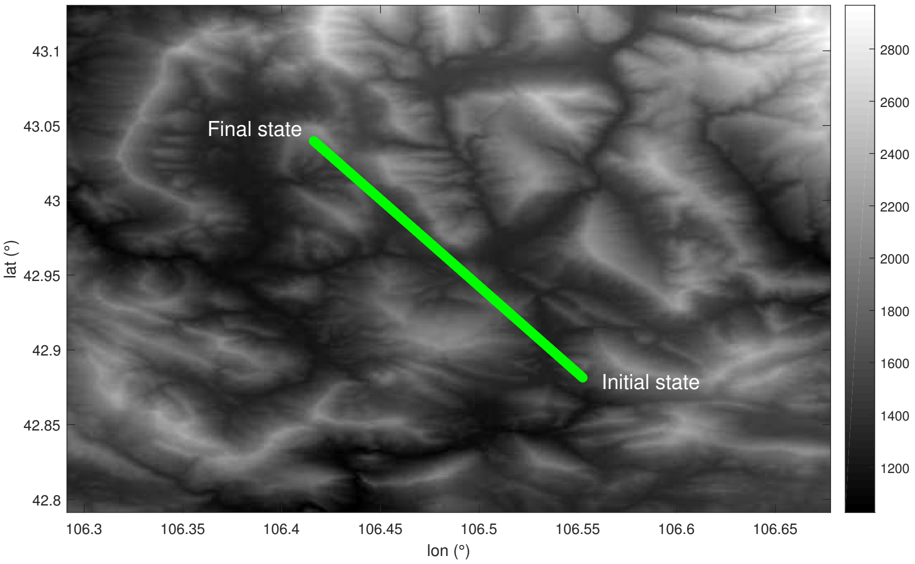


Figure 3.4: Terrain map for Scenario TAN Alps, with the terrain elevation in meters (gray scale), and trajectory (green line)

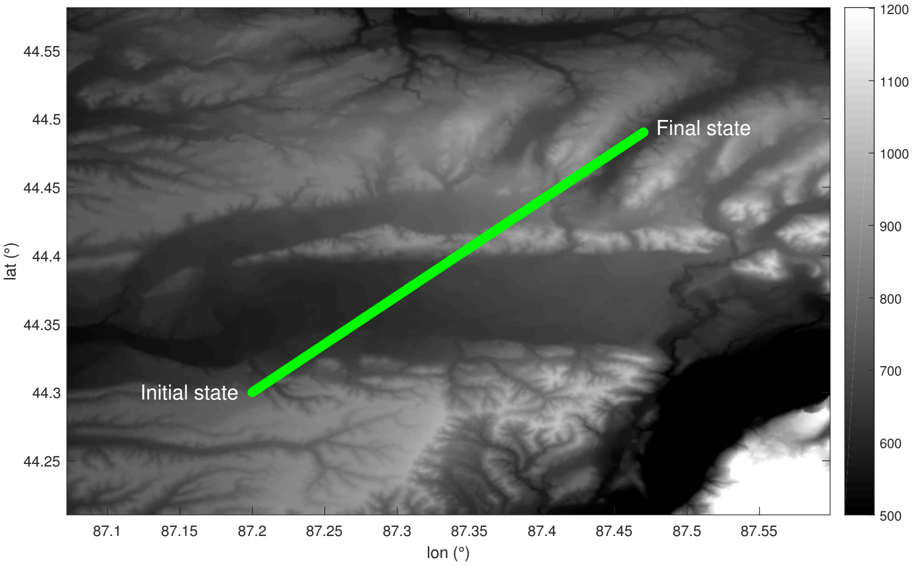


Figure 3.5: Terrain map for Scenario TAN Canyon, with the terrain elevation in meters (gray scale), and trajectory (green line)

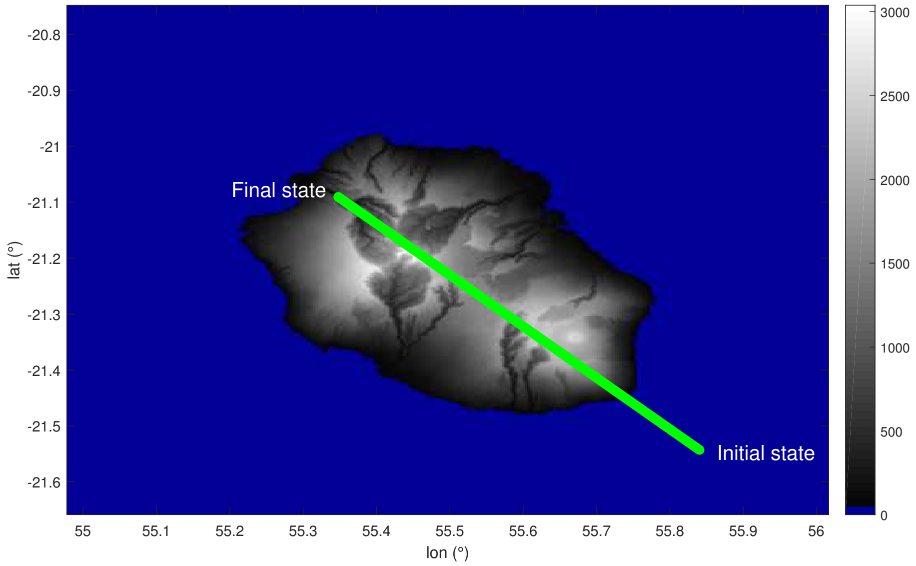


Figure 3.6: Terrain map for Scenario TAN La Reunion, with the terrain elevation in meters (gray scale), and trajectory (green line)

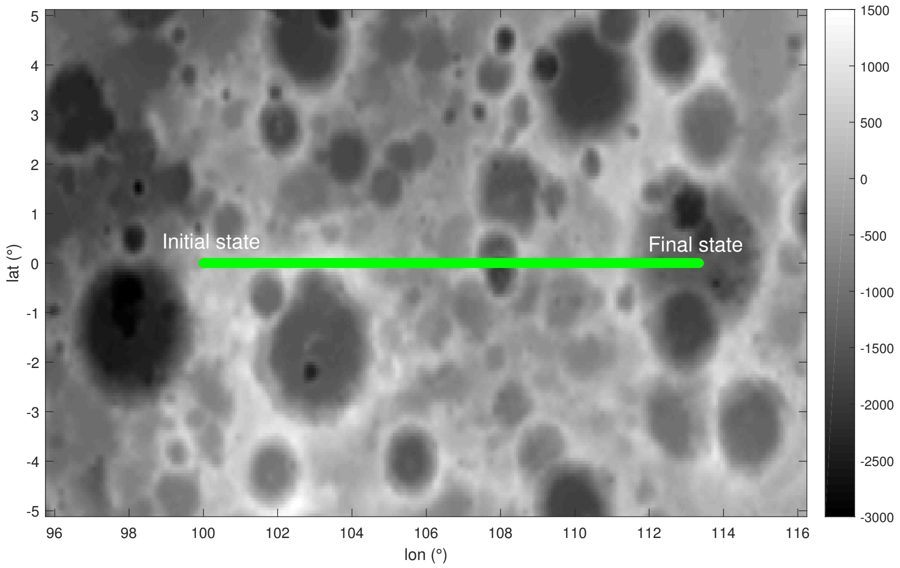


Figure 3.7: Terrain map for Scenario TAN Moon flyby, with the terrain elevation in meters (gray scale), and trajectory (green line)

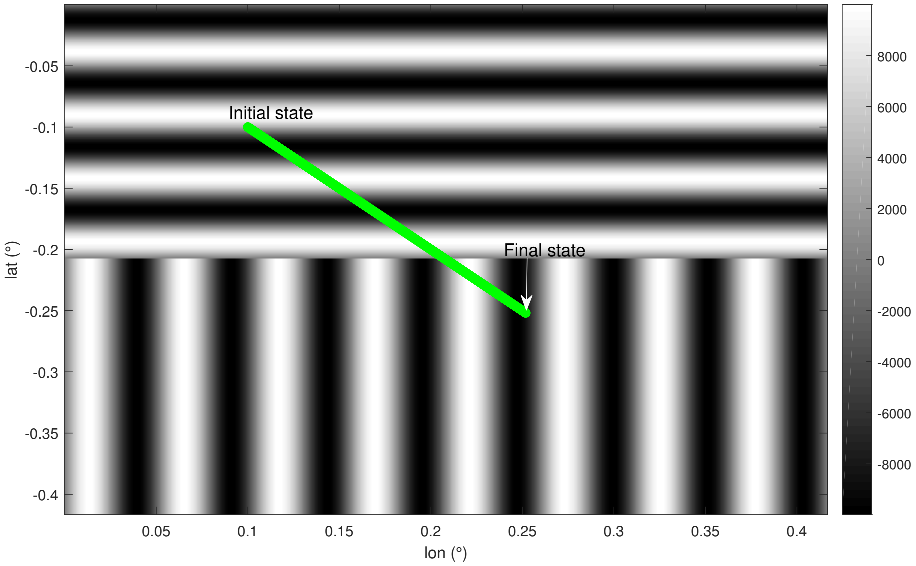


Figure 3.8: Terrain map for Scenario TAN Sinusoidal terrain, with the terrain elevation in meters (gray scale), and trajectory (green line)

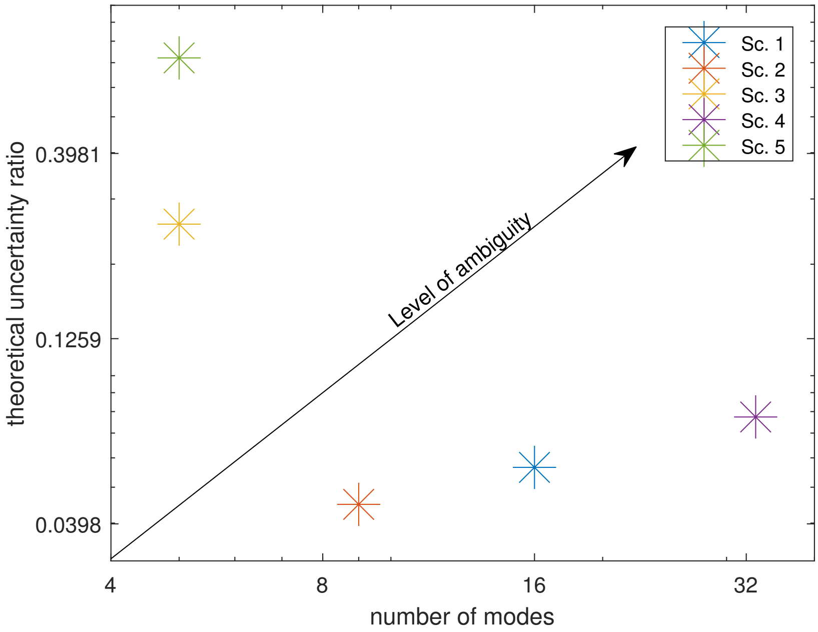


Figure 3.9: Quantification of the scenarios measurement ambiguity for Sc. 1 (Scenario TAN Alps), Sc. 2 (Scenario TAN Canyon), Sc. 3 Scenario TAN La Reunion), Sc. 4 (Scenario TAN Moon flyby), and Sc. 5 (Scenario TAN Sinusoidal terrain)

The measurements ambiguities can be evaluated in terms of informativeness (see Appendix A), and posterior multimodalities (see Appendix B). Two criteria were chosen to evaluate these aspects:

- The *theoretical uncertainty ratio* defined by $\frac{1}{K\sqrt{\rho_1}} \sum_{k=1}^{K} \sqrt{\rho_k}$, where ρ_k is the maximum eigenvalue of the posterior state density covariance $\mathrm{Cov}\left[\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{M}_k)\right]$. The covariance is evaluated at each time-step of a given scenario whose trajectory consists of K time-steps. This criterion quantifies the lack of information in the measurements.
- The highest number of modes encountered by the posterior state density $p(\mathbf{x}_k|\mathbf{M}_k)$. This criterion quantifies the ambiguity level due to the terrain non-linearities and profile similarities.

These scalar criteria have been numerically computed for all scenarios using a grid-based state density evaluation (see Zhang [ZL08]). Scenarios can therefore be positioned in a frame whose axis represent the two criteria (Figure 3.9). Ambiguity rises as the coordinates' values grow. The five considered scenarios appear to spray along a representative declination of terrain ambiguity cases.

3.4 DYNAMICAL MODELS USED FOR TRAJECTORY PLANNING

Fixed-wing aircraft unicycle model

A fixed wing aircraft motion in the horizontal plane can be described by the unicycle model (e.g. jung [JT08]):

$$\begin{bmatrix} \dot{x} & \dot{y} & \dot{V} & \dot{\psi} \end{bmatrix}^T = \begin{bmatrix} V\cos\psi & V\sin\psi & 0 & \frac{g}{V}\tan\varphi \end{bmatrix}^T \tag{3.12}$$

where x and y are the Cartesian position of the aircraft with respect to a reference point, V is the ground speed, ψ is the heading, g is the gravity field value, and φ is the rolling angle, which can be taken as the control input (also referred as *bank angle*). The assumption is made that the speed remains roughly constant.

Fixed-wing aircraft longitudinal model

A fixed wing aircraft motion in the vertical plan can be linearised around an equilibrium point (Cook [Coo12]):

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{3.13}$$

where

$$\mathbf{F} = \begin{bmatrix} X_u & X_w & 0 & -g\cos\theta_{eq} & 0 & 0 \\ Z_u & Z_w & u_{eq} & -g\sin\theta_{eq} & 0 & 0 \\ M_u + M_{\dot{w}}Z_u & M_w + M_{\dot{w}}Z_w & M_q + u_{eq}M_{\dot{w}} & -M_{\dot{w}}g\sin\theta_{eq} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & u_{eq} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.14}$$

and

$$\mathbf{B} = \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} + M_{\dot{w}}Z_{\delta_e} & M_{\delta_t} + M_{\dot{w}}Z_{\delta_t} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.15}$$

The state $\mathbf{x} = [u - u_{eq}, w, \dot{\theta}, \theta, \dot{h}, h]^T$ consists of the deviation between the longitudinal body axis velocity u and the equilibrium speed $u_{eq} > 0$, the vertical body frame velocity w, the pitch derivative $\dot{\theta}$, the pitch θ, the Earth frame vertical speed $\dot{h}$, and the altitude h. The control $\mathbf{u} = [\delta_e, \delta_\tau]^T$ consists of the elevator angle and the throttle longitudinal acceleration. Coefficients $X_u, X_w, Z_u, Z_w, M_u, M_w, M_{\dot{w}}, Z_{\dot{w}}, Z_q$ and M_q are the *stability derivatives* which depend on the aircraft's aerodynamics and structural characteristics. Coefficients $X_{\delta_e}, X_{\delta_t}, Z_{\delta_e}, Z_{\delta_t}, M_{\delta_e}$ and M_{δ_t} are the *control derivatives*.

Relative orbital motion (Clohessy-Wiltshire equations)

The Clohessy-Wiltshire relative model approaches the dynamics of the vector linking an orbital mobile (e.g. a spacecraft) with another mobile orbiting the same celestial body (in what follows, the *target*). Let $\mathbf{x} = [x, y, \dot{x}, \dot{y}]^T$ be the state vector representing the relative Cartesian position and velocity of the spacecraft with respect to the target in the relative coordinate system defined in Vallado [Val97] and illustrated in Figure 3.10. The target is assumed to cruise on a near-circular orbit.

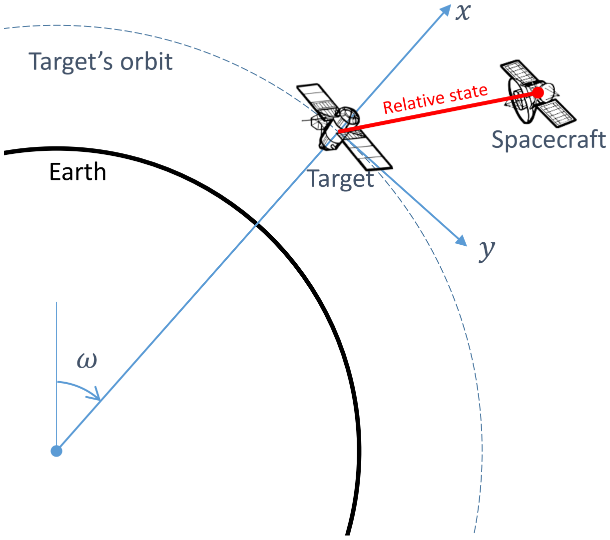


Figure 3.10: Relative coordinate system used to express the Clohessy-Wiltshire dynamics

The relative Clohessy-Wiltshire dynamical model can be derived from the orbital motion equations under the hypothesis of a close proximity between the target and the spacecraft:

$$\begin{cases} \ddot{x} = 2\omega^2\dot{y} + 3\omega^2 x + u_x \\ \ddot{y} = -2\omega^2\dot{x} + u_y \end{cases} \tag{3.16}$$

where $[u_x, u_y]^T$ is the control acceleration and ω the target's orbital angular rate. This model is useful for orbital rendezvous and collision avoidance applications.

3.5 EVALUATION CRITERIA FOR STATE ESTIMATION AND FAILURE PROBABILITY EVALUATION

In the following chapters, algorithms will be evaluated with several criteria defined in this section. Estimation algorithms outputs are a state estimate $\widehat{\mathbf{x}}_k$ and a state confidence. In this manuscript, the confidence is defined by the covariance matrix $\widehat{\mathbf{P}}_k$ of the posterior conditional state density. Estimators are evaluated by running N_p simulations describing the universe of realisations of the involved random variables (initial estimation error, process noise and measurement noise realisations).

The accuracy of an algorithms is evaluated using the *Root Mean Square Error* (RMSE) defined by:

$$RMSE_{\chi}(k) = \sqrt{\frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \|\widehat{\chi}_{k,i} - \chi_{k,i}\|^2} \tag{3.17}$$

where $\chi_{k,i}$ and $\widehat{\chi}_{k,i}$ are respectively the actual state and the estimate for simulation i at time-step k. The state $\chi_{k,i}$ can be the whole state $\mathbf{x}$ or only a sub-vector, e.g. position or velocity. The RMSE can also be used for the failure probability estimation (see Section 2.3.2).

The *average confidence* of a state estimator is obtained by:

$$\sigma_{\chi}(k) = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \text{diag}\left(\sqrt{\widehat{P}^{\chi}_{k,i}}\right) \tag{3.18}$$

where $\widehat{P}^{\chi}_{k,i}$ is the estimated covariance matrix of the state (or state sub-vector) χ for run i.

The *non-convergence rate* indicates the percentage of runs having an inconsistent estimation at a given time-step. The consistency of estimation is a binary value which indicates if the actual state belongs or not to the confidence region. For probabilistic estimators in the general case, one can define:

$$\frac{100}{N_{MC}} \sum_{i=1}^{N_{MC}} \mathbf{1}_{|\hat{\mathbf{x}}_k - \mathbf{x}_k| \geq n\sigma_{\mathbf{x}}(k)} \tag{3.19}$$

where usually $n = 3$ (i.e. 3σ, a confidence of 99% for a Gaussian distribution).

A *pessimism ratio* can be defined by the ratio between the covariance and the RMSE. This quantifies the consistency between the actual estimation error and the estimated confidence:

$$\text{pessimism}(k) = \frac{\sigma_{\chi}(k)}{RMSE_{\chi}(k)} \tag{3.20}$$

A ratio around 1 would characterize a good consistency between the estimation error and the associated confidence. If the ratio is greater than 1, the estimator is said pessimistic, which means that it results in an unnecessary large uncertainty on the estimate (also referred to as *conservatism*). If the ratio is lower than 1, the estimator is optimistic. Too much optimism may lead to non-convergences. In this thesis, the pessimism ratio is considered to be acceptable in the interval $[0.8, 1.5]$, and to be pessimist (conservative) if greater than 1.5.

An *RMSE ratio* can be defined as an improvement criterion between the initial state uncertainty (in terms of initial RMSE) and the final RMSE. This provides a dimensionless normalised criterion:

$$\text{r}(k) = \frac{RMSE_{\chi}(k_{end})}{RMSE_{\chi}(1)} \tag{3.21}$$

This criterion is of interest for problems where the measurements are regularly collected and the noise characteristics remain unchanged (e.g. rejection or no occurrence of outlying data). A ratio lower than 1 indicates that the final RMSE is lower than the initial RMSE. In the specific case of the TAN application (see Section 3.2), the ratio RMSE ratio is considered to be good if lower than 0.2 in position and than 0.8 in velocity, and very good if lower than 0.1 in position and than 0.6 in velocity.

4

THE BOX REGULARISED PARTICLE FILTER

Chapter 2 examined the background literature about state estimation. Two frameworks were introduced, namely Bayesian estimation and Interval Analysis. These two frameworks yielded the Box Particle Filter (BPF) framework.

BPF was introduced in Section 2.1.6, on the basis of previous works (Gning [Gni+13]). The BPF presents very similar features to a conventional Particle Filter, introduced in Section 2.1.4. However, instead of propagating a cloud of weighted point-wise particles, it propagates a set of weighted boxes, i.e. vectors of intervals. These box particles are dynamically propagated and updated by measurements using the Interval Analysis framework (e.g. see Jaulin [Jau09]) introduced in Section 2.1.5. The measurements and their uncertainty are described as a bounded measurements box. From a probabilistic point of view, box particles and box measurements correspond to uniform kernels' supports. The BPF appears to outperform the conventional Particle Filter in terms of robustness to ambiguities (see Appendix B) and computational load (see Appendix C), since it can reach similar performances with far less particles (e.g. a hundred times less in Gning [GRM12]). Nevertheless, BPF has several drawbacks:

- First, it is assumed that box particles and box measurements represent uniform kernels. When the measurement density is bounded but not uniform, the resulting performances may not account for this information. Several other density hypotheses have been considered in other formulations of BPF, but there is no general formulation that would allow an arbitrary density function to be applied;
- As in the conventional PF (see Section 2.1.4), BPF requires resampling steps that do not ensure that the actual state still belongs to any box particle, potentially leading to non-convergences in cases of severe ambiguities (see Section 3.5 for the definition of non-convergence);
- The BPF resampling operates subdivisions of high-weighted box particles. However, no efficient method exists for selecting the choice of dimension to subdivide;
- Last, the BPF estimated density description relies on a mixture of box particles that often exactly overlap, resulting in a jagged state density estimation. This may negatively impact the estimation accuracy.

This chapter introduces several contributions to the BPF field:

- Section 4.1 introduces a theoretical derivation of BPF which relaxes the hypotheses on the box kernels and measurement density. Indeed, these densities are no longer restricted to uniform kernels. This allows the provision of a general framework that covers previous formulations of BPF.
- Section 4.2 introduces a guaranteed version of the resampling step, named Guaranteed Resampling, which leads to the guarantee that the actual state always belongs to at least one box particle after

resampling. A theoretical proof is provided. Numerical simulations illustrate the robustness of the algorithm on the Terrain Aided Navigation (TAN) application.

- Section 4.3 introduces two different ways of subdividing box particles. The first is presented in Section 4.3.1. It is called Geometrical Subdivision and is based on a geometrical criterion. It is limited to dynamical models of vehicles whose state description is expressed in terms of Cartesian coordinates. The second is in Section 4.3.2. It relies on a Maximum Likelihood (ML) Covariance criterion and can be used whatever the state description, the dynamical model and the measurement model. Comparisons in terms of RMSE are performed between the random picking method from the literature (Gning [Gni+13]), the Geometrical Subdivision and the Maximum Likelihood Covariance subdivision.

- Section 4.4 introduces a Kernel Regularisation applied to BPF. On the basis of the Regularised Particle Filter (RPF, [MOLG01], see Section 2.1.4), the regularisation process is extended to the box particle scheme. It aims to smooth the estimated state density in order to improve the accuracy of the estimate. The regularised version of BPF theoretically is formally proven to enhance the state density estimation in terms of Mean Integrated Square Error, and simulations illustrate the impact of regularisation on estimation accuracy.

Section 4.5 concludes this chapter by comparing the performances in terms of RMSE of the Box Regularised Particle Filter (BRPF), derived from the BPF with the aforementioned contributions, with conventional algorithms, such as the SIR-PF, and the conventional BPF. A computational load study is also performed, in terms of theoretical floating-point operations per time-step (see Appendix C), and of empirical computation time.

The main application investigated in this chapter is Terrain Aided Navigation (TAN). TAN scenarios and modeling were presented in Section 3.2. For the sake of simplicity, the vehicle's dynamical model is a double integrator (see Section 3.1, equation (3.2)). More realistic models will be considered in the following chapters. The performances of the algorithms are evaluated in terms of RMSE (see equation (3.17)), estimated covariance (see equation (3.18)), RMSE ratio (see equation (3.19)), and pessimism (see equation (3.20)), introduced in Section 3.5.

4.1 A MORE GENERAL FORMULATION OF THE BOX PARTICLE FILTER

The Box Particle Filter relies on a description of the estimated conditional state density as a mixture of uniform kernels bounded by boxes. The measurement density is also assumed to be a uniform density. In this section, this framework is generalized to any kernel:

- Box particles $[\mathbf{x}_k^i]$ are viewed as supports of bounded kernels $\pi_k^i : \mathbb{R}^d \to \mathbb{R}$, without any hypothesis about the kernel's other properties;

- The measurement density, denoted π_k^m is assumed to be bounded by the box measurement $[\mathbf{m}_k]$. Nevertheless, the following equations remain valid for unbounded measurement density (the measurement set is then an open set equal to $[\mathbf{m}_k] = \mathbb{R}^{d_m}$).

4.1.1 Principle and example

The prior conditional state density at time $k-1$ is defined by a mixture of N kernels bounded by box particles $[\mathbf{x}_{k-1}^i] \in \mathbb{IR}^d$ and weighted by weights w_k^i whose sum is unity:

$$p(\mathbf{x}_{k-1}|\mathbf{M}_{k-1}) = \sum_{i=1}^{N} w_{k-1}^i \pi_{k-1}^i(\mathbf{x}_{k-1}) \mathbf{1}_{[\mathbf{x}_{k-1}^i]}(\mathbf{x}_{k-1}) \tag{4.1}$$

where each box kernel $\pi_{k-1}^i : \mathbb{R}^d \rightarrow \mathbb{R}$ satisfies $\int_{[\mathbf{x}_{k-1}^i]} \pi_{k-1}^i(\mathbf{x})\mathrm{d}\mathbf{x} = 1$.

The process noise density, i.e. the uncertainty on the dynamics, is defined by a bounded kernel $\pi_{k|k-1}^x : \mathbb{R}^d \rightarrow \mathbb{R}$ whose support is $[\mathbf{w}_k] \in \mathbb{IR}^d$, such that $\int_{[\mathbf{w}_k]} \pi_{k|k-1}^x(\mathbf{x})\mathrm{d}\mathbf{x} = 1$. The measurement density, i.e. the uncertainty on the measurements, is defined by a bounded kernel $\pi_k^m : \mathbb{R}^{d_m} \rightarrow \mathbb{R}$ whose support is $[\mathbf{v}_k] \in \mathbb{IR}^{d_m}$, such that $\int_{[\mathbf{v}_k]} \pi_k^m(\mathbf{y})\mathrm{d}\mathbf{y} = 1$.

At each step, the conditional state density $p(\mathbf{x}|\mathbf{M})$ is defined as a weighted mixture of box particle kernels. This is illustrated in Figure 4.1 for a scalar case, with four box particles. The prediction step is performed by applying the Chapman-Kolmogorov equation (2.4) to the prior density $p(\mathbf{x}_{k-1}|\mathbf{M}_{k-1}) = \sum_i w_k^i \pi_{k-1}^i(\mathbf{x}_{k-1})\mathbf{1}_{[\mathbf{x}_{k-1}^i]}(\mathbf{x}_{k-1})$. By distributivity of the summation, it is equivalent to independently applying the Chapman-Kolmogorov equation to each box kernel π_{k-1}^i. Weights are not modified during the prediction. The correction step is obtained by applying Bayes' rule to the predicted density, which is equivalent to independently applying it to each predicted box kernel π_k^i. Weights are then updated with regard to the overlapping of each box kernel with the measurement density, which can be interpreted as a consistency. The Optimal filter's equations are derived in Section 4.1.2. This leads to a general formulation of BPF equations, which covers various particular cases introduced in the literature.

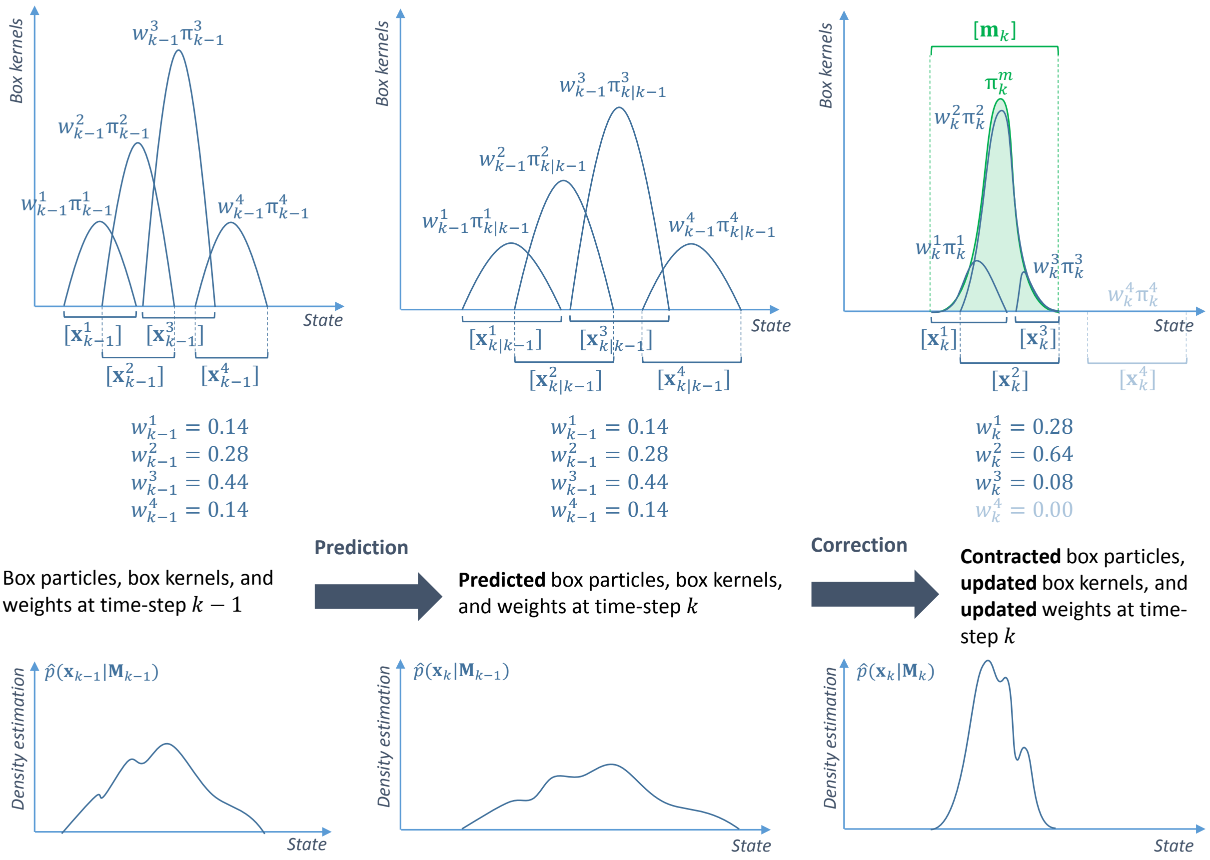


Figure 4.1: Illustration of the general box kernel formulation of the Box Particle Filter

4.1.2 Derivation of the optimal filter's equations

This section details the general BPF description by deriving the Optimal Filter's equations (introduced in Section 2.1.2). The assumption is made that the dynamical model $f : \mathbb{R}^d \to \mathbb{R}^d$ and the observation model $h : \mathbb{R}^d \to \mathbb{R}^{d_m}$ are continuous on their domain. Initial box kernels $\pi_0^i : \mathbb{R}^d \to \mathbb{R}$ are also assumed to be continuous. Likewise, the measurement kernel $\pi_k^m : \mathbb{R}^d \to \mathbb{R}^{d_m}$ is assumed to be continuous at each time-step.

Proposition 1. *Let be the prior conditional density* (4.1) *defined by a mixture of N kernels* $\pi_{k-1}^i : \mathbb{R}^d \to \mathbb{R}$ *bounded by box particles* $[\boldsymbol{x}_{k-1}^i] \in \mathbb{IR}^d$ *and weighted by weights* w_k^i *whose sum is unity. Then, the predicted conditional density is equal to:*

$$p(\boldsymbol{x}_k|\boldsymbol{M}_{k-1}) = \sum_{i=1}^{N} w_{k-1}^i \pi_{k|k-1}^i(\boldsymbol{x}_k) \mathbf{1}_{[\boldsymbol{x}_{k|k-1}^i]}(\boldsymbol{x}_k) \tag{4.2}$$

with

$$\begin{cases} [\boldsymbol{x}_{k|k-1}^i] \triangleq [f]([\boldsymbol{x}_{k-1}^i], \boldsymbol{u}_k) + [\boldsymbol{w}_k] \\ \pi_{k|k-1}^i(\boldsymbol{x}_k) \triangleq \displaystyle\int_{[\boldsymbol{x}_{k-1}^i]} \pi_{k|k-1}^x(\boldsymbol{x}_k - f(\boldsymbol{x}, \boldsymbol{u}_k)) \pi_{k-1}^i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \end{cases} \tag{4.3}$$

where $\pi_{k|k-1}^x : \mathbb{R}^d \to \mathbb{R}$ *is the process noise kernel and* $\boldsymbol{u}_k$ *is some deterministic control input. The posterior conditional density is:*

$$p(\boldsymbol{x}_k|\boldsymbol{M}_k) = \sum_{i=1}^{N} w_k^i \pi_k^i(\boldsymbol{x}_k) \mathbf{1}_{[\boldsymbol{x}_k^i]}(\boldsymbol{x}_k) \tag{4.4}$$

with

$$\begin{cases} [\boldsymbol{x}_k^i] \triangleq \left\{ \boldsymbol{x}_k \in [\boldsymbol{x}_{k|k-1}^i] \quad \middle| \quad h(\boldsymbol{x}_k) \in [\boldsymbol{m}_k] \right\} \\ w_k^i \propto w_{k-1}^i \displaystyle\int_{[\boldsymbol{x}_k^i]} \pi_k^m(\boldsymbol{m}_k - h(\boldsymbol{x})) \pi_{k|k-1}^i(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\ \pi_k^i \triangleq \dfrac{1}{\int_{[\boldsymbol{x}_k^i]} \pi_k^m \pi_{k|k-1}^i \mathrm{d}\boldsymbol{x}} \pi_k^m \pi_{k|k-1}^i \end{cases} \tag{4.5}$$

Proof. Derive the Optimal Filter's equations (see Section 2.1.2) for the prediction and correction steps.

Prediction step

Applying the Chapman-Kolmogorov equation (2.4) to the prior density (4.1), one obtains:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{M}_{k-1}) &= \int_{\mathbb{R}^d} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{M}_{k-1})\mathrm{d}\mathbf{x}_{k-1} \\ &= \int_{\mathbb{R}^d} \pi^x_{k|k-1}(\mathbf{x}_k - f(\mathbf{x}_{k-1},\mathbf{u}_k)) \sum_{i=1}^{N} w^i_{k-1}\pi^i_{k-1}\mathbf{1}_{[\mathbf{x}^i_{k-1}]}(\mathbf{x}_{k-1})\mathrm{d}\mathbf{x}_{k-1} \\ &= \sum_{i=1}^{N}\left(w^i_{k-1}\int_{[x^i_{k-1}]} \pi^x_{k|k-1}(\mathbf{x}_k - f(\mathbf{x}_{k-1},\mathbf{u}_k))\pi^i_{k-1}(\mathbf{x}_{k-1})\mathbf{1}_{[\mathbf{x}^i_{k-1}]}(\mathbf{x}_{k-1})\mathrm{d}\mathbf{x}_{k-1}\right) \end{aligned} \tag{4.6}$$

Define $\pi^i_{k|k-1}$ as the i^{th} predicted kernel, whose support is included in box particle $[\mathbf{x}^i_{k|k-1}]$:

$$\pi^i_{k|k-1}(\mathbf{x}_k) \triangleq \int_{[x^i_{k-1}]} \pi^x_{k|k-1}(\mathbf{x}_k - f(\mathbf{x}_{k-1},\mathbf{u}_k))\pi^i_{k-1}(\mathbf{x}_{k-1})\mathrm{d}\mathbf{x}_{k-1} \tag{4.7}$$

For the sake of brevity, kernel arguments will be omitted in what follows. The predicted conditional density can be written as:

$$p(\mathbf{x}_k|\mathbf{M}_{k-1}) = \sum_{i=1}^{N} w^i_{k-1}\pi^i_{k|k-1}\mathbf{1}_{[\mathbf{x}^i_{k|k-1}]} \tag{4.8}$$

with

$$[\mathbf{x}^i_{k|k-1}] \triangleq [f]([\mathbf{x}^i_{k-1}],\mathbf{u}_k) + [\mathbf{w}_k] \tag{4.9}$$

which yields (4.3).

Correction step

The correction step determines the posterior conditional distribution of the state with respect to the predictive distribution (4.8) and the measurement density $p(\mathbf{m}_k|\mathbf{x}_k) = \pi^m_k$. Bayes' rule (2.5) leads to:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{M}_k) &= \tfrac{1}{q_k} p(\mathbf{x}_k|\mathbf{M}_{k-1})p(\mathbf{m}_k|\mathbf{x}_k) \\ &= \tfrac{1}{q_k}\left(\sum_{i=1}^{N} w^i_{k-1}\pi^i_{k|k-1}\mathbf{1}_{[\mathbf{x}^i_{k|k-1}]}\right)\pi^m_k \\ &= \tfrac{1}{q_k}\sum_{i=1}^{N} w^i_{k-1}\left(\pi^i_{k|k-1}\mathbf{1}_{[\mathbf{x}^i_{k|k-1}]}\pi^m_k\mathbf{1}_{[\mathbf{m}_k]}\right) \end{aligned} \tag{4.10}$$

As described by (2.58), each box kernel support $[\mathbf{x}^i_{k|k-1}]$ is contracted to a box $[\mathbf{x}^i_k]$ defined as the smallest box that contains all the states of box $[\mathbf{x}^i_{k|k-1}]$ that are consistent with the measurements $[\mathbf{m}_k]$, defined by the Constraints Satisfaction Problem $\left\{\mathbf{x}_k \in [\mathbf{x}^i_{k|k-1}] \quad | \quad h(\mathbf{x}_k) \in [\mathbf{m}_k]\right\}$ (see Section 2.1.5). The updated box particle is obtained by applying an interval contraction algorithm. This box contraction step is the direct consequence of the product of the two indicator functions, i.e.:

$$[\mathbf{x}^i_k] = \left[\left\{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{1}_{[\mathbf{x}^i_{k|k-1}]}(\mathbf{x})\ \mathbf{1}_{[\mathbf{m}_k]}(\mathbf{m}_k - h(\mathbf{x})) \neq 0\right\}\right] \tag{4.11}$$

Thus:

$$p(\mathbf{x}_k|\mathbf{M}_k) = \frac{1}{q_k}\sum_{i=1}^{N} w_{k-1}^{i}\left(\pi_k^m \pi_{k|k-1}^{i}\mathbf{1}_{[\mathbf{x}_k^i]}\right) \tag{4.12}$$

Note that in the particular case of the measurement space being equal to the state space (i.e. $h(\mathbf{x}_k) = \mathbf{x}_k$), the contraction is equivalent to the intersection $[\mathbf{x}_k^i] = [\mathbf{x}_{k|k-1}^i] \cap [\mathbf{m}_k]$.

However, the term $\pi_k^m \pi_{k|k-1}^i \mathbf{1}_{[\mathbf{x}_k^i]}$ no longer sums to unity and is therefore not a pdf. Indeed, the support of the kernel $\pi_{k|k-1}^i \mathbf{1}_{[\mathbf{x}_{k|k-1}^i]}$ has been truncated by $[\mathbf{x}_k^i] \subset [\mathbf{x}_{k|k-1}^i]$ whose volume is lower or equal to that of $[\mathbf{x}_{k|k-1}^i]$. Furthermore, it is multiplied by the measurement kernel π_k^m, which leads to a new kernel proportional to $\pi_k^m \pi_{k|k-1}^i$. Therefore, it has to be normalised by $\int_{[\mathbf{x}_k^i]} \pi_k^m \pi_{k|k-1}^i \mathrm{d}\mathbf{x}$, leading to:

$$p(\mathbf{x}_k|\mathbf{M}_k) = \frac{1}{q_k}\sum_{i=1}^{N} w_{k-1}^{i}\left(\frac{\int_{[\mathbf{x}_k^i]} \pi_k^m \pi_{k|k-1}^i \mathrm{d}\mathbf{x}}{\int_{[\mathbf{x}_k^i]} \pi_k^m \pi_{k|k-1}^i \mathrm{d}\mathbf{x}}\pi_k^m \pi_{k|k-1}^{i}\mathbf{1}_{[\mathbf{x}_k^i]}\right) \tag{4.13}$$

thus giving,

$$p(\mathbf{x}_k|\mathbf{M}_k) = \frac{1}{q_k}\sum_{i=1}^{N}\left(w_{k-1}^{i}\int_{[\mathbf{x}_k^i]} \pi_k^m \pi_{k|k-1}^i \mathrm{d}\mathbf{x}\right)\pi_k^i \mathbf{1}_{[\mathbf{x}_k^i]} \tag{4.14}$$

where $\pi_k^i \triangleq \frac{1}{\int_{[\mathbf{x}_k^i]} \pi_k^m \pi_{k|k-1}^i \mathrm{d}\mathbf{x}} \pi_k^m \pi_{k|k-1}^i$ is the updated kernel. This kernel is only defined if $\int_{[\mathbf{x}_k^i]} \pi_k^m \pi_{k|k-1}^i \mathrm{d}\mathbf{x} \neq 0$, i.e. if box i is consistent with the measurement density. If not, it can be set by convention to $w_k^i = 0$.

Finally, the posterior conditional density is obtained:

$$p(\mathbf{x}_k|\mathbf{M}_k) = \sum_{i=1}^{N} w_k^i \pi_k^i \mathbf{1}_{[\mathbf{x}_k^i]} \tag{4.15}$$

where the update of weights is, for $k \geq 1$:

$$w_k^i \triangleq \frac{1}{q_k} w_{k-1}^i \int_{[\mathbf{x}_k^i]} \pi_k^m(\mathbf{m}_k - h(\mathbf{x}))\pi_{k|k-1}^i(\mathbf{x})\mathrm{d}\mathbf{x} \tag{4.16}$$

The integral term in (4.16) can be interpreted as a consistency term between the box kernel $\pi_{k|k-1}^i(\mathbf{x})$ whose support is restricted to box $[\mathbf{x}_k^i]$ and the measurement kernel $\pi_k^m(\mathbf{m}_k - h(\mathbf{x}))$. This integral term belongs to interval $[0,1]$. If the box particle is not consistent with the measurement density, (4.16) ensures that $w_k^i = 0$. The normalisation quotient is:

$$q_k = \sum_i w_{k-1}^i \int_{[\mathbf{x}_k^i]} \pi_k^m \pi_{k|k-1}^i \mathrm{d}\mathbf{x} \tag{4.17}$$

which yields (4.5). ☐

Remark 1: link to previous work

The above development gives a general theoretical formulation of BPF. It provides a common framework that covers a wide variety of particle and box particle filters. For example, in the case of null volume box particles, the BPF can be linked to the Particle Filter described in Gordon [GSS93]. This can be derived from (4.12) when the box kernel $\pi_k^i \mathbf{1}_{[\mathbf{x}_k^i]}$ tends to a Dirac $\delta_{\mathbf{c}_k^i}$, where $\mathbf{c}_k^i$ is the center box particle i. The updated weights are $w_k^i = \frac{1}{q_k} w_{k-1}^i \pi_k^m \left(\mathbf{m}_k - h(\mathbf{c}_k^i)\right)$.

Furthermore, in the case of uniform density functions, assuming that the state propagation of the box always results in a uniform kernel, $\pi_{k|k-1}^i \approx \mathcal{U}_{[f]([\mathbf{x}_{k-1}^i])+[\mathbf{w}_k]} \quad \forall k \geq 1$ and with a uniform measurement density $\pi_k^m = \frac{1}{|[\mathbf{m}_k]|}$, one obtains: $w_k^i = \frac{1}{q_k} w_{k-1}^i \frac{1}{|[\mathbf{m}_k]|} \frac{|[\mathbf{x}_k^i]|}{|[\mathbf{x}_{k|k-1}^i]|}$ for non-empty contracted boxes and zero for empty boxes. This is equivalent to the formulation proposed by Gning [Gni+13]:

$$w_k^i \propto w_{k-1}^i \frac{|[\mathbf{x}_k^i]|}{|[\mathbf{x}_{k|k-1}^i]|} \tag{4.18}$$

since all weights are normalised.

In the case of uniform box particles and non bounded measurement density (the box measurement is then a set equal to the open box $[\mathbf{m}_k] = \mathbb{R}^{d_m}$), the measurement update described in Blesa [BI+15], $w_k^i = \frac{1}{q_k} w_{k-1}^i \frac{1}{|[\mathbf{x}_k^i]|} \int_{[\mathbf{x}_k^i]} \pi_k^m \left(\mathbf{m}_k - h(\mathbf{x})\right) \mathrm{d}\mathbf{x}$, is recovered.

Remark 2: implementation of box-kernels

The proposed general developments presented in the last sections make it possible to have a clear overview on how the literature's derivations of BPF are positioned in the Bayesian framework, and in the assumptions done. Therefore, this general formulation of BPF has a theoretical interest.

However, the use of the exact formulation of box kernel propagation (4.3) and correction (4.5) may be limited in practice:

- The exact formulation of each box-kernel's propagation (4.3) may be intractable (i.e. deriving $\pi_{k|k-1}^i$ from $\pi_{k-1}^i \ \forall (i,k)$). For implementation simplicity, the hypothesis can be made that each box kernel is defined via a reference kernel π:

  $$\begin{aligned} \pi_{k-1}^i(\mathbf{x}) &\approx \frac{1}{|[\mathbf{x}_{k-1}^i]|} \pi \left(\frac{\mathbf{x} - \underline{\mathbf{x}}_{k-1}^i}{\overline{\mathbf{x}}_{k-1}^i - \underline{\mathbf{x}}_{k-1}^i} \right) \mathbf{1}_{[\mathbf{x}_{k-1}^i]} \ \ \forall (i,k) \\ \pi_{k|k-1}^i(\mathbf{x}) &\approx \frac{1}{|[\mathbf{x}_{k|k-1}^i]|} \pi \left(\frac{\mathbf{x} - \underline{\mathbf{x}}_{k|k-1}^i}{\overline{\mathbf{x}}_{k|k-1}^i - \underline{\mathbf{x}}_{k|k-1}^i} \right) \mathbf{1}_{[\mathbf{x}_{k|k-1}^i]} \ \ \forall (i,k) \end{aligned} \tag{4.19}$$

  where $\pi : \mathbb{R}^d \rightarrow \mathbb{R}^+$ integrates to unity on $[0,1]$, and $[\underline{\mathbf{x}}_\bullet^i, \overline{\mathbf{x}}_\bullet^i]$ represent the bounds of box i. In Gning [Gni+13], this approximation was done in the particular case of uniform kernels, which yields $\pi(\mathbf{x}) = 1 \ \forall \mathbf{x}$. Such an approximation may introduce conservatism and results in a sub-optimal propagation. A refined box propagation approach based on uniform mixtures was introduced in Gning [GMA10], but its computational complexity may limit its use in practice.

- Integrating on $\mathbb{R}^d$ the composition (4.16) of the observation model $h : \mathbb{R}^d \rightarrow \mathbb{R}^{d_m}$ with the measurement density $\pi^m : \mathbb{R}^{d_m} \rightarrow \mathbb{R}$ rapidly becomes intractable for non-linear observation models. Indeed, no explicit formulation can be derived as an expression of h, π_k^i, and π_k^m from the weights update step (4.16) in the general case. This limits the practical use of the general formulation for state estimation from ambiguous measurements, which is the purpose of this work. For example, in the Terrain Aided Navigation (TAN) problem, the observation equation is a function of a non-analytic Digital Elevation Model (DEM) stored in memory. Then, no analytic solution can be derived from (4.16).

Table 4.1: Simulation settings for the Box Particle Filter

Parameters	Values
Initial uncertainty (std: m, m, ms^{-1}, ms^{-1})	$[30, 30, 2, 2]$
Process noise (std: m, m, ms^{-1}, ms^{-1})	$[0.1, 0.1, 0.01, 0.01]$
Measurement uncertainty, position (std: m)	$[10, 10]$
Resampling threshold	0.7
Number of box particles	900

4.1.3 Numerical results

This section makes a comparison between the uniform BPF formulation (4.18) and the general formulation (4.16) applied to the linear-Gaussian case:

$$w_k^i = \frac{1}{q_k} w_{k-1}^i \frac{1}{|[\mathbf{x}_{k|k-1}^i]|} \int_{[\mathbf{x}_k^i]} \pi_k^m \left(\mathbf{m}_k - \mathbf{H}\mathbf{x}\right) \mathrm{d}\mathbf{x} \tag{4.20}$$

where $\mathbf{H} \in \mathbb{R}^{d_m \times d}$ is a known observation matrix, and with a truncated Gaussian measurement kernel $\pi_k^m(\mathbf{y}) \propto \exp\left(-\frac{1}{2}\mathbf{y}^T\mathbf{R}^{-1}\mathbf{y}\right) \mathbf{1}_{[\mathbf{v}_k]}$. In this case, $\mathbf{R}$ was chosen such that $\frac{1}{2}\delta_{[\mathbf{v}_k]} = 3\text{Diag}\left(\sqrt{\mathbf{R}}\right)$. The chosen dynamical model is the double integrator model (3.1) and the observation model is the Cartesian measurements (3.7). The following points are to be checked:

1. The impact of truncated Gaussian measurement noise realisations on both formulations;
2. The impact of uniform measurement noise realisations on both formulations.

It can be expected that truncated Gaussian measurement noise would yield lower RMSE and covariance with the Gaussian BPF formulation, while the uniform measurements would yield better results with the uniform BPF formulation.

Figure 4.2 and Table 4.2 present the performances for both formulations in terms of Root Mean Square Error (RMSE) and averaged covariance (100 Monte Carlo runs) for Gaussian measurements. By accounting for the knowledge of the measurement density, the Gaussian BPF formulation can reach a lower RMSE and a significantly lower estimated covariance than the uniform formulation, which makes it less conservative (point 1). A Kalman Filter is added for comparison, since it tends to the optimal solution in the linear-Gaussian case. Although the Gaussian BPF is closer to it than the uniform BPF, it remains significantly sub-optimal. This can be explained by approximation (4.19), and by the wrapping effect (see Section 2.1.5).

Figure 4.3 and Table 4.3 present the performances for both formulations (100 Monte Carlo runs) for uniform measurements (point 2). In this context, the Gaussian BPF formulation yields a higher RMSE than the uniform one and a potentially inconsistent covariance (non-convergence rate of 3% versus 0% for uniform BPF).

Points 1 and 2 show that taking the right hypothesis on the measurement noise yields better results. However, the general BPF formulation is difficult to implement in non-linear cases and does not appear to be applicable to state estimation under severe ambiguities. In practice, the prediction and correction steps can be computed in a sub-optimal but computationally efficient way with the uniform formulation (4.18) (Gning [Gni+13]).

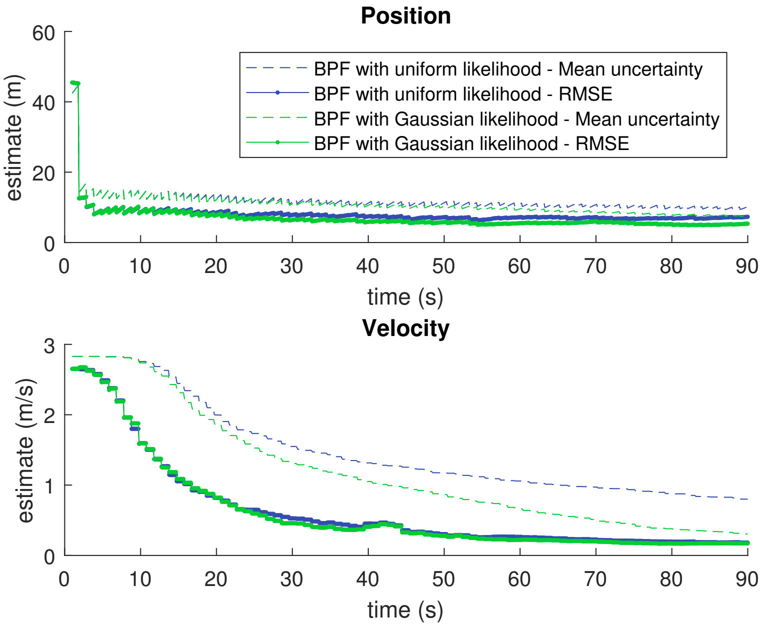


Figure 4.2: Comparison between uniform and Gaussian measurement density for the Box Particle Filter with Gaussian realisations: RMSE and filter's average confidence (estimated covariance). The Gaussian formulation of the likelihood allows the covariance to be lower, which yields less conservatism.

Table 4.2: Simulation results for point-mass dynamics with Cartesian measurements at final time-step. The Gaussian measurement noise hypothesis yields the Box Particle Filter to lower Root Mean Square Error and Covariance.

	Kalman Filter	BPF Uniform	BPF Gaussian
Non-convergence rate (%)	0	1	2
RMSE (position, m)	3.41	7.31	5.34
RMSE (velocity, ms^{-1})	0.11	0.19	0.17
Covariance (position, m)	3.17	9.47	7.41
Covariance (velocity, ms^{-1})	0.10	0.80	0.30
Pessimism (position)	0.93	4.38	1.39
Pessimism (velocity)	0.90	1.92	1.78
RMSE ratio (position)	0.075	0.16	0.11
RMSE ratio (velocity)	0.041	0.070	0.064

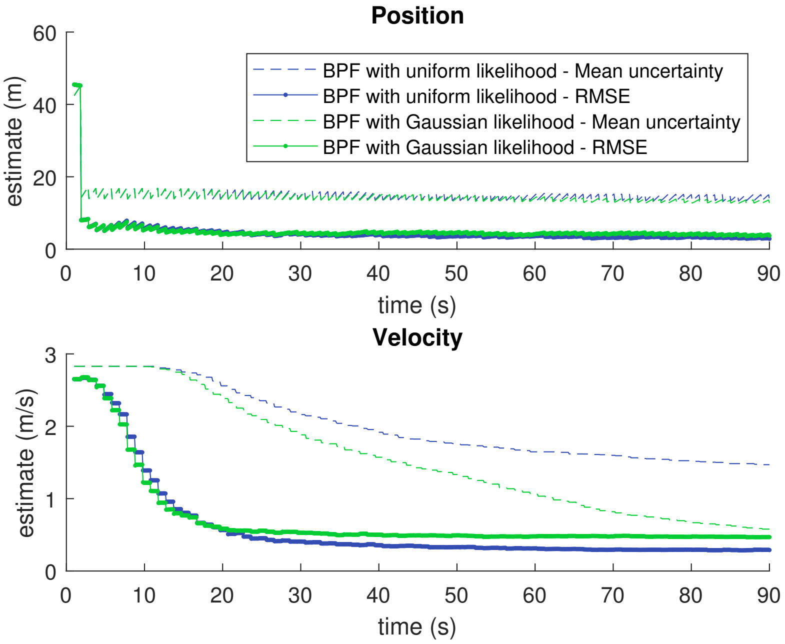


Figure 4.3: Comparison between uniform and Gaussian measurement density for the Box Particle Filter with uniform realisations: RMSE and filter's average confidence (estimated covariance). The Gaussian formulation of the likelihood yields a RMSE discrepancy and an inconsistent confidence.

Table 4.3: Simulation results for point-mass dynamics with Cartesian measurements at final time-step. The Gaussian measurement noise hypothesis yields the Box Particle Filter to higher Root Mean Square Error and a higher non-convergence rate.

	BPF Uniform	BPF Gaussian
Non-convergence rate (%)	0	3
RMSE (position, m)	2.9	3.7
RMSE (velocity, ms^{-1})	0.29	0.46
Covariance (position, m)	13.4	12.7
Covariance (velocity, ms^{-1})	1.47	0.58
Pessimism (position)	4.68	3.43
Pessimism (velocity)	5.07	1.24
RMSE ratio (position)	0.063	0.082
RMSE ratio (velocity)	0.11	0.18

4.2 A GUARANTEED VERSION OF MULTINOMIAL RESAMPLING

In order to prevent degeneracy in the particle filter algorithm, where all the particles but one have near-zero weights, resampling methods have been introduced, as mentioned in Section 2.1.4. Previous works on BPF proposed to use the Multinomial Resampling approach (see Gning [AGB07; GMA10; GRM12; GMA12; Gni+13; DF+16; MDPL16]). Multinomial Resampling however, does not take advantage of the guaranteed characteristics of box propagation and box contraction. BPF is commonly used to deal with bounded densities, which provides deterministic information about box particle consistency with past measurements, that is, if box particle i is non-zero-weighted ($w_k^i \neq 0$), then it has a non-zero probability to contain the actual state $\mathbf{x}_k$. Multinomial Resampling aims to provide more instances of high-weighted particles while removing low-weighted ones, but it does not differentiate low weighted and zero-weighted particles. Making such a distinction would not be useful in the case of a measurement density with a large or infinite distribution tail (e.g. Gaussian kernel), but is of strong interest in the case of a bounded measurement support. When measurements are highly ambiguous, as in the Terrain Aided Navigation problem, Multinomial Resampling may remove box particles containing the actual state to the benefit of box particles that do not, leading to non-convergence.

Accounting for the knowledge of the deterministic consistency of box particles yields a derived version of Multinomial Resampling, where only zero-weighted box particles would be removed and replaced. In other words, all non-zero-weighted box particles are kept or subdivided and only zero-weighted box particles are replaced with subdivisions of high-weighted box particles. The Guaranteed Resampling method to determine the number of new instances per particle is described in Algorithm 6. In what follows, Guaranteed BPF will refer to the BPF (Algorithm 5) using Guaranteed Resampling instead of Multinomial Resampling.

Algorithm 6 Guaranteed Resampling

```
Input: particle weights $\{w^i\}_{i \in [1,N]}$
Output: number of new instances per particles $\{n^i\}_{i \in [1,N]}$
Initialise the duplication counters to:
    $\begin{cases} n^i = 1 & \text{if } w_k^i \neq 0 \\ n^i = 0 & \text{if } w_k^i = 0 \end{cases}$    (4.21)
Compute the number of zero-weighted box particles ($M \leq N$):
    $M = \text{card}\left\{ i \in [1,N] \mid w_k^i = 0 \right\}$    (4.22)
for $i = 1$ to $M$ do
    Draw $u^i \sim \mathcal{U}_{[0,1]}$
    Find $j \in [1,N]$ such that $u^i \in \left] \sum_{l=1}^{j-1} w^l, \sum_{l=1}^{j} w^l \right]$
    Count $n^j = n^j + 1$
end for
Return $n^i \; \forall i \in [1,N]$
```

4.2.1 Principle and example

Figure 4.4 illustrates the effects of Guaranteed Resampling and Multinomial Resampling on an example with a three dimensional state ($\mathbf{x} = [p_x, p_y, p_z]^T$). The resampling step occurs after the correction step, which consists of a box contraction step with respect to the measurements (4.11) and a weights' update step (4.16). In this example, the actual state (the red cross) is contained in one single large box $[\mathbf{x}^{i_x}_{k|k-1}]$. Only a small subset of it $[\mathbf{x}^{i_x}_{k}] \subset [\mathbf{x}^{i_x}_{k|k-1}]$ is consistent with the set of states consistent with the measurements (the green set), defined by $h^{-1}([\mathbf{m}_k]) \triangleq \{\mathbf{x} \in \mathbb{R}^d \mid h(\mathbf{x}) \in [\mathbf{m}_k]\}$. Therefore, by application of (4.16), the associated weight $w^{i_x}_k$ will be small with respect to the box particles that are consistent with $[\mathbf{m}_k]$. This is easily understandable in the case of uniform densities, where the weight update is simply $w^i_k \propto w^i_{k-1} \frac{|[\mathbf{x}^i_k]|}{|[\mathbf{x}^i_{k|k-1}]|}$. The ratio of box particle volumes before and after contraction leads to a lower weight if only a small sub-box of the box particle i is consistent with the measurements.

As a result, performing Multinomial Resampling may lead to the destruction of low-weighted box particles, including the box particle $[\mathbf{x}^{i_x}_k]$ that actually contained the state, thus leading to non-convergence. By always keeping non-zero weighted box particles, Guaranteed Resampling ensures that the actual state $\mathbf{x}_k$ always belongs to at least one box particle. The following section (Section 4.2.2) provides a theoretical proof of this statement.

In addition, the computational load required for Guaranteed Resampling is less or equal to that required for Multinomial Resampling. Indeed, Multinomial Resampling requires a sample of N uniform random values, while Guaranteed Resampling only requires $M \leq N$ samples (see Algorithm 3 and Algorithm 6).

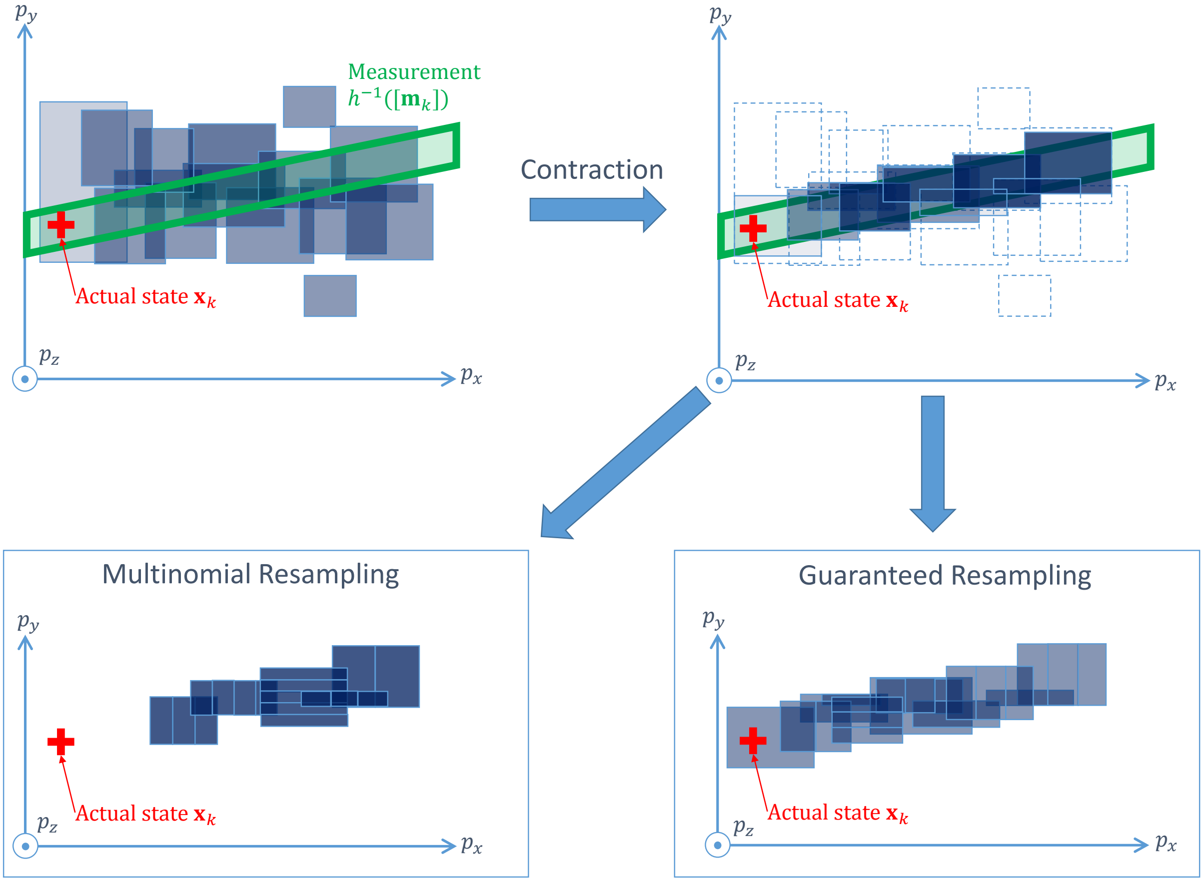


Figure 4.4: Illustration of Multinomial Resampling compared to Guaranteed Resampling, after the correction step.

4.2.2 Theoretical justification of a Guaranteed Box Particle Filter

This section proves that Guaranteed Resampling (Algorithm 6) ensures that at least one box-particle contains the actual state $\mathbf{x}_k$ at each time-step k. This proof is provided using the general framework presented in 4.1.2.

Proposition 2. *If the initial actual state belongs to the initial box particle cloud, and given that:*

- *The dynamical model f and the observation model h are continuous functions (respectively $\mathcal{C}^0(\mathbb{R}^d, \mathbb{R}^d)$ and $\mathcal{C}^0(\mathbb{R}^d, \mathbb{R}^{d_m})$),*
- *The actual process noise $\boldsymbol{x}_k - f(\boldsymbol{x}_{k-1})$ belongs to support $[\boldsymbol{w}_k] \in \mathbb{IR}^d$,*
- *The actual measurement noise $\boldsymbol{m}_k - h(\boldsymbol{x}_k)$ belongs to the interior of $[\boldsymbol{v}_k] \in \mathbb{IR}^{d_m}$, i.e. $h(\boldsymbol{x}_k)$ belongs to the interior of the set $[\boldsymbol{m}_k] \triangleq \boldsymbol{m}_k + [\boldsymbol{v}_k]$,*
- *The box particle kernels and the box measurement kernel are positive and continuous on their support,*

Therefore, the Box Particle Filter (Algorithm 5) enhanced with Guaranteed Resampling (Algorithm 6) ensures that the actual state always belong to at least one box particle:

$$\forall k \geq 0, \exists i \; \boldsymbol{x}_k \in [\boldsymbol{x}_k^i] \tag{4.23}$$

Proof. Assume that the actual state $\mathbf{x}_k^i$ is included in at least one box particle $[\mathbf{x}_0^i]$:

$$\exists i \in [1, N] \; \mathbf{x}_0 \in [\mathbf{x}_0^i] \tag{4.24}$$

and that the actual process noise belongs to support $[\mathbf{w}_k]$ at each time-step:

$$\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k) \in [\mathbf{w}_k] \tag{4.25}$$

The dynamical model f is assumed to be continuous. Furthermore, assume that the actual state is consistent with the measurement box, when a measurement is available and belongs to the interior of the measurement box $[\mathbf{m}_k]$ (i.e. cannot belong to its boundary, see Appendix D). In practice, this can be achieved by adding a infinitesimal non-zero volume box $[\varepsilon^m] \in \mathbb{IR}^{d_m}$ to the measurement box, which guarantees that $h(\mathbf{x})$ does not belong to the boundary of $[\mathbf{m}_k]$:

$$\begin{cases} h(\mathbf{x}_k) \in \mathbf{m}_k + [\mathbf{w}_k] \subset [\mathbf{m}_k] \\ [\mathbf{m}_k] \triangleq \mathbf{m}_k + [\mathbf{w}_k] + [\varepsilon^m] \\ \lambda\left([\varepsilon^m]\right) > 0 \end{cases} \tag{4.26}$$

where h is a continuous observation model.

In terms of probabilistic box-kernels $\pi_k^i(\mathbf{x}_k)$ and measurement density π^m, assume that the actual initial state has a non-zero probability to occur in box i defined in (4.24):

$$\pi_0^i(\mathbf{x}_0) \neq 0 \tag{4.27}$$

and is consistent with the process noise, i.e. $p(\mathbf{x}_k|\mathbf{x}_{k-1}) \neq 0$ for the actual $\mathbf{x}_k$. Furthermore, assume that the current actual state is consistent with the measurement density, when a measurement is available:

$$\pi^m(\mathbf{m}_k - h(\mathbf{x}_k)) \neq 0 \tag{4.28}$$

Moreover, π^i and π^m are assumed to be positive and continuous on their box-supports.

Prediction step

Assume that the prior actual stet $\mathbf{x}_{k-1}$ belongs to at least one prior box particle, i.e. $\exists i \in [1, N], \mathbf{x}_{k-1} \in [\mathbf{x}^i_{k-1}]$. By definition of the inclusion functions introduced in Section 2.1.5, the propagation of the prior actual state $\mathbf{x}_k$ by f belongs to the inclusion function output:

$$\mathbf{x}_{k-1} \in [\mathbf{x}^i_{k-1}] \Rightarrow f(\mathbf{x}_{k-1}, \mathbf{u}_k) \in [f]([\mathbf{x}^i_{k-1}], \mathbf{u}_k) \tag{4.29}$$

Provided that the actual process noise belongs to the box $[\mathbf{w}^i_k]$ (4.25), the current actual state $\mathbf{x}_k$ belongs to the predicted box $[\mathbf{x}^i_{k|k-1}]$:

$$\left\{ \begin{array}{l} f(\mathbf{x}_{k-1}, \mathbf{u}_k) \in [f]([\mathbf{x}^i_{k-1}], \mathbf{u}_k) \\ \mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k) \in [\mathbf{w}_k] \end{array} \right. \Rightarrow \mathbf{x}_k \in [f]([\mathbf{x}^i_{k-1}], \mathbf{u}_k) + [\mathbf{w}_k] = [\mathbf{x}^i_{k|k-1}] \tag{4.30}$$

Considering the box kernel propagation equation (4.3) and that the box kernel has a non-zero value for the actual state, i.e. $\pi^i_{k-1}(\mathbf{x}_{k-1}) \neq 0$, the convolution of kernel π^i_{k-1} by $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ yields:

$$\mathbf{x}_{k-1} \in [\mathbf{x}^i_{k-1}] \Rightarrow \pi^i_k(\mathbf{x}_k)\mathbf{1}_{[\mathbf{x}^i_{k|k-1}]}(\mathbf{x}_k) \neq 0 \tag{4.31}$$

As a result, the actual state $\mathbf{x}_k$ belongs to the predicted box particle:

$$\mathbf{x}_{k-1} \in [\mathbf{x}^i_{k-1}] \Rightarrow \mathbf{x}_k \in [\mathbf{x}^i_{k|k-1}] \tag{4.32}$$

Furthermore, since kernels are continuous, box kernel π^i_k is non-zero in a neighbourhood of $\mathbf{x}_k$.

Correction step

By definition of the contractors introduced in Section 2.1.5 and from (4.26), if the actual state belongs to box particle i and is consistent with measurements $[\mathbf{m}_k]$, then it still belongs to box particle i after contraction:

$$\left\{ \begin{array}{l} \mathbf{x}_k \in [\mathbf{x}^i_{k|k-1}] \\ h(\mathbf{x}_k) \in [\mathbf{m}_k] \end{array} \right. \Rightarrow \mathbf{x}_k \in \left\{ \mathbf{x} \in [\mathbf{x}^i_{k|k-1}] \mid h(\mathbf{x}) \in [\mathbf{m}_k] \right\} \subset [\mathbf{x}^i_k] \tag{4.33}$$

Provided that the actual predicted measurement $h(\mathbf{x}_k)$ always belongs to the interior of the measurement box $[\mathbf{m}_k]$ and that the observation model h is continuous, the posterior box always keeps a non-zero volume, i.e. $\lambda([\mathbf{x}^i_k]) > 0$.

Using (4.16), (4.31), and (4.28), the innovation $p_k^i(\mathbf{x}) \triangleq \pi_k^m(\mathbf{m}_k - h(\mathbf{x}))\pi_{k|k-1}^i(\mathbf{x})$ is non-zero for $\mathbf{x} = \mathbf{x}_k$. Since the kernels are positive and continueous, $p_k^i(\mathbf{x}) > 0$ on a neighbourhood of $\mathbf{x}_k$. Using (4.33), this yields:

$$\mathbf{x}_k \in [\mathbf{x}_{k|k-1}^i] \Rightarrow \mathbf{x}_k \in [\mathbf{x}_k^i] \Rightarrow \int_{[\mathbf{x}_k^i]} \pi_k^m(\mathbf{m}_k - h(\mathbf{x}))\pi_{k|k-1}^i(\mathbf{x})d\mathbf{x} > 0 \tag{4.34}$$

which finally leads to:

$$\mathbf{x}_k \in [\mathbf{x}_{k|k-1}^i] \Rightarrow w_k^i \neq 0 \tag{4.35}$$

As a result,

$$\mathbf{x}_{k-1} \in [\mathbf{x}_{k-1}^i] \Rightarrow \begin{cases} \mathbf{x}_k \in [\mathbf{x}_k^i] \\ w_k^i \neq 0 \end{cases} \tag{4.36}$$

Resampling step

In order to guarantee that at least one resampled particle $[\tilde{\mathbf{x}}_k^i]$ contains the actual state, ensure that there is at least one resampled box-particle j such that $\mathbf{x}_k \in [\mathbf{x}_k^i] \Rightarrow \mathbf{x}_k \in [\tilde{\mathbf{x}}_k^j]$. Using (4.21) and step 6 combined with (4.35) yields:

$$\mathbf{x}_k \in [\mathbf{x}_k^i] \Rightarrow n^i \geq 1 \tag{4.37}$$

which leads to a regular sub-paving of n^i box particles whose convex hull is $[\mathbf{x}_k^i]$. Therefore, there is at least one resampled particle $[\tilde{\mathbf{x}}_k^j]$ that contains $\mathbf{x}_k$:

$$n^i \geq 1 \Rightarrow \exists j \in [1, N]\ \mathbf{x}_k \in [\tilde{\mathbf{x}}_k^j] \tag{4.38}$$

By combining (4.24), (4.25), (4.26), (4.27), (4.28), (4.29), (4.30), (4.35), (4.37) and (4.38), under the hypothesis that kernels are positive and continuous and by using algorithm 6:

$$\forall k\ \exists i\ \mathbf{x}_k \in [\mathbf{x}_k^i] \tag{4.39}$$

□

Remark

The above development proves that at least one box particle contains the actual state $\mathbf{x}_k$ at each time-step k, given an initial consistency (4.24), a process consistency (4.25), a measurement consistency (4.26) and some assumptions on the box-kernels and the measurement density. However, this does not prove that the estimation error and the filter's uncertainty tend to zero or are stable (their derivatives tend to zero). Nevertheless, the use of Guaranteed Resampling theoretically ensures that the non-convergence rate (Section 3.5) is zero. Furthermore, Guaranteed Resampling only replaces box particles that became inconsistent with respect to the measurements. Therefore, the resampling frequency tuning may not significantly impact the results.

4.2.3 Numerical results

The use of Guaranteed Resampling for BPF theoretically ensures that at least one box particle contains the actual state. As a consequence, the BPF estimation is always consistent with respect to the measurements. This is of particular interest for ambiguous estimation problems such as the TAN application. In this section, experiments are designed to evaluate the following points:

1. The impact of Guaranteed Resampling on the BPF's performance in terms of RMSE and estimated covariance, compared to the original algorithm (Multinomial Resampling BPF, Gning [Gni+13]),
2. The relative impact of Guaranteed Resampling and Multinomial Resampling on the BPF's non-convergence rate,
3. The impact of the resampling frequency on Guaranteed Resampling's performances, compared to Multinomial Resampling.

The considered criteria, as introduced in Section 3.5 are: RMSE ratio (final RMSE divided by initial RMSE), pessimism rate (final estimated standard deviation divided by final RMSE), and non-convergence rate (percentage of simulation ending with an inconsistent estimation). To evaluate points 1, 2, and 3, Guaranteed Resampling BPF is run on all the scenarios presented in Section 3.2. A focus is made on Scenario TAN Moon flyby, which contains the most severely ambiguous measurements considered in this work. Unless a different setting is explicitly stated, the BPF algorithm is tuned with $N = 900$ box particles, no process noise and a resampling threshold of $\theta_{eff} = 0.7$.

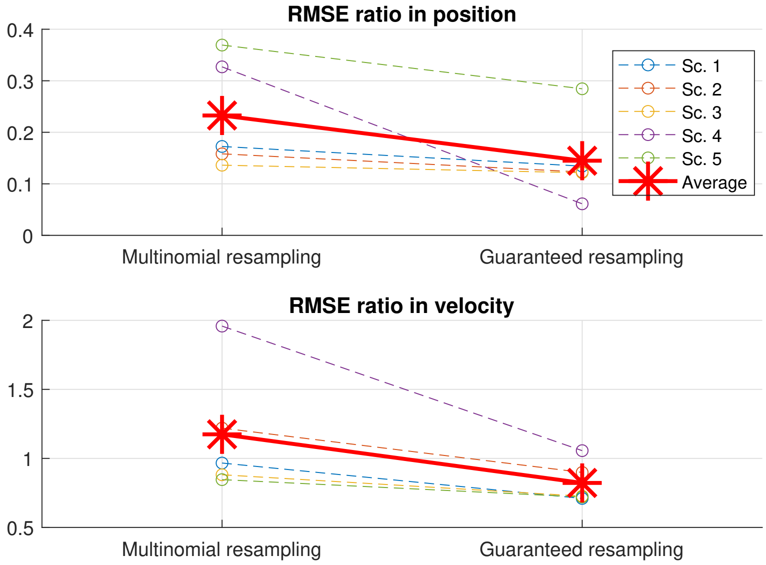
Impact of Guaranteed Resampling on RMSE, pessimism and non-convergences (points 1 and 2)

Figure 4.5 (a) shows the RMSE ratio on position and velocity, respectively defined as the norm of the three positions variables and the three velocities, as introduced in 3.5. Table 4.4 presents the averaged criteria on all scenarios, for a hundred runs. Guaranteed Resampling clearly yields a lower RMSE than Multinomial Resampling, especially in position (a mean RMSE ratio of 0.14 versus 0.23, which represents a 39% decrease). However, Guaranteed Resampling brings more conservatism than Multinomial Resampling. This yields a significant pessimism rate increase (e.g. 1.29 to 2.62 in position, which is twice as much).

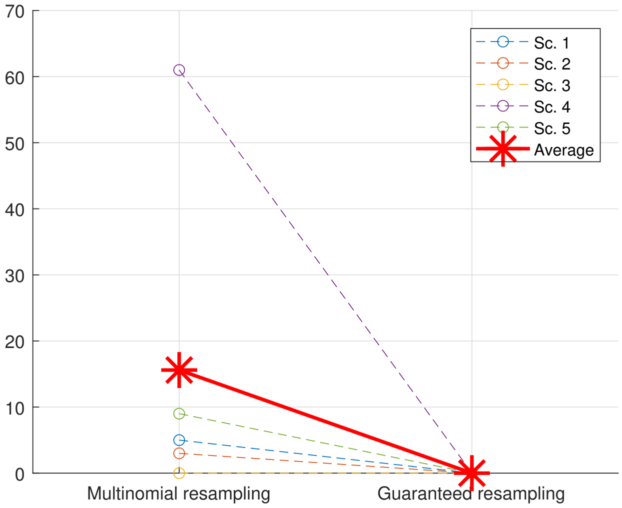
Guaranteed Resampling also has a strong impact on the non-convergence rate, which is theoretically ensured, as zero for all scenarios, as shown in Figure 4.5 (b). It can be noticed that the average computation time per time-step remains the same.

Table 4.4: Simulation results using Guaranteed Resampling and Geometrical Subdivision, averaged on all scenarios at the final time-step.

	Multinomial Resampling	Guaranteed Resampling
Non-convergence rate (%)	15.6	0
RMSE ratio (position)	0.23	0.14
RMSE ratio (velocity)	1.17	0.82
Pessimism (position)	1.29	2.62
Pessimism (velocity)	1.11	1.97
Computation time (ms)	11	11



(a) RMSE ratio



(b) Non-convergence rate

Figure 4.5: Root Mean Square Error ratios (a) and non-convergence rate (b) obtained for all scenarios, using the Multinomial Resampling and the Guaranteed Resampling approaches. The Guaranteed Resampling approach significantly outperforms Multinomial Resampling in terms of RMSE and ensures a zero non-convergence rate.

Table 4.5: Simulation results for Scenario TAN Moon flyby, at final time-step (the terrain resolution is 1000 m)

	Multinomial Resampling	Guaranteed Resampling
Non-convergence rate (%)	61	0
RMSE (position, m)	23958	2612
RMSE (velocity, ms^{-1})	13.1	7.05
Covariance (position, m)	16533	9923
Covariance (velocity, ms^{-1})	5.69	11.7
Pessimism (position)	1.18	3.79
Pessimism (velocity)	0.43	1.66
RMSE ratio (position)	0.33	0.06
RMSE ratio (velocity)	1.96	1.05

Focus on the lunar orbiter scenario

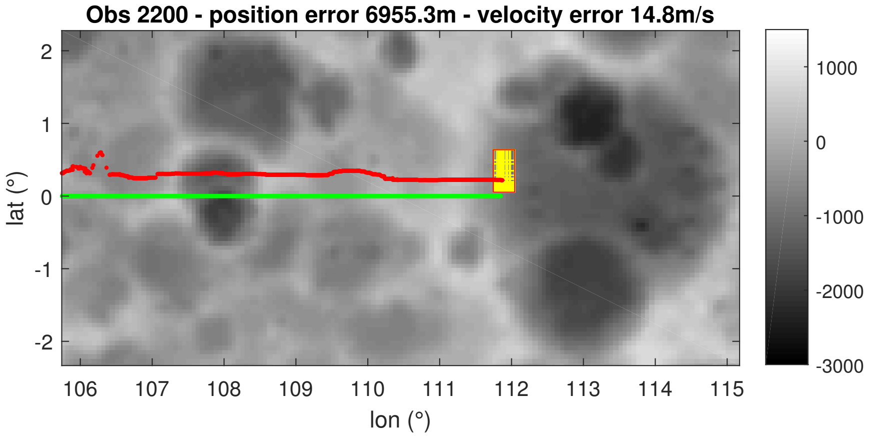
A focus is made on Scenario TAN Moon flyby, which represents the most significant improvement between Multinomial Resampling and Guaranteed Resampling. Table 4.5 presents the results at the final time-step for Scenario TAN Moon flyby, for a hundred runs. As theoretically expected, Guaranteed Resampling leads to a non-convergence rate of 0%, while 61% of the simulations diverge with Multinomial Resampling. The Guaranteed BPF RMSE is significantely lower (2km versus 24km in position, which represents a 91% decrease).

Multinomial Resampling yields a final velocity RMSE ratio of 1.96, which means that the final error is twice as much as the initial error. In other words, Multinomial Resampling downgrades the prior knowledge. Although Guaranteed Resampling does not allow the filter to lower the velocity RMSE (ratio of 1.05), it does not significantly increase the level of error, as does Multinomial Resampling. Figure 4.7 plots the RMSE in position and velocity for BPF using Multinomial Resampling (blue curves) and Guaranteed Resampling (green curves). The prior information discrepancy over time by Multinomial Resampling is clearly visible for the velocity RMSE.

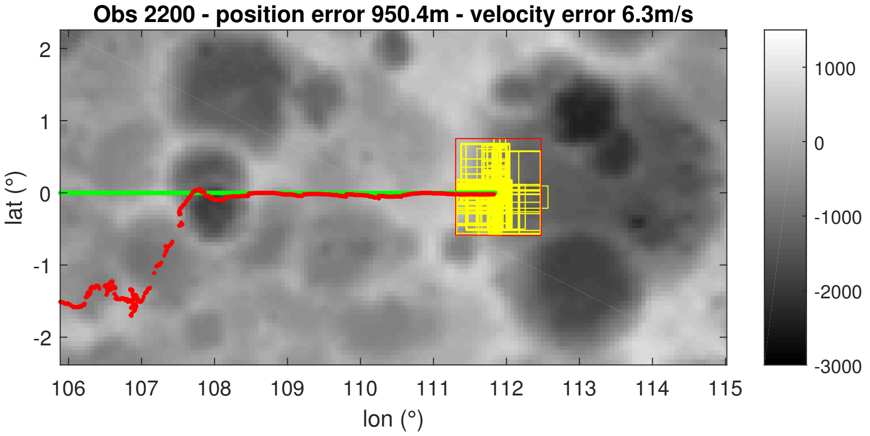
As previously seen, Guaranteed Resampling leads to a more pessimistic estimation with a pessimism ratio's increase by 3 in position (1.18 to 3.79) and by 4 in velocity (0.43 to 1.66), compared to Multinomial Resampling. Nevertheless, these pessimism rate comparisons are not strongly significant since Multinomial Resampling produces a large majority of inconsistent estimations (61% of non-convergence). Figure 4.6 illustrates the behavior of BPF using Multinomial Resampling (a) and Guaranteed Resampling (b), in the same conditions (the same initial errors and measurements realisations). Figure 4.6 (a) illustrates a case where the use of Multinomial Resampling leads to a non-convergence. Again, Guaranteed Resampling ensures the estimation consistency, but leads to a larger estimation confidence, as illustrated in Figure 4.6 (b). Nevertheless, the estimation errors are significantly lower for all state variables.

Impact of the resampling frequency on the algorithm's performances (point 3)

The resampling step is triggered by a threshold whose tuning directly impacts how often the resampling algorithm is called. This section compares the behavior of original BPF and Guaranteed BPF for several values of the *N efficient* threshold θ_{eff} (Kong and Liu [KLW94], see equation (2.22)). This threshold quantifies the lower admissible fraction of the particles whose weights are said to be *efficient*. As introduced in Chapter 2, Section 2.1.4, the resampling step is triggered if the estimated number of efficient particles is below this threshold. Therefore, the resampling frequency grows with θ_{eff}.



(a) BPF with Multinomial Resampling



(b) BPF with Guaranteed Resampling

Figure 4.6: Comparison between estimations obtained with Multinomial Resampling (a) and Guaranteed Resampling (b) on Scenario TAN Moon flyby. The actual trajectory is in green, the estimate is in red, and box particles are in yellow.

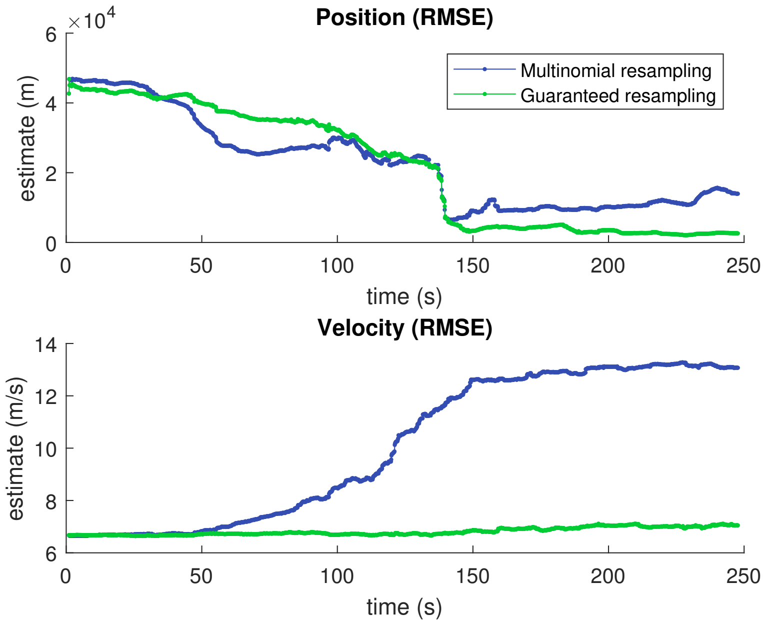


Figure 4.7: Comparison between the Root Mean Square Errors obtained with Multinomial Resampling and Guaranteed Resampling on Scenario TAN Moon flyby.

Table 4.6 compares the averaged performance of original BPF and Guaranteed BPF at the final time-step for several values of θ_{eff}, on all scenarios for a hundred runs. For Multinomial Resampling, the threshold appears difficult to tune since a balance between accuracy (low RMSE) and robustness (low non-convergence rate) must be achieved. This imposes a problem dependant tuning, since high values of θ_{eff} may lead to a high non-convergence rate (e.g. 68% for $\theta_{eff} = 0.9$). On the contrary, Guaranteed Resampling prevents RMSE discrepancy while θ_{eff} tends to 1. The RMSE ratios are globally kept equal for the different values of θ_{eff}, which shows that the resampling frequency has a low impact on the Guaranteed BPF performances (e.g. the averaged position RMSE ratio remains in the interval $[0.15, 0.18]$). The minimum value is obtained for $\theta_{eff} = 0.7$. Therefore, a tuning of $\theta_{eff} \in [0.5, 0.7]$ is advised. As stated previously, Guaranteed Resampling yields more conservatism than Multinomial Resampling: the pessimism ratio is greater than 1.8 for both position and velocity (recall that the value is considered to be acceptable between 0.8 and 1.5).

Figure 4.8 plots the non-convergence rate obtained for all the scenarios, which highlights the non-convergence risk of Multinomial Resampling. As shown in Table 4.6, Guaranteed Resampling ensures a zero non-convergence rate for all values of θ_{eff}.

Table 4.6: Simulation results for several values of the resampling frequency (set by θ_{eff}) averaged on all scenarios at the final time-step

	Multinomial Resampling				Guaranteed Resampling			
Resampling threshold θ_{eff}	0.3	0.5	0.7	0.9	0.3	0.5	0.7	0.9
Non-convergence rate (%)	0.8	4.4	17	68	0	0	0	0
RMSE ratio (position)	0.17	0.17	0.23	1.03	0.18	0.16	0.15	0.17
RMSE ratio (velocity)	1.01	1.04	1.20	2.19	0.91	0.86	0.81	0.82
Pessimism (position)	1.95	1.69	1.22	0.89	2.19	2.35	2.45	2.39
Pessimism (velocity)	1.58	1.46	1.06	0.52	1.80	1.90	1.99	1.97

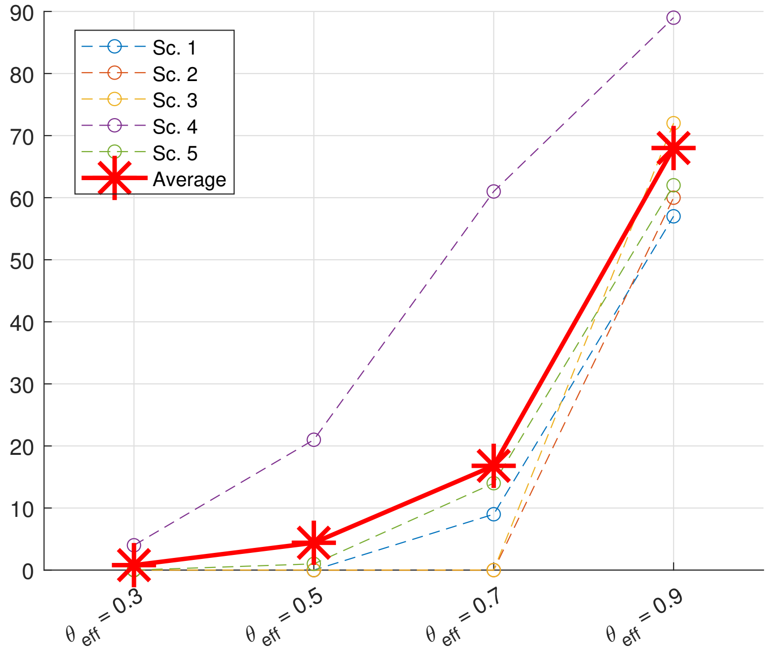


Figure 4.8: Non-convergence rate obtained with Multinomial Resampling on all scenarios for several resampling frequencies (set by θ_{eff}). The tuning of the threshold is highly case-dependant. On the contrary, Table 4.6 shows that the use of Guaranteed Resampling ensures a zero non-convergence rate whatever be the threshold tuning.

4.3 NEW WAYS OF PERFORMING BOX RESAMPLING BY SUBDIVISION

The previous section introduced a guaranteed version of Multinomial Resampling called Guaranteed Resampling. It ensures that the actual state is contained by at least one box particle, which prevents the algorithm from estimation inconsistency. It also yields a lower RMSE than using Multinomial Resampling. However, it introduces conservatism and yields some unnecessary high estimated covariance. This section introduces two new approaches that aim to lower the pessimism ratio and continues to decrease the RMSE.

The BPF resampling step consists of replacing each box particle by a sub-paving of $n^i \in [0, N]$ subdivisions of itself (see Chapter 2, Section 2.1.6). The number n^i is computed by a resampling algorithm, e.g. Multinomial Resampling (Algorithm 3), or Guaranteed Resampling (Algorithm 6), as introduced in the previous section. If n^i is zero, box particle i is removed. If n^i is equal to one, box particle i is kept as it is. If n^i is greater than one, it is subdivided and yields n^i new box particles. For the sake of simplicity, a given box particle can be subdivided along a single state dimension d^i called the *cutting dimension*. However, as developed in Section 2.1.6, the cutting dimension determination is often problem-dependent. To the best of the author's knowledge, only one general approach was proposed in the literature (see Gning [Gni+13]). It consists of a uniform random choice over the state dimensions $\{1, ..., d\}$. Although it is simple to implement, it is likely to produce degenerated box particles whose diameters are disproportioned. This may yield inaccuracy, as experimentally shown in what follows.

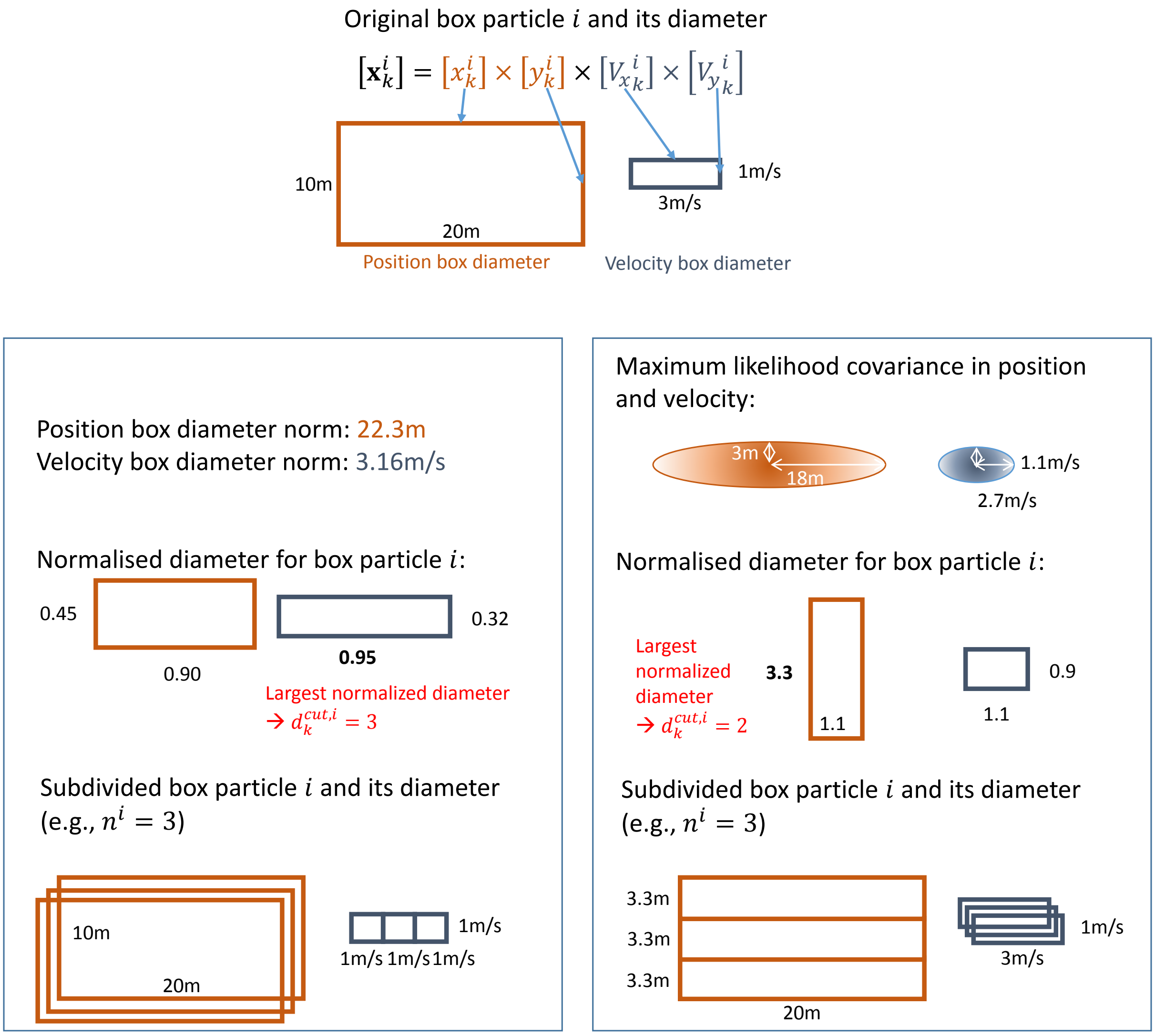
This section is dedicated to the design of two methods to choose the cutting dimension during the box resampling step. Their impact on the BPF's performance is evaluated. Intuitively, they both rely on a box diameter normalisation in order to subdivide each box along its longer normalised edge. The normalisation operation is necessary to enable comparisons of box particle diameters since they do not have the same order of magnitude in the general case. Figure 4.9 illustrates this idea.

A first approach consists of splitting the state vector into several sub-vectors w.r.t the orders of magnitude of their diameters (e.g. positions or velocities). For each box, each sub-vector of diameters can then be normalised by its euclidean norm. This results in a normalised vector of values between 0 and 1. The largest defines the subdivision dimension for this box particle. This makes it possible to maintain each sub-box squared proportional (in terms of intervals lengths in positions and velocities etc.), i.e. to keep the edge lengths roughly equal for each sub-box. This approach is described in Section 4.3.1. Figure 4.9 (a) illustrates this approach, described in the next section.

A second approach accounts for the information carried by measurements. Each box can be subdivided along the dimension which corresponds to the state variable receiving the greater amount of information from the measurements, given the box's current diameter. In other words, each box is subdivided along its side which is the larger with respect to the state estimation accuracy. This results in box particles which have smaller intervals on dimensions receiving information (i.e. currently observed dimensions), and larger intervals on non observed ones. It is achieved by normalising the box particle diameters by the standard deviation of the maximum likelihood theoretical estimator. Figure 4.9 (b) illustrates this approach. It is described in Section 4.3.2.

Whatever the subdivision method, Guaranteed Resampling (Section 4.2) retains its guaranteed property, since the whole domain previously covered by non-zero weighted particles remains covered by a sub-paving of them.

Section 4.3.3 evaluates the impact of the two subdivision approaches on the filter's performance. They are also compared with the original subdivision approach which consists of randomly picking $d_k^{cut,i}$ (Gning [Gni+13]). Subdivision methods presented in this section can be equivalently used with Multinomial Resampling or Guaranteed Resampling, since they are independent from the determination of the new instances number per particle n^i. The performance of the subdivision approaches is discussed in the context of both resampling algorithms (Multinomial and Guaranteed resampling approaches).



(a) Geometrical subdivision

(b) Maximum likelihood covariance subdivision

Figure 4.9: Illustration of the two approaches to subdivide boxes during resampling step: the Geometrical Subdivision and the Maximum Likelihood Covariance Subdivision, leading to n^i new box particles after the subdivision of particle i along dimension $d_k^{cut,i}$

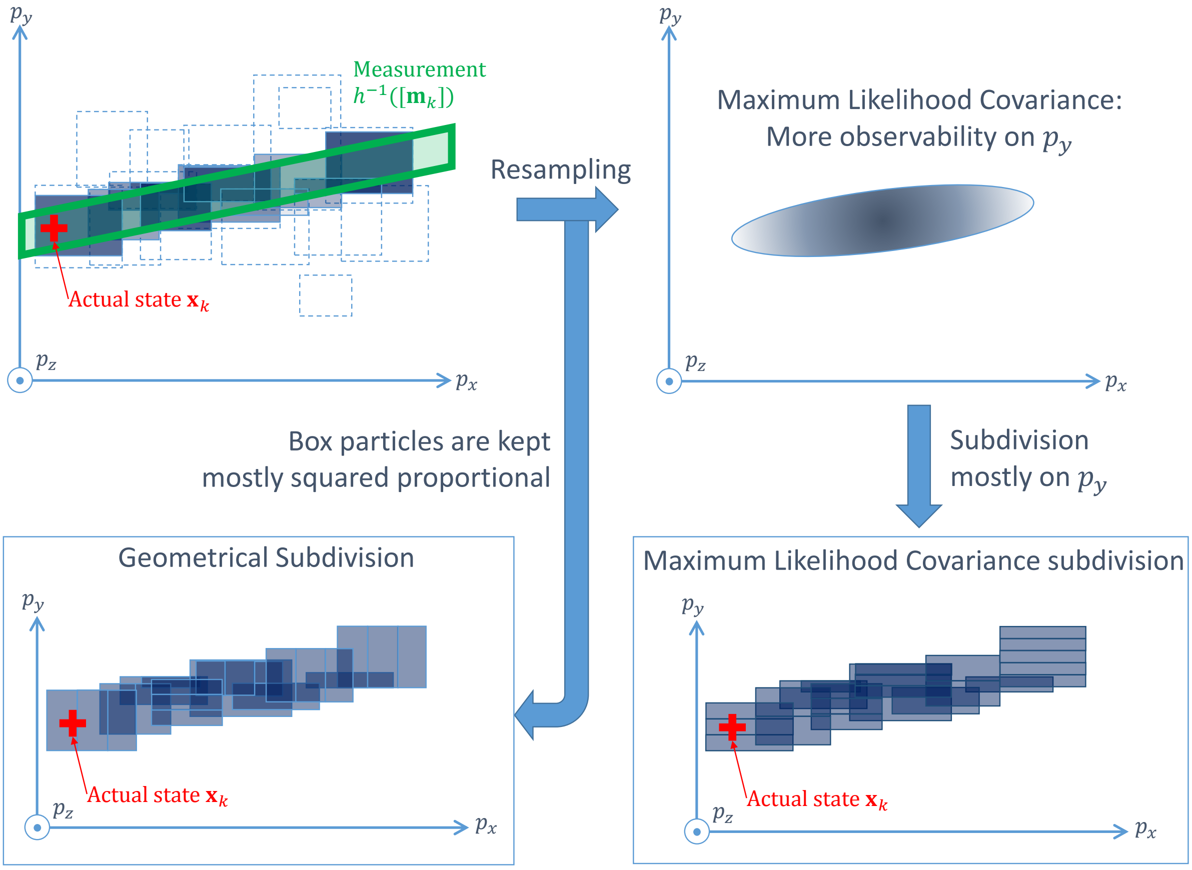


Figure 4.10: Illustration of the two approaches to subdivide boxes during resampling step, on the whole box particle cloud.

4.3.1 Geometrical Subdivision

A first solution to normalise each box diameter vector $\delta_{[\mathbf{x}_k^i]} \in \mathbb{R}^d$ is to account for each box particle diameter's orders of magnitude (see Table 2.1 in Chapter 2 for the box diameter definition). In what follows, for the sake of brevity, each box particle diameter will be denoted δ_k^i. The following proposition generalises for any box to be subdivided.

Proposition 3. *Let be a box $[\boldsymbol{x}] = [x_1] \times ... \times [x_d] \in \mathbb{IR}^d$ such that its diameters can be rearranged by $n \in \mathbb{N}^*$ sub-vectors of equal length $d' \in \mathbb{N}^*$ such that $nd' = d$:*

$$\delta \triangleq [|[x_1]|, ..., |[x_d]|]^T = \left[\delta_1^T, ..., \delta_n^T\right]^T \tag{4.40}$$

where each sub-vector p is composed of d' interval diameters:

$$\delta_p \triangleq \left[\delta_{p,1}, ..., \delta_{p,d'}\right]^T \in \mathbb{R}^{d'} \tag{4.41}$$

These coefficients are regrouped by order of magnitude, i.e.

$$\forall (i,j) \in [1,d']^2, \delta_{p,i} = \mathcal{O}(\delta_{p,j}) \tag{4.42}$$

Then, the box diameter can be normalised by:

$$\tilde{\delta} \triangleq \left[\frac{1}{\|\delta_1\|}\delta_1^T, ..., \frac{1}{\|\delta_n\|}\delta_n^T\right]^T = \left[\tilde{\delta}_1, ..., \tilde{\delta}_d\right]^T \tag{4.43}$$

where $\|\delta_j\|$ is the euclidean norm of the diameter $\delta_j \in \mathbb{R}^{d'}$ with $j \in [1,n]$. The box can be subdivided in n^i new boxes (provided by a resampling algorithm, e.g. Algorithm 6) along the dimension that maximises the normalised diameter:

$$d^{cut} = \underset{j \in [\![1,d]\!]}{\operatorname{argmax}}(\tilde{\delta}_j) \tag{4.44}$$

Remark: In practice, the box diameters can also be rearranged by physical dimensions, that often match with the orders of magnitude.

Examples: Consider a 6 dimensional state vector made of three metric positions $\mathbf{p}_k \in \mathbb{R}^3$ and three velocities $\mathbf{V}_k \in \mathbb{R}^3$. This state representation is commonplace. An example is the double integrator (3.1) (see Section 3.1). Consider one box particle i whose sub-diameters are denoted $\delta_{\mathbf{p},k}^i \in \mathbb{R}^3$ for the position and $\delta_{\mathbf{V},k}^i \in \mathbb{R}^3$ for the velocity. Then, by applying (4.43), each box particle diameter can be normalised by:

$$\tilde{\delta}_k^i = \left[\frac{1}{\|\delta_{\mathbf{p},k}^i\|}{\delta_{\mathbf{p},k}^i}^T, \frac{1}{\|\delta_{\mathbf{V},k}^i\|}{\delta_{\mathbf{V},k}^i}^T\right]^T \tag{4.45}$$

Similarly, Geometrical Subdivision can be applied to IMU hybridisation (see Section 3.1). In this case, the 15 state variables can be arranged in $n = 5$ sub-vectors of $d' = 3$ variables, namely: metric position errors, velocity errors, attitude angles errors, accelerometer bias, and gyrometers bias.

Figure 4.9 (a) illustrates the Geometrial Subdivision approach. Consider a box particle of $\mathbb{IR}^4$ having a diameter of $\delta_k^i = [20, 10, 3, 1]^T$. In this example, the state vector is made of two positions (x, y) and two velocities (V_x, V_y). It can thus be split into two sub-vectors (position $[x, y]^T$ and velocity $[V_x, V_y]^T$). The euclidean norms of the diameters of the two sub-boxes for for box particle i are $\alpha = 22.3$ m in position and $\beta = 3.16$ m in velocity. Therefore, $\delta_{[\mathbf{x}_k^i]}$ can be normalised via these diameters using (4.43). It results in the normalised diameter vector $\tilde{\delta}_k^i = [0.90, 0.45, 0.95, 0.32]^T$. Therefore, the larger normalised diameter is V_x ($d_k^{cut,i} = 3$). This results in n^i new box particles (in this example, $n^i = 3$) whose intervals on x, y, V_y are equal; and whose intervals on V_x are obtained from a regular paving of interval on V_x of initial box particle i. The resulting box shapes will then remain roughly squared proportional for each physical dimension (position, velocity). This prevents them from degenerating to *flat* boxes, i.e. when at least one edge's length of the box tends to zero. The total number of box particles remains the same.

The computational load required by Geometrical Subdivision can be evaluated in terms of floating-point operations, i.e. the total number of elementary operations (additions and multiplications). It can be calculated using the complexity analysis introduced in Appendix C. Geometrical Subdivision requires a total computational load of:

$$c_{GR} = 4Nd + o(Nd) \tag{4.46}$$

where d is the state dimension and N the number of box particles.

4.3.2 Maximum Likelihood Covariance Subdivision

In the previous section, a box subdivision approach has been introduced. It aims to select the subdivision dimension along the larger box's normalised side. The normalisation was performed by regrouping the state parameters by sub-vectors of same order of magnitude in order to keep the box well proportioned.

This section introduces a subdivision approach based on the measurement information brought to each state variable. The goal is to subdivide the boxes along their larger side with respect to the expected estimation accuracy on this dimension. This requires to compute the lowest expected uncertainty along each state dimension.

Principle and example

A first approach to compute the lowest expected uncertainty is the Cramer-Rao Lower Bound (CRLB), defined as the inverse of the Fisher Information Matrix (FIM). However, as presented in Appendix A, the FIM is not defined for bounded measurement densities, which prevents its use under the BPF's assumptions. Nevertheless, the optimal expected uncertainty can be approached by the covariance of the Maximum Likelihood (ML) estimator. In the linear case, it was shown to be equal to the CRLB. For non-linear cases, it asymptotically tends to the CRLB (Wasserman [Was13, Theorem 10.18]). Each box particle's diameter can then be normalised on each dimension by the ML estimator's covariance. For each box particle, the subdivision dimension corresponds to the largest normalised diameter's dimension.

Figure 4.9 (b) illustrates this approach in an example for one box particle. A box particle of $\mathbb{IR}^4$ has a diameter of $\delta_k^i = [20, 10, 3, 1]^T$. Assume that the Maximum Likelihood estimator's theoretical covariance is known, whose computation is introduced in the next section. In this example, assume that the lowest expected standard deviation is, for each state variable, 18 m, 3 m, 2.7 ms^{-1}, and 1.1 ms^{-1}. In this example, the measurements provide more information about dimension y than about other dimensions. Therefore, the Maximum Likelihood covariance is lower on this dimension than on other dimensions. Diameter δ_k^i can be normalised using equation (4.48). This results in the normalised diameter vector

$\tilde{\delta}_k^i = [1.1, 3.3, 1.1, 0.9]^T$. Therefore, the dimension to subdivide is y ($d_k^{cut,i} = 2$). It results in n^i new box particles (in this example, $n^i = 3$) whose intervals on x, V_x, V_y are equal and whose intervals on y are obtained from a regular paving of interval on y of initial box particle i. This can also be interpreted as an increase of state density resolution on the most observed state dimension. The total number of box particles remains the same.

Figure 4.10 (right side) illustrates this approach in an example for the whole box particle cloud. The use of the Maximum Likelihood covariance aims to increase the resolution of the density description on the dimensions whose ML's covariance is the lowest while making the filter more stable on poorly observed dimensions. Intuitively, this can be interpreted as *waiting to get sufficient information* before subdividing boxes on a particular dimension. A higher accuracy can be expected, at least while using Multinomial resampling, and a higher robustness. However, in the context of Guaranteed Resampling, which naturally keeps all non-zero box particles, this may introduce some conservatism.

The next section describes the evaluation of the lowest expected uncertainty in terms of maximum likelihood estimator covariance.

Estimation of the lowest expected uncertainty and determination of $d_k^{cut,i}$

Proposition 4. *During the box subdivision resampling, each box can be subdivided along the edge that is the most pessimistic compared to a lowest expected variance. Provided that:*

- *The measurement noise (associated with the likelihood density) has a covariance* $\boldsymbol{R} \in \mathbb{R}^{d_m \times d_m}$*,*
- *The observation model* h *is differentiable*

The subdivision dimension d^{cut} *is chosen by picking the largest coefficient of a normalised box particle's diameter* $\tilde{\delta}_k^i = [\tilde{\delta}_k^{i,1}, ..., \tilde{\delta}_k^{i,d}]^T$*:*

$$d_k^{cut,i} = \underset{j \in [\![1,d]\!]}{\operatorname{argmax}}(\tilde{\delta}_k^{i,j}) \tag{4.47}$$

The normalised diameter is computed from the box particle's prior diameter $\delta_{[x_k^i]}$ *and the inverse of the Cholesky decomposition of a semi-definite positive matrix* $\boldsymbol{\Sigma}_k$ *which represents the lowest expected uncertainty in terms of covariance:*

$$\tilde{\delta}_k^i \triangleq \operatorname{Chol}\left(\boldsymbol{\Sigma}_k\right)^{-1} \delta_k^i \in \mathbb{R}^d \tag{4.48}$$

where $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}$ *can be evaluated by:*

$$\boldsymbol{\Sigma}_k \triangleq \widehat{\boldsymbol{P}}_k + \frac{1}{2} \sum_{i=1}^{N} w_k^i \left(\boldsymbol{\Sigma}_k^i + \boldsymbol{V}_k^i \right) \tag{4.49}$$

where

$$\boldsymbol{\Sigma}_k^i = \operatorname{Vec}^{-1} \left(\left(\lambda \boldsymbol{T}_k^i \otimes \boldsymbol{T}_k^i + (1-\lambda) \boldsymbol{I}_{d^2} \right)^{-1} \operatorname{Vec} \left(\lambda {\boldsymbol{H}_k^i}^T \boldsymbol{R}_k \boldsymbol{H}_k^i + (1-\lambda) \boldsymbol{\Delta}_k^i \right) \right) \tag{4.50}$$

where $\boldsymbol{V}_k^i$ *is obtained from the polar decomposition of* $\boldsymbol{\Sigma}_k^i$*,* $\boldsymbol{T}_k^i = \left(\frac{\partial h}{\partial x}\right)^T \left(\frac{\partial h}{\partial x}\right)$, Vec *is the vectorization operator, and* $\boldsymbol{\Delta}_k^i$ *is a diagonal matrix whose diagonal terms are the squared box's diameters* ${\delta_k^i}^2$.

Proof. In the following, the analytic expression of $\boldsymbol{\Sigma}_k$ is derived, leading to equation (4.64).

The measurement density $p(\mathbf{m}_k | \mathbf{x}_k) = \pi_k^m \left(\mathbf{m}_k - h(\mathbf{x}_k)\right)$ is assumed to have a single maximum $\widehat{\mathbf{x}}_k^i$ inside the subset $[\mathbf{x}_k^i]$ which satisfies:

$$\widehat{\mathbf{x}}_k^i = \underset{\mathbf{x}_k}{\operatorname{argmax}} \left(p(\mathbf{m}_k | \mathbf{x}_k) \right) \tag{4.51}$$

Thus, one can link $h(\widehat{\mathbf{x}}_k^i)$ and $\mathbf{m}_k$ as $h(\widehat{\mathbf{x}}_k^i) = \mathbf{m}_k$. The measurement variance $\operatorname{Cov}\left[\mathbf{m}_k\right] = \mathbf{R}_k \in \mathbb{R}^{d_m \times d_m}$ is assumed to be known. As a result, the maximum likelihood satisfies:

$$\operatorname{Cov}\left[h(\widehat{\mathbf{x}}_k^i) \right] = \operatorname{Cov}\left[\mathbf{m}_k\right] = \mathbf{R}_k \tag{4.52}$$

On the other hand, the observation function h can be locally linearised to first order as follows:

$$h(\widehat{\mathbf{x}}_k^i) = h(\mathbf{x}_k^i) + \mathbf{H}_k^i (\widehat{\mathbf{x}}_k^i - \mathbf{x}_k^i) + o(\widehat{\mathbf{x}}_k^i - \mathbf{x}_k^i) \tag{4.53}$$

with $\mathbf{H}_k^i \triangleq \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k^i} \in \mathbb{R}^{d_m \times d}$ and $\mathbf{x}_k^i = E\left[\pi_k^i\right]$.
This implies that:

$$h(\widehat{\mathbf{x}}_k^i) - h(\mathbf{x}_k^i) \approx \mathbf{H}_k^i(\widehat{\mathbf{x}}_k^i - \mathbf{x}_k^i) \tag{4.54}$$

and that

$$\left(h(\widehat{\mathbf{x}}_k^i) - h(\mathbf{x}_k^i)\right)\left(h(\widehat{\mathbf{x}}_k^i) - h(\mathbf{x}_k^i)\right)^T \approx \mathbf{H}_k^i(\widehat{\mathbf{x}}_k^i - \mathbf{x}_k^i)(\widehat{\mathbf{x}}_k^i - \mathbf{x}_k^i)^T\mathbf{H}^T \tag{4.55}$$

Thus, by taking the expectancy of both hand sides:

$$\begin{aligned} E&\left[\left(h(\widehat{\mathbf{x}}_k^i) - h(\mathbf{x}_k)\right)\left(h(\widehat{\mathbf{x}}_k^i) - h(\mathbf{x}_k)\right)^T\right] \\ &\approx E\left[\mathbf{H}_k^i(\widehat{\mathbf{x}}_k^i - \mathbf{x}_k)(\widehat{\mathbf{x}}_k^i - \mathbf{x}_k)^T{\mathbf{H}_k^i}^T\right] \\ = \mathbf{H}_k^i E&\left[(\widehat{\mathbf{x}}_k^i - \mathbf{x}_k)(\widehat{\mathbf{x}}_k^i - \mathbf{x}_k)^T\right]{\mathbf{H}_k^i}^T = \mathbf{H}_k^i \mathrm{Cov}[\widehat{\mathbf{x}}_k^i]{\mathbf{H}_k^i}^T \end{aligned} \tag{4.56}$$

Thus, the local maximum likelihood must satisfy:

$$\mathrm{Cov}\left[h(\widehat{\mathbf{x}}_k^i)\right] \approx \mathbf{H}_k^i \boldsymbol{\Sigma}_k^i {\mathbf{H}_k^i}^T \tag{4.57}$$

with $\Sigma_k^i \triangleq \mathrm{Cov}[\widehat{\mathbf{x}}_k^i]$. Therefore, by combining (4.52) and (4.57), one can write:

$$\mathbf{H}_k^i \boldsymbol{\Sigma}_k^i {\mathbf{H}_k^i}^T \approx \mathbf{R}_k \tag{4.58}$$

In practice, since the actual state $\mathbf{x}_k$ is unknown, $\mathbf{H}_k^i$ can be evaluated from $\mathbf{H}_k^i \approx \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{c}_k^i}$ with $\mathbf{c}_k^i$ the center of the i^{th} box particle. Equation (4.58) imposes a constraint on $\boldsymbol{\Sigma}_k^i$ which depends on the observation equation.

However, if the rank of $\mathbf{R}_k$ is less than the rank of $\boldsymbol{\Sigma}_k^i$, some additional information needs to be added. Indeed, (4.58) only affects the coefficients of $\boldsymbol{\Sigma}_k^i$ that explicitly depend on the measurement in the observation function h. In order to calculate the other coefficients, which are linked together through the dynamical model $f()$, a possible solution is to introduce a dynamical information with $\boldsymbol{\Delta}_k^i \triangleq \mathrm{Diag}\left({\delta_k^i}^2\right)$. Function Diag() transforms a vector of $\mathbb{R}^n$ into a diagonal matrix of $\mathbb{R}^{n\times n}$. Such techniques are known as *regularisation* techniques (Neumaier [Neu98]). Therefore, the problem to solve is a trade-off between the observation constraint and the state constraint:

$$\begin{cases} \boldsymbol{\Sigma}_k^i = \mathrm{argmin}\left(J(\boldsymbol{\Sigma})\right) \\ J(\boldsymbol{\Sigma}) = \lambda \left\|\mathbf{H}_k^i \boldsymbol{\Sigma} {\mathbf{H}_k^i}^T - \mathbf{R}_k\right\|_F^2 + (1-\lambda)\left\|\boldsymbol{\Sigma} - \boldsymbol{\Delta}_k^i\right\|_F^2 \\ \boldsymbol{\Sigma} > 0 \end{cases} \tag{4.59}$$

where $\lambda \in (0,1)$ is a tuning coefficient. Equation (4.59) can be solved by computing the derivative of J:

$$\begin{array}{rcl} \frac{\partial J}{\partial \mathbf{\Sigma}} & = & \frac{\partial}{\partial \mathbf{\Sigma}} \left[\lambda tr\left((\mathbf{H}_k^i \mathbf{\Sigma} {\mathbf{H}_k^i}^T - \mathbf{R}_k)^T (\mathbf{H}_k^i \mathbf{\Sigma} {\mathbf{H}_k^i}^T - \mathbf{R}_k)\right) + (1-\lambda) tr\left((\mathbf{\Sigma} - \mathbf{\Delta}_k^i)^T (\mathbf{\Sigma} - \mathbf{\Delta}_k^i)\right)\right] \\ & = & 2\lambda \frac{\partial}{\partial \mathbf{\Sigma}} tr\left((\mathbf{H}_k^i \mathbf{\Sigma} {\mathbf{H}_k^i}^T)^T (\mathbf{H}_k^i \mathbf{\Sigma} {\mathbf{H}_k^i}^T)\right) - 2\lambda \frac{\partial}{\partial \mathbf{\Sigma}} tr\left(\mathbf{H}_k^i \mathbf{\Sigma} {\mathbf{H}_k^i}^T \mathbf{R}_k\right) \\ & & +(1-\lambda) \frac{\partial}{\partial \mathbf{\Sigma}} tr\left(\mathbf{\Sigma}^2\right) - 2(1-\lambda) \frac{\partial}{\partial \mathbf{\Sigma}} tr\left(\mathbf{\Sigma} \mathbf{\Delta}_k^i\right) \\ & = & 2\lambda \left({\mathbf{H}_k^i}^T \mathbf{H}_k^i \mathbf{\Sigma} {\mathbf{H}_k^i}^T \mathbf{H}_k^i\right) - 2\lambda \left({\mathbf{H}_k^i}^T \mathbf{R}_k \mathbf{H}_k^i\right) + 2(1-\lambda)\mathbf{\Sigma} - 2(1-\lambda)\mathbf{\Delta}_k^i \end{array} \tag{4.60}$$

Finally, the equation $\frac{\partial J}{\partial \mathbf{\Sigma}} = 0$ is equivalent to:

$$\lambda \mathbf{T}_k^i \mathbf{\Sigma} \mathbf{T}_k^i + (1-\lambda)\mathbf{\Sigma} = \mathbf{\Omega}_k^i \tag{4.61}$$

with $\mathbf{T}_k^i = {\mathbf{H}_k^i}^T \mathbf{H}_k^i$ and $\mathbf{\Omega}_k^i \triangleq \lambda {\mathbf{H}_k^i}^T \mathbf{R}_k \mathbf{H}_k^i + (1-\lambda)\mathbf{\Delta}_k^i$. Using the Kronecker product properties, denoted $\otimes$, derive (4.61) as follows:

$$(\lambda \mathbf{T}_k^i \otimes \mathbf{T}_k^i + (1-\lambda)\mathbf{I}_{d^2}) \mathrm{Vec}(\mathbf{\Sigma}) = \mathrm{Vec}(\mathbf{\Omega}_k^i) \tag{4.62}$$

where $\mathrm{Vec}() : \mathbb{R}^{d \times d} \to \mathbb{R}^{d^2}$ stands for the column-wise concatenation of a matrix.

As a result, the solution $\mathbf{\Sigma}_k^i$ can be obtained by:

$$\mathbf{\Sigma}_k^i = \mathrm{Vec}^{-1}\left((\lambda \mathbf{T}_k^i \otimes \mathbf{T}_k^i + (1-\lambda)\mathbf{I}_{d^2})^{-1} \mathrm{Vec}(\mathbf{\Omega}_k^i)\right) \tag{4.63}$$

where $\mathrm{Vec}^{-1}() : \mathbb{R}^{d^2} \to \mathbb{R}^{d \times d}$ gives a $d \times d$ matrix representation of a d^2 vector whose elements are taken column-wise.

However, the resulting $\mathbf{\Sigma}_k^i$ matrix might not be positive definite. It can then be approximated by the nearest positive definite matrix, in terms of Frobenius norm, by $\overline{\mathbf{\Sigma}}_k^i = \frac{1}{2}\left(\mathbf{\Sigma}_k^i + \mathbf{V}_k^i\right)$, where $\mathbf{V}_k^i \in \mathbb{R}^{d \times d}$ is obtained from the polar decomposition of $\mathbf{\Sigma}_k^i$, i.e $\mathbf{\Sigma}_k^i = \mathbf{U}_k^i \mathbf{V}_k^i$ with ${\mathbf{U}_k^i}^T \mathbf{U}_k^i = \mathbf{I}_d$. This theorem is developed in the work of Higham [Hig88].

The Maximum Likelihood covariance can therefore be approached by:

$$\mathbf{\Sigma}_k \triangleq \widehat{\mathbf{P}}_k + \sum_{i=1}^{N} w_k^i \overline{\mathbf{\Sigma}}_k^i \tag{4.64}$$

Each box diameter is normalised by the Cholesky decomposition of $\mathbf{\Sigma}$, using (4.48) and the choice of d^{cut} is done using (4.44). ☐

Remark

The computational load required by Maximum Likelihood Covariance Subdivision is obtained by complexity analysis (see Appendix C):

$$c_{MLCS} = Nd^6 + 4Nd^4 + (3 + 2d_m)Nd^2 + o(Nd) \tag{4.65}$$

where d is the state dimension, d_m the measurements dimension and N the number of box particles. The cost c_{MLCS} is thus of $\mathcal{O}(Nd^6)$, that is significantly higher than the computation load required by Geometrical Subdivision, which is about $\mathcal{O}(Nd)$. This high computational load is mainly due to the use of Kronecker products and their inversion for each box particle in (4.63).

4.3.3 Numerical results

This section compares three box-subdivision approaches for BPF resampling. The first, named Random Subdivision, was introduced in Section 2.1.6 in reference to Gning [Gni+13]. It consists of randomly picking $d^{cut,i}$ in $\{1, ..., d\}$ for each box particle with a discrete uniform distribution. The second, called Geometrical Subdivision (GS), is introduced in Section 4.3.1 and consists of choosing the largest box diameter after euclidean normalisation. The third is the Maximum Likelihood Covariance Subdivision approach (MLCS), introduced in Section 4.3.2. It consists of choosing the largest box diameter compared with the lowest expected uncertainty on each state dimension. Experiments were designed to evaluate the following points:

1. The relative impact of each subdivision approach on the filter's RMSE and pessimism, in the context of Guaranteed Resampling;
2. The relative impact of the resampling algorithm (Multinomial and Guaranteed Resampling approaches) on each subdivision method's performances;
3. The impact of measurement information on the RMSE in the context of Maximum Likelihood Covariance Subdivision.

The considered criteria, as introduced in Section 3.5 are: RMSE ratio (final RMSE divided by initial RMSE, quantifying the error of estimation), pessimism rate (final estimated standard deviation divided by final RMSE, quantifying the conservatism) and non-convergence rate (percentage of simulation ending with an inconsistent estimation). The empirical computational load is not discussed in this section and will be tackled in Section 4.5 (see Table 4.13). To evaluate points 1, 2, and 3, the considered BPF configurations are run on all of the scenarios presented in Section 3.2. Unless a different setting is explicitly stated, the BPF algorithm is tuned with $N = 900$ box particles, no process noise and a resampling threshold of $\theta_{eff} = 0.7$.

Impact of each subdivision approach on the Guaranteed BPF's performances (point 1)

The Guaranteed Resampling introduced in Section 4.2 prevents the filter from diverging. However, it also yields a larger estimated covariance than Multinomial Resampling (and a higher pessimism ratio). This section evaluates the impact of the three subdivision approaches on accuracy and conservatism.

Figure 4.11 and Table 4.7 show the RMSE ratios, non-convergence rates and pessimism ratios obtained with the three approaches while using Guaranteed Resampling on all scenarios, for a hundred runs. The random subdivision (Gning [Gni+13]) yields the highest RMSE ratio (0.15 in position, 0.81 in velocity). Maximum Likelihood Covariance Subdivision slightly improves the averaged velocity RMSE ratio (0.74, which represents a 9% decrease), but yields the same averaged position RMSE ratio (0.15). In addition, it yields a slightly higher pessimism than the random picking approach (e.g. in position, 2.51 versus 2.45, which represents a 2.4% increase). Therefore, Maximum Likelihood Covariance Subdivision appears to slightly introduce conservatism. Geometrical Subdivision yields the lowest RMSE in position (0.091, which represents a 39% decrease) and in velocity (a 17% decrease). In addition, it yields a lower pessimism in position (a 34% decrease). However, it fails to reduce the pessimism in velocity which remains equal (a 1% increase). The three approaches appear to be conservative (recall that the value is considered to be acceptable between 0.8 and 1.5).

The conservatism of Maximum Likelihood Covariance Subdivision is illustrated in Figure 4.12 (b) for Scenario TAN Sinusoidal terrain. In this case, box particles have been mostly subdivided along the latitude dimension during the first part of the trajectory. During the second part of the trajectory, box particles have to be subdivided along the longitude dimension. The filter needs more time to converge than with Geometrical Subdivision (Figure 4.12 (a)), which keeps the box particles squared-proportional along the whole trajectory. Multimodalities are then kept longer (two disjoint groups of box particles in yellow can be noticed) and ambiguities may fail to be removed in a fixed time period. This also highlights a drawback of the least square estimator for particle estimation (LS, see Chapter 2, Section 2.1.4). The LS box particle state estimator is obtained by the weighted sum of the box particles expectations, i.e. their centers, and is represented in red in Figure 4.12. The hypothesis is often made that the estimated density tends to a unimodal density. In such a case, the least square estimator tends to the Maximum A Posteriori estimator (MAP, see see Chapter 2, Section 2.1.2). However, in the case of multimodalities, it may provide some estimate which does not belong to the conditional density support. Although MAP evaluation approaches have been proposed for the Particle Filter as discussed in Chapter 2, they often yield a high computational load and will not be considered in this work.

These simulations on different terrain types make it possible to conclude that Guaranteed Resampling with Geometrical Subdivision provides lower RMSE and pessimism than the other approaches.

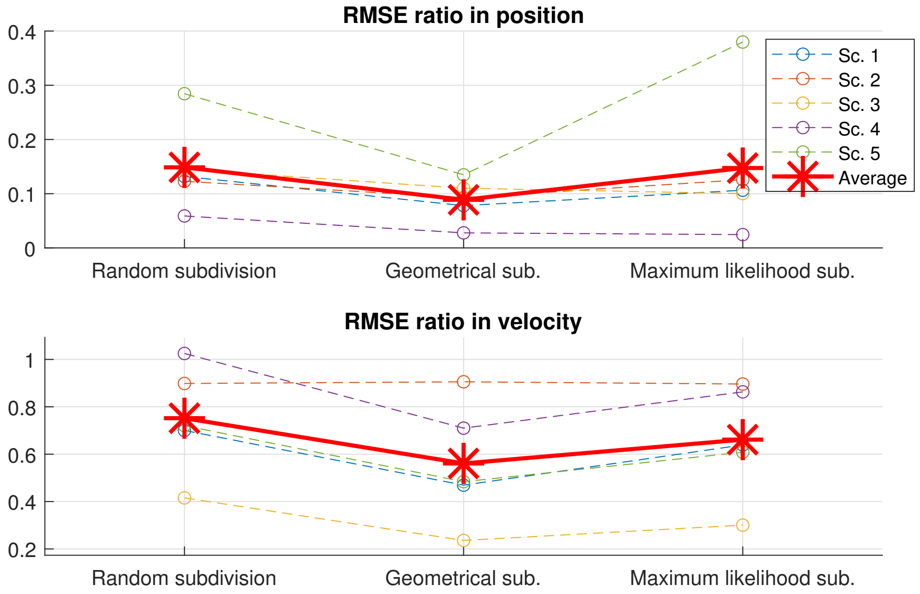
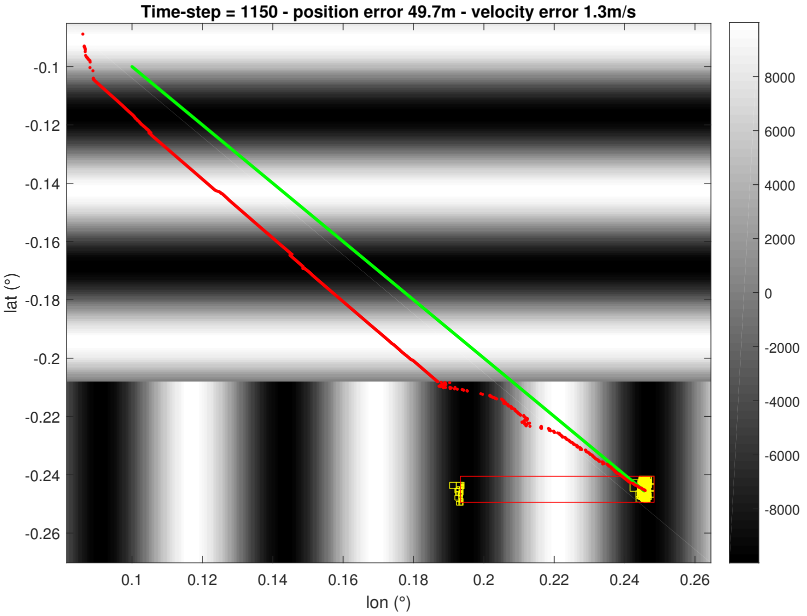


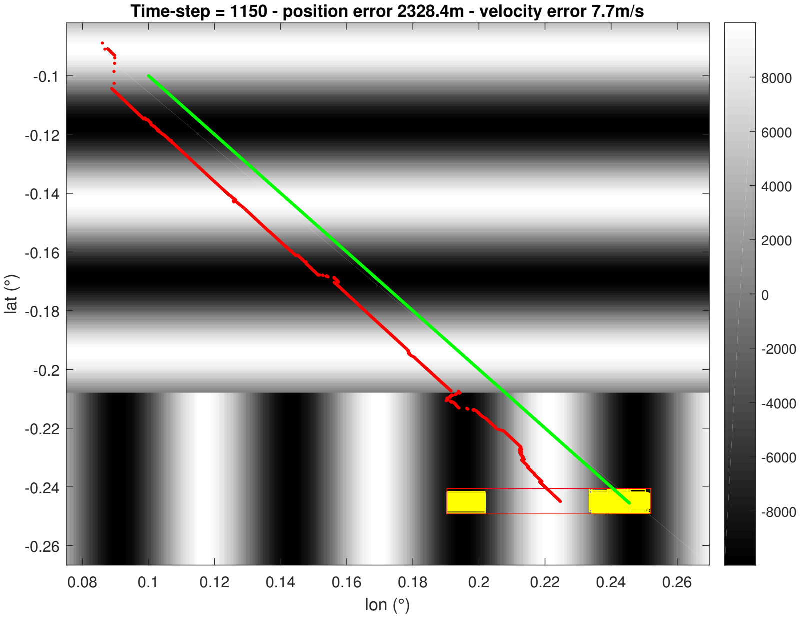
Figure 4.11: Root Mean Square Error ratios for the three subdivision approaches using Guaranteed Resampling on all scenarios. Geometrical Subdivision associated to Guaranteed resampling is able to handle severe ambiguities and provides better accuracy than other approaches.

Table 4.7: Simulation results for the three subdivision approaches using Guaranteed resampling, averaged on all scenarios

	Random subdivision	Geometrical subdivision	Maximum Likelihood subdivision
Non-convergence rate (%)	0	0	0
RMSE ratio (position)	0.15	0.091	0.15
RMSE ratio (velocity)	0.81	0.67	0.74
Pessimism (position)	2.45	1.61	2.51
Pessimism (velocity)	1.99	1.79	2.12



(a) Geometrical Subdivision



(b) Maximum Likelihood Covariance Subdivision

Figure 4.12: Example of estimation trajectory obtained with the Geometrical Subdivision approach (a) and the Maximum Likelihood Covariance Subdivision approach (b), on Scenario TAN Sinusoidal terrain, using Guaranteed Resampling. Actual state is in green, estimate in red, box particles in yellow, and the box hull of all non-zero weighted particles in red. Associated to Guaranteed Resampling, Maximum Likelihood Covariance Subdivision introduces too much conservatism, which may prevent it to remove ambiguities.

Table 4.8: Simulation results for the three subdivision approaches using Multinomial Resampling and Guaranteed Resampling. RMSE ratios are averaged on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion.

	Multinomial Figure 4.13 (a)			Guaranteed Figure 4.13 (b)		
	Random subdivision	Geom. subdivision	Max. Lik. subdivision	Random subdivision	Geom. subdivision	Max. Lik. subdivision
RMSE ratio (pos)	0.15	0.13	0.12	0.13	0.096	0.11
RMSE ratio (vel)	0.99	0.89	0.84	0.67	0.54	0.61
Pessimism (pos)	1.45	1.05	1.22	3.42	2.67	3.58
Pessimism (vel)	1.33	1.06	1.34	2.46	2.82	2.87

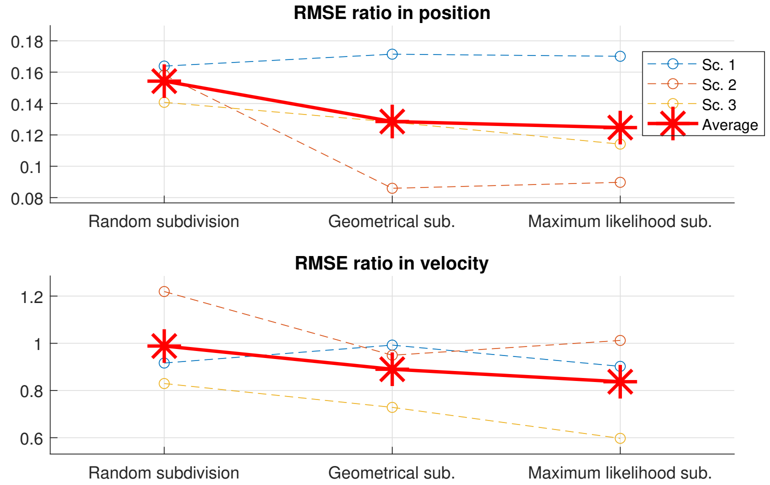
Impact of the resampling algorithm on each subdivision approach (point 2)

In order to evaluate point 2, Multinomial BPF and Guaranteed BPF are compared in three scenarios: Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion. The two other scenarios (Scenario TAN Moon flyby, Scenario TAN Sinusoidal terrain) are excluded from this comparison because Multinomial Resampling yields too many non-convergences on them (see Section 4.2). They would not allow fair comparisons in terms of RMSE and pessimism ratios.

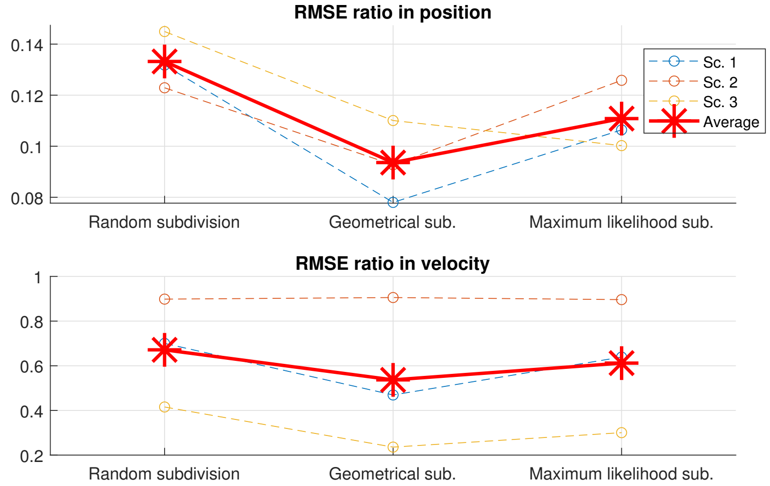
Figure 4.13 and Table 4.8 show the RMSE ratios obtained with the three approaches while using Multinomial Resampling (a) and Guaranteed Resampling (b), for a hundred runs. For all subdivision approaches, using Guaranteed Resampling yields a lower RMSE ratio and higher pessimism than using Multinomial Resampling. The most extreme example is the Geometrical Subdivision case which holds a significant improvement in velocity RMSE (a 39% decrease) and a huge increase in velocity pessimism (1.7 times higher).

In the context of Multinomial Resampling, it can also be noticed that Maximum Likelihood Covariance Subdivision yields slightly lower RMSE than Geometrical Subdivision. The Maximum Likelihood Covariance Subdivision approach aims to favour box subdivision on the most observed state dimensions while limiting it on the others. This is of interest in the context of Multinomial Resampling which may delete some consistent box particles, since it increases the filter's stability by making each box larger on uncertain state dimensions. However, in the context of Guaranteed Resampling, the stability is ensured by construction. Therefore, as seen in the previous section, Maximum Likelihood subdivision introduces too much conservatism, which leads to lower accuracy.

To conclude, Multinomial Resampling yields non-convergences and a higher RMSE than Guaranteed Resampling for all subdivision methods. Guaranteed Resampling however introduces a significant conservatism.



(a) Multinomial resampling



(b) Guaranteed resampling

Figure 4.13: Comparison between Root Mean Square Error ratios for the three subdivision approaches using Multinomial Resampling (a) and Guaranteed resampling (b). Maximum Likelihood Covariance Subdivision provides a better accuracy while using Multinomial Resampling, but is outperfomed by Geometrical Subdivision with Guaranteed Resampling. RMSE ratios are presented on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion.

Impact of measurement information on the BPF's RMSE (point 3)

This section evaluates the influence of the measurement information on the filter's performances in the context of the Maximum Likelihood Covariance Subdivision approaches (Section 4.3.2).

The influence of measurement information is tuned by parameter λ, in (4.59). High values of λ will favour the subdivision of state variables that hold the highest information. Giving λ a high value is equivalent to increasing the resolution of the box particle cloud along the state variable that has the greatest information from the current measurements. As a result, a lower RMSE can be expected for high values of λ than for low values of λ.

Figure 4.14 shows the RMSE ratios obtained for $\lambda \in [0.1, 0.7]$ (using Guaranteed Resampling) on all TAN scenarios. As expected, the RMSE ratio decreases when λ increases, for example 0.18 to 0.15 in position for $\lambda = 0.1$ and $\lambda = 0.7$ respectively, which represents a 16% decrease. Increasing λ also yields less pessimism on observed dimensions, in this case positions, with a pessimism ratio decrease of 6.5%. However, estimation on non-observed dimensions becomes more conservative, with a pessimism increase of 10% in velocity. However, this approach remains conservative (recall that the pessimism ratio is considered to be acceptable between 0.8 and 1.5).

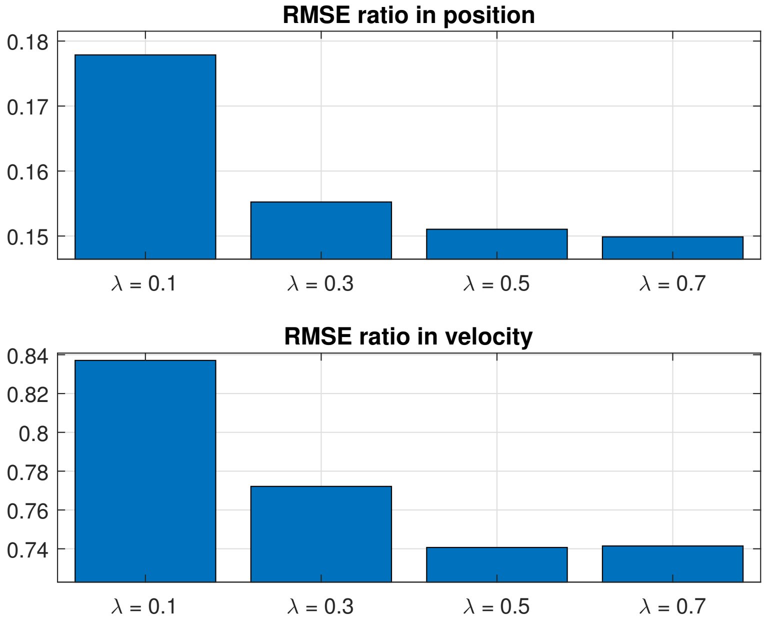
To conclude, increasing the tuning parameter λ yields a slightly better accuracy but introduces some conservatism on non-observed state variables. Note that although the RMSE ratio tends to decrease with λ on average, the observed variations remain small compared to the range of values, as shown by Figure 4.14 (b). Therefore, it can be concluded that the tuning of λ has a limited impact on the filter's performances.

Table 4.9: Simulation results for several values of λ, averaged on all scenarios, using Guaranteed resampling.

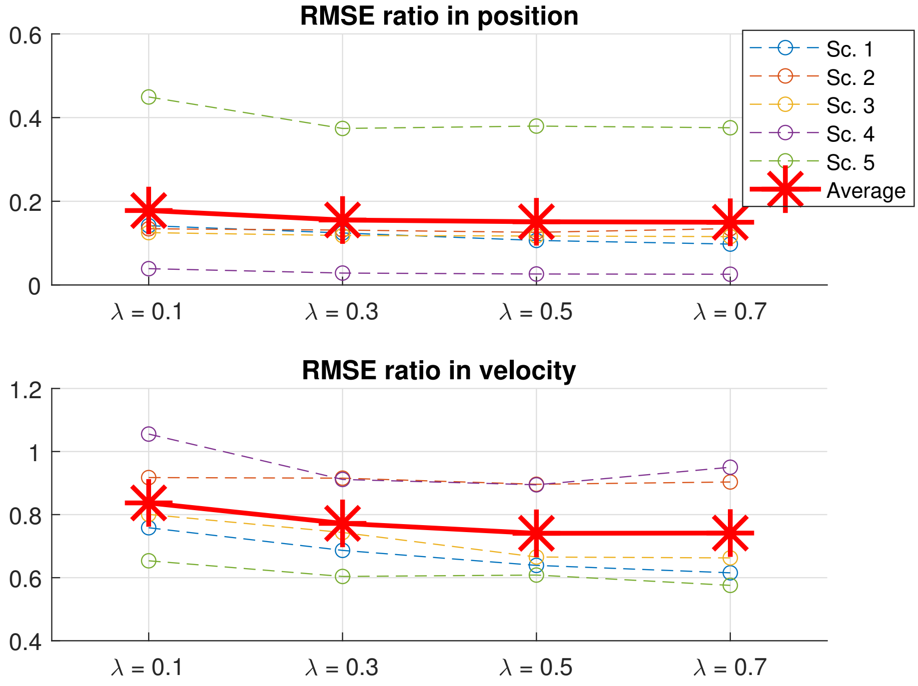
Coefficient λ	0.1	0.3	0.5	0.7
Non-convergence rate (%)	0	0	0	0
RMSE ratio (position)	0.178	0.155	0.151	0.149
RMSE ratio (velocity)	0.837	0.772	0.741	0.741
Pessimism (position)	2.61	2.51	2.51	2.44
Pessimism (velocity)	1.93	2.05	2.12	2.13

4.3.4 Conclusion on subdivision methods

To conclude, two resampling subdivision approaches have been introduced and compared with the existing random subdivision method. Table 4.10 summarises the pros and the cons of each approach, that was previously discussed in more detail. These results tend to show that accounting for observability (i.e. the use of MLCS) is not necessary as long as the filter is robust by construction (ensured by Guaranteed Resampling).



(a) Averaged RMSE ratio



(b) RMSE ratio

Figure 4.14: Comparison of RMSE ratios averaged on TAN scenarios with several values of λ (Maximum Likelihood Covariance Subdivision in the context of Guaranteed Resampling): (a) averaged RMSE ratio, (b) RMSE ratios for all scenarios.

Table 4.10: Subdivision methods pros and cons

Method	Pros	Cons
Random Subdivision (Gning [Gni+13])	Simple to implement	Yields poor performance with respect to other methods, Introduces some random sample generation which makes it non-deterministic.
Geometrical Subdivision	Relatively simple to implement ($\mathcal{O}(Nd)$ operations), Yields the best RMSE when used in conjunction with Guaranteed Resampling, in the context of TAN, Fully deterministic.	Downgraded performances when used in conjunction with Multinomial Resampling, in the context of TAN, Limited to state representations satisfying Proposition 3 axioms (although this covers a large variety of estimation problems).
Maximum Likelihood Covariance	Yields the best RMSE when used in conjunction with Multinomial Resampling, in the context of TAN, Yields more robustness than other methods by subdividing boxes along the most observed dimensions, Can be generalised to any state description, dynamical model, and observation model, Fully deterministic.	Introduces too much conservatism when used in conjunction with Guaranteed Resampling, which makes it less accurate than Geometrical Subdivision, in the context of TAN, More complex to implement ($\mathcal{O}(Nd^6)$ operations).

4.4 THE BOX REGULARISATION

In the previous sections, a guaranteed resampling approach and two deterministic ways to subdivide box particles have been introduced. However, whatever be the resampling methods, this operation ends either with exactly superposed box particles or a regular paving of the state space. It yields a jagged density approximation whose resolution is coarse. As a result, the posterior density $p(\mathbf{x}_k|\mathbf{M}_k)$ is often poorly approximated, which may lead to a biased estimate. A possible solution to enhance the posterior density approximation is by smoothing it. A smoothing approach, called kernel regularisation, was presented by Musso [MOLG01] to improve the robustness of conventional particles filters and was presented in Section 2.1.4. It is based on the theory of Silverman [Sil86] on kernel smoothing and density estimation.

The objective of this section is to determine an optimal smoothing kernel applied to the box parameters (centers and diameters), in terms of the Mean Integrated Square Error (MISE) criterion. An adaptation of the kernel regularisation method to BPF is developped, which leads to an algorithm called Box Regularised Particle Filter (BRPF). The regularisation takes place after the resampling operation and the correction step that was previously presented. It is always triggered consecutively after the resampling step.

4.4.1 Principle and example

Each box particle (in $\mathbb{IR}^d$) can be described by its center ($\mathbf{c}_k^i \in \mathbb{R}^d$) and its diameter ($\mathbf{d}_k^i \in \mathbb{R}^d$), as introduced in Section 2.1.5. Box particles are the d-dimensional supports of bounded kernels. The estimated state conditional density is a weighted sum of these kernels, as developed in Section 4.1.2. After a resampling by box subdivisions, the box particle cloud results in one or several groups of strictly aligned or superposed box particles. The estimated density is thus quite jagged. This aliasing can be interpreted as a high correlation between the particle parameters (centres and diameters). Regularisation consists of smoothing the density by noising the box parameters with a carefully defined stochastic noise. This noise is chosen to minimize the MISE criterion defined in Section 2.1.4, which quantifies a distance between the theoretical conditional density and the estimated one. Theoretically, this results in a more accurate estimation and a lower RMSE (Musso [MOLG01]).

Figure 4.15 illustrates the Box Regularisation principle on an example with three state variables ($\mathbf{x} = [p_x, p_y, p_z]^T$) and 10 box particles. Weights are represented by the box opacity. Regularisation occurs after a resampling step, where 9 box particles were removed and replaced by 10 subdivisions of the same box particle along the p_z dimension. The estimated density (the blue curve) is illustrated before and after regularisation. For readability, the density is illustrated as a marginalisation on one state dimension, for example p_x. After resampling and before regularisation, the estimated density results in the mixture of 10 exactly superposed box particles kernels (in this example, uniform kernels). Therefore, the estimated density is similar to a step function, which does not fit with the theoretical conditional density (the red curve) obtained with the optimal filter. By smoothing the distribution of the box particles' parameters, regularisation makes the estimated density fit more closely to the theoretical conditional density. Note that Guaranteed Resampling (Section 4.2) does not retain its guaranteed property, since the box particles may be shifted. However, it can be assumed that the actual state remains covered by the box particle cloud if the number of box particles is sufficiently high. The next section develops the formalism of Box Regularisation.

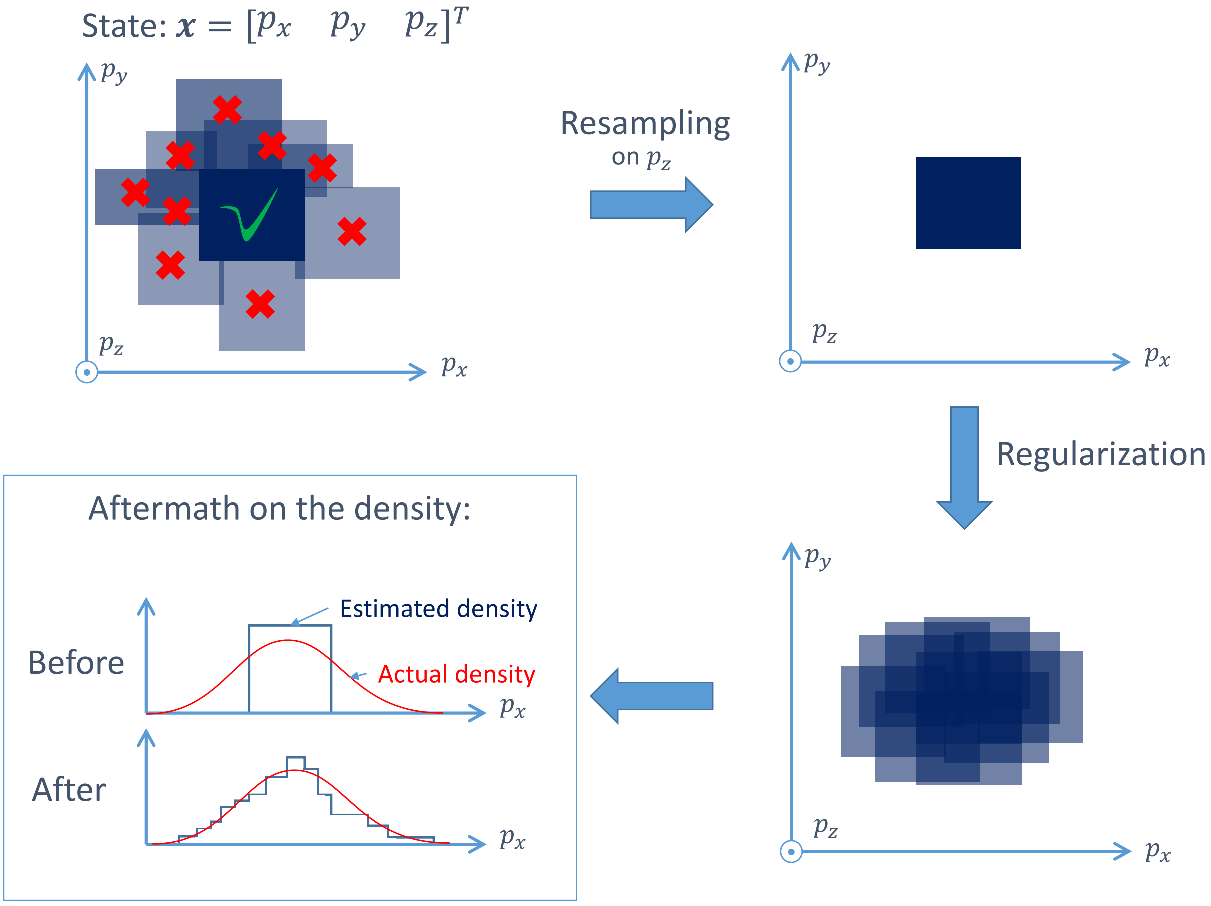


Figure 4.15: Illustration of the Box Regularisation principle, after a resampling involving a cloud of 10 box particles. The estimated density (blue curve) is plotted before and after regularisation and compared with the theoretical conditional density (red curve). For the sake of clarity, the density is marginalized on one dimension.

4.4.2 Kernel Regularisation applied to the Box Particle Filter

This section presents the Kernel Regularisation framework in the context of BPF.

Each box particle $[\mathbf{x}_k^i]$ can be characterised by a vector of $\mathbb{R}^{2d}$ which consists of their center $\mathbf{c}_k^i \in \mathbb{R}^d$ and their diameter $\delta_k^i \in \mathbb{R}^d$. A vector description of a box particle is:

$$\boldsymbol{\xi}_k^{i\,T} = \left[\mathbf{c}_k^{i\,T}, \delta_k^{i\,T}\right] \in \mathbb{R}^{2d} \tag{4.66}$$

Let $\boldsymbol{\xi}_k^T = \left[\mathbf{c}_k{}^T, \delta_k{}^T\right]$ be the random vector representing the boxes parameters (center and diameter). Therefore, a new expression of the density's approximation can be written, using a kernel density centered on each box particle $\boldsymbol{\xi}_k^i$.

$$\widehat{p}(\boldsymbol{\xi}_k|\mathbf{M}_k) = \widehat{p}(\mathbf{c}_k, \delta_k|\mathbf{M}_k) \approx \sum_{i=1}^{N} w_k^i K_h(\boldsymbol{\xi}_k - \boldsymbol{\xi}_k^i) \tag{4.67}$$

where

$$\begin{cases} K_h : \mathbb{R}^{2d} \to \mathbb{R} \\ K_h(\boldsymbol{\xi}) = \frac{1}{h^{2d}} K\left(\frac{1}{h}\boldsymbol{\xi}\right) \end{cases} \tag{4.68}$$

is the re-scaled kernel density $K(.)$ and $h \in \mathbb{R}^{+*}$ the kernel bandwidth. The kernel density is a symmetric probability density function.

The regularisation step therefore consists of noising each box parameter:

$$\boldsymbol{\xi}_k^i \leftarrow \boldsymbol{\xi}_k^i + \mathrm{Chol}(\widehat{\mathbf{S}})\boldsymbol{\epsilon}_k^i \tag{4.69}$$

where $\boldsymbol{\epsilon}_k^i \sim K_h(\boldsymbol{\epsilon}_k^i)$ and $\widehat{\mathbf{S}}$ is the empirical covariance matrix of parameters $\boldsymbol{\xi}_k^i$.

The kernel $K(.)$ and bandwidth h are chosen to minimise the Mean Integrated Square Error (MISE, equation (2.27)) between the theoretical posterior density and the corresponding regularised density. As introduced in Section 2.1.4 (Silverman [Sil86]), an optimal choice of the kernel is the bounded Epanechnikov kernel in terms of MISE asymptotic minimisation. It is assumed that all the box particles have the same weight, which is the case during the resampling step; and that the actual posterior density is asymptotically unimodal. For multimodal cases, a multiplicative coefficient $\mu \in]0,1]$ is introduced to limit the regularisation impact on the filter's stability (equation (2.33)). After kernel smoothing regularisation, the box particle kernels mixture is guaranteed to better fit the optimal posterior density. In practice, this will result in an improved estimation accuracy (RMSE), as will be experimentally shown in the next section. The theoretical asymptotic BRPF posterior density can be approached by replacing each box particle with an Epanechnikov kernel.

Proposition 5. *Let us approach the asymptotic regularised BRPF posterior density by a mixture of Epanechnikov kernels:*

$$\widehat{p}_{\infty}^{BRPF} \triangleq \lim_{k\to\infty} \widehat{p}^{BRPF}(\boldsymbol{x}_k|\boldsymbol{M}_k) \approx \sum_{i=1}^{N} w_k^i K_h(\boldsymbol{x}_k - \boldsymbol{c}_k^i) \tag{4.70}$$

where $\boldsymbol{c}_k^i$ *are the box particles centres and* h *the optimal bandwidth defined in (2.30). Let us approach the asymptotic BPF posterior density without regularisation by a mixture of uniform kernels:*

$$\widehat{p}_\infty^{BPF} \triangleq \lim_{k\to\infty} \widehat{p}^{BPF}(\boldsymbol{x}_k|\boldsymbol{M}_k) \approx \sum_{i=1}^{N} w_k^i \mathcal{U}_{\delta_k^i}(\boldsymbol{x}_k - \boldsymbol{c}_k^i) \tag{4.71}$$

If all weights are equal (during the resampling step), and if the actual density tends to a twice differentiable function defined by $p_\infty \triangleq \lim_{k\to\infty} p(\boldsymbol{x}_k|\boldsymbol{M}_k)$*, then the resulting MISE of the BRPF is lower than the MISE of the original BPF,* $\forall N, d \in \mathbb{N}^*$*:*

$$\mathrm{MISE}\left(\widehat{p}_\infty^{BRPF}, p_\infty\right) < \mathrm{MISE}\left(\widehat{p}_\infty^{BPF}, p_\infty\right) \tag{4.72}$$

Proof. The MISE obtained with a uniform mixture is minimised if the uniform kernels diameters are equal to the optimal bandwidth $h' > 0$ defined by (2.30), which provides a lower bound for the BPF's MISE:

$$\mathrm{MISE}\left(\widehat{p}_\infty^{BPF}, p_\infty\right) \geq \underline{\mathrm{MISE}}^{BPF} \triangleq \mathrm{MISE}\left(\sum_{i=1}^{N} w_k^i \mathcal{U}_{h'}(\mathbf{x}_k - \mathbf{c}_k^i), p_\infty\right) \tag{4.73}$$

For any kernel mixture, the lower reachable MISE is obtained by combining equation (2.28) with (2.30), which yields:

$$\mathrm{MISE}\left(\bullet, p_\infty\right) = \left(\int_{\mathbb{R}^d} (\nabla^2 p(\mathbf{x}))^2 \mathrm{d}\mathbf{x}\right)^{\frac{d}{d+4}} N^{\frac{-4}{d+4}} \alpha^{\frac{2d}{d+4}} \beta^{\frac{4}{d+4}} \left(\frac{1}{4} d^{\frac{4}{d+4}} + d^{\frac{-4}{d+4}}\right) \tag{4.74}$$

where $d > 0$ is the number of dimensions, α and β are the Kernel's parameters defined in (2.29). The ratio of two optimal MISE obtained with different kernel mixtures is thus:

$$r = \left(\frac{\alpha}{\alpha'}\right)^{\frac{2d}{d+4}} \left(\frac{\beta}{\beta'}\right)^{\frac{4}{d+4}} \tag{4.75}$$

where (α, β) are the coefficients corresponding to a first kernel (Epanechnikov in what follows) and (α', β') correspond to a second kernel (uniform in the following). The uniform kernels are approached by beta kernels whose coefficients tend to unity. A more detailed description of the kernels coefficients can be found in Appendix E. Applying this relation to the Epanechnikov and the uniform kernels yields:

$$\frac{\mathrm{MISE}\left(\widehat{p}_\infty^{BRPF}, p_\infty\right)}{\underline{\mathrm{MISE}}^{BPF}} = \left(\frac{3}{4+d}\right)^{\frac{2d}{4+d}} \left(\frac{2(d+2)\Gamma\left(\frac{d}{2}+1\right)}{\pi^{\frac{d}{2}}(4+d)}\right)^{\frac{4}{4+d}} \tag{4.76}$$

where $\Gamma : \mathbb{R} \to \mathbb{R}$ is the Gamma function. This ratio remains strictly bounded between 0 and 1, which yields:

$$\mathrm{MISE}\left(\widehat{p}_\infty^{BRPF}, p_\infty\right) < \underline{\mathrm{MISE}}^{BPF} \leq \mathrm{MISE}\left(\widehat{p}_\infty^{BPF}, p_\infty\right) \tag{4.77}$$

$\square$

Remark 1

The ratio between the BRPF theoretical asymptotic MISE and the BPF lower reachable MISE defined in equation (4.76) is plotted in Figure 4.16 for $d \in [1, 100]$. The MISE enhancement between the regularised asymptotic density (4.70) and the lower reachable value (4.73) for the BPF MISE (4.71) appears to be significant. The next section will evaluate if the regularisation step improves the filter's RMSE in practice.

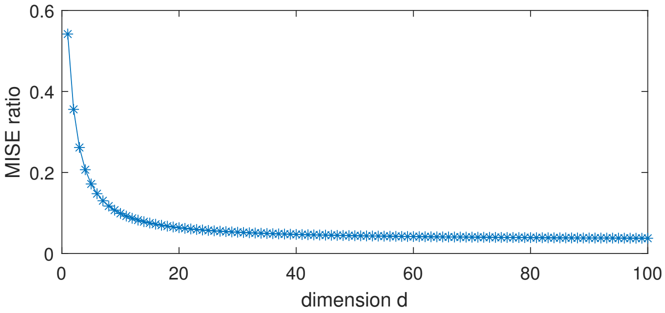


Figure 4.16: Ratio (4.76) between the Box Regularised Particle Filter's theoretical asymptotic Mean Integrated Square Error (MISE) and the original Box Particle Filter lower reachable MISE

Remark 2

The computational load (see Appendix C) required to perform the regularisation is:

$$c_{reg} = 4Nd^2 + 2Ndc_{random} + o(Nd) \tag{4.78}$$

where d is the state dimension, N the number of box particles, and c_{random} the computational load required to produce one random sample.

4.4.3 Numerical results

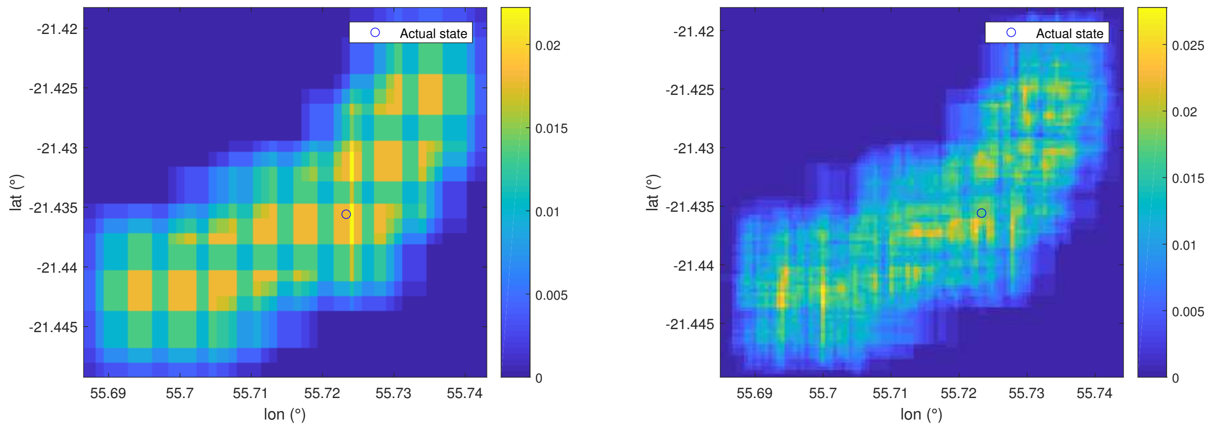
This section evaluates the following points:

1. The impact of regularisation on the filter's performances, in the context of Guaranteed Resampling and the Geometrical Subdivision approach;
2. The impact of regularisation on the filter's robustness.

The considered criteria, as introduced in Section 3.5 are: RMSE ratio (final RMSE divided by initial RMSE), pessimism rate (final estimated standard deviation divided by final RMSE), and non-convergence rate (the percentage of simulation ending with an inconsistent estimation). Points 1 and 2 are evaluated on all scenarios using Guaranteed Resampling and Geometrical Subdivision. Results for the first three scenarios (Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion) are discussed separately from those of the last two scenarios (Scenario TAN Moon flyby, Scenario TAN Sinusoidal terrain). Indeed, their measurement ambiguities are stronger, leading to a different behaviour. Unless a different setting is explicitly stated, the BRPF algorithm is tuned with $N = 900$ box particles, no process noise and a resampling threshold of $\theta_{eff} = 0.7$.

Kernel Regularisation aims to smooth the estimated conditional density in an optimal way, as developed in the previous section and as illustrated in Figure 4.15. The smoothing effect of Kernel

Regularisation can be checked in simulation. Figure 4.17 shows the conditional density estimated by BRPF for one run on Scenario TAN La Reunion, (a) before regularisation and (b) afterward. Box kernels are taken as uniform densities (i.e. $\pi_k^i = \frac{1}{|[\mathbf{x}_k^i]|}\mathbf{1}_{[\mathbf{x}_k^i]}$ in (4.5)). The regularisation kernel is an Epanechnikov density, as described in (2.32). The conditional density after regularisation clearly appears smoothed, which refines the estimation of its highest peaks corresponding to the most probable states, including the actual state (blue circle). In order to gain a two dimensional illustration, the plotted density corresponds to the marginalised density on latitude and longitude dimensions.



(a) Before regularisation

(b) After regularisation

Figure 4.17: Estimated conditional state density before and after one regularisation step (with $\mu = 0.1$) for Scenario TAN La Reunion. The regularisation step refines the resolution of the density estimation.

The RMSE ratios (the ratio between the final value of RMSE and the initial value) are displayed in Figure 4.19 and Table 4.11 for Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion; using Guaranteed Resampling and Geometrical Subdivision, for a hundred runs. A somewhat linear decreasing progression with respect to the regularisation magnitude μ is obtained for the averaged position RMSE. Velocity averaged RMSE is improved for $\mu = 0.1$, but may suffer some discrepancy for higher values, while the position remains improved. This highlights the importance of parameter μ whose values must be kept around 0.1 in practice to avoid instabilities for the TAN application. The reason why the actual optimal kernel bandwidth (2.33) must be lowered by μ is that the MISE minimisation is performed under a unimodal asymptotic density hypothesis. In practice, regularisation is performed when the density contains a number of modes (peaks), whose individual standard deviation in the TAN context usually has an order of magnitude of $\frac{1}{10}$ of the whole box particle cloud standard deviation. Giving higher values to μ may result in non-convergence in the presence of multimodalities (i.e. ambiguities). Figure 4.20 illustrates this tendency for $\mu \geq 0.3$. A significant improvement can be observed in the final position RMSE between $\mu = 0$ and $\mu = 0.2$. The velocity RMSE is also slightly improved until $\mu = 0.1$ It then suffers some discrepancy for $\mu > 0.1$. Higher values of μ lead to non-convergences. A possible approach to make regularised particle filters more robust to multimodalities is to perform some independent regularisation on each mode (Murangira [Mur+11]). However, such methods are costly and are beyond the scope of this thesis. Note that regularisation does not significantly reduce the conservatism (the pessimism ratios remain higher than 1.5).

Figure 4.18 shows the RMSE obtained in position for $\mu = 0$ to $\mu = 0.2$ along the whole trajectory on Scenario TAN Alps. Again, although regularisation helps to reduce the final error, it may introduce some instabilities if μ is too large, which can be the cause of a greater RMSE variability in the presence

Table 4.11: Simulation results for different values of the regularisation parameter. Results are averaged on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion, at final time-step

Regularisation coefficient μ	0	0.1	0.2	0.3	0.4
Non-convergence rate (%)	0	0	0	16.3	33.8
RMSE ratio (position)	0.097	0.094	0.090	1.79	3.32
RMSE ratio (velocity)	0.620	0.614	0.630	3.14	14.8
Pessimism (position)	2.08	2.12	2.07	1.32	0.54
Pessimism (velocity)	2.37	2.39	2.32	2.00	2.60
Computation time[1] (ms)	12	12	12	12	12

[1] For $\mu = 0$, the regularisation computation is not performed. The computation time difference between $\mu = 0$ and $\mu > 0$ appears negligible.

of multimodalities. In the present case, this phenomenon arises for $\mu = 0.2$, although it does not yield non-convergence until $\mu \approx 0.3$ (Figure 4.20).

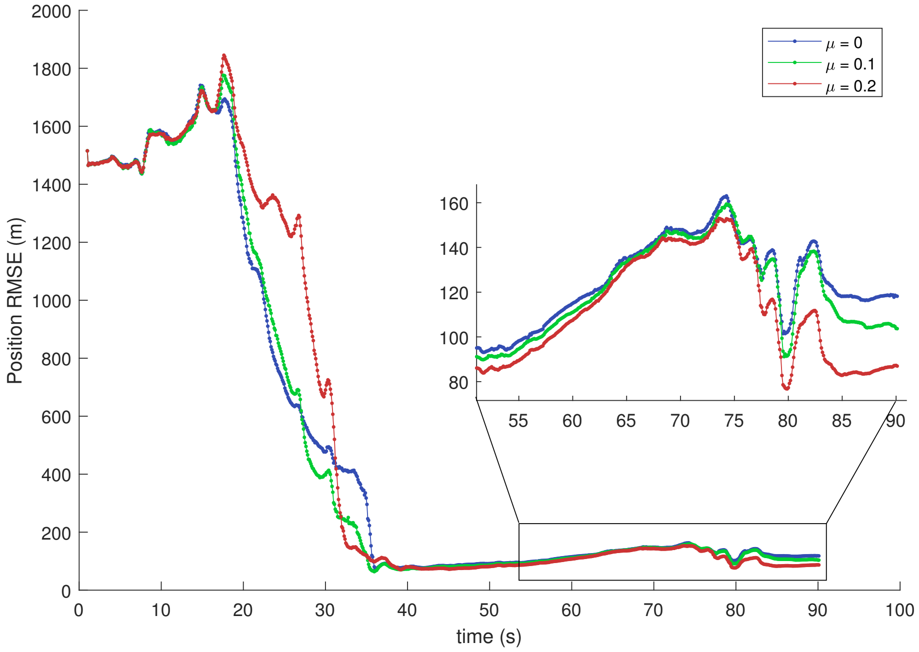


Figure 4.18: Position RMSE obtained for several values of μ on Scenario TAN Alps. The accuracy enhancement clearly appears during the last part of the trajectory (zoomed plot).

However, for the most severe ambiguity cases (as encountered in Scenario TAN Moon flyby and Scenario TAN Sinusoidal terrain) even low values of μ may yield non-convergences. Table 4.12 presents the averaged results of these two scenarios for $\mu \in \{0, 0.1\}$, for a hundred runs. These instabilities can be explained by the fact that in both cases, the posterior box kernels mixture density is not unimodal. Therefore, the regularisation hypotheses are not met, which makes the method unsuitable for such cases. To conclude, regularisation can improve the RMSE but needs to be carefully tuned, and should be avoided in the presence of strongly multimodal state densities.

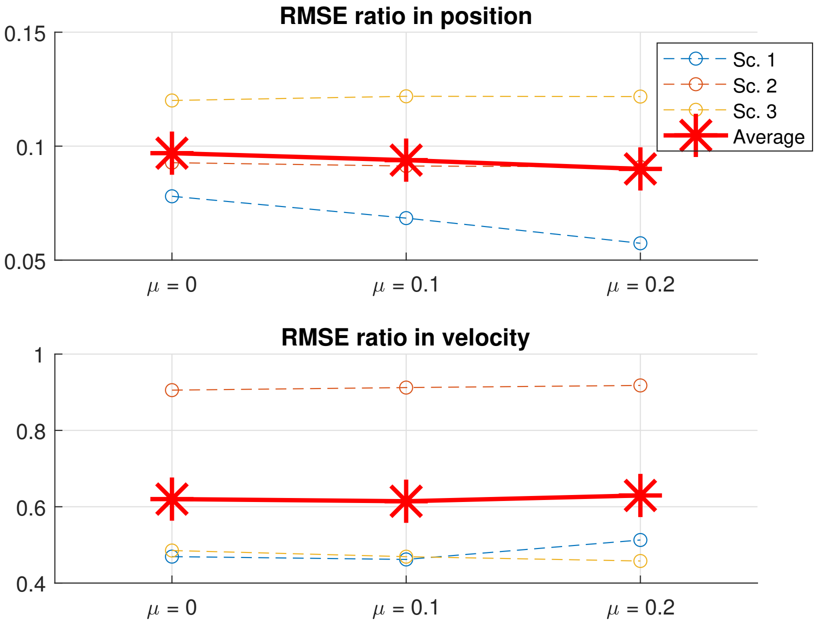


Figure 4.19: [
Root Mean Square Error ratios obtained for several values of the regularisation parameter μ (using Guaranteed Resampling and Geometrical Subdivision)]Root Mean Square Error ratios obtained for several values of the regularisation parameter μ (using Guaranteed Resampling and Geometrical Subdivision), on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion.

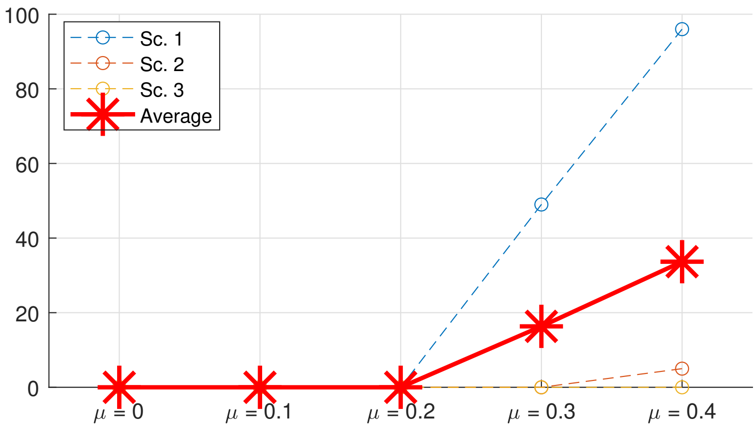


Figure 4.20: Non-convergence percentage obtained for several values of the regularisation parameter μ (using Guaranteed Resampling and Geometrical Subdivision) on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion.

Table 4.12: Averaged results for the impact of regularisation on the most ambiguous scenarios (Scenario TAN Moon flyby, Scenario TAN Sinusoidal terrain) at final time-step

Regularisation coefficient μ	0	0.1
Non-convergence rate (%)	0	40
RMSE ratio (position)	0.077	0.569
RMSE ratio (velocity)	0.582	1.91

4.5 OVERVIEW OF THE BOX REGULARISED PARTICLE FILTER

This section summarises the chapter's contributions to the Box Particle Filter field. The above developments are summarised in Algorithm 7, which is then called the Box Regularised Particle Filter. In order to provide an overview of the BRPF's performances with respect to existing approaches, simulations are run to evaluate the following points:

1. The relative performances of BPF [Gni+13] and the different configurations of BRPF with respect to the number of box particles;
2. The required computational loads for PF [GSS93], BPF [Gni+13] and the different configurations of BRPF for a given number of (box) particles;
3. The computational load difference between BRPF and conventional Particle Filter for a similar performance budget;
4. The impact of a non-symmetric unknown-but-bounded measurement density on BRPF.

Unless a different setting is explicitly stated, the BRPF algorithm is tuned with $N = 900$ box particles, no process noise and a resampling threshold of $\theta_{eff} = 0.7$.

Relative performances of BPF [Gni+13] and BRPF (point 1)

Figure 4.21 presents the RMSE ratios obtained for several values of the box particles number N, averaged on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion. A comparison is made between the original BPF (Gning, [Gni+13]) in blue and four configurations of BRPF: The Maximum Likelihood Covariance Subdivision (MLCS) with Multinomial Resampling (MR) in orange, the Geometrical Subdivision (GS) with MR in yellow, the MLCS using Guaranteed Resampling (GR) in purple, and the GS using RS in green. The regularisation is set up with $\mu = 0.1$ (2.33). The use of Guaranteed Resampling with both proposed subdivision methods (MLCS, GS) yields a better accuracy than the use of Multinomial Resampling. In particular, original BPF is outperformed in both position and velocity estimation, even for low values of N (e.g. $N = 100$). The configuration of BRPF involving Guaranteed Resampling and Geometrical Subdivision appears to be the most accurate for the TAN application. The obtained RMSE lowers as N increases. It can be noticed that the use of BRPF may help to significantly reduce the number of box particles with respect to the BPF for equivalent accuracy. For example, BRPF using GR and GS needs only 400 box particles to obtain similar RMSE in position as BPF using 1600 box particles. According to Table 4.13, the theoretical computational load can then be reduced by 75%.

By considering both performance budget and computational load in the context of the TAN application, it appears that the most advantageous algorithm is the BRPF using Geometrical Subdivision and Guaranteed Resampling.

Computational load and complexity analysis (point 2)

Table 4.13 provides the orders of magnitude of the computational load for the conventional Particles filter (SIR-PF), the conventional BPF and two configurations of BRPF. See Section 4.3.1 for Geometrical Subdivision and see Section 4.3.2 for Maximum Likelihood Covariance Subdivision. Note that the computational difference between Multinomial resampling and Guaranteed Resampling (see Section 4.2) is negligible with respect to other terms. Regularisation is enabled and set up with $\mu = 1$ (see Section 4.4). Although EKF cannot robustly handle severe ambiguities and non-linearities such as those encountered in the TAN application, its computational load is provided to give a comparison.

Theoretical computational loads are given in terms of the total number of floating-point operations (flops, see Appendix C) for prediction, correction, resampling, estimate, regularisation and the total load, obtained as the summation of the loads of all steps. The assumptions are made that the observation equation load is lower than the squared dimension, i.e. $c_h < d^2$, as well as the random sample cost $c_{random} < d^2$. It can be noticed that the total load is usually driven by the prediction step, which is the most costly. The order of magnitude of the computation time per time-step is globally the same for the Particle Filter, BPF and BRPF using Geometrical Resampling (about 10 ms). An exception arises for BRPF using Maximum Likelihood Covariance Subdivision, whose load is significantly higher than the prediction's (see (4.65) for a detailed analysis). Numerical evaluations of the flops' orders of magnitude are provided, assuming that particle filters use $N = 10^3$ particles (or box particles), the state dimension is $d = 6$ and the measurements dimension is lower than the state dimension $d_m < d$ (for TAN, $d_m = 1$).

Average empirical computation time per time-step is provided using a 1.6 GHz CPU running Matlab® for the TAN application (non-linear measurement, see (3.9)) and double integrator linear dynamics (see (3.1)). Computation times are averaged for time-steps when a resampling occurs, time-steps when only prediction and correction step occur and for every time-step. Empirically, a resampling occurs on average 2 to 5 % of the time, which makes its impact on the total computation load often insignificant. Indeed, the averaged computation time for all time-steps remains of the order of magnitude of the prediction and correction steps' computation times. The BRPF using MLCS averagely takes about 40 ms per time-step, which represents a majority of 10 ms steps and a few 1 s steps when a resampling occurs.

The empirical computation times are globally consistent with the theoretical computation loads in terms of the relative order of magnitude. However, empirical values must be carefully interpreted and may be biased by some intern parallelism of Matlab®. Also note that those times can be optimised by taking advantage of the high level of parallelism of particle filtering, using for example GPU (Graphics Processing Unit) or FPGA (Field Programmable Gate Array) chips, which is beyond the scope of this thesis.

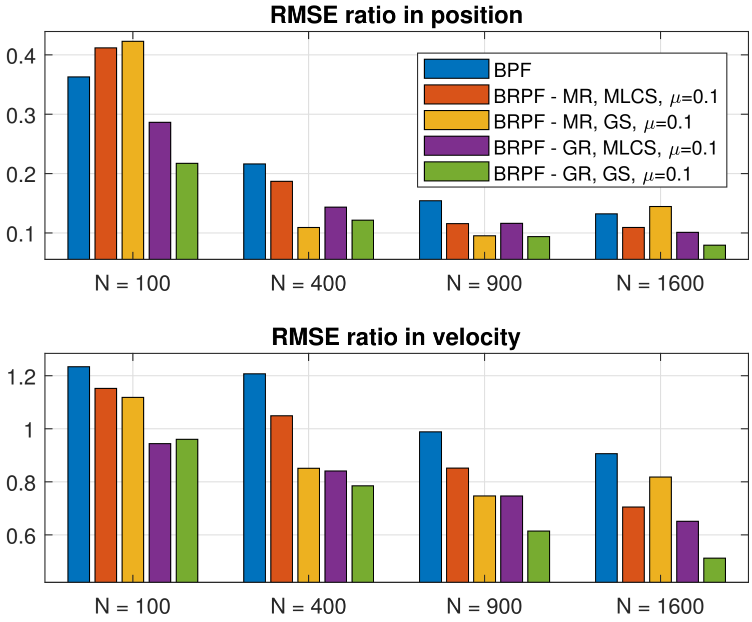


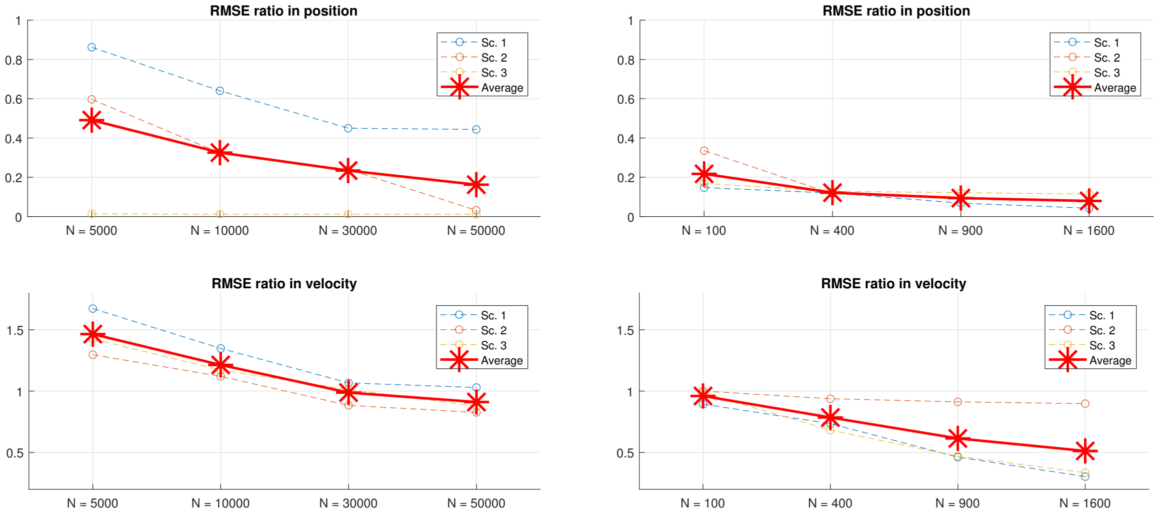
Figure 4.21: Root Mean Square Error ratios obtained for several values of the box particles number N, for the original BPF (Gning, [Gni+13]) in blue, and four configurations of BRPF: the Maximum Likelihood Covariance Subdivision (MLCS) with Multinomial Resampling (MR) in orange, the Geometrical Subdivision (GS) with MR in yellow, the MLCS using Guaranteed Resampling (GR) in purple, and the GS using GR in green. The last configuration appears to be the most accurate for the TAN application, and the resulting RMSE gets lower as N increases. RMSE ratios are averaged on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion.

Table 4.13: Computational load (flops) needed by BRPF, compared to previous works. The BRPF is declined in two configurations: using Geometrical Subdivision (GS, see Section 4.3.1), and Maximum Likelihood Covariance Subdivision (MLCS, see Section 4.3.2). A numerical example is provided for $N = 10^3$ particles, $d = 6$, $d_m < d$, $c_h < d^2$, $c_{random} < d^2$. The empirical computation time is obtained with a 1.6 GHz CPU running Matlab®. The average computation time is provided for one time-step where a resampling occurs, and for one time-step where only prediction and correction steps occur. The average of all time-steps without discrimination is also provided.

Algorithm	EKF	SIR-PF	BPF	BRPF	
	Verhaegen	Karlsson	Gning	GS	MLCS
	[VVD86]	[KSG05]	[AGB07]	Section 4.3.1	Section 4.3.2
Prediction	$\mathcal{O}(d^3)$	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^2)$
Correction	$\mathcal{O}(d^2 d_m)$	$\mathcal{O}(Nc_h)$	$\mathcal{O}(Nc_h)$	$\mathcal{O}(Nc_h)$	$\mathcal{O}(Nc_h)$
Resampling	-	$\mathcal{O}(N)$	$\mathcal{O}(Nc_{random})$	$\mathcal{O}(Nd)$	$\mathcal{O}(Nd^6)$
Regularisation	-	-	-	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^2)$
Estimate	$\mathcal{O}(d^3)$	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^2)$
Total	$\mathcal{O}(d^3)$	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd^6)$
e.g.	$\mathcal{O}(10^2)$	$\mathcal{O}(10^4)$	$\mathcal{O}(10^4)$	$\mathcal{O}(10^4)$	$\mathcal{O}(10^6)$
Empirical with resampling (ms)	-	11	18	30	10^3
Emp. without resampling (ms)	4	7	9	9	9
Empirical average (ms)	4	10	11	12	40

Relative computation load of SIR-PF [GSS93] and BRPF for similar performances (point 3)

Simulations have been run to test the conventional Particle Filter (SIR-PF) performances on the same scenarios as the BRPF, for different values of the number of particles and box particles N on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion. The SIR-PF has been set-up with a truncated Gaussian measurement noise assumption, which corresponds to the actual measurement noise used for the simulation. Note that the RMSE have been computed keeping all the Monte Carlo runs, including the non-convergent ones. Table 4.14 and Figure 4.22 present the averaged performances obtained with the SIR-PF and the BRPF, for a hundred runs. The SIR-PF yields approximatively the same averaged RMSE ratios with 30,000 particles than the BRPF with 100 box particles. Therefore, there is a ratio of about $\mathcal{O}(10^2)$ particles between SIR-PF and BRPF, which corresponds to a 99% of computational load reduction for an equal state dimension, according to Table 4.13. The empirical computation time per time-step is 74 ms for the SIR-PF with $30,000$ particles and 2.5 ms for the BRPF with 100 box particles. This represents a computation time gain of 97%, which tends to confirm the theoretical gain prediction. Note that the BRPF also brings more robustness with a zero non-convergence rate.



(a) SIR-PF (b) BRPF

Figure 4.22: Root Mean Square Error ratios obtained for several values of the (box) particles number N, for the SIR-PF (a), and BRPF (Geometrical Subdivision using Guaranteed Resampling) in (b). A similar RMSE ratio can be reached for SIR-PF and BRPF for respectively $50,000$ particles and 400 box particles, which represents approximately a ratio of 10^2 particles. RMSE ratios are averaged on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion.

Table 4.14: Averaged results for SIR-PF and BRPF on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion, at final time-step.

	SIR-PF				BRPF			
N	5,000	10,000	30,000	50,000	100	400	900	1,600
Non-convergence rate (%)	67	77	12	15	0	0	0	0
RMSE ratio (position)	0.49	0.33	0.23	0.16	0.21	0.12	0.09	0.08
RMSE ratio (velocity)	1.46	1.21	0.98	0.91	0.96	0.78	0.61	0.51
Emp. computation time (ms)	14	27	74	127	2.5	6.1	12	19

Impact of the measurement density on BRPF (point 4)

Several bounded measurement densities were designed in order to check point 4:

- Truncated Gaussian density: This density was used in all simulations until then. It is defined by (3.10).
- Truncated Rayleigh density: In this case, this density represents a noise whose expectation is biased from -10 m with respect to an unbiased observation model expected value ($\mathbf{m}_k = h(\mathbf{x}_k)$). Furthermore, the probability of $\mathbf{m}_k \geq h(\mathbf{x}_k)$ is zero. The truncated Rayleigh density used in the simulations is defined by:

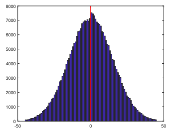
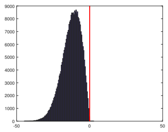
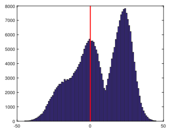
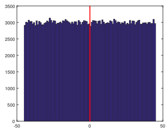
$$\begin{cases} p(\mathbf{m}_k|\mathbf{x}_k) = 0 & \text{if } \mathbf{m}_k - h(\mathbf{x}_k) \in [0, \infty[\\ p(\mathbf{m}_k|\mathbf{x}_k) \propto -(\mathbf{m}_k - h(\mathbf{x}_k))\mathbf{B}^{-1} \exp\left(-\frac{1}{2}(\mathbf{m}_k - h(\mathbf{x}_k))^T \mathbf{B}^{-1}(\mathbf{m}_k - h(\mathbf{x}_k))\right) \mathbf{1}_{[\mathbf{v}_k]} & \text{else} \end{cases} \tag{4.79}$$

  where $\mathbf{B} = 10^2$ m^2 is the parameter corresponding both to a variance and a bias. Although BRPF must handle any density provided that its support is bounded by $[\mathbf{v}_k]$, a higher BRPF estimation error can be expected due to the measurement bias.

- Multimodal density: This density is obtained by the addition of three Gaussian densities corresponding to the normal laws $\mathcal{N}\left(20, \left(\frac{15}{2}\right)^2\right)$, $\mathcal{N}(5, 5^2)$, $\mathcal{N}\left(-20, \left(\frac{15}{2}\right)^2\right)$, and one Rayleigh density (4.79) of parameter $\mathbf{B} = 10^2$ m^2. This density represents a bimodal measurement noise. A higher BRPF estimation error may be expected due to the measurement modes.
- Uniform density: This density corresponds to the BRPF measurement noise assumption (see Section 4.1, equation (4.18)). Therefore, a lower BRPF estimation error can be expected in the context of an actual uniform measurement density than for other densities.

Table 4.15 and Figure 4.23 present the obtained results for a hundred runs. The BRPF appears robust to unknown-but-bounded measurement densities, since all four bounded densities yield a zero non-convergence rate. The RMSE ratios are not impacted in a significant way. One can notice a slightly downgraded RMSE in the presence of systematically biased measurements (Rayleigh density). When the actual measurement density matches with the BRPF assumption (i.e. uniform density), the RMSE is slightly better than with other densities.

Table 4.15: Averaged results for different measurement densities on all scenarios, at final time-step. An empirical histogram of each density is provided (deep blue bars). The red line indicates the abscissa zero, which corresponds to a perfect measurement $\mathbf{m}_k = h(\mathbf{x}_k)$.

Actual measurement density	Gaussian	Rayleigh	Multimodal	Uniform
				
BRPF assumed measurement density	Uniform			
Non-convergence rate (%)	0	0	0	0
RMSE ratio (position)	0.09	0.11	0.09	0.08
RMSE ratio (velocity)	0.61	0.61	0.58	0.53
Pessimism (position)	1.79	1.86	1.77	1.69
Pessimism (velocity)	2.26	2.26	2.25	2.25

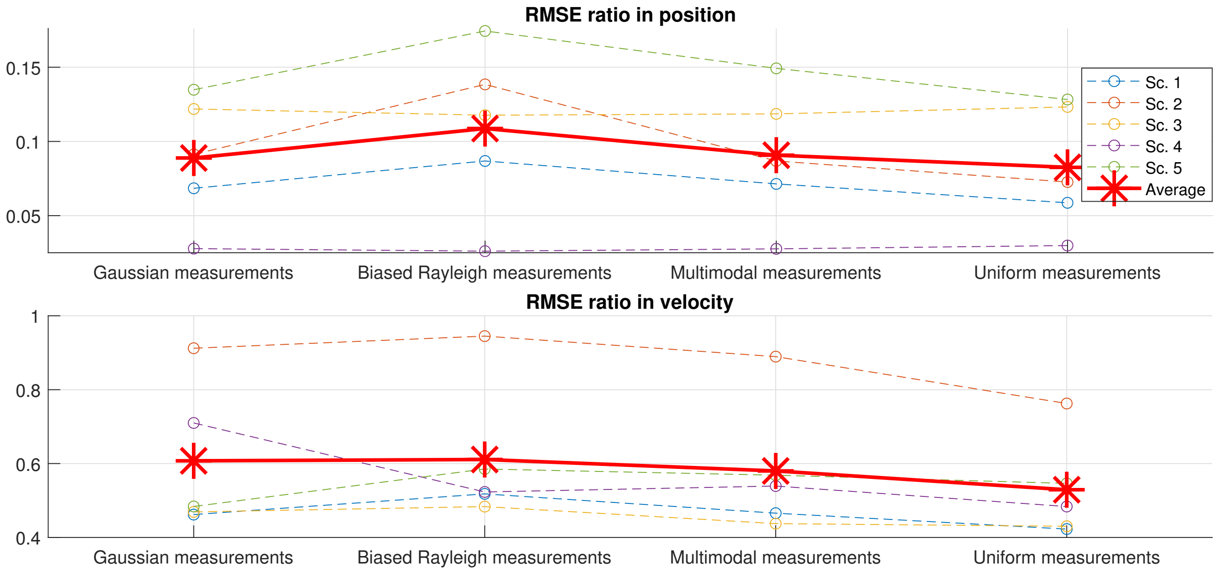


Figure 4.23: Root Mean Square Error ratios obtained for different measurement densities (using Guaranteed Resampling and Geometrical Subdivision), on all scenarios. BRPF appears to be robust to biased and multimodal measurement densities.

Algorithm 7 Box Regularised Particle Filter

```
*Inputs*: initial state density $p(\mathbf{x}_0)$, process noise box $[\mathbf{w}_k]$ $\forall k$, measurement noise box $[\mathbf{v}_k]$ $\forall k$.
*Outputs*: state estimate $\widehat{\mathbf{x}}_k$ $\forall k$, estimated covariance $\widehat{\mathbf{P}}_k$ $\forall k$.
**Initialisation**: The initial box particle set (or box particle cloud) $\{[\mathbf{x}_0^i] \in \mathbb{IR}^d, w_0^i \in \mathbb{R}^{+*}\}_{i\in[1,N]}$ is drawn such that the initial estimated density (2.56) approaches the initial actual density $p(\mathbf{x}_0)$. In practice, it can be initialised with a regular paving on the state density's support, which makes it similar to an histogram.
**for** each time-step $k$ **do**
    **Prediction step**: Propagate box particles using the dynamics $[\mathbf{x}^i_{k|k-1}] = [f]\left([\mathbf{x}^i_{k-1}]\right) + [\mathbf{w}_k]$
    **Correction step**: Contract each particle with respect to the measurement box $[\mathbf{m}_k]$ (see contractors in Section 2.1.5), such that: $[\mathbf{x}_k^i] = \left[\left\{\mathbf{x}_k \in [\mathbf{x}^i_{k|k-1}] \mid h(\mathbf{x}_k) \in [\mathbf{m}_k]\right\}\right]$
    Update weights with the consistency term (4.16). Under the assumption of uniform box kernels and uniform measurement density, the consistency term is the ratio between the volumes of the box after and before contraction (2.60).
    Compute the state estimate $\mathbb{E}\left[\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)\right]$. It can be approximated by a least-square estimator: $\widehat{\mathbf{x}}_k = \sum_i w_k^i \mathbf{c}_k^i$ with $\mathbf{c}_k^i$ the center of each particle (see Table 2.1).
    Compute a confidence indicator, e.g. the covariance $\widehat{\mathbf{P}}_k = \sum_i w_k^i(\mathbf{c}_k^i - \widehat{\mathbf{x}}_k)(\mathbf{c}_k^i - \widehat{\mathbf{x}}_k)^T$.
    **if** a resampling criterion is satisfied, e.g. $N_{eff} < \theta_{eff} N$, see (2.22) **then**
        Draw a new set of box particles $\{[\mathbf{x}_k^i], w_k^i\}_{i\in[1,N]}$ using a resampling method, e.g. Multinomial Resampling (Algorithm 3), or Guaranteed Resampling (Algorithm 6) to determine the number of new instances per box particles $n^i$.
        Replace low-weighted (or zero-weighted) box particles with $n^i$ subdivisions of high-weighted particles using Geometrical Subdivision (4.43), or Maximum Likelihood Covariance Subdivision, (4.48) (4.47).
        Reset all weights to $1/N$.
        Perform a regularisation on box particles' centers and diameters by generating samples via the Epanechnikov Kernel (2.32).
    **end if**
**end for**
Return $\widehat{\mathbf{x}}_k, \widehat{\mathbf{P}}_k$ $\forall k$.
```

4.6 SUMMARY

This chapter showed that the BPF can be formalised for any box kernel mixture (research question 1). It yields less pessimism, as shown in the linear-Gaussian case. However, it becomes rapidly intractable in the presence of non-linearities and is therefore not of practical use to tackle ambiguous measurements.

A guaranteed formulation of Multinomial Resampling was then introduced which ensures that at least one box particle contains the actual state (research question 2). This proves efficient in practice and yields a lower RMSE while ensuring a zero non-convergence rate. However, it brings some conservatism (higher pessimism rate).

The impact of two new subdivision resampling approaches was also studied (research question 3): the Geometrical Subdivision (GS) based on a sub-box normalisation and the Maximum Likelihood Covariance Subdivision (MLCS) based on observability. Although GS is less general than MLCS, it yields a significantly lower computational load and a lower RMSE when used in conjunction with Guaranteed Resampling. This tends to show that accounting for observability is not necessary as long as the filter's robustness is ensured by construction.

The kernel regularisation concept was introduced into the BPF scheme, leading to the Box Regularised Particle Filter (BRPF, research question 4). It was proved that the BRPF's MISE is necessarily lower than the original BPF's. In practice, it translates into a lower RMSE and does not have a significant impact on the total computational load. However, regularisation must be tuned carefully in the presence of multimodalities and may introduce instability.

Finally, the BRPF was compared to previous approaches. BRPF can reduce the computational load by 75% (4-fold reduction) compared to BPF and by 97% (33-fold) compared to the SIR-P, for a similar state-estimation accuracy. A theoretical complexity analysis of the algorithms is provided. The BRPF was also run with different actual measurement noise, while the expected measurement distribution remains uniform. The RMSE and pessimism rates are not significantly impacted by multimodal measurement noise or biased noise, which confirms that BRPF can tackle *unknown-but-bounded* measurements.

5

BOX REGULARISED PARTICLE FILTER IN FEDERATED ARCHITECTURES

Chapter 4 introduced a new approach to the Box Particle Filter called the Box Regularised Particle Filter (BRPF). BRPF is able to perform state estimation with high robustness to severely ambiguous and non-linear measurements. The state is estimated via a cloud of weighted box particles. In probabilistic terms, the estimated state density is approximated by a weighted mixture of bounded kernels whose supports are the box particles. The measurement is assumed to be included in a box, and box particles are contracted to keep only a subset of them which is consistent with the measurement set. For every box, the weights quantify the probability to contain the actual state given the measurements. In order to regroup box particles in the state space regions that are the most likely to contain the actual state, a resampling step can be triggered. It consists of replacing low-weighted box particles with subdivisions of high-weighted ones. The total number of box particles is kept.

By combining Interval Analysis and the probabilistic framework (see Chapter 2), BRPF can robustly handle state density multimodalities (see Appendix B). In particular, the use of Guaranteed Resampling introduced in Chapter 4, Section 4.2 ensures that the actual state always belongs to at least one box particle. This ensures a zero non-convergence rate (see Section 3.5). As a result, BRPF is a suitable solution for severely ambiguous estimation problems, such as Terrain Aided Navigation (TAN). BRPF achieves a better accuracy than the original BPF (Gning, [Gni+13]) and the conventional Particle Filter (Gordon, [GSS93]); and requires a significantly lower computation load for similar performances.

However, for other types of measurements (e.g. GNSS or magnetometer), the BRPF's estimate may be sub-optimal with respect to the Optimal Filter MAP estimator (see Section 2.1.2):

- Taking into account specific noise density's hypotheses is not straightforward in BRPF. Chapter 4, Section 4.1 introduced a general formulation of BPF which theoretically allows accounting for any probabilistic knowledge from the measurements. However, this formulation can be rapidly intractable in practice, e.g. for non-linear measurements, which limits its use. As a consequence, the practical formulation of BRPF assumes that the measurement's density is a uniform kernel. This is a suitable approximation for unknown-but-bounded densities, but is sub-optimal for example for actually Gaussian densities.

- The dynamical propagation of box particles is based on inclusion functions (see Chapter 2, Section 2.1.5). Dynamical inclusion propagation returns a propagated box containing the output set corresponding to the propagation of the prior box. However, correlation between state variables is lost at the box particle level, since the box's edges are aligned with the state space basis. Therefore, during the contraction step, BRPF cannot explicitly establish any link between explicitly measured state variables and indirectly observed ones, as does a Kalman gain, by accounting for non-diagonal coefficients of the predicted covariance matrix (see Chapter 2, equation (2.13)). In other words, BRPF (and BPF in general) can only contract each box particle along the directly measured dimensions (explicit inputs of function h), and not along all the observed dimensions, which depend on h and the dynamics f, (see for example Robenack [Röb03]). In the TAN application for example, only the position variables are explicitly involved in the observation model. Therefore, velocities and possibly other variables such as attitude angles, are estimated in a sub-optimal way.

Section 2.2 examined the literature about collaborative state estimation architectures, in particular federated architecture, where measurements are processed in a first layer by several dedicated local filters whose local estimates are fused by a Master Filter in a second layer. A possible approach to tackle multi-sensor fusion in the presence of ambiguities would be to integrate BRPF to such an architecture. Therefore, severely ambiguous measurements would be tackled by a dedicated BRPF, while other measurements would be processed by other local filters (e.g. Kalman Filters). The local estimates would then be fused in a Master Filter (for example a collaborative Kalman Filter [OS09] or a collaborative Information Filter [Man93]).

This chapter is organized as follows: Section 5.1 discusses the conditions for the BRPF density output to be unimodal. Section 5.2 studies the integration of BRPF in a centralised federated architecture. Numerical results illustrate the practical interest of such an architecture. Section 5.3 applies this scheme to a distributed architecture. Simulations illustrate the interest of this approach, in terms of RMSE and covariance stability, in the context of collaborative navigation in a fleet of vehicles.

5.1 ASYMPTOTIC UNIMODALITY OF THE BOX REGULARISED PARTICLE FILTER

In order to implement a BRPF as a component of a Bayesian federated architecture (using for example a Kalman Filter as a master filter), a unimodal approximation is needed for the BRPF posterior state density. It was shown in Chapter 4 that the state conditional density estimated by the BRPF may be multimodal, i.e. may have several maxima. It clearly appears in Figure 4.12 where the particle cloud is made of the union of two disjoint box particle clusters of non-zero weights. In what follows, a *cluster* will refer to the union of several box particles $\bigcup_{i \in I} [\mathbf{x}_k^i]$ where I is an indexing set of box particles belonging to the cluster (see Appendix D). A cluster can be reduced to a single box particle.

The box particle cloud represents the estimated state density support $\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)$ It can be deduced that the estimated density may have several local maxima, called modes. When these modes are relatively distant from one another as in Figure 4.12, the unimodal assumption is no longer valid.

The causes of multimodalities are discussed in Appendix B. Multimodalities in the posterior conditional state density can be caused by (a) a nonlinear observation equation h, (b) a multimodal measurement density $p(\mathbf{m}_k|\mathbf{x}_k)$, or (c) some non observed state variables. In the context of BRPF, case (b) has no impact on the box particle cloud connectedness since the measurement density support is assumed to be contained in the measurement box $[\mathbf{m}_k]$ (see Section 4.5). BRPF's robustness to this case was experimentally shown in Table 4.15. Case (c) does not strictly produce multiple local maxima, but may yield a density support discrepancy along unobserved dimensions. Provided that h is injective and continuous on the particle cloud domain and $p(\mathbf{m}_k|\mathbf{x}_k)$ is unimodal, the contracted box particle cloud remains connected. The focus will then be on non-linear and non-injective observations.

5.1.1 Asymptotic unimodality using Multinomial Resampling

Even if the estimated state density is multimodal, it is possible to make an asymptotic unimodality assumption. The BRPF (Algorithm 7) consists of iterating three main steps: a box prediction (4.9), a weight update (4.16) and, if needed, a resampling step by replacing low-weighted boxes with subdivisions of high-weighted boxes. The resampling step can be performed using several methods. In the literature, the most commonly used method is Multinomial Resampling, presented in Section 2.1.4, Algorithm 3. Chapter 4, Section 4.2 introduced a guaranteed version of Multinomial Resampling called Guaranteed Resampling (Algorithm 6).

In the context of the Particle Filter, King [KF00] showed that Multinomial Resampling yields to asymptomatic state density unimodality. In other words, even in the case of an actual multimodal state density, all samples of a finite particle cloud collapse in the neighbourhood of one of the modes that has a non-zero probability of being the wrong mode, which may lead to a non-convergence, as k tends to infinity. This asymptotic behaviour is illustrated in Figure 5.1. Since Multinomial Resampling only takes weights as inputs, these results are directly applicable to BPF and BRPF. Unfortunately, this is not the case for Guaranteed Resampling.

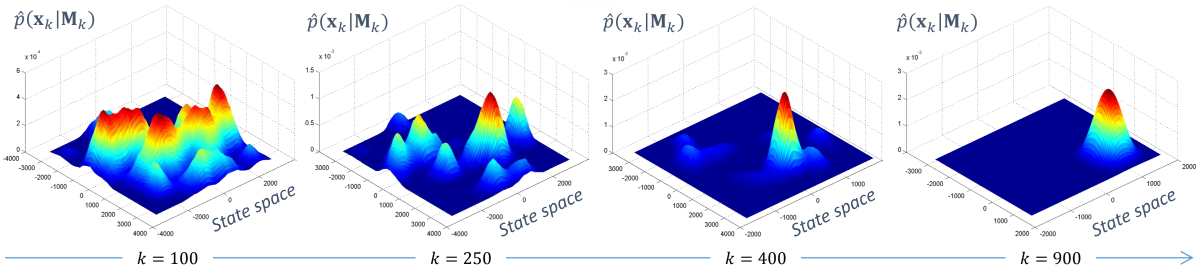


Figure 5.1: Posterior conditional state density obtained with a Box Regularised Particle Filter using Multinomial Resampling in the terrain measurements case (Scenario TAN Alps) for one run. The state density converges to a single mode (which may be the wrong one), as shown in King [KF00].

5.1.2 Asymptotic multimodality using Guaranteed Resampling

Guaranteed Resampling (Algorithm 6) ensures that all box particles $[\mathbf{x}_k^i]$ that have a non-zero probability to contain the actual state $\mathbf{x}_k$ are kept in the cloud. This yields more robustness, but also ensures that every box particle $[\mathbf{x}_k^i]$ containing at least one state consistent with the measurements box (i.e. $\exists \mathbf{x} \in [\mathbf{x}_k^i], h(\mathbf{x}) \in [\mathbf{m}_k]$) is guaranteed to be kept in the cloud.

The initial box particle cloud is assumed to form a connected set $\bigcup_{i=1}^{N} [\mathbf{x}_0^i]$ containing the actual state $\mathbf{x}_0$. This can be achieved in practice by initialising box particles as a regular paving containing the initial state density support $\{\mathbf{x} \in \mathbb{R}^d \mid p(\mathbf{x}) > 0\}$. However, in the event of multimodalities occurring in the posterior conditional state density $p(\mathbf{x}_k|\mathbf{M}_k)$ (in particular, the non monotony of h), the box particle cloud may not remain connected.

In order to provide an asymptotic unimodal approximation of the estimated state density $\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)$, the focus will first be on the existence of a unique connected cluster containing the actual state. Finally, the posterior density unimodality characteristics will be discussed.

5.1.3 Existence, connectedness, and uniqueness of the actual cluster

It is first shown that there is always a unique connected box particle cluster containing the actual state. Each box particle forms a connected set by construction. It is assumed that the dynamical model f is continuous.

Proposition 6. *Consider a BRPF (Algorithm 7) using Guaranteed Resampling (Algorithm 6) and no regularisation ($\mu = 0$ in (2.33)), initialised with a connected box particle cloud whose union contains the* *actual state. The dynamical model f and the observation model h are assumed to be continuous. Then, for every time-step k, there exists a unique larger connected cluster $\{\chi_k\} \subset \mathbb{R}^d$ containing the actual state $\mathbf{x}_k$, called the actual cluster.*

Proof. Define the prior *actual cluster* $\{\chi_{k-1}\} = \bigcup_{i \in I_{k-1}} [\mathbf{x}_{k-1}^i]$ where I_{k-1} is an indexing set. Box particles indexed by I_{k-1} are assumed to satisfy the following axioms. At least one of them contains the actual state $\mathbf{x}_{k-1}$, their union $\{\chi_{k-1}\}$ is connected and is disjoint from any box particle which does not belongs to it. In other words, $\{\chi_{k-1}\}$ is the largest connected cluster containing the actual state. It is assumed that the initial box particle cloud is connected and contains the actual state. Then, the indexing set $I_0 = [1, N]$ contains every box particle. The prediction step keeps the predicted actual cluster $\{\chi_{k|k-1}\}$ as a connected set, provided that the dynamics f is continuous (see Appendix D), and is defined by:

$$\{\chi_{k|k-1}\} \triangleq \bigcup_{i \in I_{k|k-1}} [\mathbf{x}_{k|k-1}^i] \tag{5.1}$$

where the indexing set is kept:

$$I_{k|k-1} = I_{k-1} \tag{5.2}$$

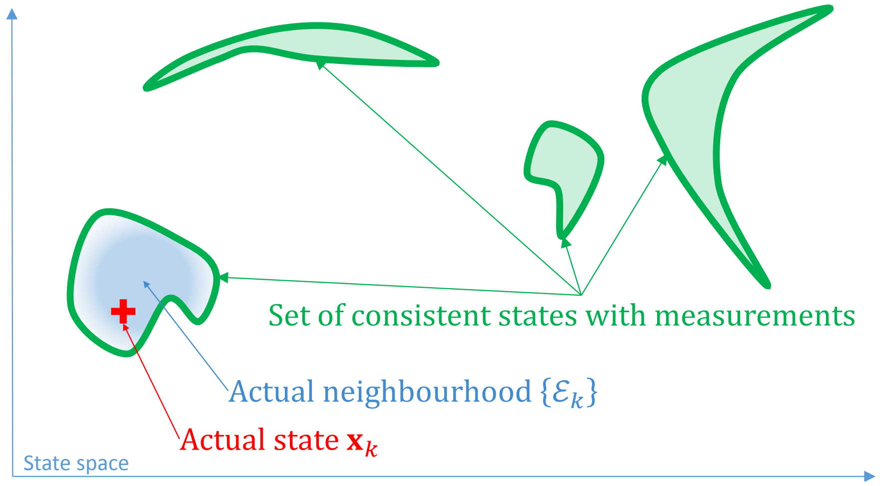
When a measurement $\mathbf{m}_k^i \in \mathbb{R}^{d_m}$ is available, each box particle is contracted with respect to the measurement box $[\mathbf{m}_k^i] \in \mathbb{IR}^{d_m}$ (see Section 4.1). Unfortunately, the resulting cluster $\bigcup_{i \in I_{k|k-1}} [\mathbf{x}_k^i]$ may result in a non-connected set after applying the contraction step to box particles of the predicted actual cluster $\{\chi_{k|k-1}\}$. Nevertheless, assume that there exists a neighbourhood $\mathcal{E}_k$ of the actual state which satisfies:

$$\begin{aligned} &\mathbf{x} \in \mathcal{E}_k \Rightarrow \mathbf{x} \in \{X_k\} \\ &\mathbf{x}_k \in \mathcal{E}_k \end{aligned} \tag{5.3}$$

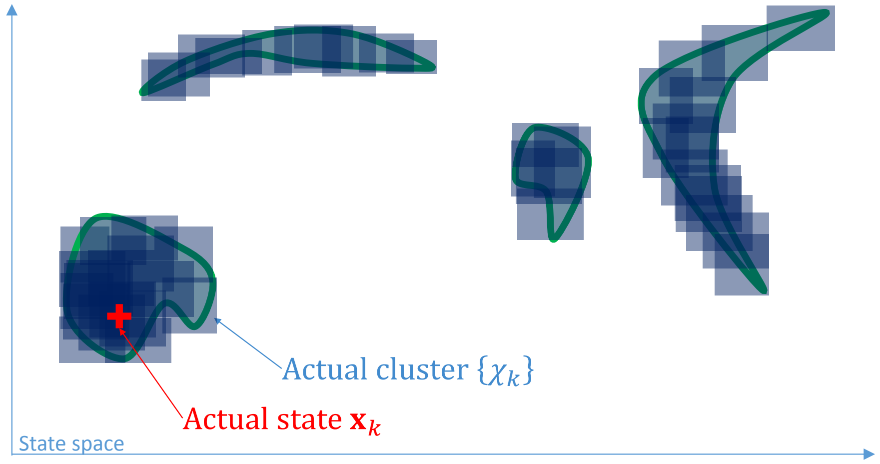
where $\{X_k\}$ is the theoretical set containing all states being consistent with the measurements since $k = 0$ (see Section 2.1.5, equation (2.36)). In what follows $\mathcal{E}_k$ will refer to the *actual neighbourhood*. This concept is illustrated in Figure 5.2 (a). The actual neighbourhood is an open subset of the consistent set $\{X_k\}$ containing all prior states being consistent with the measurements. Such a neighbourhood $\mathcal{E}_k$ exists if the observation model h is continuous and if the actual predicted measurement $h(\mathbf{x}_k)$ always belongs to the interior of the measurement box $[\mathbf{m}_k]$, i.e. it cannot belong to its boundary, see Appendix D. Note that these assumptions are also required for the proof of guarantee of Guaranteed Resampling (see Section 4.2.2). In what follows, it will be considered that these two conditions are satisfied.

Figure 5.2 (b) illustrates the resulting box particle cloud after at least one contraction step. The box particle cloud may not remain connected. It therefore results in the union of several disjoint clusters. Figure 5.3 illustrates the consistent set and the resulting empirical box particle clusters for a simulation on Scenario TAN Alps. The contraction step ensures that each box particle containing a consistent point $\mathbf{x} \in \mathbb{R}^d$ such that $h(\mathbf{x}) \in [\mathbf{m}_k]$ before contraction keeps this point after contraction. Formally, $[\mathbf{x}_k^i] \triangleq [\{\mathbf{x} \in [\mathbf{x}_{k|k-1}^i] \mid h(\mathbf{x}) \in [\mathbf{m}_k]\}]$ yields:

$$\forall \mathbf{x} \in [\mathbf{x}_{k|k-1}^i], h(\mathbf{x}) \in [\mathbf{m}_k] \Rightarrow \mathbf{x} \in [\mathbf{x}_k^i] \tag{5.4}$$

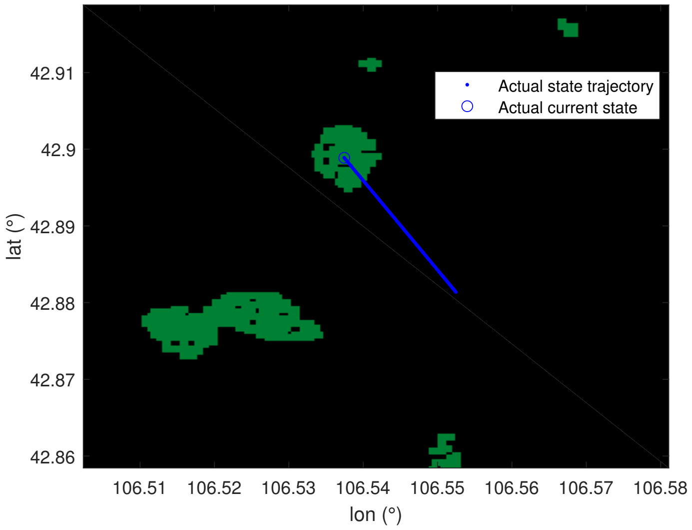


(a) Actual state, actual neighbourhood, and consistent set

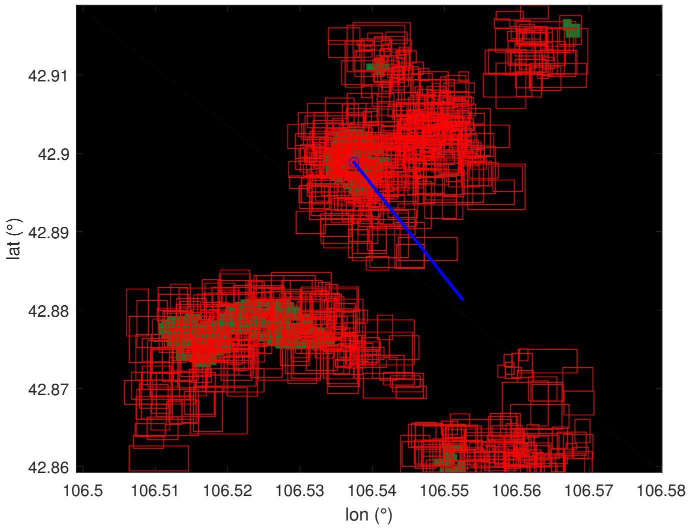


(b) Resulting box particle cloud and clusters

Figure 5.2: Illustration of (a) the actual state, the actual neighbourhood, and the consistent set (all states consistent with prior measurements). This configuration yields a box particle cloud made of several disjoint clusters (b). One of them necessarily contains the actual state and is called the actual cluster.



(a) Theoretical posterior consistent set (green areas) and actual trajectory



(b) Resulting box particle cloud and clusters

Figure 5.3: Simulation of the actual state (green line), and the consistent set (green areas) containing the actual neighbourhood, obtained for Scenario TAN Alps, at $t = 10$ s (a). This configuration yields a box particle cloud made of several disjoint clusters (red boxes) (b). One of them necessarily contains the actual state and is called the actual cluster.

In particular, every particle containing at least one element of the actual neighbourhood keeps it after contraction:

$$\forall \mathbf{x} \in [\mathbf{x}^i_{k|k-1}], \mathbf{x} \in \mathcal{E}_k \Rightarrow \mathbf{x} \in [\mathbf{x}^i_k] \tag{5.5}$$

Then, every pair of box particles belonging to the connected cluster $\{\chi_{k|k-1}\}$ and having a non-empty intersection which contains at least one element of the actual neighbourhood, has a non-empty intersection after contraction and then forms a connected union:

$$\begin{cases} \mathbf{x} \in [\mathbf{x}^i_{k|k-1}] \cap [\mathbf{x}^j_{k|k-1}], \forall i,j \in I_{k|k-1} \\ \mathbf{x} \in \mathcal{E}_k \end{cases} \Rightarrow \mathbf{x} \in [\mathbf{x}^i_k] \cap [\mathbf{x}^j_k], \forall i,j \in I_{k|k-1} \tag{5.6}$$

Then, an updated cluster can be derived as a subset of $\{\chi_{k|k-1}\}$ defined by: $\{\chi_k\} \triangleq \bigcup_{i \in I_k} [\mathbf{x}^i_k]$ with I_k the new indexing set of all box particles satisfying that their union is connected, and one of them contains the actual state $\mathbf{x}_k$. This results in the following indexing set:

$$I_k = \left\{ i \in I_{k|k-1} \mid [\mathbf{x}^i_k] \cap \mathcal{E}_k \neq \emptyset \right\} \tag{5.7}$$

In particular, the contraction step ensures that if a box particle $[\mathbf{x}^i_{k|k-1}]$ contains the actual state $\mathbf{x}_k$ before the contraction step, it still contains it afterwards, as shown by (4.33) in Section 4.2. Then, cluster $\{\chi_k\}$ contains at least the box particle that contains the actual state. Cluster $\{\chi_k\}$ also contains all the pairs of box particles whose intersection contains at least one element of $\mathcal{E}_k$. To conclude, the contraction step yields the largest connected subset of the predicted actual cluster that contains the actual state. This new cluster can be defined as an updated actual cluster.

Guaranteed Resampling ensures that the actual state always belongs to at least one box particle (Section 4.2). In addition, it guarantees that all the other states $\mathbf{x}_k \in \mathbb{R}^d$ such that the state density is non-zero $p(\mathbf{x}_k|\mathbf{M}_k) > 0$ still belong to at least one box particle after resampling. In particular, box particles containing at least one element of neighbourhood $\mathcal{E}_k$ are always kept or subdivided. Then, all box particles of cluster $\{\chi_k\}$ are kept or subdivided. If a box particle i is subdivided, it results in a subpaving of n^i new box particles. By definition, a subpaving is a connected set, since consecutive boxes of the paving have non-empty intersections that consist of at least a common edge (Jaulin [Jau01]). Since none of the box particles belonging to the actual cluster are removed, and since subdivided box particles result in a connected subpaving, the actual cluster remains connected after Guaranteed Resampling is applied. A resampled cluster $\{\chi'_k\} \triangleq \bigcup_{i \in I'_k} [\mathbf{x}^i_k]$ can be defined, where I'_k is a new indexing set:

$$I'_k \triangleq \left\{ i \in [1,N] \mid [\mathbf{x}^i_k] \cap \mathcal{E}_k \neq \emptyset, \exists j \in I_k, [\mathbf{x}^i_k] \cap [\mathbf{x}^j_k] \neq \emptyset \right\} \tag{5.8}$$

□

It was shown that there is always a largest connected union of box particles, called cluster $\{\chi_k\}$, which contains the actual state $\mathbf{x}_k$. Furthermore, this cluster is unique. This can easily be proved by contradiction: assume that there are two disjoint connected clusters $\{\chi^1_k\}$ and $\{\chi^2_k\}$ that contain the actual state $\mathbf{x}_k$. Then, they have at least one element in common and they are not disjoint. Their union is connected and has a non-empty intersection with the actual cluster: $\left(\{\chi^1_k\} \cup \{\chi^2_k\}\right) \cap \{\chi_k\} \neq \emptyset$.

5.1.4 Asymptotic convergence to a unimodal posterior density via regularisation

The previous section showed that the actual state always belongs to a connected cluster of box particles. In other words, there is always a neighbourhood of the actual state belonging to at least one non-zero weighted box particle. Therefore, the posterior estimated density $\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)$ is always non-zero in the actual state's neighbourhood.

This section shows that regularisation (see Section 4.4) makes the posterior estimated density to asymptotically tend to a unimodal density.

Proposition 7. *Let us consider the BRPF set up with a regularisation parameter $\mu > 0$, for a unidimensional state. Then the posterior state density tends to a unimodal function for a sufficiently high number of regularisation steps (i.e. resampling steps).*

Proof. It was shown in Oudjane [Oud00] that the regularisation step is equivalent to a convolution of the posterior state density $\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)$ with a kernel $K : \mathbb{R} \to \mathbb{R}^+$ satisfying:

$$\int K(t)\mathrm{d}t = 1, \int tK(t)\mathrm{d}t = 0 \tag{5.9}$$

Therefore, it can be shown that $n \in \mathbb{N}^*$ convolutions with K of variance $\mathbf{V}_K$ are equivalent to one convolution with a kernel of variance $n\mathbf{V}_K$. Let $\mathcal{K}(X) = \int K(t) \exp(2i\pi Xt)\, dt$ be the Fourier transform of K. The first and second order derivatives of $\mathcal{K}$ are:

$$\begin{aligned} \nabla\mathcal{K}(X) &= 2\pi i \int tK(t) \exp(2i\pi Xt)\, dt \\ \nabla^2\mathcal{K}(X) &= -4\pi^2 \int t^2 K(t) \exp(2i\pi Xt)\, dt \end{aligned} \tag{5.10}$$

With $X = 0$:

$$\begin{aligned} &\mathcal{K}(0) = 1 \\ &\nabla\mathcal{K}(0) = 0 \\ &\nabla^2\mathcal{K}(0) = -4\pi^2 \int t^2 K(t)dt = -4\pi^2 \mathrm{Var}\left[t \sim K\right] = -4\pi^2 \mathbf{V}_K \end{aligned} \tag{5.11}$$

The Fourier transform of the n convolutions of K, denoted $\circledast_n K$, is the n power of K's Fourier transform, which yields:

$$\nabla^2\mathcal{K}^n(0) = -4\pi^2 \mathrm{Var}\left[t \sim \circledast_n K\right] \tag{5.12}$$

On the other hand, the differentiation of $\mathcal{K}^n$ yields:

$$\nabla^2\mathcal{K}^n = n(n-1)\mathcal{K}^{n-2}(\nabla\mathcal{K})^2 + n\mathcal{K}^{n-1}\nabla^2\mathcal{K} \tag{5.13}$$

Combining (5.13) with (5.11) for $X = 0$ yields:

$$\nabla^2\mathcal{K}^n(0) = n\nabla^2\mathcal{K}(0) = -4n\pi^2\mathbf{V}_K \tag{5.14}$$

Finally, combining (5.12) with (5.14) yields:

$$\mathrm{Var}\left[t \sim \circledast_n K\right] = n\mathbf{V}_K \tag{5.15}$$

Therefore, n convolutions of $\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)$ with K are equivalent to one convolution with a kernel of variance $n\mathbf{V}_K$. The assumption is made that the posterior estimated density $\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)$ is non-zero in the actual state's neighbourhood (Proposition 6). As a result, for n sufficiently high, the posterior density is smoothed at a point that makes it unimodal. $\square$

Remark

The above proof is provided for $d = 1$ dimension. In what follows, the same result for $d > 1$ dimensions will be admitted. A numerical study is provided in the next section for the TAN application.

5.1.5 Numerical study: application to Terrain Aided Navigation

This section experimentally checks the following hypotheses:

1. Multinomial Resampling ensures an asymptotic unimodality of the posterior density,
2. Regularisation acts as a smoothing operation on $\widehat{p}(\mathbf{x}_k|\mathbf{M}_k)$ and makes it unimodal.

To do so, define two criteria: the percentage of runs ending with more than one connected cluster at the final time-step, called the *non-connected rate* and the percentage of runs ending with more than one local maximum, called the *multimodality rate*. According to hypothesis 1, lower multimodality and non-connected rates can be expected for Multinomial Resampling than for Guaranteed Resampling. According to hypothesis 2, lower multimodality and non-connected rates can be expected with regularisation ($\mu > 0$ in equation (2.33)) than without ($\mu = 0$). The study was performed with four settings for BRPF: with Multinomial or Guaranteed Resampling and with $\mu = 0$ or $\mu = 0.1$. Table 5.1 shows the obtained criteria on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion, on a hundred runs.

Hypothesis 1 is not validated in this case as Multimodal Resampling leads to a significantly higher multimodality rate than Guaranteed Resampling. However, trajectories consist of a thousand time-steps, which leads only to about $\mathcal{O}(10)$ resampling triggering per run. This is not enough to evaluate an asymptotic behaviour. Nevertheless, it can be concluded that Guaranteed Resampling yields lower non-connected and multimodality rates in practice. This is probably due to its conservatism, leading to a higher pessimism rate, as shown in Chapter 4.

Hypothesis 2 is validated, despite the low number of resampling and regularisation steps per simulation. Indeed, performing regularisation ($\mu = 0.1$) yields significantly less multimodality. It can be concluded that regularisation makes the BRPF more compatible with Gaussian federated architectures than a fully deterministic Box Particle Filter, as long as it does not threaten the filter's stability (see Section 4.4).

Table 5.1: Non-connected rate (percentage of runs ending with more than one connected cluster) and Multimodality rate (percentage of runs ending with more than one local maximum), on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion, at final time-step. Guaranteed Resampling and regularisation tend to lower the number of modes.

	Multinomial Resampling		Guaranteed Resampling	
Regularisation parameter	$\mu = 0$	$\mu = 0.1$	$\mu = 0$	$\mu = 0.1$
Non-connected rate (%)	9.3	4.7	3.0	2.3
Multimodality rate (%)	21.3	16.7	16.0	6.0

5.2 CENTRALISED FEDERATED ARCHITECTURE

This section studies the integration of BRPF in centralised federated architectures with a Gaussian Master Filter (e.g. Kalman Filter). The problem is first described in Section 5.2.1. Section 5.1 studied the conditions for the BRPF's posterior conditional state estimated density to be unimodal, at least in an asymptotic way. Under this hypothesis, the BRPF can be integrated to a federated architecture.

5.2.1 Principle

This section states the BRPF integration scheme in federated architectures, on the basis of the framework introduced in Chapter 2, Section 2.2. Figure 5.4 illustrates the centralised case where at least one BRPF is involved to tackle TAN ambiguities in conjunction with other sensors.

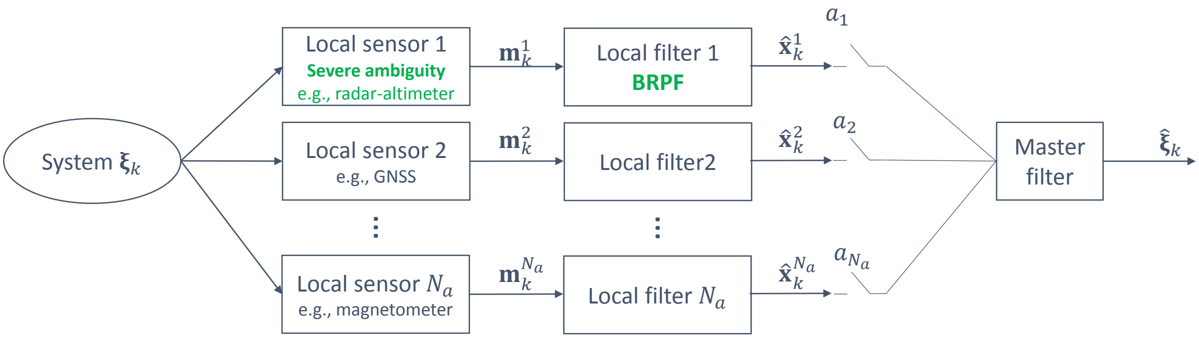


Figure 5.4: Box Regularised Particle Filter integrated to a federated architecture. Nonlinearities and ambiguities are dealt with in a first layer, and local estimates are fused together in a second layer. Sensor measurements $\mathbf{m}_k^i$ are preprocessed in a dedicated local filter in the first layer. Then, local estimates $\mathbf{x}_k^i$ are sent to the Master Filter to be fused. Communication existence between each node i (sensor and local filter) and the Master Filter is modelled by coefficients a_i. Local filters can also be identity transfer functions, i.e. some local measurements can be directly sent to the Master Filter without preprocessing.

Consider a system represented by its state $\xi_k \in \mathbb{R}^d$. Denote f the system's dynamical model:

$$\xi_k = f(\xi_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \tag{5.16}$$

where $\mathbf{w}_k$ is the system's process noise and $\mathbf{u}_k$ the system's deterministic control input.

Local sensors

The measurements' relationship with the system's state is defined as follows:

$$\left[\mathbf{m}_k^{1\,T}, ..., \mathbf{m}_k^{N_a\,T}\right]^T = h(\boldsymbol{\xi}_k) + \mathbf{v}_k \tag{5.17}$$

where h is the system's observation model and $\mathbf{v}_k$ the measurement noise. The observation model may contain severely non-linear terms, which might make it difficuly to estimate the state in a single filter, or may yield sub-optimal estimation. The state vector $\boldsymbol{\xi}$ is assumed to be globally identifiable (Ljung [LG94]).

The system is able to gather measurements from N_a *local sensors*. This scheme can be applied to various cases like IMU hybridisation or collaborative navigation for formation flying. Figure 5.4 illustrates the case of an aerial vehicle navigation system equipped with various sensors that can be used during different phases of a mission. For example, the GNSS may not be available because of jamming or signal loss. The TAN radar-altimeter can be used in the case of GNSS denied situation provided that the aircraft flies over a known terrain. Other sensors can also be used, such as magnetometers.

Local filters

A dedicated local filter can be associated with each local sensor i. Define a local state $\mathbf{x}_k^i \in \mathbb{R}^{d^i}$. Its dynamical model is defined as follows:

$$\mathbf{x}_k^i = f^i(\mathbf{x}_{k-1}^i, \mathbf{u}_k) + \mathbf{w}_k^i \tag{5.18}$$

where f^i is the local dynamical model and $\mathbf{w}_k^i$ the process noise. Local measurements are modelled as follows:

$$\mathbf{m}_k^i = h^i(\boldsymbol{\xi}_k) + \mathbf{v}_k^i \tag{5.19}$$

where h_k^i is the local observation model and $\mathbf{v}_k^i$ the local measurement noise. This results in several decoupled observation models that are easier to process independently. Non-linearities can then be tackled by dedicated filters which can be run in parallel. This has the benefit of avoiding the involvement of the global observation model h (5.17).

Coefficient $a_i \in \{0, 1\}$ quantifies if node i (local sensor and local filter) is enabled ($a_i = 1$) or disabled ($a_i = 0$). They can be associated with adjacency coefficients of a graph (see Section 2.2.1). Local filters provide local estimates covariance $\widehat{\mathbf{x}}_k^i$ associated with local covariances $\widehat{\mathbf{P}}_k^i$.

Master filter

The Master Filter fuses all the available local estimates together with the system dynamics (5.16). The local estimates form a set of virtual measurements that are assumed to depend on the state in a linear way:

$$\widehat{\mathbf{x}}_k^i = \mathbf{H}^i \boldsymbol{\xi}_k + \widehat{\boldsymbol{v}}_k^i \tag{5.20}$$

where $\mathbf{H}^i \in \mathbb{R}^{d^i \times d}$ and $\widehat{\boldsymbol{v}}_k^i \in \mathbb{R}^{d_m^i}$ a noise of covariance $\widehat{\mathbf{P}}_k^i$. This results in a federated estimate $\widehat{\boldsymbol{\xi}}_k$. Equation (5.20) can be extended to non-linear dependency, but this is beyond the scope of this work. The objective of this architecture is dealing with non-linearities in the first layer only, at a local level.

5.2.2 Application to multi-sensor navigation

This section implements the BRPF in federated architectures to illustrate two advantages of this approach with respect to a single level BRPF, dealing with all the measurements together:

1. BRPF permits robust state estimation from ambiguous, non-linear, non-Gaussian measurements, but is sub-optimal for Gaussian measurements. Dealing with Gaussian measurements in dedicated Gaussian filters would yield a near-optimal solution,
2. BRPF computational load depends on the square state dimension d^2 (see Chapter 4, Table 4.13). Therefore, it is of interest to reduce the BRPF dynamics dimension as much as possible. Federated architectures allow the BRPF to focus on state variables involved in non-linearities, while the Master Filter's output can be of a higher dimension. This can be interpreted as an implicit marginalisation. Furthermore, Gaussian measurements are tackled in a near optimal way by dedicated Gaussian filters.

Federated architectures are compared with the BRPF only architecture where a BRPF processes all the measurements together, under the uniform measurement hypothesis (see Chapter 4, Section 4.1, equation (4.18)). In this section, all BRPF implementations are set up with Guaranteed Resampling (see Section 4.2), Geometrical Subdivision (see Section 4.3.1) and Kernel Regularisation (with $\mu = 0.1$, see Section 4.4). Unless a different setting is explicitly stated, the BRPF algorithm is tuned with $N = 900$ box particles, no process noise and a resampling threshold of $\theta_{eff} = 0.7$.

Accounting for Gaussian measurements

The first example implements a 6-dimensional double integrator dynamical model with Terrain Aided Navigation (TAN) and velocity measurements. The control input is assumed to be zero. The state vector is modelled by three positions $\mathbf{p}_k \in \mathbb{R}^3$ and three velocities $\mathbf{v}_k \in \mathbb{R}^3$:

$$\boldsymbol{\xi}_k = \left[\mathbf{p}_k^T, \mathbf{v}_k^T\right]^T \in \mathbb{R}^6 \tag{5.21}$$

The TAN radar altimeter measurements, denoted $\mathbf{m}_k^1 \in \mathbb{R}$ in what follows, are assumed to have bounded uncertainty. They are modelled as presented in Section 3.2, equation (3.7). The velocity measurements, denoted $\mathbf{m}_k^2 \in \mathbb{R}^3$ in what follows, are assumed to be linear-Gaussian:

$$\mathbf{m}_k^2 = \mathbf{v}_k + \boldsymbol{v}_k^2 \tag{5.22}$$

where $\boldsymbol{v}_k^2 \in \mathbb{R}^3$ is associated with a random vector following a Normal law. Measurement frequencies for each sensor may be different. An active sensor i is associated with an adjacency coefficient $a_i = 1$, while an inactive sensor yields $a_i = 0$. The proposed federated architecture illustrated in Figure 5.5 consists of:

- One BRPF, processing the radar-altimeter measurements in conjunction with a Digital Elevation Model (DEM, see Section 3.2);
- One Master Filter (here, a Kalman Filter), estimating the system's state $\boldsymbol{\xi}_k$ from the BRPF's output (position $\widehat{\mathbf{p}}_k^1$ only) and the velocity measurement $\mathbf{m}_k^2$.

Both filters use the 6-dimensional double integrator dynamical model (See Section 3.1, equation (3.1)). Note that the BRPF estimates a 6-dimensional vector, but only the position dimensions are used by the Master Filter. Simulation settings (noises and frequencies) are presented in Table 5.2. The regularisation step is enabled for Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion, with $\mu = 0.1$. It is disabled for Scenario TAN Sinusoidal terrain and Scenario TAN Moon flyby. The computational load required by the Kalman Filter is negligible with respect to that required by the BRPF (see Section 4.5 and Appendix C): $d^3 \ll Nd^2$ with $d = 6$ and $N = 900$. Therefore, it can be considered that both architectures (BRPF only and the federated architecture) have a similar computational load.

Table 5.2: Federated architecture settings

Parameter	Value
Radar altimeter uncertainty bounds	±45 m
Radar altimeter actual density	Truncated Gaussian (st.d 15 m)
Radar altimeter frequency	10 Hz
Velocity sensor (standard deviation)	$[2, 2, 2]$ ms^{-1}
Velocity sensor frequency	1 Hz
Process noise in position (st.d)	$[20, 20, 5]$ m
Process noise in velocity (st.d)	$[0.1, 0.1, 0.1]$ ms^{-1}
Initial uncertainty in position (st.d)	$[1000, 1000, 100]$ m
Initial uncertainty in velocity (st.d)	$[3, 3, 1]$ ms^{-1}

Figure 5.7 illustrates the RMSE obtained on Scenario TAN Alps for the federated architecture's output (the green curve), and BRPF alone (all the measurements are processed together by the BRPF, the blue curve). Velocity RMSE converges in a faster way, which illustrates the interest of processing Gaussian measurements in a Gaussian filter. These results are confirmed for all the scenarios, as illustrated in Figure 5.6 and Table 5.3, on a hundred runs. RMSE in position appears slightly downgraded, but not in a significant way. The velocity RMSE is significantly improved.

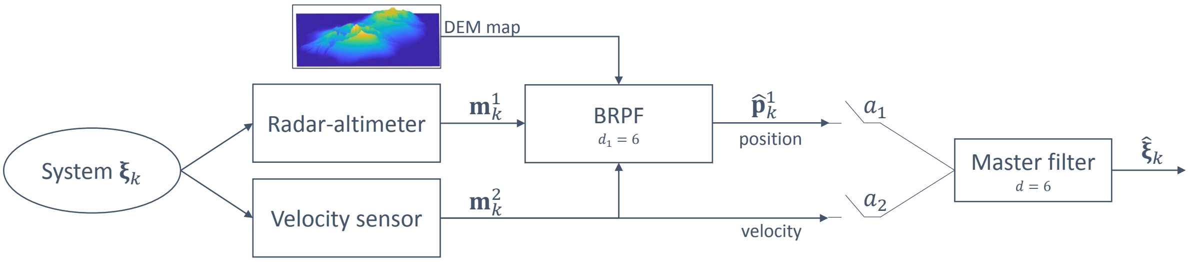


Figure 5.5: Box Regularised Particle Filter integrated to a federated architecture. BRPF tackles the ambiguous and non-linear TAN problem and provides a position estimate to the Gaussian Master Filter (e.g. Kalman Filter). The Master Filter estimates the whole state vector (position, measurement) from the BRPF's position estimation under Gaussian asymptotic hypothesis and the Gaussian velocity measurements. The BRPF and the Master Filter use the same 6-dimensional dynamical model.

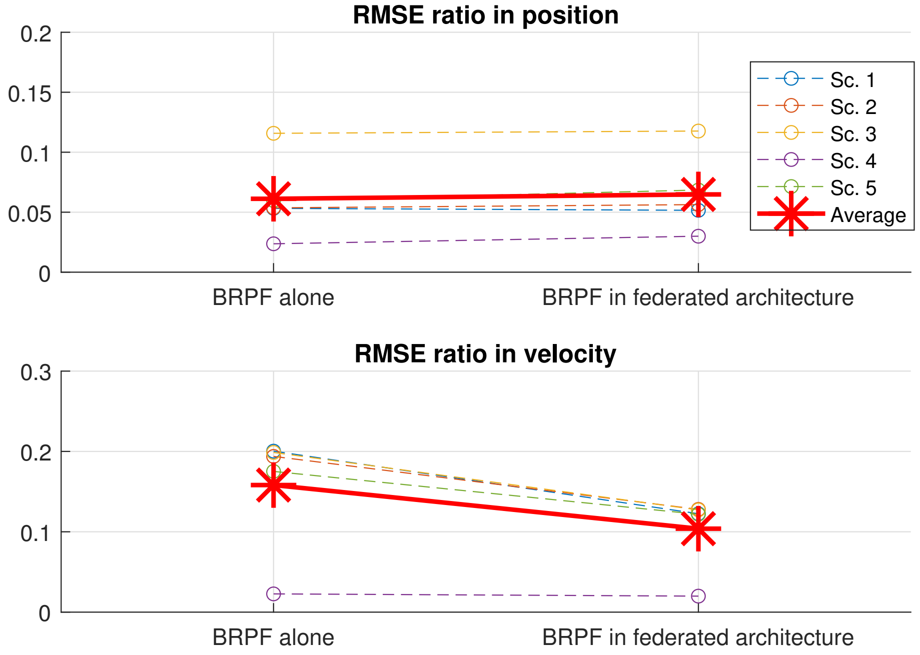


Figure 5.6: Root Mean Square Error ratios obtained for the Box Regularised Particle Filter in a federated architecture described in Figure 5.5, and BRPF alone (all measurements are processed together by one BRPF), on all scenarios.

Table 5.3: Averaged results for one Box Regularised Particle Filter alone and the federated architecture on Scenario TAN Alps, Scenario TAN Canyon, and Scenario TAN La Reunion, at final time-step.

	Single BRPF	Federated architecture
Non-convergence rate (%)	0	0
RMSE ratio (position)	0.061	0.065
RMSE ratio (velocity)	0.16	0.10
Pessimism (position)	1.3	1.3
Pessimism (velocity)	1.26	1.49

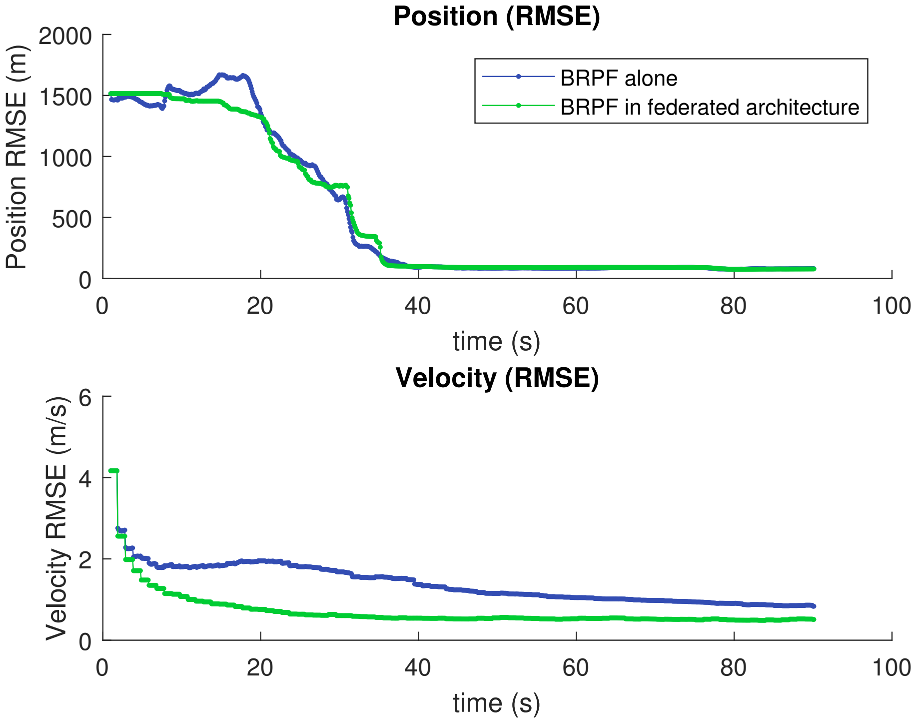


Figure 5.7: Root Mean Square Error obtained for the Box Regularised Particle Filter in a federated architecture described in Figure 5.5, (green curve), and BRPF alone (all measurements are processed together by the BRPF, blue curve), on Scenario TAN Alps. The federated architecture allows the Kalman Filter to process Gaussian measurements (velocity) while the BRPF deals with non-linearities and ambiguities (TAN problem). BRPF only proves to be sub-optimal in Gaussian measurement filtering, which yields a higher velocity RMSE than for the federated architecture.

Extending the state vector without increasing the computational load

A second example implements a 15-dimensional IMU hybridisation dynamical model, updated with TAN measurements, velocity measurements, and attitude measurements. The aircraft trajectory is assumed to be a straight and level flight.

The TAN radar altimeter measurements are denoted $\mathbf{m}_k^1 \in \mathbb{R}$, the velocity measurements are denoted $\mathbf{m}_k^2 \in \mathbb{R}^3$ and are assumed to be linear-Gaussian. The attitude measurements are denoted $\mathbf{m}_k^3 \in \mathbb{R}^3$ and are assumed to be linear-Gaussian:

$$\mathbf{m}_k^3 = [\psi_k, \theta_k, \varphi_k]^T + \boldsymbol{v}_k^3 \tag{5.23}$$

where ψ_k, θ_k, and φ_k are the attitude Euler angles (respectively heading, pitch, roll), and $\boldsymbol{v}_k^3 \in \mathbb{R}^3$ is associated with a random vector following a Normal law. The IMU measurements are denoted $\mathbf{m}_k^\alpha \in \mathbb{R}^3$ for the acceleration and $\mathbf{m}_k^\omega \in \mathbb{R}^3$ for the angular rate. The proposed federated architecture illustrated in Figure 5.8 consists of:

- One double integrator transfer function integrating the IMU measurements and providing a state IMU estimate $\widehat{\boldsymbol{\xi}}_k^{IMU}$ which consists of the vehicle IMU position, the IMU velocity, and the IMU attitude, given an initial IMU state;
- One BRPF, updating the IMU position from the radar-altimeter measurements, in conjunction with a Digital Elevation Model, the IMU estimate, and the velocity sensor;
- One Master Filter (here, a Kalman Filter), estimating the system's state $\boldsymbol{\xi}_k$ from the BRPF's output (position $\widehat{\mathbf{p}}_k^1$ only), the velocity measurements $\mathbf{m}_k^2$, and the attitude measurements $\mathbf{m}_k^2$. The estimate $\widehat{\boldsymbol{\xi}}_k$ consists of the nine updated IMU state variables (positions, velocities, and attitude angles) and six estimated biases (accelerometer and gyrometer biases).

The Master Filter uses the 15-dimensional IMU error propagation model (See Section 3.1, equation (3.6)). The BRPF uses a simplified 6-dimensional IMU error propagation model, whose process noise consists of a marginalisation of the attitude and biases drift. Simulation settings (noises and frequencies) are presented in Table 5.4, for a hundred runs. This federated architecture appears to have a significantly lower computational load when compared to the single BRPF (15-dimensional BRPF dealing with all measurements togather). Indeed, the BRPF computational load per time-step is about Nd^2 (see Section 4.5 and Appendix C). Since the federated architecture yields a significant BRPF state dimension reduction from 15 to 6 with respect to a unique BRPF, the theoretical computation gain is therefore of 60% for an equal number of box particles.

Figure 5.9 illustrates the RMSE obtained on Scenario TAN Alps for the federated architecture's output (the green curve), and BRPF alone (all measurements are processed together by the BRPF, the blue curve), for a hundred runs. Velocity RMSE converges in a faster way, which illustrates the interest of processing Gaussian measurements in a Gaussian filter. Pitch and roll angles also converge in a faster way than for BRPF only. Nevertheless, the heading angle does not benefit from the federated architecture. Indeed, when a vehicle performs a straight and level flight, the heading is dynamically decoupled from the other state variables, and thus is not observable (see Nordlund [Nor02]). Therefore, the slow convergence observed on the heading dimension RMSE is only due to the attitude sensor (st.d. of 1 °). On the contrary, the pitch and roll angles benefit from their dynamical correlations with the velocity and position state variables, which allow them to converge faster than the heading in a Gaussian Master Filter (here, a KF). However, when attitude is only estimated by a BRPF, the dynamical couplings are not explicitly accounted (see the *wrapping effect* in Section 2.1.5). It yields a slower and less accurate convergence of these variables that only benefit from the attitude sensor. These results are confirmed in all scenarios, as shown in Table 5.5. The RMSE ratios yielded by the federated architecture remain acceptable (globally lower or equal than 0.2), except for the heading which is only observed via the attitude sensor. The pessimism ratios are significantly reduced with respect to the single BRPF in position and velocity (respectively a 87% and a 26% decrease). The attitude angles' pessimism ratios increase due to the dramatic improvement of the estimation accuracy but remain lower than 1.5 (maximum value of 1.24 for the pitch). The empirical computation gain between a 15-dimensional BRPF alone and a federated architecture including a 6-dimensional BRPF is about 60% (12 ms versus 30 ms per time-step), which confirms the theoretical computational load predictions.

To conclude, the Gaussian federated architecture yields a lower computational load while significantly improving the estimation accuracy and reducing the conservatism.

Table 5.4: Federated architecture settings

Parameter	Value
Radar altimeter uncertainty bounds	±45 m
Radar altimeter actual density	Truncated Gaussian (st.d 15 m)
Radar altimeter frequency	10 Hz
Velocity sensor (standard deviation)	$[2, 2, 2]$ ms^{-1}
Velocity sensor frequency	1 Hz
Attitude sensor (st.d)	$[1, 1, 1]$ °
Attitude sensor frequency	10 Hz
Accelerometer biases (st.d)	$[3, 3, 3]\ 10^{-5}$ ms^{-2}
Gyrometer biases (st.d)	$[5, 5, 5]\ 10^{-4}$ °s^{-1}
Process noise in position (st.d)	$[20, 20, 5]$ m
Process noise in velocity (st.d)	$[0.1, 0.1, 0.1]$ ms^{-1}
Process noise in attitude (st.d)	$[1, 1, 1]\ 10^{-8}$ °
Process noise in acc. biases (st.d)	$[5, 5, 5]\ 10^{-8}$ ms^{-2}
Process noise in gyr. biases (st.d)	$[1, 1, 1]\ 10^{-8}$ °s^{-1}
Initial uncertainty in position (st.d)	$[1000, 1000, 100]$ m
Initial uncertainty in velocity (st.d)	$[3, 3, 1]$ ms^{-1}
Initial uncertainty in attitude (st.d)	$[3, 3, 3]$ °

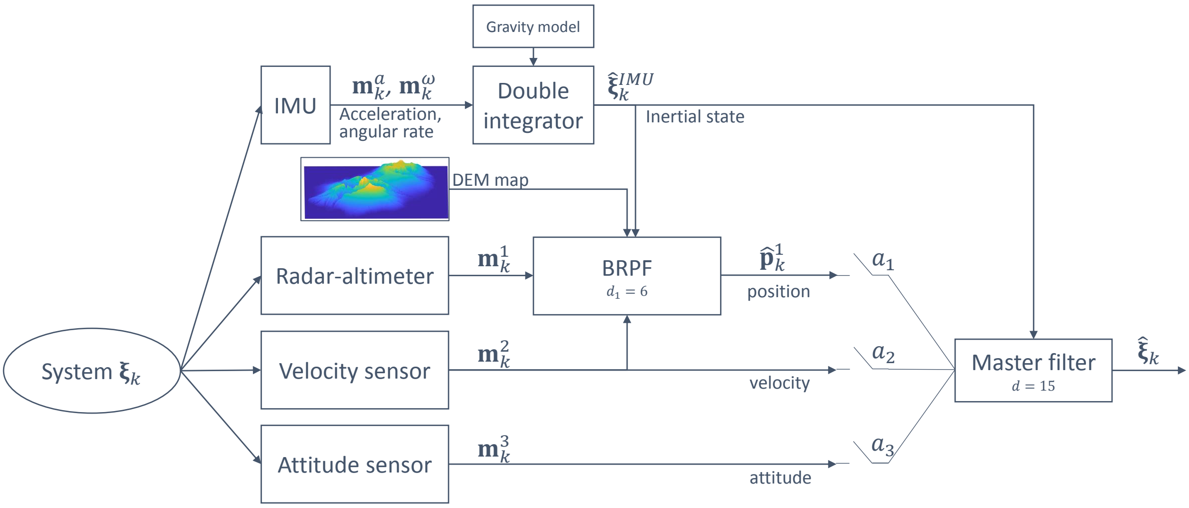


Figure 5.8: Box Regularised Particle Filter integrated to a federated architecture in the context of IMU hybridisation with TAN, velocity, and attitude measurements. In addition to separating severely ambiguous measurements (radar-altimeter) from Gaussian ones, this architecture allows the BRPF to only run a 6-dimensional dynamical model, while the Master Filter estimate the whole state vector (15 dimensions). The federated architecture allows the computational load to be significantly reduced with respect to a 15-dimensional BRPF estimation that would be, in addition, sub-optimal.

Table 5.5: Averaged results for the single Box Regularised Particle Filter and the federated architecture on all scenarios, at final time-step.

	Single BRPF	Federated architecture
Non-convergence rate (%)	0	0
RMSE ratio (position)	0.15	0.21
RMSE ratio (velocity)	0.49	0.14
RMSE ratio (heading)	0.51	0.55
RMSE ratio (pitch)	0.53	0.03
RMSE ratio (roll)	0.53	0.03
Pessimism (position)	2.74	1.36
Pessimism (velocity)	1.37	1.01
Pessimism (heading)	0.55	1.01
Pessimism (pitch)	0.62	1.24
Pessimism (roll)	0.48	1.11
Time (ms)	30	12

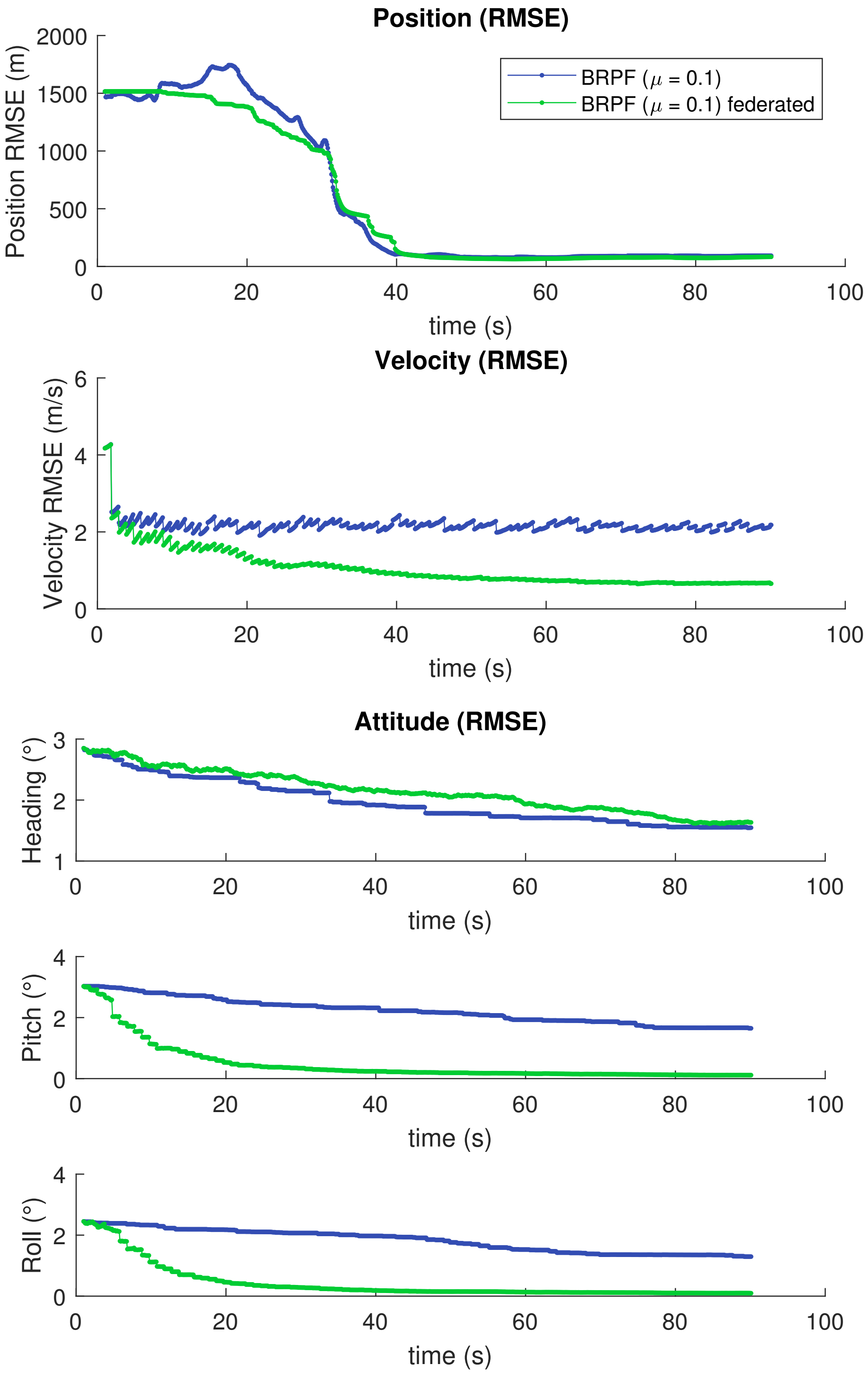


Figure 5.9: Root Mean Square Error obtained for Box Regularised Particle Filter in the federated architecture described by Figure 5.8 (green curve), and BRPF alone (all measurements are processed together by the BRPF, blue curve), on Scenario TAN Alps. The federated architecture allows Gaussian measurements to be processed in a near-optimal way by the Master Filter (Kalman Filter) while BRPF deals with ambiguities (TAN problem).

5.3 DISTRIBUTED FEDERATED ARCHITECTURES

5.3.1 Principle

Consider $N_a \in \mathbb{N}^*$ agents gathering local measurements, relative measurements, and communicated information to retrieve their own state vector $\mathbf{x}_k^i \in \mathbb{R}^d$, as illustrated in Figure 5.10.

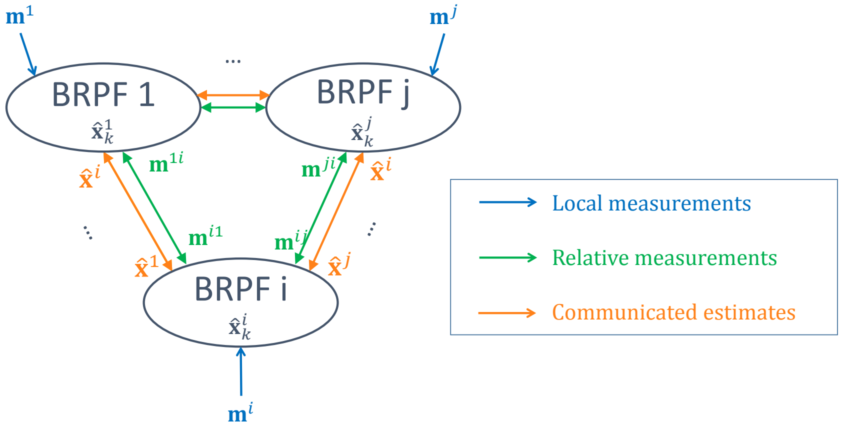


Figure 5.10: Collaborative estimation scheme: each system i uses a BRPF to estimate its own state $\mathbf{x}_k^i$ from its local measurements $\mathbf{m}_k^i$. In order to improve its estimation quality, it is allowed to exchange its estimate $\hat{\mathbf{x}}_k^i$ with its connected neighbors (e.g. j) and to get some relative measurements $\mathbf{m}_k^{ij}$.

Define a generic scheme where measurements and communications consist of:

- A measurement vector $\mathbf{m}_k^i \in \mathbb{R}^{d_m}$, which may be highly nonlinear and ambiguous with a potentially unknown-but-bounded uncertainty.
- $N_a - 1$ relative measurement vectors $\mathbf{m}_k^{ij}$ $(j \neq i)$. They are obtained from the relative states $\mathbf{x}_k^{ij} \triangleq \mathbf{x}_k^j - \mathbf{x}_k^i$, such that:

  $$\mathbf{m}_k^{ij} = h^r(\mathbf{x}_k^{ij}) + \mathbf{v}_k^{ij} \tag{5.24}$$

  The relative dynamics is denoted:

  $$\mathbf{x}_k^{ij} = f^r(\mathbf{x}_{k-1}^{ij}, \mathbf{u}_k^i) + \mathbf{w}_k^{ij} \tag{5.25}$$

  where $f^r : \mathbb{R}^d \times \mathbb{R}^{d_c} \to \mathbb{R}^d$ is the relative dynamics and $\mathbf{w}_k^{ij}$ is associated with a random vector modelling the model uncertainty, as well as the uncertain control input of agent j.
- Communicated information consists of the other agents BRPF outputs $\hat{\mathbf{x}}_k^j$ associated with covariance $\hat{\mathbf{P}}_k^j \in \mathbb{R}^{d\times d}$.

Agent i collects all the available estimates from agents $j \neq i$ and combines them with the corresponding relative estimates $\widehat{\mathbf{x}}_k^{ij}$ (associated with covariance $\widehat{\mathbf{P}}_k^{ij} \in \mathbb{R}^{d \times d}$) to retrieve a local estimate of its state i given the knowledge obtained from j, i.e.:

$$\widehat{\mathbf{x}}_k^{i|j} = \widehat{\mathbf{x}}_k^{j} - \widehat{\mathbf{x}}_k^{ij} \tag{5.26}$$

which can be used as a pseudo-measurement to update the state of agent i. It can be associated with the following covariance, under the assumption that $\widehat{\mathbf{x}}_k^{j}$ and $\widehat{\mathbf{x}}_k^{ij}$ are statistically independent:

$$\widehat{\mathbf{P}}_k^{i|j} = \widehat{\mathbf{P}}_k^{j} + \widehat{\mathbf{P}}_k^{ij} \tag{5.27}$$

In a distributed federated architecture, each measurement vector (local and relative) can be processed by dedicated filters. If the local absolute measurement vector $\mathbf{m}_k^i$ is highly ambiguous and nonlinear with respect to the state, it can be associated with a box $[\mathbf{m}_k^i] \in \mathbb{IR}^{d_m}$ and can be processed by a local BRPF (e.g. in the TAN case). If some local absolute measurements are linear-Gaussian, they can be processed by additional Gaussian filters or be directly sent to the Master Filter, as developed in the previous section. For the sake of brevity, this case will not be detailed here.

If the relative estimation problem is linearisable and near-Gaussian, each relative state $\mathbf{x}_k^{ij} \triangleq \mathbf{x}_k^{j} - \mathbf{x}_k^{i}$ between agent j and agent i can be estimated by a KF or an EKF, as illustrated in Figure 5.11.

The Master Filter, taken as an information filter (see Section 2.2.2), can estimate the agent's state $\mathbf{x}_k^i$ from all the available local estimates, namely, the local BRPF i estimate $\mathbf{x}_k^i$ and the conditional local estimates from other agents $\widehat{\mathbf{x}}_k^{i|j}$ ($j \neq i$):

$$\begin{cases} \widehat{\mathbf{x}}_{k,MF}^{i} = \widehat{\mathbf{P}}_{k,MF} \left({\widehat{\mathbf{P}}_{k|k-1,MF}^{i}}^{-1} \widehat{\mathbf{x}}_{k|k-1,MF}^{i} + {\widehat{\mathbf{P}}_k^{i}}^{-1} \widehat{\mathbf{x}}_k^{i} + \sum_{\substack{j=1 \\ j\neq i}}^{N_a} a_{ij} \, {\widehat{\mathbf{P}}_k^{i|j}}^{-1} \widehat{\mathbf{x}}_k^{i|j} \right) \\ \widehat{\mathbf{P}}_{k,MF} = \left({\widehat{\mathbf{P}}_{k|k-1,MF}^{i}}^{-1} + {\widehat{\mathbf{P}}_k^{i}}^{-1} + \sum_{\substack{j=1 \\ j\neq i}}^{N_a} a_{ij} \, {\widehat{\mathbf{P}}_k^{i|j}}^{-1} \right)^{-1} \end{cases} \tag{5.28}$$

where the predicted estimates are obtained by:

$$\begin{aligned} \widehat{\mathbf{x}}_{k|k-1,MF} &= f\left(\widehat{\mathbf{x}}_{k-1,MF}, \mathbf{u}_k\right) \\ \widehat{\mathbf{P}}_{k|k-1,MF} &= \mathbf{F}_{k-1} \widehat{\mathbf{P}}_{k-1,MF} \mathbf{F}_{k-1}^T + \mathbf{Q}_k \end{aligned} \tag{5.29}$$

where f is the dynamical model and $\mathbf{Q}_k \in \mathbb{R}^{d \times d}$ the process noise covariance.

This distributed federated architecture holds the previously presented advantages: near-optimal processing of Gaussian measurements and potential dimension reduction (the Master Filter being able to extend the state dimension to additional state variables).

The availability of communication and relative sensing can be modelled by two graphs (see Chapter 2, Section 2.2.1). For the sake of simplicity, assume that the communication graph and the relative sensing graph are equal and that each relative filter ij produces the associated relative estimate $\widehat{\mathbf{x}}_k^{ij}$ at a greater or equal rate than the communication's rate from j to i occurs. No delays are considered for communications and for measurements. The adjacency coefficients are denoted $a_{ij} \in \{0, 1\}$.

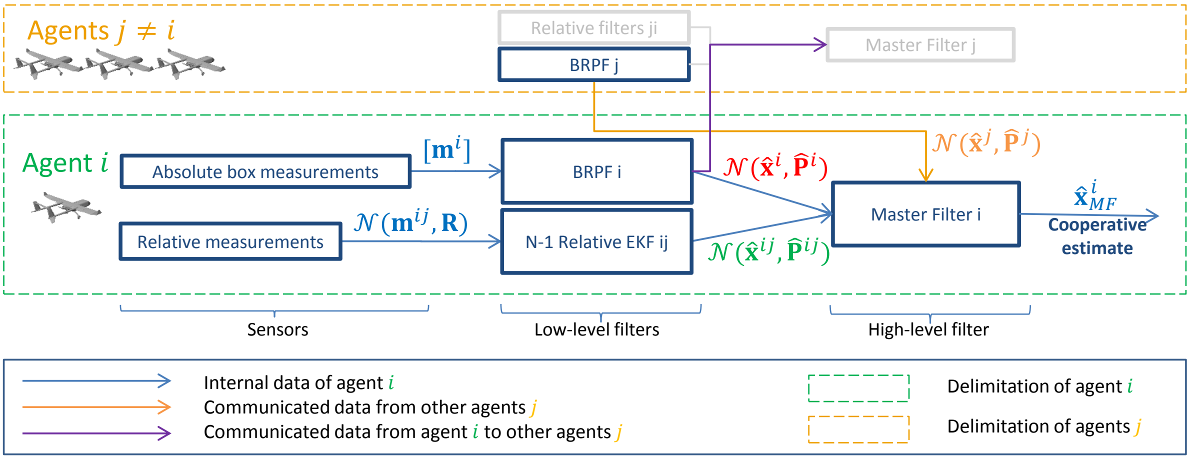


Figure 5.11: Two stages fusion architecture for distributed BRPF: each agent i gets its local measurement box $[\mathbf{m}^i_k]$. A first estimate $\hat{\mathbf{x}}^i_k$ is done by the BRPF. Simultaneously, an Extended Kalman Filter provides an estimation of the relative states $\hat{\mathbf{x}}^i_k$ using the relative measurements $\mathbf{m}^{ij}_k$. Then, the Master Filter fuses a unimodal approximation of the own state estimate, the relative estimates, and unimodal approximations of the communicated states in a linear way.

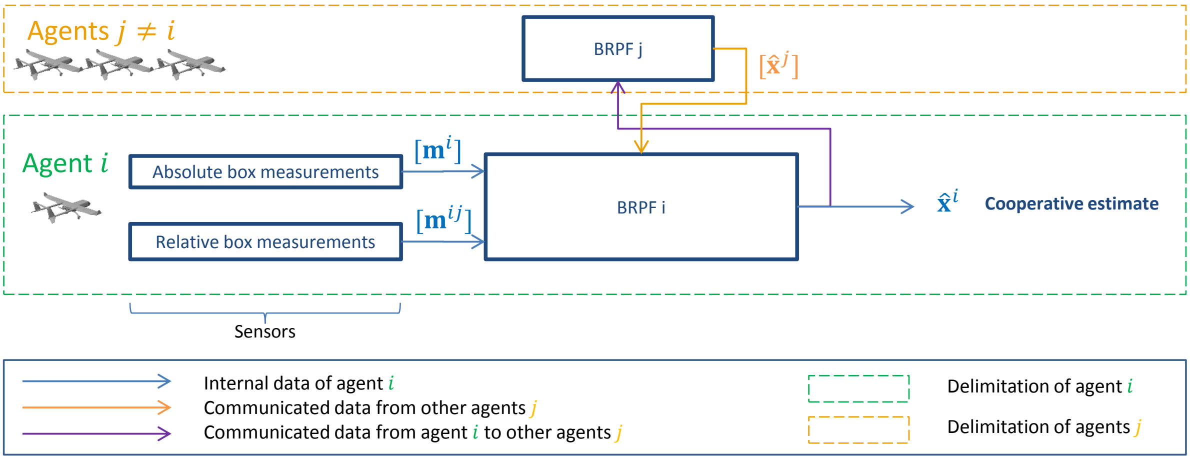


Figure 5.12: One stage fusion architecture for distributed BRPF: each agent i gets its local measurement box $[\mathbf{m}^i_k]$, the relative measurement boxes $[\mathbf{m}^{ij}_k]$ with respect to other agents $j \neq i$, and the communicated estimation box $[\hat{\mathbf{x}}^j_k]$. By combining these sets, viewed as a whole measurement box, $[\mathbf{m}^i_k] \times [\mathbf{m}^{ij}_k] \times [\hat{\mathbf{x}}^j_k]$, the filter can improve its estimate quality. However, such a single state BRPF appears less accurate than a BRPF integrated to a federated architecture.

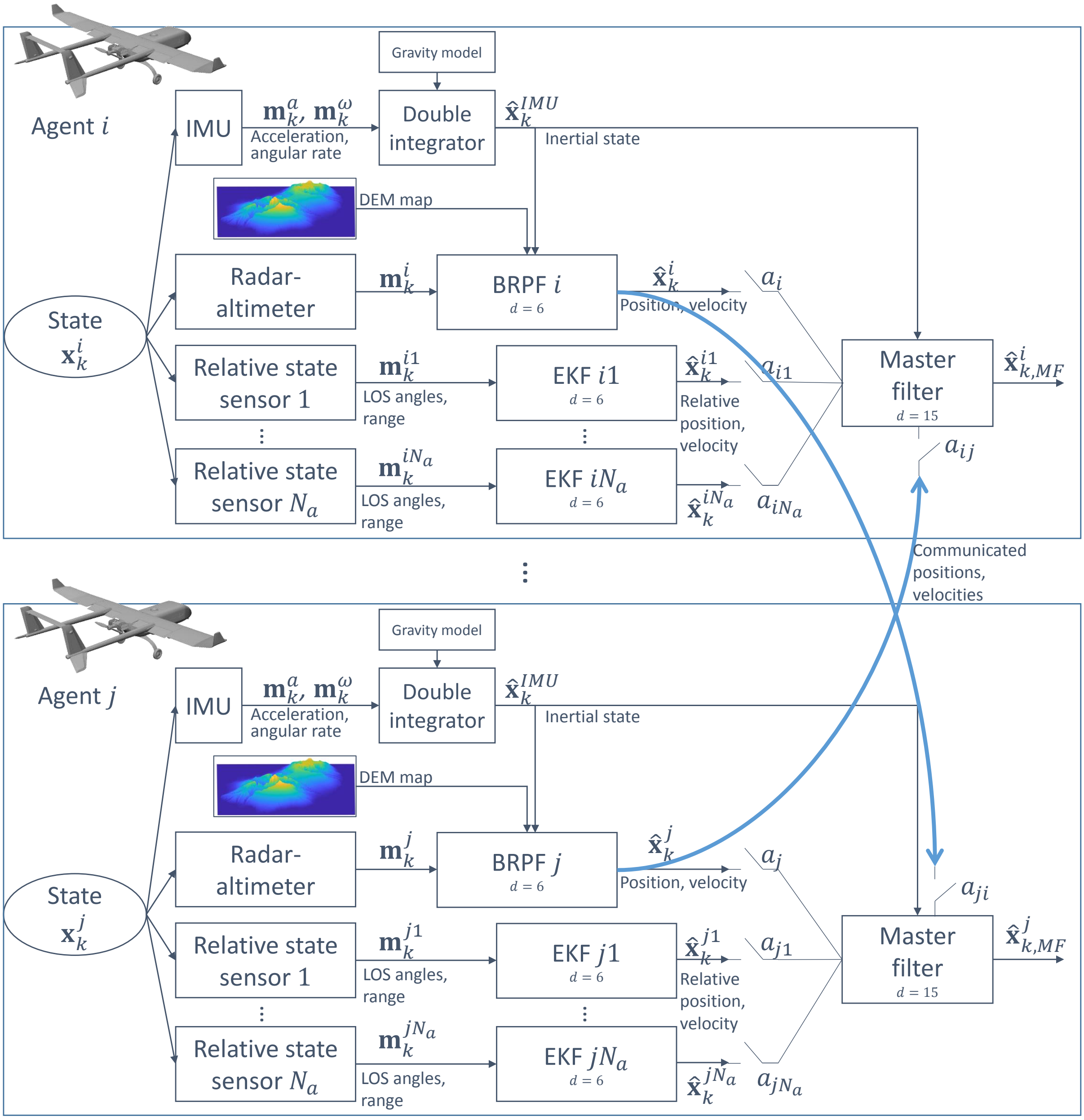


Figure 5.13: BRPF integrated to a distributed federated architecture in the context of IMU hybridisation with TAN, relative measurements, and communicated BRPF estimates. Two agents i and j among the N_a vehicles are represented, with $j \neq i$. Local absolute measurements $\mathbf{m}_k^i$ are obtained from a radar altimeter, local relative measurements $\mathbf{m}_k^{ij}$ $(j \neq i)$ are Line Of Sight (LOS) angles and range, and communicated information are BRPF estimates $\hat{\mathbf{x}}_k^i$. The BRPF is assumed to run with a 6-dimensional IMU hybridisation model. The $N_a - 1$ relative EKF filters run with a 6-dimensional relative state model. The Master Filter runs with a 15-dimensional IMU hybridisation model and fuses local filters outputs in a linear way.

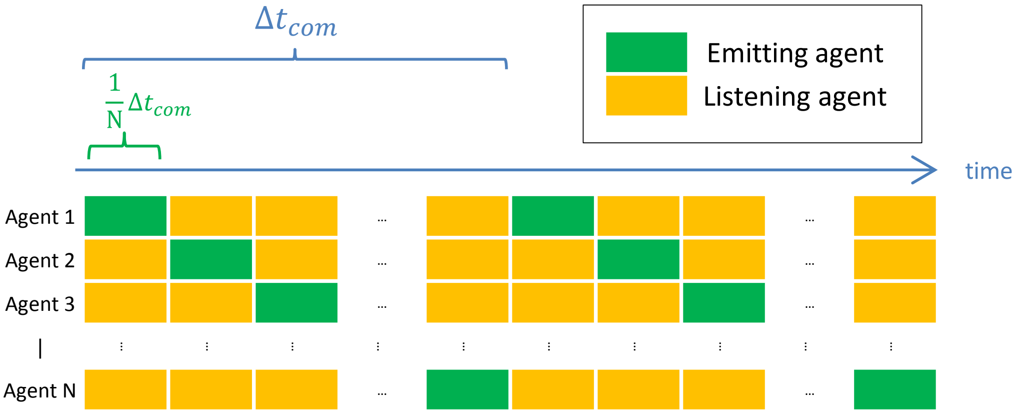


Figure 5.14: Communication protocol

5.3.2 Application to cooperative navigation in a fleet of vehicles

This section applies the distributed federated architecture introduced in Section 5.3.1 to a formation flying case. Absolute measurements $\mathbf{m}_k^i$ are radar altimeter for Terrain Aided Navigation (TAN), in the context of IMU hybridisation (see Section 3.1). Relative measurements $\mathbf{m}_k^{ij}$ are polar measurements (See Section 3.2). The range measurements are obtained with a low frequency and a high uncertainty, which makes the relative sensing coarse. Communicated information consists of the BRPF output in terms of estimated position and velocity $\widehat{\mathbf{x}}_k^i$, as well as the associated covariances $\widehat{\mathbf{P}}_k^i$. The BRPF model can be reduced to a 6-dimensional model, as introduced in Section 5.2. Since the vehicles fly in formation and in near straight and level flight evolutions, the relative dynamics can be approximated as a double integrator (see Section 3.1). It is assumed that the communication graph is fully connected. However, in practice, a vehicle cannot receive and emit information at the same time. The chosen communication protocol consists of periodic rotations illustrated in Figure 5.14. Other protocols could also be used, such as event-triggered communications (e.g. Zhang [Zha+14], Viel [Vie+17]), but this is outside the scope of this work.

Comparison to a single layer fusion architecture

In this section, the distributed federated architecture is compared with a single layer BRPF, where the LOS measurements, the communicated states and the terrain navigation measurements are fused at the same time. Therefore, in addition to the basic cost $c_{BRPF} = \mathcal{O}(Nd^2)$ (See Section 4.5), the box contraction operation derived from the polar tracking observation model (see Section 3.2) requires about $\mathcal{O}(10^2 N n_k^i)$ elementary operations, with N the number of box particles and n_k^i the number of received communications at a current time. With $N = 10^3$, $n_k^i = 1$, and $d = 15$, the single layer architecture would require about $\mathcal{O}(10^5)$ operations per time-step. Since this architecture does not require any additional filtering layer and fuses only one communicated measurement at a time, the computation load is independent from the total number of vehicles N_a. However, it has a high cost that can be dramatically reduced by the federated strategy.

A conventional Kalman Filter requires $c_{EKF}(d, d_m) = 4d^3 + 3d^2 d_m + 2dd_m^2 + d_m^3/6$ operations, with d and d_m the state dimension and the measurements dimension respectively. Based on the same methodology, the proposed Master Filter requires $c_{MF}(d, d_m, n_k^i) = 4d^2 + 6d^3 + d_m^3 n_k^i - d_m^3 + 3d_m^2 n_k^i - 2d_m^2 - 2d + d_m$ operations. Then, the proposed architecture requires the following number of operations: $c_{total} = (N_a - 1)c_{EKF}(d_{EKF}, d_{LOS}) + c_{BRPF}(d_{BRPF}, N) + c_{MF}(d_{MF}, d_{EKF}, n_k^i)$, with $d_{EKF} = 6$, $d_{LOS} = 3$, $d_{MF} =$

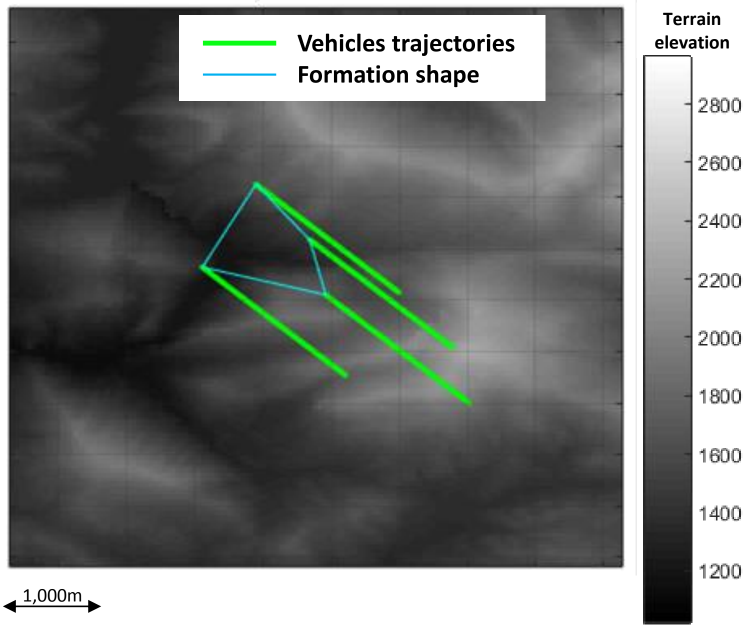


Figure 5.15: UAVs trajectories (green) above a mountainous terrain (gray-scale)

15, N_a the number of vehicles and $n_k^i = 1$. Under these conditions, each EKF computational load is about $c_{EKF}(6,3) = \mathcal{O}(10^2)$, the local BRPF computational load is about $c_{BRPF}(6,10^3) = \mathcal{O}(10^4)$, and the Master Filter computational load is about $c_{MF}(15,6,1) = \mathcal{O}(10^4)$. Therefore, the whole architecture requires a total number of operations of about $\mathcal{O}(10^2(N_a - 1) + 10^4)$, which is equivalent to $\mathcal{O}(10^4)$ if $N_a \ll 10^3$, which is the case for the considered type of scenario ($N_a = \mathcal{O}(10)$). Therefore, one less order of magnitude operations can be expected per time-step than with the single layer BRPF architecture.

Simulation results

A numerical simulation is considered with four fixed-wing UAVs flying in formation over a mountainous terrain, as illustrated in Figure 5.15. Table 5.6 describes the simulation parameters. In the multi-agent context, the global RMSE is defined, which consists of the average of the N_a agent's RMSE:

$$RMSE_k = \frac{1}{N_a} \sum_{i=1}^{N_a} RMSE_k^i \tag{5.30}$$

Likewise, a global pessimism ratio can be defined. Unless a different setting is explicitly stated, the BRPF algorithm is tuned with $N = 900$ box particles, no process noise and a resampling threshold of $\theta_{eff} = 0.7$.

Table 5.7 presents the RMSE, the estimated uncertainty (standard deviations), and the resulting pessimism ratios obtained with BRPF alone (Figure 5.12) and the federated architecture (Figure 5.11 and Figure 5.13), for a hundred runs. A significant improvement can be noticed in RMSE with the federated architecture compared to BRPF alone: 14% decreasing in position and 15% in velocity, while the attitude RMSE remains the same. The federated architecture also appears more accurate in terms of conservatism: 67% decreasing of the pessimism ratio in position, 64% in velocity and roughly 60% in attitude. This yields a significantly less conservative estimation. The pessimism ratio remains lower than 1.5 for the federated architecture, while the BRPF yields a pessimism ratio greater than 2. The computational load is reduced by 42%. These results confirm the ability of federated architecture to

Table 5.6: Simulation configuration

General	Value
Communication update rate	$\Delta t_{com} = 1\,\text{s}$
Number of vehicles	$N_a = 4$
Relative distance	$10^3\,\text{m}$
Absolute velocity	$200\ \text{ms}^{-1}$
IMU initial position error (std)	$\left[10^3, 10^3, 10^2\right]\ \text{m}$
IMU initial velocity error (std)	$[2, 2, 1]\ \text{ms}^{-1}$
IMU initial attitude error (std)	$\left[5 \times 10^{-3}, 5 \times 10^{-3}, 5 \times 10^{-3}\right]\ \text{rad}$
IMU accelerometer biases (std)	$\left[10^{-2}, 10^{-2}, 10^{-2}\right]\ \text{ms}^{-2}$
IMU gyrometer biases (std)	$\left[10^{-4}, 10^{-4}, 10^{-4}\right]\ \text{rad s}^{-2}$
Measurements	
Radar-altimeter error (support)	$v_k \in [-45, +45]\,\text{m}$
Relative angles error (std)	$\sigma_\beta = 1^\circ$ and $\sigma_\varepsilon = 1^\circ$
Relative range error (std)	$\sigma_r = 500\,\text{m}$
Radar-altimeter update rate	$\Delta t_{RA} = 0.1\,\text{s}$
Relative angles update rate	$\Delta t_\beta = 0.1\,\text{s}$ and $\Delta t_\varepsilon = 0.1\,\text{s}$
Relative range update rate	$\Delta t_r = 5\,\text{s}$
Navigation Filter (BRPF)	$N = 900$, $\theta_{eff} = 0.5$
Master Filter	
Process noise in position (std)	$[80, 80, 10]\ \text{ms}^{-1}$
Process noise in velocity (std)	$[2, 2, 1]\ \text{ms}^{-1}$
Process noise in attitude (std)	$\left[10^{-3}, 10^{-3}, 10^{-3}\right]\ \text{rad s}^{-2}$
Accelerometer biases (std)	$\left[10^{-6}, 10^{-6}, 10^{-6}\right]\ \text{ms}^{-2}$
Gyrometer biases (std)	$\left[10^{-8}, 10^{-8}, 10^{-8}\right]\ \text{rad s}^{-2}$

refine the BRPF estimate while reducing its computational load. Furthermore, the approach remains valid for coarse relative measurements.

The pessimism ratio reduction is of strong interest for the formation flying application. Figure 5.16 plots the formation estimation uncertainties at the final time-step. The solid thin ellipses represent the estimation uncertain zone (3σ) for the BRPF alone, while the the dotted ellipses represent the Master Filter's estimated uncertainty (3σ). In the studied case, the relative measurements are very coarse, which result in highly uncertain relative estimates. Figure 5.17 shows the $(N_a - 1)$ EKF outputs from the point of view of agent 1 (with $N_a = 4$). Despite these uncertainties, the absolute navigation estimation performed by the Master Filter appears to be significantly refined.

Table 5.7: Averaged results for the single Box Regularised Particle Filter and the federated architecture on all scenarios, at final time-step.

	Single BRPF	Federated architecture
Non-convergence rate (%)	0	0
RMSE (position, m)	126	108
RMSE (velocity, ms^{-1})	4.8	4.1
RMSE (heading, degree)	0.38	0.38
RMSE (pitch, degree)	0.41	0.41
RMSE (roll, degree)	0.41	0.41
Estimated st.d (position, m)	495	161
Estimated st.d (velocity, ms^{-1})	11.4	4.12
Estimated st.d (heading, degree)	0.77	0.30
Estimated st.d (pitch, degree)	0.77	0.30
Estimated st.d (roll, degree)	0.68	0.30
Pessimism (position)	3.93	1.48
Pessimism (velocity)	2.37	0.99
Pessimism (attitude)	2.02	0.73
Time (ms)	24	14

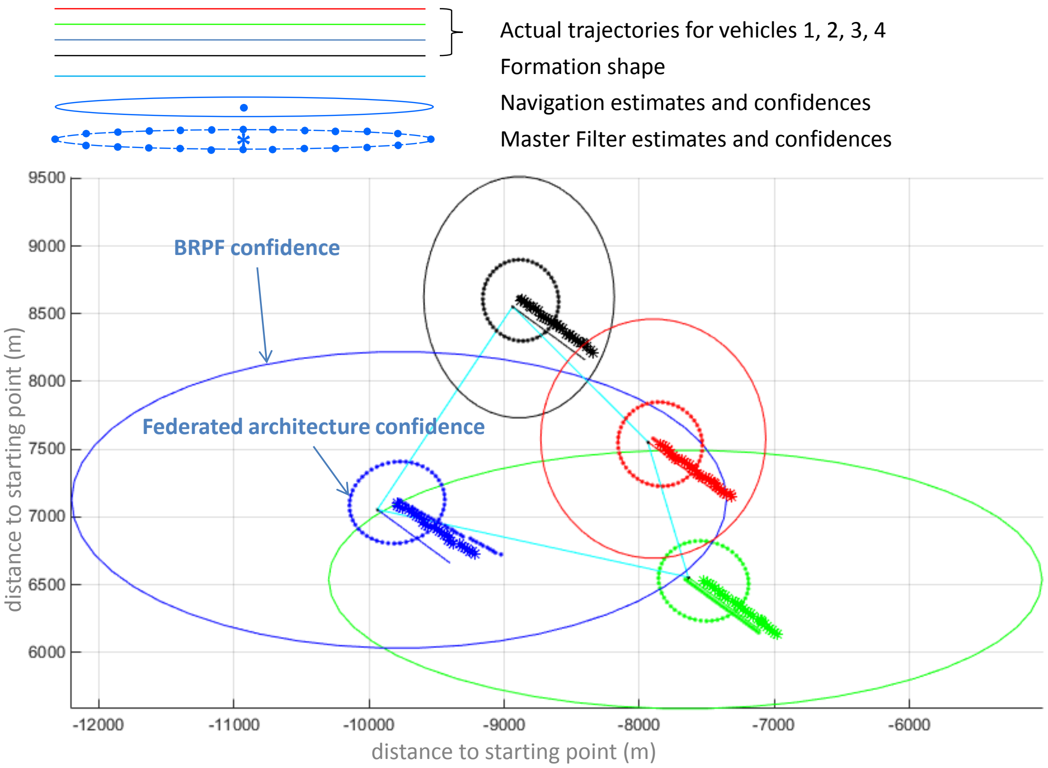


Figure 5.16: Confidence estimation with the federated architecture (Master Filter output) and with BRPF only, for vehicles 1, 2, 3 and 4.

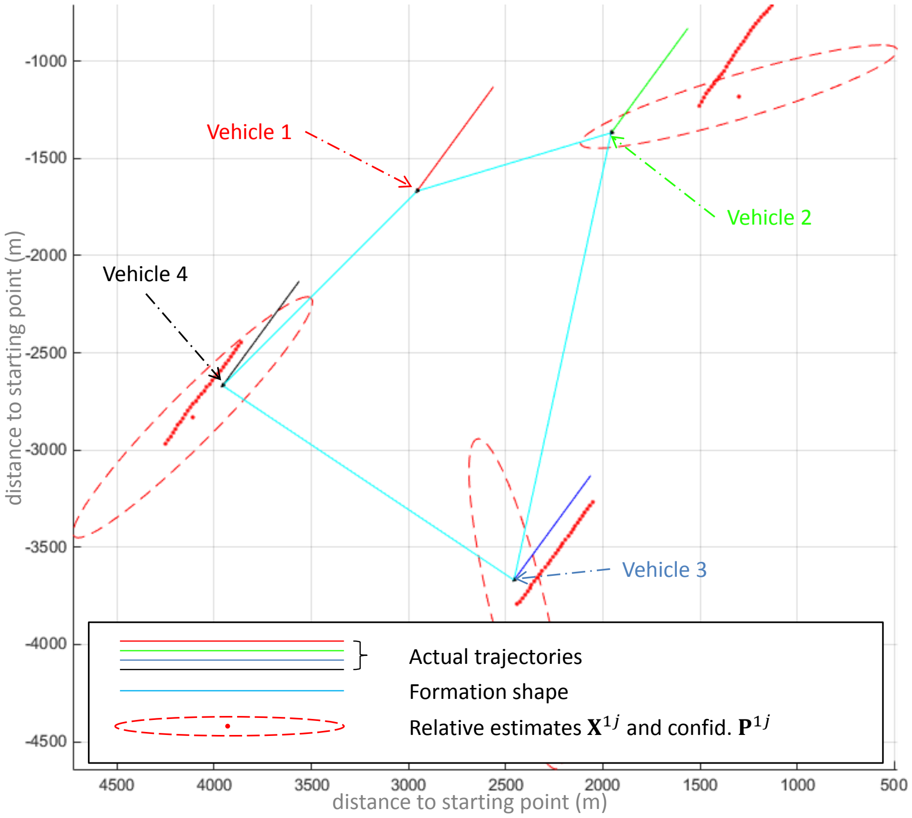


Figure 5.17: Relative estimates and associated uncertainties for vehicle 1 toward vehicles 2, 3 and 4. This figure illustrates how inaccurate the relative estimates can be (relative Extended Kalman Filters outputs).

5.4 SUMMARY

This chapter showed that BPF-like algorithms can be integrated to federated architectures, provided that the estimated posterior conditional density tends to a unimodal function (research question 5). It was first recalled that Multinomial Resampling theoretically yields an asymptotic unimodal posterior density. However, this does not appear in practice, since the number of resampling steps is not sufficient during the considered scenarios (about $\mathcal{O}(10)$). Furthermore, it was shown in Chapter 4 that Multinomial Resampling may threaten the filter's robustness. It was then stated that Guaranteed Resampling tends to keep all possible modes, which may yield an asymptotic multimodal posterior density. In practice, it yields less multimodality than Multinomial Resampling, due to its conservatism and to the low number of resampling steps. Finally, regularisation was shown to act as a smoothing operation which theoretically transforms the posterior density into a unimodal function. This is confirmed in practice for both resampling approaches. To conclude, BRPF can be implemented in federated architecture and benefits from regularisation.

BRPF was then implemented in several federated architectures, where it tackled TAN measurements while dedicated filters dealt with Gaussian measurements. First, such an architecture yields a more accurate estimation from Gaussian measurements, which would suffer from the wrapping effect if only tackled in the BRPF. In addition, the conservatism is significantly reduced. Then, federated architectures make it possible to marginalise the state vector estimated by the BRPF to focus on the state variable involved in measurement ambiguities (e.g. TAN measurements). This results in a significantly lower computational load than a BRPF estimating a full dimension state. Finally, this scheme can be extended to distributed federated architecture for collaborative estimation in a fleet. The architecture remains robust, accurate and computationally efficient for non-linear and coarse Gaussian measurements coming in complement to TAN measurements.

6

TRAJECTORY OPTIMISATION UNDER STATE UNCERTAINTY AND CONSTRAINTS: THE BOX PARTICLE CONTROL

Chapter 2 introduced the approach of Chance Constrained predictive control in the presence of state uncertainty. The problem consists of finding the optimal sequence of control inputs that minimises a cost function while keeping the probability of state constraint violation below a desired threshold. At each time-step k, a constrained optimisation is performed along a predicted state trajectory (e.g. going from k to $k+n$ for receding horizon control with $n > 1$, or going from k to k_f for optimal control). To perform the Chance Constrained optimisation, a failure probability estimator must be designed. In the context of non-Gaussian densities and non-convex constraints, possible approaches are the sampling methods (Pagnoncelli, [PAS09]), which consist of propagating a cloud of weighted particles. However, sampling methods (e.g. Particle Control by Blackmore [Bla+10]), suffer from the fact that a large number of particle samples (theoretically infinite) is needed to meet the constraints in a guaranteed way. Reducing the number of particles leads to an uncontrolled loss of accuracy in the constraints evaluation and may cause unforeseen constraint violation. In addition, the point-wise particle description of the state pdf leads to non-differentiable constraints. Blackmore [Bla+10] has thus proposed a Mixed-Integer-Linear-Programming formulation of the problem, which limits it to the linear case.

This chapter introduces a Chance Constraint estimator based on box particle propagation, on the basis of the framework introduced in Chapter 4. The main differences between a Box Particle Filter and a box particle Chance Constraint estimation are that the state density is propagated along a trajectory of several time-steps and that no measurements are introduced during the predicted trajectory. The focus will be on the failure probability estimation problem, and the Chance Constrained optimisation for trajectory planning. The integration of the trajectory planning approach in a Receding Horizon Control or an optimal control scheme is out of the scope of this work and can be considered as future research. The chapter is organised as follows:

- In Section 6.1, the propagation of box particles on several time-steps is discussed. A minimal inclusion propagation formulation is introduced for linear time-varying dynamical models.
- In Section 6.2, the box kernel based Chance Constraint formulation is introduced. Numerical comparisons are done with the point-wise particle based chance constraint existing approaches.
- In Section 6.3, the box kernel Chance Constraint estimator integration in a differentiable optimisation is studied, leading to the Box Particle Control method (BPC). The main point is to ensure that the constraints are differentiable with respect to the sequence of future control inputs, which constitute the optimisation variables. Numerical comparisons are done against the point-wise particle based chance constraint approach (Blackmore [Bla+10])), and the ability of BPC to handle non-linear models is experimentally demonstrated.

6.1 BOX PARTICLE DENSITY PROPAGATION ON A TRAJECTORY

Chapter 4 introduced the box kernel mixture formulation of a state density:

$$p(\mathbf{x}_k|\mathbf{M}_k) = \sum_{i=1}^{N} w_k^i \pi_k^i(\mathbf{x}_k) \mathbf{1}_{[\mathbf{x}_k^i]}(\mathbf{x}_k) \tag{6.1}$$

where $\mathbf{x}_k \in \mathbb{R}^d$ is the state at time-step k, $\mathbf{M}_k$ is the measurements' trajectory until k, and $\pi_k^i : \mathbb{R}^d \rightarrow \mathbb{R}$ a kernel that sums to unity on the box domain $[\mathbf{x}_k^i]$.

In practice, each box kernel π_k^i can be approached using a reference kernel π, as stated in Section 4.1.2:

$$\pi_k^i(\mathbf{x}) \approx \frac{1}{|[\mathbf{x}_k^i]|} \pi \left(\frac{\mathbf{x} - \underline{\mathbf{x}}_k^i}{\overline{\mathbf{x}}_k^i - \underline{\mathbf{x}}_k^i} \right) \mathbf{1}_{[\mathbf{x}_k^i]} \quad \forall (i,k) \tag{6.2}$$

where $\pi : \mathbb{R}^d \rightarrow \mathbb{R}^+$ integrates to unity on $[0,1]$, and $[\underline{\mathbf{x}}_k^i, \overline{\mathbf{x}}_k^i]$ represents the bounds of box i. Under this assumption, π_k^i can be described via its box support $[\mathbf{x}_k^i]$.

The propagation of probability density (6.1) along a trajectory from time-step k to time-step $k+n$ can be done using n times the interval propagation (2.43) introduced in Chapter 2. However, even if the one-step-ahead inclusion function is minimal, the propagation of each box $[\mathbf{x}_k^i]$ on several time-steps may not be minimal due to an accumulation of *wrapping effects* (see Section 2.1.5, equation (2.45) on inclusion functions composition).

6.1.1 Problem formulation and example

In Chapters 4 and 5, inclusion functions were used for one-step-ahead predictions. Assuming that some interval contraction regularly occurs thanks to the measurement's information, it was sufficient to derive an inclusion function from the recursive state propagation equation (2.2). Inclusion functions were then assumed to be minimal or near-minimal. However, in the context of trajectory planning, inclusion functions are used for a whole trajectory, which yields a number of propagation steps.

From Theorem 1, Section 2.1.5, inclusion functions are minimal if they do not have any formal repetitions of the input variables (dependency effect) and if the function is continuous. In addition, the composition of two inclusion functions is not equal to the inclusion function of their composition (see equation (2.45)).

However, when an inclusion propagation is performed recursively (e.g. k time-steps from time 0), the inclusion function is composed n times with itself, which may introduce an accumulation of the wrapping effect. Indeed, consider Moor's example (e.g. see Nickel [Nic86]) which consists of a two dimensional linear dynamical model defined by $\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1}$ with:

$$\mathbf{F} = \begin{bmatrix} 1 & u \\ -u & 1 \end{bmatrix} \tag{6.3}$$

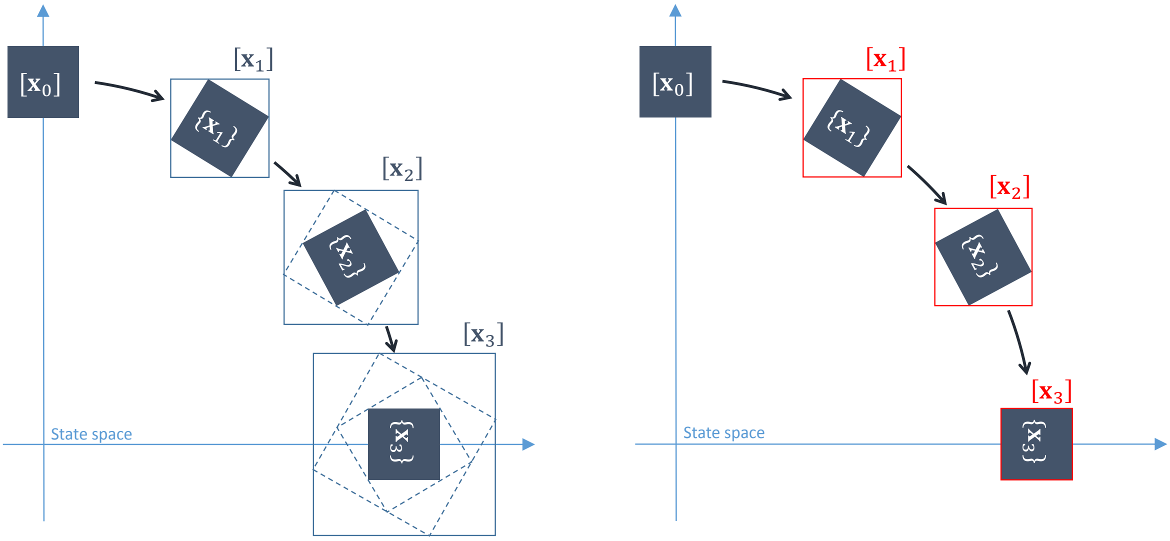
where $u > 0$. Figure 6.1 (a) illustrates the successive boxes (blue boxes $[\mathbf{x}_k]$) obtained with the recursive inclusion function defined by:

$$[\mathbf{x}_k] = \left[\left\{ \mathbf{F}\mathbf{x} \mid \mathbf{x} \in [\mathbf{x}_{k-1}] \right\} \right] \tag{6.4}$$

The filled blue areas $\{\mathbf{x}_k\}$ are the actual propagation of the initial box $[\mathbf{x}_0]$ defined by:

$$\{\mathbf{x}_k\} = \{\mathbf{Fx} \mid \mathbf{x} \in \{\mathbf{x}_{k-1}\}\} \tag{6.5}$$

Although it is minimal for the first propagation step, the inclusion function defined in (6.4) is not minimal for $k \geq 2$ because of wrapping effect accumulation. The output box $[\mathbf{x}_k]$ can rapidly become excessively large, which yields a huge conservatism. Figure 6.1 (b) illustrates the successive boxes (red boxes) that would be obtained with a minimal inclusion function. The next section proposes the formulation of a minimal inclusion propagation for linear time-varying systems.



(a) Recursive inclusion propagation (b) Minimal inclusion propagation

Figure 6.1: Moor's example (e.g. see Nickel [Nic86]): output $[\mathbf{x}_k]$ (blue thin boxes) of the non-minimal inclusion function obtained with the naive wrapping of the recursive propagation (a), and minimal inclusion function output (red boxes) (b). The propagated set $\{\mathbf{x}_k\}$ (blue filled areas) is obtained by exact set propagation of the initial box $[\mathbf{x}_0]$.

6.1.2 Minimal inclusion trajectory propagation

Let $[\mathbf{x}_{k-1}] \in \mathbb{R}^d$ be the box containing the state $\mathbf{x}_{k-1}$ and $f : \mathbb{R}^d \times \mathbb{R}^{d_c} \rightarrow \mathbb{R}^d$ the discrete dynamical model. Then, by applying (2.35), the one-step-ahead prediction equation of an interval-based estimation algorithm can be defined:

$$\left[\mathbf{x}_{k|k-1}\right] = [f]\left([\mathbf{x}_{k-1}], \mathbf{u}_k\right) + [\mathbf{w}_k] \tag{6.6}$$

with

$$[f]\left([\mathbf{x}_{k-1}], \mathbf{u}_k\right) = \left[\left\{f\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) \mid \mathbf{x}_{k-1} \in [\mathbf{x}_{k-1}]\right\}\right] \tag{6.7}$$

For a prediction on $n > 1$ time-steps, the dynamical model f has to be composed n times with itself as follows:

$$\mathbf{x}_{k+n|k} = f(...(f(\mathbf{x}_k, \mathbf{u}_{k+1}) + \mathbf{w}_k, ...), \mathbf{u}_{k+n}) + \mathbf{w}_{k+n} \tag{6.8}$$

By applying property (2.45), this may result in an increasing wrapping effect.

For nonlinear cases, there is no general solution except carefully designing the one-step-ahead prediction (6.6) using Theorem 1, or deriving an analytic solution of the dynamics differential equation denoted $[F]([\mathbf{x}_0, \mathbf{U}_k])$ in what follows and defined by:

$$[F]([\mathbf{x}_k], \mathbf{U}_{k:k+n}, [\mathbf{W}_{k:k+n}]) = \left[\left\{f(...f(\mathbf{x}, \mathbf{u}_k) + \mathbf{w}_k, ..., \mathbf{u}_{k+n}) + \mathbf{w}_{k+n} \;\middle|\; \mathbf{x} \in [\mathbf{x}_k], \mathbf{w}_i \in [\mathbf{w}_i] \;\forall i \in [k, k+n]\right\}\right] \tag{6.9}$$

where $\mathbf{U}_{k:k+n} = [\mathbf{u}_k^T, ..., \mathbf{u}_{k+n}^T]^T \in \mathbb{R}^{dk}$ is the sequence of control inputs, and $[\mathbf{W}_{k:k+n}] = [\mathbf{w}_k^i] \times ... \times [\mathbf{w}_{k+n}^i] \in \mathbb{IR}^{dk}$ the sequence of process noise box supports.

For time-varying linear systems, an analytic formulation obtained by recursion can be derived. Let a dynamical model be defined by:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_{k+1} + \mathbf{w}_{k+1} \tag{6.10}$$

The n times composition of (6.10) to propagate state $\mathbf{x}_k$ to time-step $k+n$ ($n > 0$) is equivalent to:

$$\mathbf{x}_{k+n|k} = \sum_{j=k}^{k+n-1} \left(\prod_{l=j+1}^{k+n-1} \mathbf{F}_l\right) \left(\mathbf{B}_j \mathbf{u}_{j+1} + \mathbf{w}_{j+1}\right) + \left(\prod_{l=k}^{k+n-1} \mathbf{F}_l\right) \mathbf{x}_k \tag{6.11}$$

This formulation has the great advantage of directly mapping $\mathbf{x}_k$ to $\mathbf{x}_{k+n}$ in an analytic way, which helps avoiding multiple one-step-ahead compositions. It can now be wrapped into intervals:

$$\left[\mathbf{x}_{k+n|k}\right] = \sum_{j=k}^{k+n-1} \left(\prod_{l=j+1}^{k+n-1} \mathbf{F}_l\right) \left(\mathbf{B}_j \mathbf{u}_{j+1} + [\mathbf{w}_{j+1}]\right) + \left(\prod_{l=k}^{k+n-1} \mathbf{F}_l\right) [\mathbf{x}_k] \tag{6.12}$$

which is the minimal inclusion function for a multiple-steps prediction of the model (6.10), by applying Theorem 1. It can be applied to locally linearised non-linear systems. It is of strong interest in state estimation when several consecutive prediction steps occur without receiving any measurement (pure prediction). It is also worth using in predictive control, while predicting a future trajectory. In the case of a linear system such that $\mathbf{F}_k = \mathbf{F}$ and $\mathbf{B}_k = \mathbf{B}$ $\forall k$, then (6.12) becomes:

$$\left[\mathbf{x}_{k+n|k}\right] = \sum_{j=k}^{k+n-1} \left(\mathbf{F}^{n-1} \left(\mathbf{B}\mathbf{u}_{j+1} + [\mathbf{w}_{j+1}]\right)\right) + \mathbf{F}^n [\mathbf{x}_k] \tag{6.13}$$

6.2 BOX PARTICLE CHANCE CONSTRAINT ESTIMATION

This section aims to evaluate the probability of constraint violation (failure probability) via the propagation of a box particle cloud of bounded kernels. For the sake of brevity, the initial time-step of the predicted trajectory is denoted $k = 0$ and the prediction horizon is K.

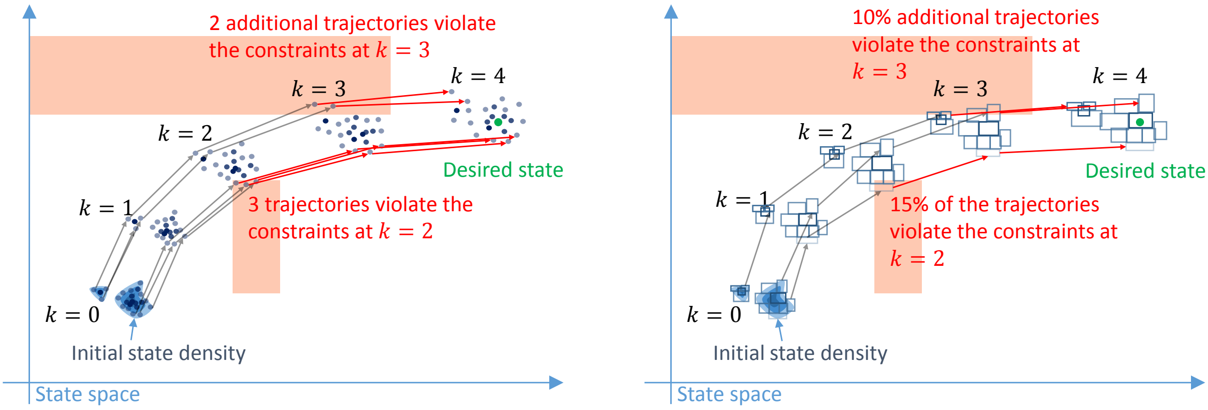
6.2.1 Principle and example

Chapter 2 introduced the Particle Control (Blackmore [Bla+10]) which relies on a sample-based Chance Constraint calculation. Consider the example illustrated in Figure 6.2 (a). The initial state density $p(\mathbf{x}_0)$ is approximated as a mixture of N weighted Dirac functions whose weights are denoted w_0^i with $\sum_i w_0^i = 1$.

The cloud of N particles is propagated for each time-step, yielding N particle trajectories. Since no measurement input is considered, the weights w_k^i remain the same as initial weights. It can then be considered that each trajectory is associated with a timeless weight $w^i \triangleq w_0^i$. The number of trajectories violating the constraints for all time-steps is then computed. For example, three trajectories violate the constraints at time-step $k = 2$ and two additional trajectories violate the constraints at time-step $k = 3$. Consider that the weights w^i of those five trajectories are $0.08, 0.02, 0.05, 0.03, 0.07$ respectively. The estimated failure probability is the sum of the weights of the trajectories violating the constraints: $\widehat{p} = 0.25$.

This chapter introduces a box particle kernel based Chance Constraint estimation. As illustrated in Figure 6.2 (b), it relies on a continuous integration of the propagated state box densities on the constraints domain. The initial state density $p(\mathbf{x}_0)$ is approximated by a mixture of N weighted bounded kernels whose weights are denoted w_0^i with $\sum_i w_0^i = 1$. Each box particle is then propagated, leading to N box trajectories $[\mathbf{x}_0^i] \times ... \times [\mathbf{x}_K^i]$ until a final time-step K. At each time-step, each box kernel π_k^i is integrated on the constraints domain (a set of forbidden states). Note that for uniform box kernels, the integration corresponds to the ratio of the area of the the intersection of box particle i with the constraints domain, by the box particle area. The results of this integration are then weighted by weights w^i and summed. For example, consider that one box particle intersects the constraint set at time-step $k = 2$. Consider that the integration of its box kernel on the constraint domain yields the value of 0.75, weighted by a weight of 0.2. Consider that two additional boxes violate the constraints at time-step $k = 3$, yielding a kernel integration of 0.6 and 0.4 respectively. Their weights are both equal to 0.1. Finally, the weighted sum of the box kernel integrations is $\widehat{p} = 0.25$.

The box kernel approach has the advantage of being continuous, thanks to the smooth state density approximation by the bounded kernels mixture. It can then be used with less sampling parameters than particle based chance constraint approaches. Under some hypotheses that will be discussed in Section 6.3, it can be implemented in differentiable optimisation schemes, which is not possible with the sample-based chance constraint.



(a) Particle based chance-constraint estimation (b) Box particle based chance-constraint estimation

Figure 6.2: The particle based chance-constraint estimation consists of the weighted sum of indicator functions associated with each particle-trajectory violating the constraints (a). The box particle chance constraint evaluation relies on the integration of box kernels on the constraints' domain (b).

6.2.2 Theoretical description

Consider that the initial time-step of the trajectory prediction is $k = 0$. Note the predicted state and predicted box particles as $\mathbf{x}_k \triangleq \mathbf{x}_{0+k|0}$ and $[\mathbf{x}_k^i] \triangleq [\mathbf{x}_{0+k|0}^i]$. Let K denote the prediction horizon.

The use of a mixture of bounded kernels implies that the failure probability has to be computed from a continuous sum of functions, which makes it *a priori* intractable in formulation (2.80). This section introduces a derivation of the failure probability which makes the problem tractable. Intuitively, it consists of evaluating the proportion of possible trajectories which fall out of the feasible set, as illustrated in Figure 6.2.

Proposition 8. *Let us consider the state density* (6.1) *at time-step* 0. *Considering a feasible domain* $\mathcal{F} \subset \mathbb{R}^{d(K+1)}$ *for the state trajectory* $\mathbf{X}$ *representing the set of admissible trajectories w.r.t.* $K+1$ *constraints* $g_i(\boldsymbol{x}_i) \leq 0$*:*

$$\mathcal{F} \triangleq \left\{ \mathbf{X}_k = [\boldsymbol{x}_0^T, ..., \boldsymbol{x}_K^T]^T \;\middle|\; \begin{matrix} g_0(\boldsymbol{x}_0) \leq 0 \\ \vdots \\ g_K(\boldsymbol{x}_K) \leq 0 \end{matrix} \right\} \tag{6.14}$$

Then, the probability of $\mathbf{X}$ *violating the constraints (failure probability) can be approached by:*

$$\begin{gathered} \widehat{P}(\mathbf{X} \notin \mathcal{F}) = \sum_{i=1}^{N} w_0^i \int_{[\boldsymbol{x}_0^i] \cap \mathcal{C}_0} \pi_0^i(\boldsymbol{x}_0) \mathrm{d}\boldsymbol{x}_0 \\ + \sum_{k=1}^{K} \sum_{i=1}^{N} w_k^i \left(\int_{[\boldsymbol{x}_k^i] \cap \mathcal{C}_k} \pi_k^i(\boldsymbol{x}_k) \mathrm{d}\boldsymbol{x}_k - \int_{\{[f]([\boldsymbol{x}_{k-1}^i] \cap \mathcal{C}_{k-1}, \boldsymbol{u}_k) + [\boldsymbol{w}_k]\} \cap \mathcal{C}_k} \pi_k^i(\boldsymbol{x}_k) \mathrm{d}\boldsymbol{x}_k \right) \end{gathered} \tag{6.15}$$

where $\mathcal{C}_i \in \mathbb{R}^d$ *represents the forbidden set of states at time-step i.*

$$\mathcal{C}_i \triangleq \{ \boldsymbol{x} \mid g_i(\boldsymbol{x}) > 0 \} \tag{6.16}$$

Proof. In a continuous state and discrete time scheme, the theoretical failure probability $P(\mathbf{X} \notin \mathcal{F})$ (2.80) can be interpreted as the sum, for each time-step k, of the conditional probability of current state $\mathbf{x}_k$ violating the constraints ($\mathbf{x}_k \in \mathcal{C}_k$), given that it has always satisfied them:

$$P(\mathbf{X} \notin \mathcal{F}) = \sum_{k=1}^{K} P\left(\mathbf{x}_k \in \mathcal{C}_k \mid \mathbf{X}_{k-1} \in \mathcal{F}_{k-1}\right) \tag{6.17}$$

where $\mathbf{X}_{k-1} = [\mathbf{x}_0^T, ..., \mathbf{x}_{k-1}^T]^T$ is the trajectory from initial time-step 0 to time-step $k-1 < K$, and $\mathcal{F}_{k-1}$ is the feasible domain along this trajectory, i.e.:

$$\mathcal{F}_{k-1} \triangleq \left\{ \mathbf{X}_{k-1} = [\mathbf{x}_0^T, ..., \mathbf{x}_{k-1}^T]^T \;\middle|\; \begin{matrix} g_0(\mathbf{x}_0) \leq 0 \\ \vdots \\ g_{k-1}(\mathbf{x}_{k-1}) \leq 0 \end{matrix} \right\} \tag{6.18}$$

The instantaneous failure probability can be defined as a probability measure of the instantaneous propagated state densities $p(\mathbf{x}_k)$ on a domain $\mathcal{I}_k$:

$$P\left(\mathbf{x}_k \in \mathcal{C}_k \mid \mathbf{X}_{k-1} \in \mathcal{F}_{k-1}\right) = \int_{\mathcal{I}_k} p(\mathbf{x}_k)\mathrm{d}\mathbf{x}_k \tag{6.19}$$

where the integration domain is:

$$\mathcal{I}_k = \left\{ \mathbf{x}_k \in \mathcal{C}_k \;\middle|\; \begin{array}{ll} \mathbf{x}_j \notin \mathcal{C}_j & \forall j < k \\ \mathbf{x}_j = f^j(\mathbf{x}_0, \mathbf{U}_j, \mathbf{W}_j) & \forall j \leq k \end{array} \right\} \tag{6.20}$$

where $f^{i:j}(\mathbf{x}_i, \mathbf{U}_{i:j}, \mathbf{W}_{i:j}) \triangleq f(...f(\mathbf{x}_i, \mathbf{u}_{i+1}) + \mathbf{w}_{i+1}, ..., \mathbf{u}_j) + \mathbf{w}_j$ is the state trajectory from $\mathbf{x}_i$ to $\mathbf{x}_j$, with the control sequence $\mathbf{U}_{i+1:j} = [\mathbf{u}_{i+1}^T, ..., \mathbf{u}_j^T]^T$ and the process noise realisations $\mathbf{W}_{i+1:j} = [\mathbf{w}_{i+1}^T, ..., \mathbf{w}_j^T]^T$ with $i < j$. Domain $\mathcal{I}$ is equal to:

$$\mathcal{I}_k = \{\mathbf{x}_k \in \mathcal{C}_k\} \setminus \mathcal{X}_k \tag{6.21}$$

with $\mathcal{X}_k = \left\{ \mathbf{x}_k \mid \bigvee_{j<k} \mathbf{x}_j \in \mathcal{C}_j, \mathbf{x}_k = f^{(j:k)}(\mathbf{x}_j, \mathbf{U}_{j:k}, \mathbf{W}_{j:k}) \right\}$, in other words, $\mathcal{I}_k$ represents the set of all states $\mathbf{x}_k$ that violate the current constraints minus the set $\mathcal{X}_k$ of states $\mathbf{x}_k$ that had not previously satisfied the constraints (at least once). This yields the following failure probability formulation:

$$P(\mathbf{X} \notin \mathcal{F}) = \int_{\mathbf{x}_0 \in \mathcal{C}_0} p(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0 + \sum_{k=1}^{K} \left(\int_{\mathbf{x}_k \in \mathcal{C}_k} p(\mathbf{x}_k)\mathrm{d}\mathbf{x}_k - \int_{\mathbf{x}_k \in \mathcal{X}_k} p(\mathbf{x}_k)\mathrm{d}\mathbf{x}_k \right) \tag{6.22}$$

which represents the probability measure of initial states violating the initial constraints $g_0(\mathbf{x}_0) \leq 0 \Leftrightarrow \mathbf{x}_0 \notin \mathcal{C}_0$, plus the incremental probability measures of additional trajectories violating the constraints $g_k(\mathbf{x}_k) \leq 0 \Leftrightarrow \mathbf{x}_k \notin \mathcal{C}_k$ at time-steps $k \in [1, K]$.

Now, using the proposed box density approximation (6.1) for $p(\mathbf{x}_k)$, and by distributivity of the integration, an estimation of the failure probability is obtained:

$$P(\mathbf{X} \notin \mathcal{F}) = \sum_{i=1}^{N} w_0^i \int_{[\mathbf{x}_0^i] \cap \mathcal{C}_0} \pi_0^i(\mathbf{x}_0)\mathrm{d}\mathbf{x}_0 + \sum_{k=1}^{K} \sum_{i=1}^{N} w_k^i \left(\int_{[\mathbf{x}_k^i] \cap \mathcal{C}_k} \pi_k^i(\mathbf{x}_k)\mathrm{d}\mathbf{x}_k - \int_{[\mathbf{x}_k^i] \cap \mathcal{X}_k} \pi_k^i(\mathbf{x}_k)\mathrm{d}\mathbf{x}_k \right) \tag{6.23}$$

with $w_k^i = w_0^i \; \forall k$. However, this formulation is still difficult to evaluate because of the integration domain $\mathcal{X}_k$. Thus, make the Markovian assumption that $\mathcal{X}_k$ is equivalent to the set of states that have violated the constraints at the previous time-step:

$$\mathcal{X}_k \approx \{\mathbf{x}_k \mid \mathbf{x}_{k-1} \in \mathcal{C}_{k-1}, \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)\} \tag{6.24}$$

Therefore, the integration domain $[\mathbf{x}_k^i] \cap \mathcal{X}_k$ can be approached by:

$$[\mathbf{x}_k^i] \cap \mathcal{X}_k \approx \{[f]([\mathbf{x}_{k-1}^i] \cap \mathcal{C}_{k-1}, \mathbf{u}_k) + [\mathbf{w}_k]\} \cap \mathcal{C}_k \tag{6.25}$$

which yields (6.15). ☐

Remark 1:

The integral of a known kernel $\pi_k^i : \mathbb{R}^d \to \mathbb{R}$ on a box domain $[\mathbf{d}] = [\underline{\mathbf{d}}, \overline{\mathbf{d}}] \in \mathbb{IR}^d$ is equal to the difference of the Cumulative Density Function (CDF) $\Pi_k^i : \mathbb{R}^d \to \mathbb{R}$ of π_k^i on the domain boundaries:

$$\int_{\mathcal{D}} \pi_k^i(\mathbf{x}_k)\mathrm{d}\mathbf{x}_k = \Pi_k^i(\overline{\mathbf{d}}) - \Pi_k^i(\underline{\mathbf{d}}) \tag{6.26}$$

This formulation has the advantage of being computationally simple. A box approximation of the integration domain $[\mathbf{x}_k^i] \cap \mathcal{C}_k$ yields an outer-approximation of the integral $\int_{[\mathbf{x}_k^i] \cap \mathcal{C}_k} \pi_k^i(\mathbf{x}_k)\mathrm{d}\mathbf{x}_k$. The second domain formulation (6.25) is also a box approximation. Under those box integration domain approximations and by the use of (6.26), the theoretical complexity of the NK kernel integrations in equation (6.15) may be neglected.

Remark 2:

If box kernels are uniform functions, i.e.:

$$\pi_k^i(\mathbf{x}) = \frac{1}{|[\mathbf{x}_k^i]|}\mathbf{1}_{[\mathbf{x}_k^i]}(\mathbf{x}) \tag{6.27}$$

Then the box particle failure probability estimation (6.15) becomes:

$$\widehat{P}(\mathbf{X} \notin \mathcal{F}) = \sum_{i=1}^{N} w_0^i \frac{|[\mathbf{x}_0^i] \cap \mathcal{C}_0|}{|[\mathbf{x}_0^i]|} + \sum_{k=1}^{K}\sum_{i=1}^{N} w_k^i \frac{|[\mathbf{x}_k^i] \cap \mathcal{C}_k| - |\{[f]([\mathbf{x}_{k-1}^i] \cap \mathcal{C}_{k-1}, \mathbf{u}_k) + [\mathbf{w}_k]\} \cap \mathcal{C}_k|}{|[\mathbf{x}_k^i]|} \tag{6.28}$$

Remark 3:

A complexity analysis (see Appendix C) of one complete trajectory prediction leads to a number of elementary operations of approximately $2NKd^3$. By a comparison with the Particle Control (Blackmore [Bla+10]), the ratio between BPC (with N_{BPC} box particles) and PC (with N_{PC} particles) computational loads is:

$$r \triangleq \frac{2N_{BPC}Kd^3}{N_{PC}Kd^3} = 2\frac{N_{BPC}}{N_{PC}} \tag{6.29}$$

By ensuring that the support of the state density is included in the union of all the box particles, the BPC is expected to require fewer (box) particles than the PC, typically resulting in $r < 1$.

6.2.3 Numerical results

This section compares the failure probability estimation performance of the particle based estimation (Blackmore [Bla+10]) and the box particle based estimation defined by equation (6.15). Numerical simulations were defined to evaluate the following points:

1. The relative performance of each failure probability estimation approach in terms of probability RMSE;
2. The relative empirical computation time for both approaches, which is expected to be described by (6.29).

For both approaches, the particle and box particle clouds are initialised from a given initial state $\mathbf{x}_0$ associated with a given density $p(\mathbf{x}_0)$. The particle and box particle clouds are propagated via a given dynamical model associated to a given process noise density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. A hundred Monte Carlo runs are performed by using different initial state realisations taken as samples of $p(\mathbf{x}_0)$ and different process noise realisations, taken as truncated Gaussian samples. Therefore, each run yields a different random trajectory $\mathbf{X}^i$. For each run i, the estimated failure probability $\widehat{P}(\mathbf{X}^i \notin \mathcal{F})$ is compared to a reference failure probability value $P(\mathbf{X}^i \notin \mathcal{F})$ computed via a representative particle based estimation consisting of 10^4 trajectories. The number of particles and box particles under consideration does not exceed 10^3 particles, which makes it possible to consider that the set of 10^4 reference trajectories is representative of the actual state density. The failure probability RMSE is defined by:

$$RMSE_{\widehat{P}} = \sqrt{\frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \left(\widehat{P}(\mathbf{X}^i \notin \mathcal{F}) - P(\mathbf{X}^i \notin \mathcal{F})\right)^2} \tag{6.30}$$

where $N_{MC} = 100$ is the number of runs. The initial state density $p(\mathbf{x}_0)$ is chosen so that actual failure probability covers the range of all its possible values between 0 and 1.

The first example consists of a uniform initial density $p(\mathbf{x}_0) = \mathcal{U}_{[\mathbf{x}_0]}$, where $[\mathbf{x}_0] \in \mathbb{IR}^4$ is the initial support of the state $\mathbf{x}$. The state is defined by:

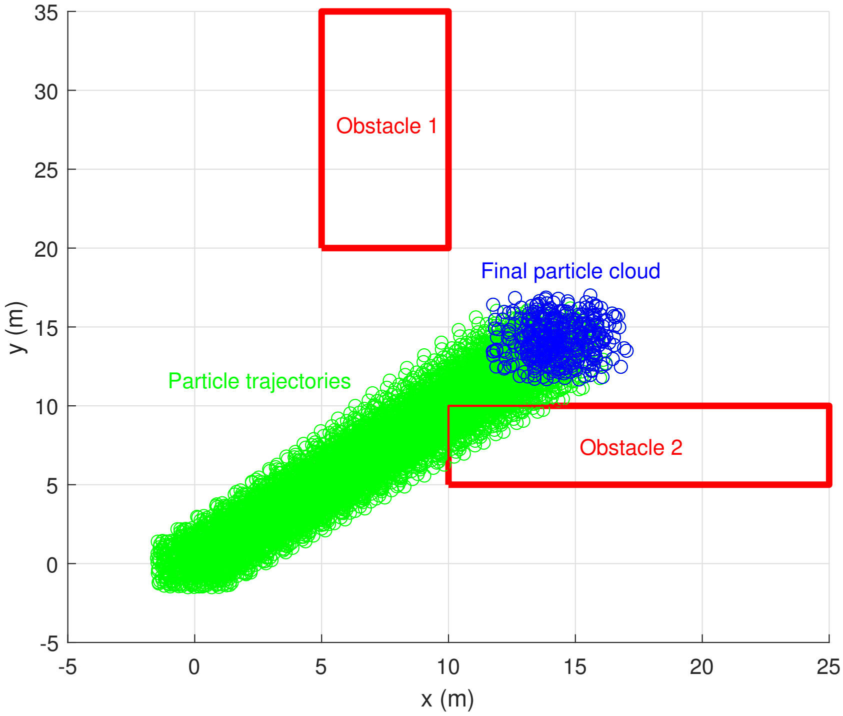
$$\mathbf{x} = [x, y, V_x, V_y]^T \tag{6.31}$$

The dynamical model under consideration is the double integrator (3.1) introduced in Section 3.1. Figure 6.3 shows one run example for the particle based approach (a) and the box particle estimation (b). The initial state density support is $[-1.5, 1.5]$ m $\times[-1.5, 1.5]$ m $\times[1.35, 1.65]$ ms^{-1} $\times[1.35, 1.65]$ ms^{-1}. The initial box particle cloud is defined by a regular paving. Each box particle weight is computed via a histogram method. The control input is assumed to be zero for all runs in the whole trajectory. The control optimisation will be considered in the next section. No process noise is considered in this example.

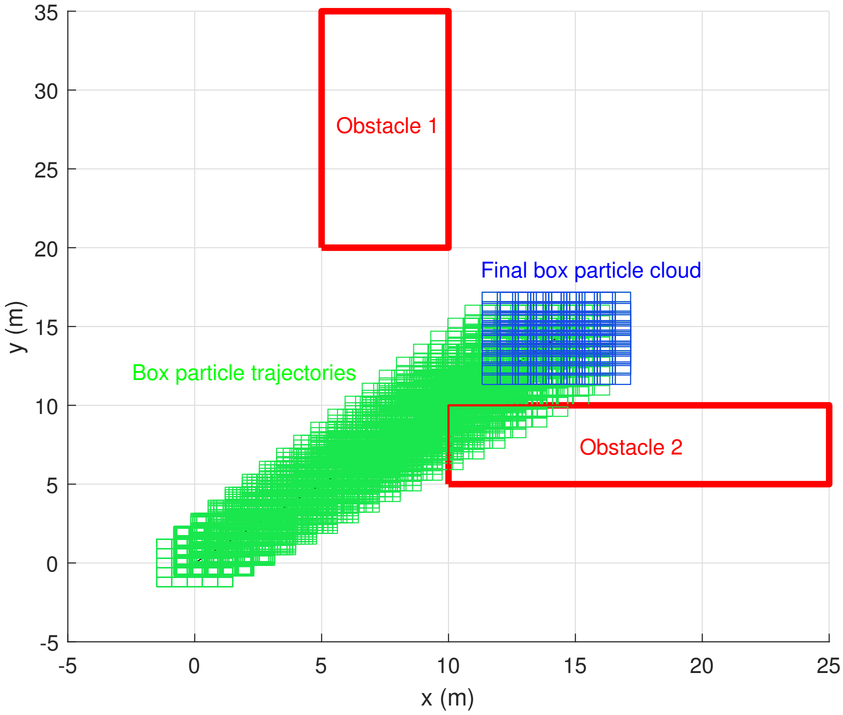
Table 6.1 presents the RMSE obtained for the probability failure estimation $\widehat{P}(\mathbf{X} \notin \mathcal{F})$ for both approaches, for several values of the particle or box particle number N. The box particle based approach outperforms the particle estimation by providing a lower RMSE of about one order of magnitude, even for low values of N (0.0042 for the box particle approach and 0.0858 for the particle based approach, with $N = 15$). This highlights the interest of the kernel density approximation which does not require many kernels to approach a given density. The computation time ratio between box particle estimation and particle estimation is slightly higher that 2 (as theoretically expected for the same number of particles and box particles). This can be explained by the computational load of the kernel integral computation in (6.15) that was neglected in the theoretical complexity analysis.

Another example consists of an initial box particle cloud obtained from the Box Regularised Particle Filter (BRPF) introduced in Chapter 4, in the context of Terrain Aided Navigation (TAN). The guidance problem consists of determining a trajectory that satisfies some keep-out-zones constraints at an early stage of the navigation process, where the state estimation problem is not yet solved. In a real TAN scenario, the forbidden areas could represent urban zones that must not be flown-over.

For each run, the initial estimate is drawn from a uniform density function in the same way as in Chapter 4. After 100 time-steps ($t = 10$ s) of TAN estimation, the BRPF box particle cloud output is taken as the initial box distribution for trajectory propagation, as illustrated in Figure 6.4. The box particle cloud is then propagated along a 50 time-steps trajectory (with a time sample of 1 s, which represents a prediction horizon of 50 s).



(a) Particle based chance-constraint estimation



(b) Box particle based chance-constraint estimation

Figure 6.3: (a) Particle based chance-constraint estimation from particle trajectories (in green, with the final cloud in blue at the prediction horizon). (b) Box particle based chance-constraint estimation from box particle trajectories. Two disjoints set of states are to be avoided, represented by red obstacles areas. For this particular run, the failure probability is $P(\mathbf{X} \notin \mathcal{F}) = 0.412$

Table 6.1: RMSE obtained for the failure probability estimation (100 runs) on the case presented in Figure 6.3, with respect to the number of particles and box particles. Empirical computation times are provided for one trajectory propagation of 20 time-steps.

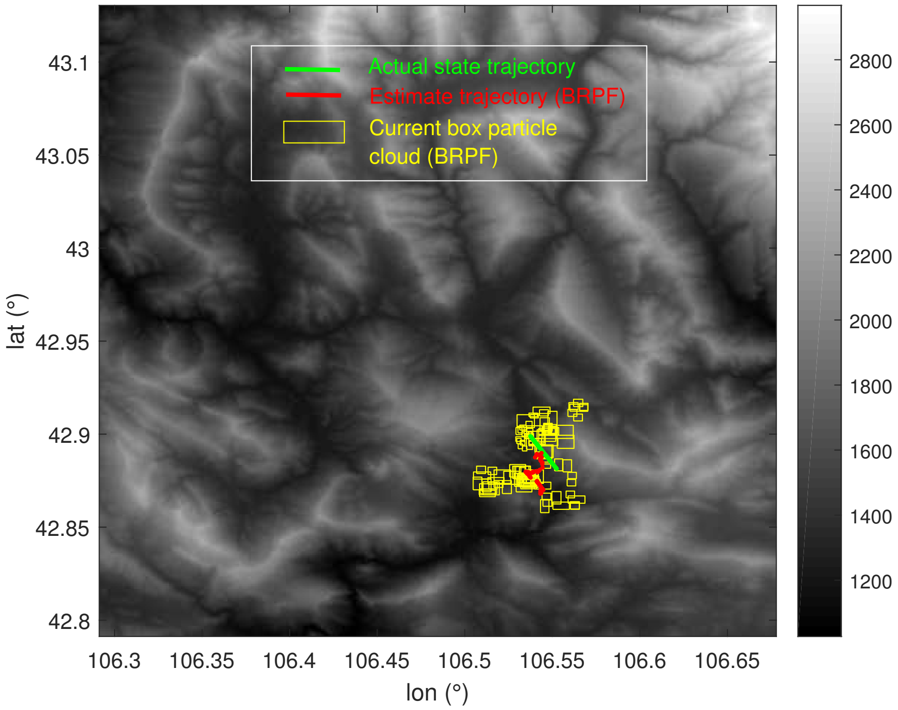
Number of particles or box particles N	15	30	100	250	400	600	900
RMSE for particle based estimation	0.0858	0.0613	0.0404	0.0279	0.0231	0.0184	0.0147
RMSE for box particle estimation	0.0042	0.0037	0.0036	0.0022	0.0022	0.0020	0.0019
Computation time for particle estimation (ms)	2	2	3	4	5	7	9
Computation time for box particle estimation (ms)	7	8	8	10	11	14	19

Table 6.2: RMSE obtained for the failure probability estimation (100 runs) on the TAN case presented in Figure 6.4, with respect to the number of particles and box particles. Empirical computation times are provided for one trajectory propagation of 50 time-steps

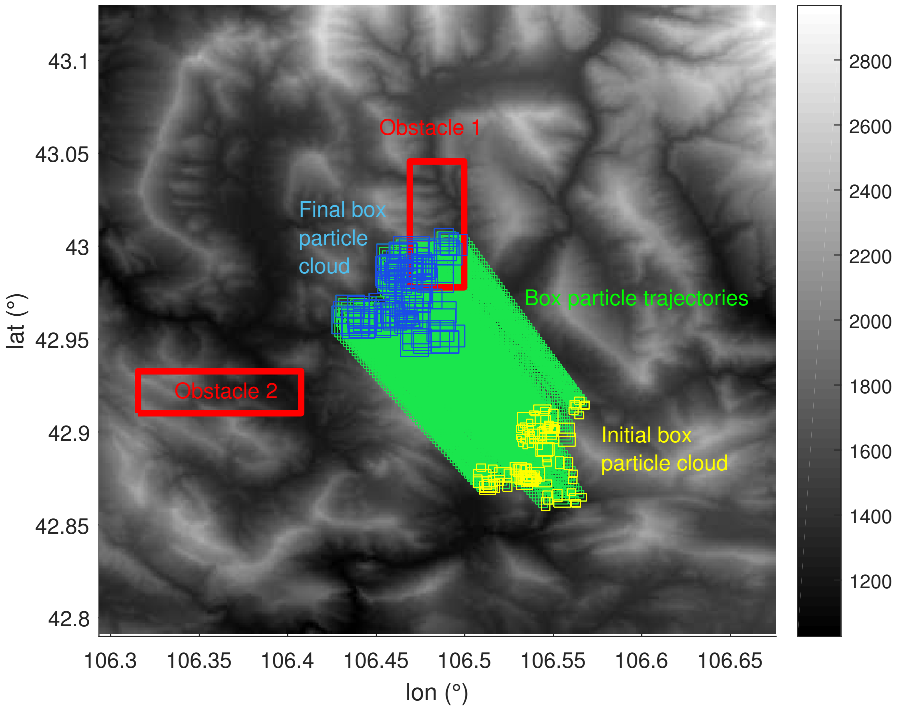
Number of particles or box particles N	100	400	900
RMSE for particle based estimation	0.0138	0.0068	0.0037
RMSE for box particle estimation	0.0019	0.0019	0.0018
Time for particle estimation (ms)	5	12	22
Time for box particle estimation (ms)	15	30	48

Figure 6.4 (b) shows the initial box particle cloud in yellow (corresponding to the BRPF output at current time, $t = 10$ s), the box trajectories in green (corresponding to $K = 50$ successive predicted box particle clouds) and the final box particle cloud at the prediction horizon time in blue. Two fixed forbidden areas are represented in red and referred to as obstacles. The dynamical model is a double integrator with a Gaussian process noise whose standard deviation is $[10, 10]$ m in position and $[0.5, 0.5]$ ms^{-1} in velocity. The control input is assumed to be zero for all runs.

Table 6.2 presents the RMSE obtained for the probability failure estimation $\widehat{P}(\mathbf{X} \notin \mathcal{F})$ for both approaches, for several values of the particle or box particle number N, initialised in the Box Particle Filter at $t = 0$. The box particle based approach again outperforms the particle estimation in terms of RMSE, for all values of N (e.g. 0.0019 for the box particle approach and 0.0138 for the particle based approach, with $N = 100$). For this example, it can be considered that the box particle probability estimation requires a number of box particles about 10 times lower than the number of particles required by the original approach. The theoretical complexity analysis (6.29) suggests that the box particle approach requires 5 times less operations than the particle estimation approach. However, the empirical computation time per trajectory yields a less optimistic ratio (about 1.5 times less), which still suggest that the box particle chance constraint estimation approach yields a computational load reduction with respect to the original method.



(a) BRPF current estimation ($t = 10$ s)



(b) Box particle trajectory prediction from BRPF estimation (from $t = 10$ s to $t = 60$ s), and forbidden areas

Figure 6.4: BRPF current estimation (a) and box trajectories prediction (b) with two areas to bed avoided (referred as obstacles, in red). The box trajectories are in green, and their end-points are in yellow for the initial cloud (current estimation) and in blue for the final cloud (at the prediction horizon).

6.3 BPC INTEGRATION WITHIN A DIFFERENTIABLE OPTIMISATION SCHEME

In the previous sections, a failure probability estimation was defined (6.23) using an approximation of the state density $p(\mathbf{x}_k)$ by a weighted kernels mixture (6.1) bounded by box particles. This section studies the integration of the box chance constraint estimation in a differentiable optimization scheme. The objective is to determine an optimal control sequence trajectory $\mathbf{U}$ which minimises a cost function $J(\mathbf{X}, \mathbf{U})$ while satisfying the chance constraint on the state trajectory $\widehat{P}(\mathbf{X} \notin \mathcal{F}) < \delta$, where δ is specified. It is stated that:

- The cost function expectation $\widehat{J}(\mathbf{X}, \mathbf{U})$ is differentiable with respect to the control sequence if the deterministic cost function J is differentiable;
- The chance constraint $\widehat{P}(\mathbf{X}(\mathbf{U}) \notin \mathcal{F}) - \delta$ is differentiable if box kernels π_k^i are differentiable on their box domain $[\mathbf{x}_k^i]$ and if the boundaries of integration domains in (6.15) are differentiable. To do so, a smooth formulation of min and max functions is introduced via sigmoid functions.

6.3.1 Problem and example

The constrained trajectory optimisation problem can be formalised as follows:

$$\min_{\mathbf{U}} \widehat{J}(\mathbf{X}, \mathbf{U}), \text{ s.t. } \widehat{P}(\mathbf{X}(\mathbf{U}) \notin \mathcal{F}) - \delta \leq 0 \tag{6.32}$$

Note that a desired failure probability $\delta = 0$ is equivalent to an optimal robust control approach. Indeed, the box trajectory propagation ensures that the actual trajectory remains in the union of all boxes.

The cost function $\widehat{J}(\mathbf{X}, \mathbf{U})$ represents the expectation of the original cost function J taken as a random variable. It can be approximated by:

$$\widehat{J}(\mathbf{X}, \mathbf{U}) = \sum_{i=1}^{N} w_0^i J(\mathbf{C}^i, \mathbf{U}) \approx \mathbb{E}\left[J(\mathbf{X}, \mathbf{U})\right] \tag{6.33}$$

with $\mathbf{C}^i = [\mathbf{c}_0^i, ..., \mathbf{c}_K^i]$ the trajectory of the center of box particle i. This definition is an extension of the cost proposed for point-wise particle representing Dirac functions in Blackmore [Bla+10]. In what follows, assume that $\widehat{J}(\mathbf{X}, \mathbf{U})$ is differentiable, which is true if the initial cost J is differentiable. A typical and practical way of defining the cost function is the quadratic formulation (2.76) which imposes a tradeoff between control effort and guidance efficiency.

However, in the general case, the constraints $\widehat{P}(\mathbf{X}(\mathbf{U}) \notin \mathcal{F}) - \delta$ are not differentiable with respect to the control sequence. The estimated failure probability $\widehat{P}(\mathbf{X}(\mathbf{U}) \notin \mathcal{F})$ consists of a sum of kernels integrations on two domains: $[\mathbf{x}_k^i] \cap \mathcal{C}_k$ which represents the intersection between box particle i with the constraints domain at time-step k, and $[\mathbf{x}_k^i] \cap \mathcal{X}_k$ which represents the intersection of $[\mathbf{x}_k^i]$ with the set of all states belonging to trajectories that have violated the constraints in the past.

The bounds of each box particle $[\mathbf{x}_k^i]$ depend on the dynamical inclusion function $[f]([\mathbf{x}_{k-1}, \mathbf{u}_k])$, or $[F]([\mathbf{x}_0], \mathbf{U}_k, [\mathbf{W}_k])$ if a minimal analytic inclusion solution exists (see (6.12) for linear systems). Therefore, the bounds of the integration domains $[\mathbf{x}_k^i] \cap \mathcal{C}_k$ and $[\mathbf{x}_k^i] \cap \mathcal{X}_k$ depend on the control inputs sequence $\mathbf{U}$. They are computed via interval intersection, using the non-differentiable min and max functions:

$$[a] \cap [b] = \begin{cases} [\max(\underline{a}, \underline{b}), \min(\overline{a}, \overline{b})] & \text{if} \max(\underline{a}, \underline{b}) \leq \min(\overline{a}, \overline{b}) \\ \emptyset & \text{else} \end{cases} \tag{6.34}$$

This may result in non-smooth boundaries with respect to the control sequence $\mathbf{U}$. In addition, in non-linear cases, the bounds of $[\mathbf{x}_k^i]$ may depend on additional non-smooth interval operations, such as the interval multiplication:

$$[a][b] = [\min(\underline{a}\overline{b}, \overline{a}\overline{b}, \underline{a}\underline{b}, \overline{a}\underline{b}), \max(\underline{a}\overline{b}, \overline{a}\overline{b}, \underline{a}\underline{b}, \overline{a}\underline{b})] \tag{6.35}$$

In what follows, a smoothing method of the failure probability constraints is proposed in order to compute their gradient.

6.3.2 Differentiable failure probability constraint

The failure probability constraint (6.23) can be differentiated with respect to the control sequence $\mathbf{U}$ using the Leibniz integral rule. This implies that:

1. The kernel Π has to be differentiable on its support;
2. The boundaries of the integration domains must be differentiable.

Condition 1 can be achieved using a smooth kernel Π, e.g. a beta kernel. Condition 2 requires two statements. First, the intersection of the forbidden set $\mathcal{C}_k$ and the current box $[\mathbf{x}_k^i]$ may produce non-smooth boundaries. To tackle this issue, approximate the intersection $[c] = [\underline{c}, \overline{c}]$ of two intervals $[a] = [\underline{a}, \overline{a}]$ and $[b] = [\underline{b}, \overline{b}]$ by a smooth analytic function. The exact intersection $[c] = [a] \cap [b]$ can be formulated as:

$$\begin{cases} \underline{c} = \max(\underline{a}, \underline{b}) = \mathbf{1}_{\underline{a} > \underline{b}} \underline{a} + (1 - \mathbf{1}_{\underline{a} > \underline{b}}) \underline{b} \\ \overline{c} = \min(\overline{a}, \overline{b}) = (1 - \mathbf{1}_{\overline{a} > \overline{b}}) \overline{a} + \mathbf{1}_{\overline{a} > \overline{b}} \overline{b} \end{cases} \tag{6.36}$$

The indicator function $\mathbf{1}_{x>y}$ can be approximated by a smooth function such that:

$$\mathbf{1}_{x>y} \approx \frac{1}{2} \left(1 + \tanh \left(\frac{x - y}{\varepsilon} \right) \right) \tag{6.37}$$

where $\varepsilon > 0$ is a scale parameter. This formulation converges to the actual intersection when $\varepsilon \to 0$. It can be generalised to d dimensions by performing it independently on each dimension. In addition, the box particle bounds defined by the inclusion propagation must be differentiable. This is guaranteed for linear models by the use of (6.12). Although this condition does not prevent the use of the method for nonlinear models, a case by case analysis must be done.

In the next section, numerical results are presented for one control sequence optimisation, to highlight the gain in robustness and the potential computational load reduction.

6.3.3 Numerical results

This section evaluates the BPC trajectory optimisation scheme introduced in Sections 6.2 and 6.3 in terms of accuracy and the ability to handle non-linear dynamics. A comparison with the Particle Control approach (PC, Blackmore [Bla+10]) is provided for linear cases. Experiments were designed to evaluate the following points:

1. The relative performance of BPC and PC in terms of actual failure probability for linear dynamical models in the presence of a convex feasible set $\mathcal{F}$;
2. The relative performances of both approaches for linear dynamical models in the presence of a non-convex feasible set $\mathcal{F}$;
3. The BPC behavior in the presence of a non-linear dynamical model.

As mentioned in Section 6.2, the actual failure probability is computed from a representative set of 10^4 reference trajectories whose initial states are samples of the initial state density $p(\mathbf{x}_0)$ and controlled by the control sequences computed by BPC and PC approaches.

Two linear applications are first presented to illustrate the performance of BPC compared to PC [Bla+10]: an aircraft flight envelope protection guidance problem, and a spacecraft orbital collision avoidance. A third example consists of an aircraft trajectory planning modelled by a non-linear unicycle dynamics, in the context of Terrain Aided Navigation. The maximum failure probability δ is chosen equal to 0.1, for consistency with the results presented in [Bla+10]. It is then switched to a lower value to evaluate how the BPC scales with δ. BPC is implemented with a Sequential Quadratic Programming (SQP) optimisation method (see Section 2.3.1).

Application to aircraft flight envelope protection (point 1)

The first example illustrates the method's performance for a convex feasible set. Consider a fixed-wing autonomous aircraft whose longitudinal dynamics is described by the following model (see Section 3.4), corresponding to a McDonnell-Douglas F-4C Phantom, linearised around an equilibrium point $\mathbf{x}_{eq} \in \mathbb{R}^6$ (Cook [Coo12]):

$$\dot{\mathbf{x}} = \begin{bmatrix} -0.068 & -0.011 & 0 & -9.81 & 0 & 0 \\ 0.023 & -2.10 & 375 & 0 & 0 & 0 \\ 0.011 & -0.160 & -2.20 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 377 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} -0.41 & 1.00 \\ -77.0 & -0.09 \\ -61.0 & -0.11 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{u} \tag{6.38}$$

where the state $\mathbf{x} = [u - u_{eq}, w, \dot{\theta}, \theta, \dot{h}, h]^T$ consists of the deviation between the longitudinal body axis velocity u and the equilibrium speed $u_{eq} = 377$ ms⁻¹, the vertical body frame velocity w, the pitch derivative $\dot{\theta}$, the pitch θ, the Earth frame vertical speed $\dot{h}$ and the altitude h. The control $\mathbf{u} = [\delta_e, \delta_\tau]^T$ consists of the elevator angle and the throttle longitudinal acceleration. The initial state $\mathbf{x}_0$ is assumed to be close to the equilibrium $\mathbf{x}_{eq}$, which consists of $u \approx u_{eq}, h \approx 300$ m and all other values close to zero.

The objective in the considered scenario is to achieve a step altitude input up to $1,000$ m. This problem is constrained by a flight envelope which consists of a convex hull defined by: $300 \leq u \leq 640$ ms⁻¹, $|\dot{h}| \leq 30$ ms⁻¹, $|\delta_e| \leq 1$ rad, $0 \leq \delta_\tau \leq 3$ ms⁻². The cost function (2.76) is defined by $\mathbf{R} = 0.1, \mathbf{Q} = 10^{-7}, \mathbf{S} = 1$.

For the sake of simplicity, the multimodal initial state density is constructed from a mixture of two Gaussian densities $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$. The initialization of the box particle cloud is done by a regular paving of the state's support. Weights are initialised by a histogram approach. The initial state density is chosen as: $\boldsymbol{\mu}_1 = \mathbf{x}_0$, $\boldsymbol{\mu}_2 = \mathbf{x}_0 + [0,0,0,0,-0.5,-5]^T$, $\boldsymbol{\sigma}_1 = [0.1, 0.1, 0.001, 0.0001, 0.1, 3]^T$, $\boldsymbol{\sigma}_2 = [0.1, 0.1, 0.001, 0.0001, 0.1, 3]^T$. Such a multimodality on h and $\dot{h}$ can be encountered when the altitude is measured by a radar altimeter [Ber99]. It can be caused by the rebounding of the waves on surfaces of different altitudes such as the ground or a forest canopy.

Figure 6.5 shows the empirical failure probability and the actual failure probability standard deviation, versus the number of particles and box particles. Similar results were reported in [Bla+10] for the Particle Control. The Box Particle Control appears more robust than the Particle Control, and requires a fewer number of particles for similar performances. As expected, it tends to overestimate the failure probability, which makes it a little more conservative. Nevertheless, this ensures a security margin in the constraint satisfaction, which makes it more robust to potential estimation errors, as illustrated by the standard deviation box around the empirical failure probability.

Figure 6.6 shows the time evolution of one of the constrained variables $\dot{h}$ and the actual altitude h obtained by applying the resulting control inputs in open loop. The desired state is achieved despite a significant state uncertainty along the trajectory prediction.

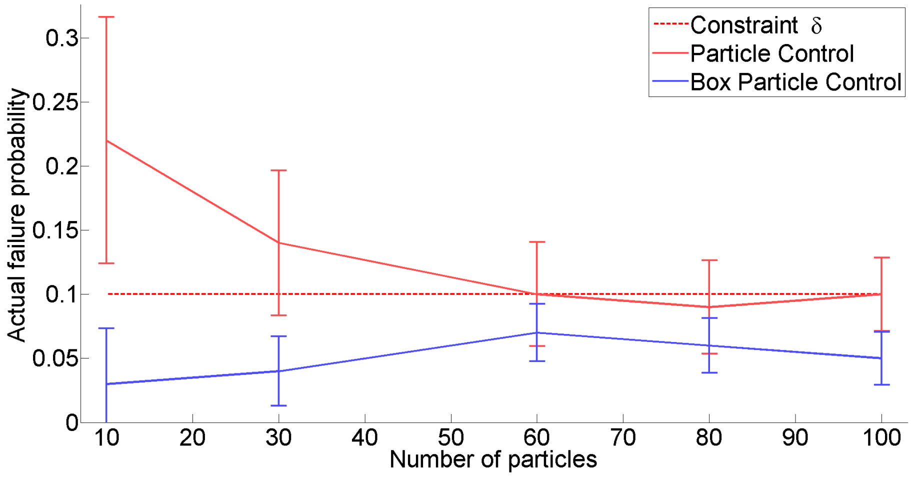


Figure 6.5: Actual failure probability with both methods, associated to its standard deviation, for a maximum desired failure probability of $\delta = 0.1$.

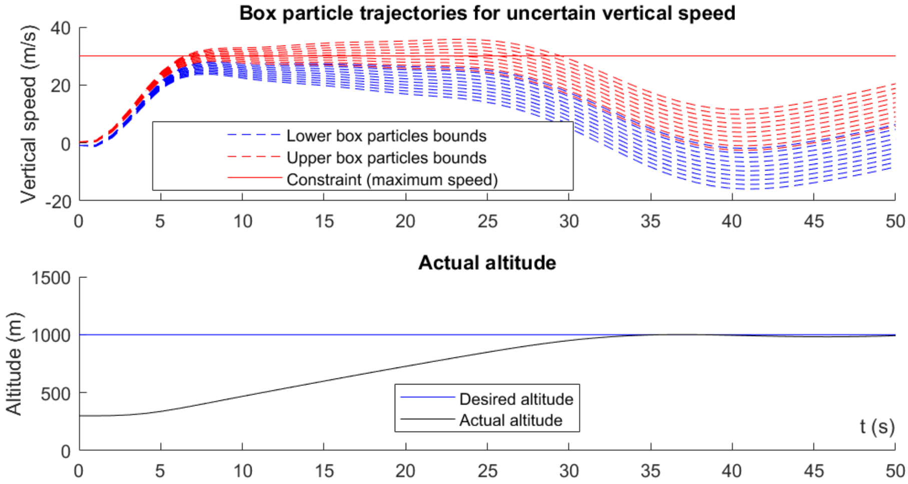


Figure 6.6: Box particle trajectories for the $\dot{h}$ dimension and actual altitude (open loop).

Application to spacecraft orbital collision avoidance (point 2)

A second example illustrates the method performances for a non-convex feasible set. The problem consists of a spacecraft orbital collision avoidance. Let $\mathbf{x} = [x, y, \dot{x}, \dot{y}]^T$ be the state vector representing the relative Cartesian position and velocity of the spacecraft with respect to a non-maneuvering obstacle (e.g. a satellite). The obstacle is assumed to be located on a near-circular low earth orbit. The considered relative dynamical model is Clohessy-Wiltshire (see Section 3.4), derived under the hypothesis of a close proximity between the obstacle and the spacecraft. The orbital angular rate is set to $\omega = 0.00117744\text{rad/s}$, which corresponds to a 90 minutes low Earth orbit.

The initial state is chosen such that it leads to a collision with the obstacle (located at the origin) if no maneuver is performed, $\mathbf{x}_0 = [-42.5, -75, -0.00024, 0.099]^T$. The cost function (2.76) is determined by $\mathbf{R} = 1, \mathbf{Q} = \mathbf{S} = 0$. The feasible set is defined by $[x, y]^T \in \mathbb{R}^2 \setminus [-10, 10] \times [-10, 10]$ m, and $|u_{x,y}| \leq 10^{-5}$ ms-2.

The multimodal initial state density is constructed from a mixture of two Gaussian densities $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$. The initial state density is defined by: $\mu_1 = \mathbf{x}_0$, $\mu_2 = \mathbf{x}_0 + [0.4, 0.4, 0, 0]^T$, $\sigma_1 = [0.1, 0.1, 10^{-4}, 10^{-4}]^T$, $\sigma_2 = [0.3, 0.3, 10^{-4}, 10^{-4}]^T$. Such a multimodality can be encountered for several configurations of orbital relative sensing (Corazzini [CH99]).

Figure 6.7 shows the actual failure probability and its standard deviation, versus the number of particles and box particles. The conclusion is the same as in the first example. This illustrates the ability of BPC to tackle non-convex feasible sets. The authors of [Bla+10] suggest using a number of particles N_{PC} from 100 to 200 for similar cases. For the BPC, it is suggested to use about $20 \leq N_{BPC} \leq 40$ box particles for the presented cases. Thus, the computational ratio between the two methods defined by (6.29) is about $r \leq 0.4$, which represents at least a 60% reduction in terms of the number of elementary operations. Fewer box particles can be used, which may lead to a slightly more conservative behaviour while still guaranteeing that $P(\mathbf{X} \notin \mathcal{F}) \leq \delta$.

Figure 6.8 shows the relative trajectory of the spacecraft with respect to the obstacle to avoid, as well as the control inputs.

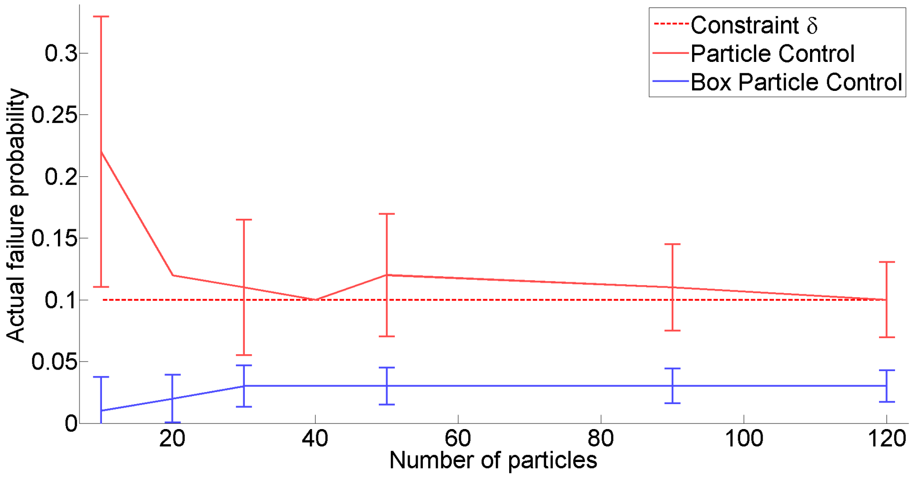


Figure 6.7: Actual failure probability with both methods, associated to its standard deviation, for a maximum desired failure probability of $\delta = 0.1$.

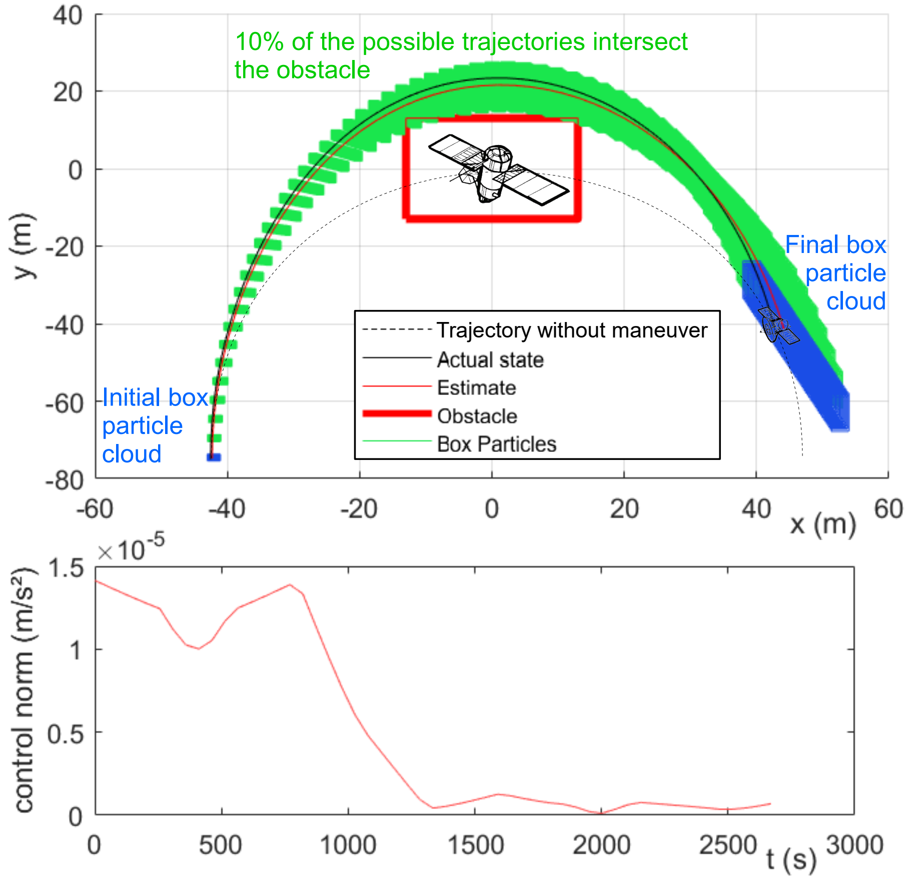


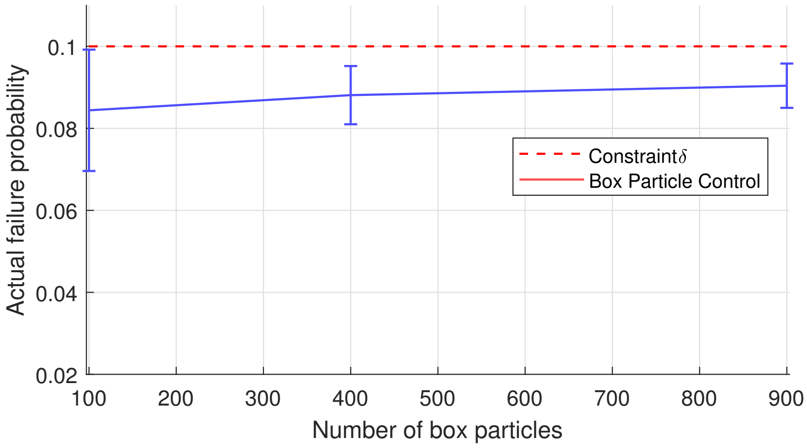
Figure 6.8: Box particle cloud trajectory for an orbital collision avoidance with an obstacle (red box) and control input norm. The box trajectories prediction corresponding to the optimal control sequence which minimises the cost function J while satisfying the chance-constraint $P(\mathbf{X} \notin \mathcal{F}) < \delta = 0.1$. The feasible domain $\mathcal{F}$ corresponds to all trajectories that do not intersect the obstacle area. The actual trajectory obtained without performing the manoeuvre (black dash curve) would not have satisfied the constraints since it yields a collision. The actual trajectory obtained by applying the optimal control sequence satisfies the chance-constraint.

Application to bank to turn control with TAN measurements inputs (point 3)

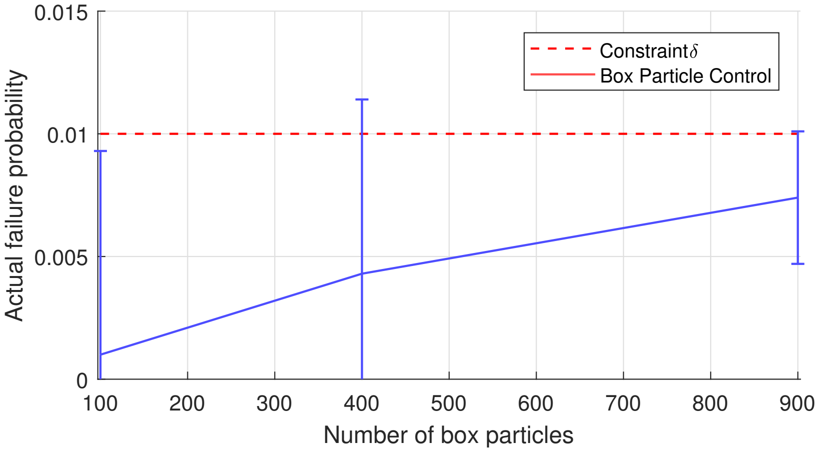
A third example presents a non-linear bank to turn trajectory planning for an autonomous aircraft whose navigation is tackled by a BRPF receiving Terrain Aided Navigation (TAN) measurements. The state density is initialised in the same way as the TAN simulation introduced in Section 6.2, with the same initial uncertainty and the same process noise. The initial state corresponds to the navigation scenario Scenario TAN Alps ($t = 10$ s), i.e. a position of 42.89 ° in latitude and 106.5 ° in longitude (recalling that the geographic coordinates of this scenario are fictitious). The initial speed is 231 ms^{-1} with a heading of 328 °. The uncertainty on the state has a variance around 10^3 m in position and 3 ms^{-1} in velocity. The dynamics of the aircraft is described by the non-linear unicycle model introduced in Section 3.4. The forbidden state areas are the same as those defined in Section 6.2 and can represent no-flight-zones (e.g. urban areas). The control input is the aircraft's bank angle (or rolling angle) and is limited to ±25 °. The cost function is set up with $\mathbf{Q} = 0$ (no desired state or trajectory) and $\mathbf{R} = \mathbf{S} = 1$. Since the Particle Control (Blackmore [Bla+10]) is limited to the linear case, only BPC is studied in this section in order to evaluate its behaviour in the presence of a non-linear dynamical model.

Figure 6.9 (a) shows the averaged actual failure probability and its standard deviation while applying the BPC control sequences computed on a hundred Monte Carlo runs, with a maximum desired failure probability of 0.1. The BPC yields a slightly conservative solution, which appears reasonable with respect to the actual failure probability standard deviation. In other words, BPC ensures that the chance constraint $P(\mathbf{X} \notin \mathcal{F}) < \delta$ is satisfied with a consistent and reasonable security margin. Conservatism decreases when the number of box particles increases. Figure 6.9 (b) shows how the BPC solution scales with $\delta = 0.01$. The standard deviation does not scale proportionally to δ, which is probably due to the wrapping effect of the box particle density propagation (see Chapter 2, Section 2.1.5). Nevertheless, the actual failure probability distribution remains consistent with the desired failure probability. The BPC solution still meets the chance constraint $P(\mathbf{X} \notin \mathcal{F}) < \delta$.

Figure 6.10 (a) shows the box trajectory prediction corresponding to the optimal control sequence plotted in Figure 6.10 (b) for one run with $\delta = 0.1$. The obtained trajectory has drifted from its original course (without manoeuvre, Figure 6.4 (b)) to meet the chance constraint. Figure 6.10 (b) plots the corresponding bank control input (rolling angle) which satisfies the control constraints (rolling angle limited to ±25 °).

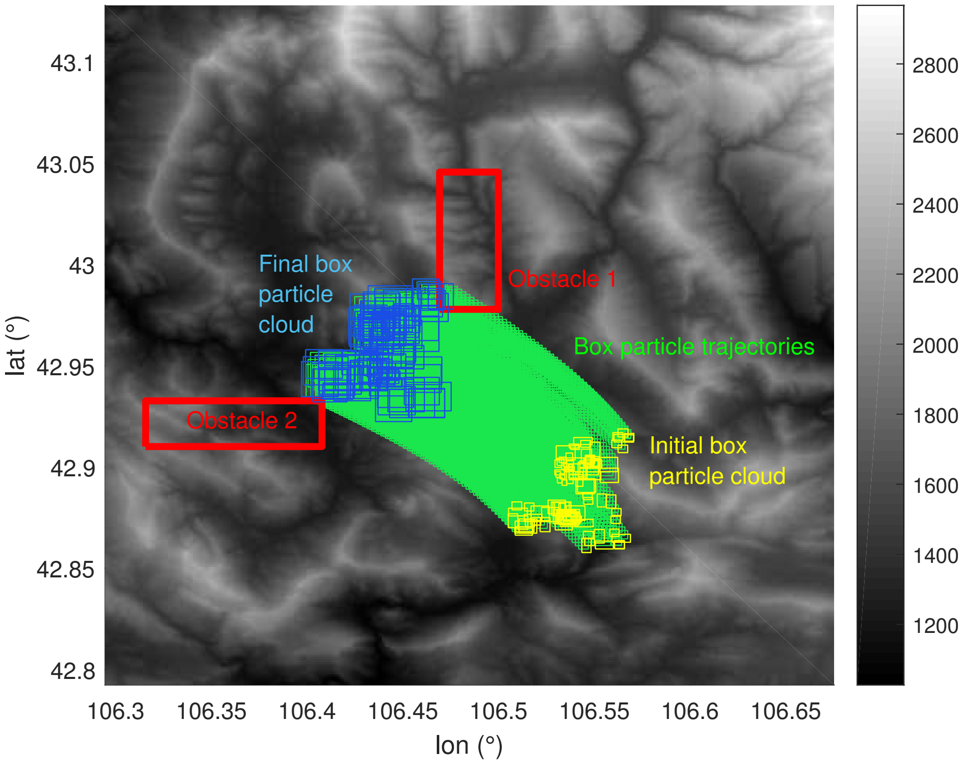


(a) Actual failure probability for BPC ($\delta = 0.1$)

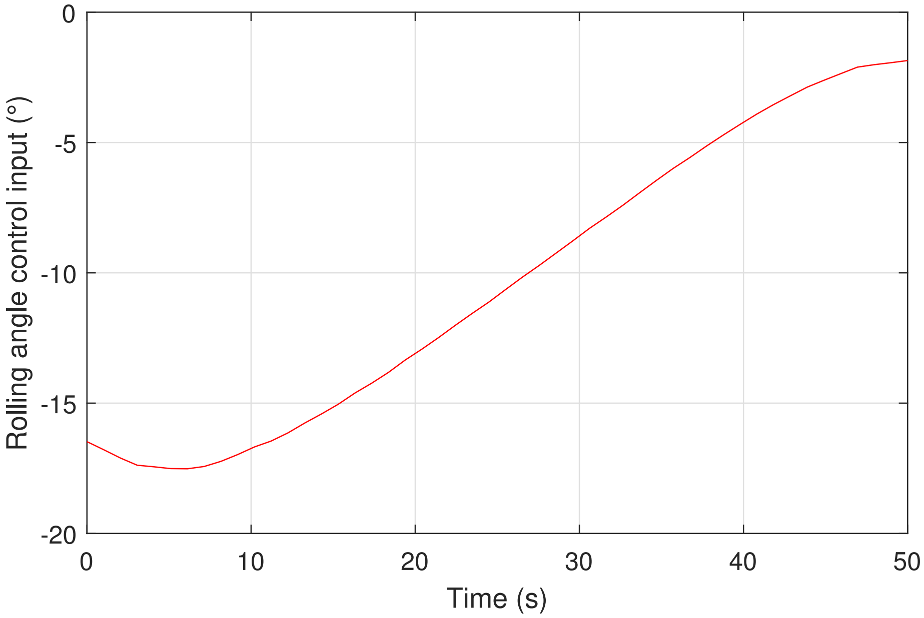


(b) Actual failure probability for BPC ($\delta = 0.01$)

Figure 6.9: Actual failure probability for BPC, associated to its standard deviation, for a maximum desired failure probability of $\delta = 0.1$ (a) and $\delta = 0.01$ (b).



(a) Box particle trajectory prediction from BRPF estimation



(b) Optimal control sequence

Figure 6.10: Box trajectories prediction (a) corresponding to the optimal control sequence (b) which minimises the cost function J while satisfying the chance-constraint $P(\mathbf{X} \notin \mathcal{F}) < \delta = 0.1$. The feasible domain $\mathcal{F}$ corresponds to all trajectories that do not intersect the two red areas referred to as obstacles.

6.4 SUMMARY

This chapter introduced a box particle based chance constrained trajectory optimisation scheme called Box Particle Control (BPC). An analytic formulation of the box chance constraint estimator is provided in (6.15). The BPC approach appears more accurate than the Particle Control approach (Blackmore [Bla+10]) and requires a lower computational load for the same performances (about a 30% reduction), as evaluated by several examples (research question 6).

The integration of estimator (6.15) in a differentiable optimisation scheme is studied, leading to a smooth formulation of the constraint gradient (research question 7). BPC is able to tackle non-linear dynamical models, such as a unicycle bank to turn dynamics. Finally, BPC appears to be a good trade-off between robustness to state uncertainty and computational load reduction for a wide variety of problems, including non-linear dynamics with non-Gaussian state uncertainty and a non-convex feasible set.

The BPC was evaluated for a single trajectory planning occurrence and appears suitable for integration in a Receding Horizon Control (RHC) such as a Model Predictive Control scheme (MPC). The study of the convergence of BPC in such a scheme will be tackled in future works. Chance constraint approaches are also suitable for integration in uncertain dynamics control problems, such as hybrid systems and jump Markov models (e.g. see some application examples in Blackmore [Bla+10]). The study of the integration of BPC in such uncertain dynamics is also a future axis of research.

7

CONCLUSION

This thesis investigated the use of kernel mixtures bounded by box particles for state estimation and trajectory planning. Based on this idea, a state estimation algorithm called the Box Regularised Particle Filter (BRPF) was introduced, as well as a chance constrained trajectory planning method called the Box Particle Contol (BPC). These approaches aim to tackle severe measurements ambiguities and safe control under a constrained and uncertain trajectory.

Chapter 4 described the BRPF approach, mainly based on the original Box Particle Filter (BPF [AGB07], see Chapter 2, Section 2.1.6) and on the Regularised Particle Filter (RPF [MOLG01], Chapter 2, Section 2.1.4). The assumption is made that uncertainties are bounded. Chapter 5 studied the implementation of BRPF within a federated architecture [Car88] (Chapter 2, Section 2.2.3). Chapter 6 described the BPC approach, derived from the Particle Control trajectory planning approach (PC [Bla+10], Chapter 2, Section 2.3.2)

BRPF was evaluated on the Terrain Aided Navigation scheme (TAN), which consists of a severely ambiguous problem (a highly non-linear and non-injective observation model) and was compared to previous approaches (BPF [AGB07], and conventional Particle Filter SIR-PF [GSS93]). For similar performances, BRPF yields a computational load reduction of 75% with respect to the original BPF, and of 97% with respect to the SIR-PF. BRPF is shown to be robust to severe ambiguities and unknown-but-bounded measurement uncertainties. BPC was applied to optimisation for a non-convex feasible set with non-Gaussian state uncertainty, for example flight envelope protection, spacecraft collision avoidance and fixed-wing aircraft trajectory determination. In particular, trajectory planning from an initial multimodal BRPF estimation output was investigated. The BPC Computational load was found to be reduced by 30% compared to the PC for similar failure probability estimation performance.

The thesis answered the following research questions:

1. *Can BPF be formalised for any box kernel and is it of practical use to do so?*
   BPF can be theoretically formalised for any box kernel mixture and any measurement density (Section 4.1). Formally accounting for the right measurement density increases the estimation accuracy. However, it becomes rapidly intractable in the presence of non-linearities and is therefore not of practical use to tackle ambiguous measurements such as the TAN case. In addition, measurement density may be unknown. In practice, the original BPF uniform kernel formulation is robust enough to handle such cases as experimentally shown in Section 4.5.

2. *Can the original BPF be modified to ensure that the actual state belongs to at least one box particle?*
   BPF can be modified such that it always yields a consistent estimate. A guaranteed formulation was derived from Multinomial Resampling which ensures that at least one box particle contains the actual state (Guaranteed Resampling, Section 4.2). This proves efficient in practice by ensuring a zero non-convergence rate (Section 4.2.3). In addition, it yields a higher estimation accuracy than the original BPF.

3. *What is the impact of the choice of box resampling subdivision dimension on the filter's performance? In particular, does accounting for observability enhance performance?*
   Two subdivision resampling approaches were introduced: the Geometrical Subdivision (GS, Section 4.3.1), based on a sub-box normalisation and the Maximum Likelihood Covariance Subdivision (MLCS, Section 4.3.2), based on observability. Although GS is less general than MLCS, it yields a significantly lower computational load and a higher accuracy when used in conjunction with Guaranteed Resampling (see Section 4.3.3). This tends to show that accounting for observability is not necessary as long as the filter's robustness is ensured by construction.

4. *Does regularisation improve the BPF's accuracy?*
   The kernel regularisation concept was derived for the BPF scheme, leading to the Box Regularised Particle Filter (BRPF, Section 4.4). It was proved that the BRPF is necessarily better than the original BPF in terms of density estimation (Mean Integrated Square Error criterion, Proposition 5). In practice, it translates into a higher accuracy, and does not have a significant impact on the total computational load (see Section 4.4.3). However, regularisation must be tuned carefully in the presence of multimodalities and may introduce instability.

5. *Can BPF-like algorithms be integrated to larger estimation architectures such as Gaussian federated filters?*
   It was shown showed that BPF algorithms can be integrated within Gaussian federated architectures, provided that the estimated posterior conditional density tends to a unimodal function (see Section 5.1). In particular, BRPF asymptotically yields an estimated unimodal density (Proposition 7). BRPF was implemented in several federated architecture, where it tackles TAN measurements while dedicated filters deal with Gaussian measurements. Such an architecture allows a more accurate estimation from Gaussian measurements, which would suffer from the wrapping effect if only tackled in the BRPF. The conservatism is also significantly reduced. In addition, federated architectures make it possible to marginalise the state vector estimated by the BRPF to focus on the state variable involved in measurements ambiguities (e.g. TAN measurements, Section 5.2). This results in a significantly lower computational load than a BRPF estimating a full dimension state. Finally, this scheme can be extended to distributed federated architecture for collaborative estimation in a fleet (Section 5.3).

6. *Does a box kernel mixture based failure probability estimation require a lower computational load than a Dirac mixture method for similar performance?*
   An analytic formulation of a box kernel failure probability estimator was introduced (Section 6.2). In practice, this estimation method is more accurate than the original Particle Control approach (Blackmore [Bla+10]) and requires a lower computational load for the same performances (see Section 6.2.3).

7. *Is the box kernel failure probability estimation approach compatible with the differentiable optimisation scheme for trajectory planning?*
   The integration of the box kernel failure probability estimator in a differentiable optimisation scheme yields a smooth formulation of the constraint gradient (Section 6.3). BPC is able to tackle non-linear dynamical models, such as a unicycle bank to turn dynamics. Finally, BPC appears to be a good trade-off between robustness to state uncertainty and computational load reduction for a wide variety of problems, including non-linear dynamics with non-Gaussian state uncertainty and a non-convex feasible set.

Several future research directions and challenges can be drawn from this work. Theoretical and empirical comparisons were made between Multinomial Resampling and its guaranteed derivation. A comparison between different resampling approaches could be done, as well as the feasibility and impact of their guaranteed derivations. For example, techniques such as Residual Resampling, Stratified Resampling, or Metropolis-Hastings could be studied (see for example Douc [DC05] and Li [LBD15]). The box subdivision step could also be further studied. It would be interesting to investigate if some observability based box subdivision method could yield a better accuracy than the GS method and a lower computational load than the MLCS approach. Regularisation was theoretically shown to enhance the estimated density fitting with the actual density, which empirically translates into a higher accuracy. Further theoretical studies could tackle the formal convergence proof of this approach, for example in terms of estimation error bound. In addition, the BRPF could be studied on other classes of problems than state estimation. For example, it would be interesting to evaluate its impact on parametric estimation for uncertain dynamics, for example to estimate some model parameters (e.g. stability and control derivatives [Coo12]), or some disturbance parameters (e.g. wind effects on an aircraft). Furthermore, the interest of BRPF could be evaluated on Simultaneous Localisation And Mapping (SLAM) problems, on the basis of previous works done on BPF (see for example Luo [LQ18a]).

Future axes of research can also be mentioned for BPC. This thesis was focused on the trajectory planning problem, using box particle estimation. Further research could implement BPC in a feedback loop, coupled with a stochastic estimator. In particular, BPC could be studied in a Receding Horizon Control scheme (e.g. Model Predictive Control) in terms of stability. This would allow constrained feedback control problems to be tackled together with multimodal state densities. The robustness of the obtained control scheme could be studied with respect to the initial state uncertainty, the transition uncertainty, and the measurement noise. The integration of sample-based chance constrained approaches in RHC has already been tackled for the original point-wise Particle Control, which could be used as a basis for its extension to the box particle scheme. In this thesis, BPC was studied in the context of linear and non-linear dynamics. It would be of interest to evaluate how the approach scales with more severely non-linear dynamics (e.g., aerodynamic models). In addition, sample-based chance constrained optimisation can also tackle uncertain dynamical models, for example Markov Jump models, for which some parameters have a given probability to change their value. This could extend the BPC approach to fault tolerant trajectory planning and control.

State estimation and control under uncertainty and ambiguity are linchpins to the success of many modern applications in the aerospace field and autonomous robotics. This thesis provides a significant improvement in the quality of these functions at minimal computational cost.

A

INTRODUCTION TO INFORMATION THEORY

Information theory aims to mathematically quantify the *informativeness* carried by a random vector (e.g. a measurement or an estimate). Information is usually derived from the associated probability density function and quantifies its compactness. A review of information theory can be found in Grocholsky [Gro02]. In this appendix, two ways of defining information are introduced. The link between information and the Maximum Likelihood estimator is then introduced.

Definitions of information

Let $\mathbf{y}$ be a random vector taking values in $\mathbb{R}^m$ associated with density $p(\mathbf{y})$. A scalar definition of information can be found in Papoulis [PP91]. It is based on the definition of entropy H:

$$H \triangleq -E_{\mathbf{y}}\left[\log(p(\mathbf{y}))\right] \tag{A.1}$$

where $E_{\mathbf{y}}[f(\mathbf{y})] = \int_{\mathbb{R}^m} f(\mathbf{y})p(\mathbf{y})\mathrm{d}\mathbf{y}$ is the mathematical expectation. The entropic information i is defined as the opposite of H:

$$i = -H \tag{A.2}$$

However, this scalar value does not independently capture the information of each variable in the vectorial case.

Another definition of information is the Fisher Information Matrix (FIM) for conditional densities. Let $\mathbf{y}$ be a random vector taking values in $\mathbb{R}^m$ associated with conditional density $p(\mathbf{y}|\boldsymbol{\theta})$ with $\boldsymbol{\theta}$ taking values in $\mathbb{R}^n$. The FIM is defined (e.g. see Tichavsky [Tic+98]) by:

$$\mathcal{I} = -E_{\mathbf{y}|\boldsymbol{\theta}}\left[\frac{\partial^2}{\partial\boldsymbol{\theta}^2}\log\left(p(\mathbf{y}|\boldsymbol{\theta})\right)\right] \in \mathbb{R}^{n\times n} \tag{A.3}$$

where $E_{\mathbf{y}|\boldsymbol{\theta}}[f(\mathbf{y},\boldsymbol{\theta})] = \int_{\mathbb{R}^m} f(\mathbf{y},\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta})\mathrm{d}\mathbf{y}$ is the mathematical expectation. It can be generalised to N independent random vectors $\mathbf{y}^i$. The conditional density is then $\prod_{i=1}^{N} p(\mathbf{y}^i|\boldsymbol{\theta})$. It can be interpreted as the expectation of the likelihood curvature with respect to $\boldsymbol{\theta}$. The larger is the likelihood's support, the less informative is the measurement. Entropic information can be derived from the FIM's determinant (e.g. see Grocholsky [Gro02]).

The FIM is defined if the following axioms are satisfied (e.g. see Lu [Lu+17]):

- Existence and absolute integrability w.r.t $\mathbf{y}$ of the first two derivatives of $p(\mathbf{y}|\boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$,
- The support of $p(\mathbf{y}|\boldsymbol{\theta})$ is independent from $\boldsymbol{\theta}$.

Those conditions are met for a large variety of densities, including Gaussian distribution. However, they do not hold for bounded densities whose support boundaries depend on the conditional variable $\boldsymbol{\theta}$, e.g. bounded measurements, as found in the Box Particle Filter scheme (see Section 2.1.6).

For Gaussian densities such that $p(\mathbf{y}|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\mathbf{P}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{H}\boldsymbol{\theta})^T\mathbf{P}^{-1}(\mathbf{y}-\mathbf{H}\boldsymbol{\theta})\right)$, where $\mathbf{H} \in \mathbb{R}^{m\times n}$, the FIM is equal to the inverse of covariance $\mathbf{P}$:

$$\mathcal{I} = \mathbf{P}^{-1} \tag{A.4}$$

The FIM in the context of state estimation

An information formulation can be derived from the state estimation framework. Under Gaussian assumptions, it led to the Information Filter, introduced by Maybeck [May82] and derived by Manyika [Man93].

The FIM is also used to quantify the information brought by the measurements to the state. In this case, the conditional density is the likelihood $p(\mathbf{m}|\mathbf{x})$, where $\mathbf{m}$ is the measurement, taking values in $\mathbb{R}^{d_m}$, and $\mathbf{x}$ the state, taking values in $\mathbb{R}^d$. The inverse of the FIM is usually referred to as the Cramér-Rao Lower Bound (CRLB) and quantifies the lowest covariance reachable by any estimator for a given estimation problem. In that context, the conditional density is the joint density of all measurements' densities. Its vectorial form in the discrete-case has been introduced by Galdos [Gal80]. In order to reduce the computational cost, an iterative formulation for recursive estimation has been derived by Tichavsky [Tic+98]. It provides an analytic expression of the minimal covariance reachable at any time-step by an estimator, starting from a given initial state covariance.

FIM and CRLB computation for bounded likelihood remain an active area of research. An extention of (A.3) to this case has been introduced by Lu [Lu+17], but this formulation is complex to compute in practice. Nevertheless, it relies on the Maximum Likelihood estimator (see (2.7) in Section 2.1.2), whose covariance asymptotically approaches the CRLB (Wasserman [Was13, Theorem 10.18]).

B

MULTIMODAL DENSITIES AND MEASUREMENTS AMBIGUITIES

This appendix introduces the concept of multimodality. A density is said to be *multimodal* if it has several local maxima. It is said to be *unimodal* if it has only one maximum. This manuscript focuses on posterior conditional state density multimodalities caused by ambiguous measurements, on the way of designing robust estimators.

Link between state density, likelihood, and measurements ambiguities

Multimodalities in the posterior conditional state density $p(\mathbf{x}_k|\mathbf{M}_k)$ can have three causes:

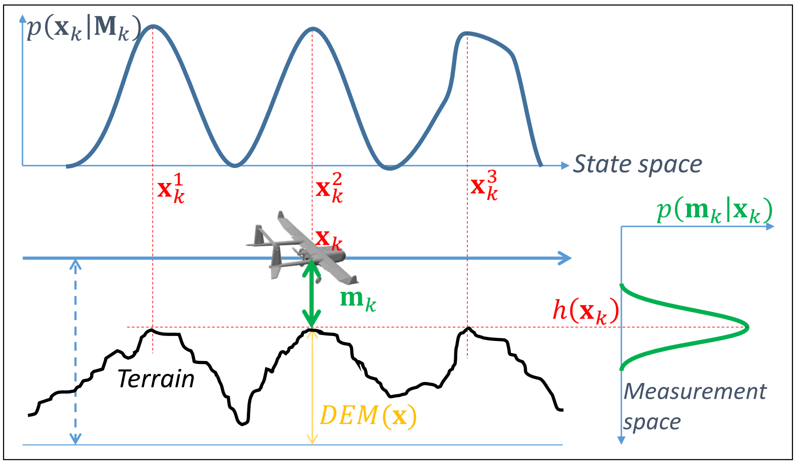
- The measurement model h is not injective or is not continuous, which may lead to several possible states $\{\mathbf{x}_k^1, ..., \mathbf{x}_k^n\}$ being consistent with one single value of measurements $\mathbf{m}_k = h(\mathbf{x}_k^i)\ \forall i$. The Bayes correction step (2.5) from the prior density will produce a conditional posterior density having n local maxima whose values $\mathbf{x}_k^i$ are called *modes*. Figure B.1 (a) illustrates this case for the Terrain Aided Navigation (TAN) case, assuming that the state is made of a single variable $x \in \mathbb{R}$. In practice, the unknown state may consist of three positions, three velocities and potentially other variables such as the angles of attitude.

- The measurement density $p(\mathbf{m}|\mathbf{x})$ may itself be multimodal, as illustrated in Figure B.1 (b). Again, Bayes' rule leads to a multimodal posterior conditional density.

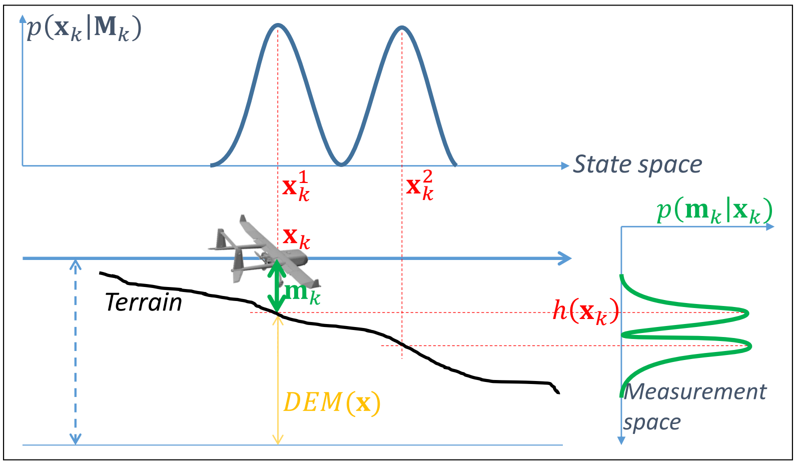
- The measurement model may be incomplete, in the sense that it does not explicitly involve all state variables to be estimated. An illustration is provided in Figure B.1 (c), where the state is made of two positions $\mathbf{x} = [p_1, p_2]^T$ and the measurement model only involves p_2: $h(\mathbf{x}) = p_2$. In such a case, the posterior conditional density does not result in a multimodal density, but has an infinite number of maxima. This case can be linked to a lack of information on one dimension. It can be quantified by the information theory (Appendix A).

In practice, an ambiguous problem may consist of a combination of these three cases. For example, the TAN application consists of a severely non-linear measurement model depending on a Digital Elvation Model (DEM) (3.9). The DEM non-linearities may cause non-injective observations and state multimodalities, as illustrated in Figure B.1 (a). Furthermore, in practice, the measurement density may depend on the overflown terrain. For example, a sand desert may produce a unimodal measurement noise, while a forest often yields a bimodal density, due to the double rebound of the electromagnetic wave on the ground and on the canopy, as stated in Bergman [Ber99]. This adds the case illustrated in Figure B.1 (b) to the previous state multimodalities. In addition, the TAN model (3.9) only involves position variables (latitude, longitude, altitude). Velocity variables are not explicitly observed, which again adds some ambiguity (a case similar to Figure B.1 (c)).

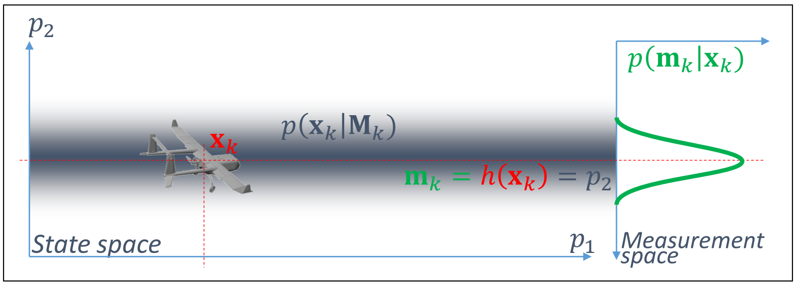
This thesis focuses on case (a), while remaining robust to cases (b) and (c), with the Box Particle Filter (Section 2.1.6). Figure B.2 shows the evolution of $p(\mathbf{x}_k|\mathbf{M}_k)$ in case (a) for Scenario TAN Alps,



(a) Non injective-continuous measurement model case



(b) Multimodal likelihood case



(c) Incomplete measurements case

Figure B.1: Posterior density multimodalities can be caused by three types of measurement ambiguity: non injective measurements (a), multimodal likelihood (b), and incomplete measurements (c). In (a) and (b), the posterior state density is in blue and holds several modes caused by measurements non-linearities or measurement multimodalities. Cases (a) and (b) are Terrain Aided Navigation (TAN) examples. In (c), the posterior density is represented in blue shades in the state space and has an infinite number of maxima (incomplete Cartesian measurements). Note that the posterior density $p(\mathbf{x}_k|\mathbf{M}_k)$ is represented here assuming that no prior knowledge is available (diffuse prior conditional density $p(\mathbf{x}_k|\mathbf{M}_{k-1}) \approx \mathcal{U}_{\mathbb{R}}$), in order to highlight the impact of ambiguities (e.g. first gathered measurement).

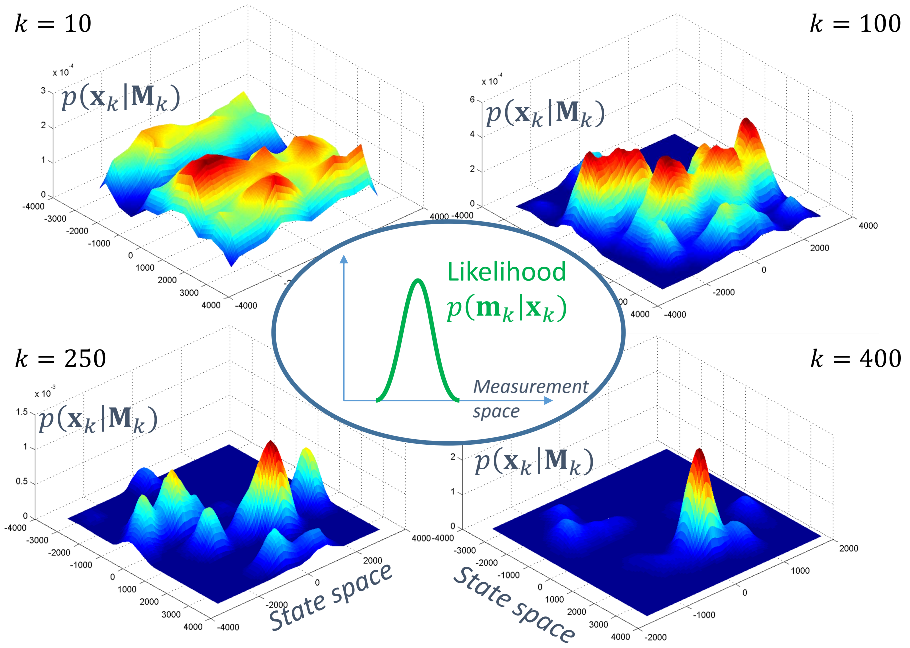


Figure B.2: Posterior conditional state density obtained on a non injective terrain measurement case (Scenario TAN Alps). The posterior density can be multimodal even if the likelihood is unimodal (Gaussian for this example). This highlights that the likelihood may not be representative of multimodalities. Nevertheless, if the measurements asymptotically contain enough information, the posterior density may converge to a unimodal density.

with a Gaussian measurement density. This illustrates the impact of non-injective measurements on the posterior density's multimodality. Figure B.1 (a) also illustrates that the likelihood (defined either as the measurement density or as the joint density of several measurements) may not be representative of the problem's ambiguity. This highlights the fact that mathematical criteria such as the Fisher Information Matrix (A.3) do not account for multimodalities, even though they account for non-linearities.

Two approaches are possible in order to quantify the level of ambiguity for a given problem. First, it is possible to numerically estimate the posterior conditional density $p(\mathbf{x}_k|\mathbf{M}_k)$ and count the number of local maxima, for example by Monte Carlo Simulations coupled to a clustering algorithm (e.g. Murangira [Mur+11]).

C

COMPLEXITY ANALYSIS

The complexity of algorithms can be evaluated in terms of their computational load. In this work, the computational load is defined as the total number of elementary operations (namely, additions and multiplications) required to perform them. Elementary operations are here considered as floating-point operations (also referred as *flops*).

This complexity criterion is often used in the state estimation area. It has been used to evaluate the cost of various implementations of the Kalman Filter, see Verhaegen [VVD86], or to compare several derivations of the Particle Filter, see Karlsson [KSG05]. Table C.1 presents the flops number for several matrix and box operations used in this work. The total number of flops is obtained by adding the number of multiplications and additions. Note that subtractions are associated with additions, and that divisions are associated with multiplications.

Table C.1: Complexity analysis for matrix and box operations

Operation	Size	Multiplications	Additions
Matrix addition $\mathbf{A}+\mathbf{B}$	$\mathbf{A},\mathbf{B}\in\mathbb{R}^{n\times m}$		nm
Matrix multiplication $\mathbf{AB}$	$\mathbf{A}\in\mathbb{R}^{n\times m},\mathbf{B}\in\mathbb{R}^{m\times l}$	nml	$(m-1)ln$
Matrix inversion $\mathbf{A}^{-1}$	$\mathbf{A}\in\mathbb{R}^{n\times n}$	n^3	
Kronecker product $\mathbf{A}\otimes\mathbf{B}$	$\mathbf{A}\in\mathbb{R}^{n\times m},\mathbf{B}\in\mathbb{R}^{p\times q}$	$nmpq$	
Box center $\mathbf{c}_{[\mathbf{x}]}$	$[\mathbf{x}]\in\mathbb{IR}^n$	n	n
Box diameter $\delta_{[\mathbf{x}]}$	$[\mathbf{x}]\in\mathbb{IR}^n$		n
Box volume $\lvert[\mathbf{x}]\rvert$	$[\mathbf{x}]\in\mathbb{IR}^n$	$n-1$	n
Box Propagation $\mathbf{A}[\mathbf{x}]$	$\mathbf{A}\in\mathbb{R}^{n\times m},[\mathbf{x}]\in\mathbb{IR}^m$	$2nm$	$2(m-1)n$
Box Addition $[\mathbf{x}]+[\mathbf{y}]$	$[\mathbf{x}]\in\mathbb{IR}^n,[\mathbf{y}]\in\mathbb{IR}^n$		$2n$
Box Multiplication $[\mathbf{x}][\mathbf{y}]$	$[\mathbf{x}]\in\mathbb{IR}^n,[\mathbf{y}]\in\mathbb{IR}^n$	$8n$	

However, some operations cannot be quantified in terms of flops, for example the random number generation, or very nonlinear functions. For consistency with previous works (e.g. Karlsson [KSG05]), the theoretical cost of one random sample is denoted c_{random} (without distinction of distributions). The computational load of nonlinear dynamical propagation is noted $c_f\in\mathbb{N}$. Similarly, the observation model computational load is noted $c_h\in\mathbb{N}$. Furthermore, some computational costs may be neglected, for example a 1 flop scalar operation with respect to a 10^3 flops of a 10×10 matrix inversion. In such a case, the neglected terms will be indicated by a $o()$, e.g. $o(n^3)$. Furthermore, for algorithm comparisons, only the most significant terms will be considered. Significant terms are denoted by a $\mathcal{O}()$, e.g. $\mathcal{O}(n^3)$.

For example, the naive implementation of the KF yields the following number of flops per time-step, as stated by Verhaegen [VVD86]:

$$c_{KF} = \frac{3}{2}d^3 + d^2\left(3d_m + \frac{1}{2}d\right) + d\left(\frac{3}{2}d_m^2 + d^2\right) + \frac{1}{6}d_m^3 \tag{C.1}$$

where d is the state dimension and d_m the measurements dimension.

The Particle Filter yields the following number of flops per time-step, assuming a linear dynamics, as stated in Karlsson [KSG05]:

$$c_{PF} = N\left(4d^2 + d + dc_{random} + c_h\right) + o(Nd) \tag{C.2}$$

Note that Karlsson [KSG05] made the computation for the Marginalized PF. However, the computational load of conventional PF can be easily derived from it.

D

ELEMENTS OF TOPOLOGY

In what follows, some useful topological properties on Euclidean sets are recalled. All definitions are provided in the particular case of $\mathbb{R}^d$ (see Simmons [Sim63] for more general definitions).

Metric: A metric d on an Euclidean set (e.g. $\mathbb{R}^d$) is an application from $\mathbb{R}^d$ to $\mathbb{R}^+$ which satisfies the following conditions. Let $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^d$: Non-negativity: $d(\mathbf{x}, \mathbf{y}) \geq 0$, identity: $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$, symmetry: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$, and triangle inequality: $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

neighbourhood: A neighbourhood of a point $\mathbf{x} \in \mathbb{R}^d$ is a subset of $\mathbb{R}^d$ which contains at least an open set containing $\mathbf{x}$.

Open/closed set: In $\mathbb{R}^d$, an open set is a subset of $\mathbb{R}^d$ which contains at least one open ball centered on each of its elements. Intuitively, it is a set which does not contain its boundary. A closed set is a set that contains its boundary.

Interior: Let $\mathcal{S}$ be a subset of an euclidean set (e.g, $\mathbb{R}^d$). The interior of $\mathcal{S}$ is the set containing all points that do not belong to its boundary. The interior is an open subset of $\mathcal{S}$.

Ball: An open ball on a metric set (e.g. $\mathbb{R}^d$ with distance d) is a set defined by a center $\mathbf{p} \in \mathbb{R}^d$ and a radius $r \in \mathbb{R}^{+*}$: $B(\mathbf{p}, r) = \left\{ \mathbf{x} \in \mathbb{R}^d \mid d(\mathbf{p}), \mathbf{x}) < r) \right\}$. A closed ball is defined by: $B(\mathbf{p}, r) = \left\{ \mathbf{x} \in \mathbb{R}^d \mid d(\mathbf{p}, \mathbf{x}) \leq r) \right\}$.

Topology: Let $\mathcal{S} \neq \emptyset$ be a non-empty subset of $\mathbb{R}^d$. A topology on $\mathcal{S}$ is a subset $\tau \subseteq \mathcal{P}(\mathcal{S})$ of the power set that satisfies the following axioms: The union of arbitrary subsets of τ is an element of τ, the intersection of any two elements of τ is an element of τ, $\mathcal{S}$ is an element of τ. $\mathcal{S}$ is said to be a topological set (e.g. $\mathbb{R}^d$).

Power set: The power set of a set $\mathcal{S}$ is the set containing all of the subsets of $\mathcal{S}$. It is noted: $\mathcal{P}(\mathcal{S}) \triangleq \{\mathcal{T} \mid \mathcal{T} \subseteq \mathcal{S}\}$.

Connected/disconnected set: A set $\mathcal{S}$ is connected if there exists a continuous function ϕ from $[0, 1]$ to $\mathcal{S}$ which satisfies:

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{S} \begin{cases} \phi(0) = \mathbf{x} \\ \phi(1) = \mathbf{y} \end{cases} \tag{D.1}$$

In other words every pair of elements of $\mathcal{S}$ can be linked by a continuous path which belongs to $\mathcal{S}$. Here are some properties:

- The output set of a continuous function is connected if the input set is connected,
- The union of two connected sets is connected if their intersection is non-empty,
- A topological set (e.g. a subset of $\mathbb{R}^d$) is said to be disconnected if it is the union of two disjoint nonempty open sets.

Disjoint sets: Two sets are said to be disjoint sets if they have no element in common.

Convex set: Let $\mathcal{S}$ be an euclidean set (e.g. a subset of $\mathbb{R}^d$). It is convex if for every pair of its points, every point on the straight line segment that joins the pair of points is also within $\mathcal{S}$.

Indexing set: An indexing set I of another set $\mathcal{S}$ is a set whose elements label (or index) the elements of $\mathcal{S}$. A mapping is done from I to $\mathcal{S}$.

Mapping: Let $\mathcal{S}$ and $\mathcal{T}$ be two sets. A mapping from $\mathcal{S}$ to $\mathcal{T}$ is a binary relation on $\mathcal{S} \times \mathcal{T}$ which associates each element of $\mathcal{S}$ with exactly one element of $\mathcal{T}$.

E

KERNELS CHARACTERISTICS

Epanechnikov kernel

The Epanechnikov kernel is defined in d dimensions by:

$$K_e(\mathbf{x}) = \begin{cases} \frac{d+2}{2c_d}\left(1 - \|\mathbf{x}\|^2\right) & \text{if } \|\mathbf{x}\| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{E.1}$$

where c_d is the volume of the unit hypersphere in $\mathbb{R}^d$ and $d \in \mathbb{N}^*$ the dimension. If a multivariate function can be expressed in terms of a radial function of radius $r = \sqrt{\sum_{i=1}^d x_i^2}$ which is invariant by angles $\theta \in [0, 2\pi]$ and $\phi_i \in [0, \pi], \forall i \in [1, d]$, then its multivariate integral can be expressed by (Miller [Mil64]):

$$\int_{\mathbb{R}^d} K(\mathbf{x}) \mathrm{d}\mathbf{x} = \int f(r) r^{d-1} \mathrm{d}\mathbf{r} \int_0^{2\pi} \mathrm{d}\theta g(d) \tag{E.2}$$

where f is the function expression only depending on $r > 0$, and $g(d)$ is:

$$g(d) \triangleq \prod_{i=1}^{d-2} \int_0^{\pi} \sin^{d-1-i} \phi_i \mathrm{d}\phi_i = \frac{\pi^{\frac{d}{2}-1}}{\Gamma\left(\frac{d}{2}\right)} \tag{E.3}$$

The Epanechnikov kernel satisfies this condition, and can thus be rewritten:

$$K_e(r) = \begin{cases} \frac{d+2}{2c_d}\left(1 - r^2\right) & \text{if } r \in [0, 1] \\ 0 & \text{otherwise} \end{cases} \tag{E.4}$$

Applying (E.2) to the α and β parameters definition (2.29) yields:

$$\alpha = \int_{\mathbb{R}^d} x_1^2 K_e(\mathbf{x}) \mathrm{d}\mathbf{x} = \frac{\pi^{\frac{d}{2}}}{c_d(4+d)\Gamma\left(\frac{d}{2}+1\right)} \tag{E.5}$$

where $\Gamma : \mathbb{R} \to \mathbb{R}$ is the Gamma function (Artin [Art15]), and:

$$\beta = \int_{\mathbb{R}^d} K_e(\mathbf{x})^2 \mathrm{d}\mathbf{x} = \frac{2(d+2)\pi^{\frac{d}{2}}}{c_d^2(4+d)\Gamma\left(\frac{d}{2}+1\right)} \tag{E.6}$$

Beta kernel

The symmetric Beta kernel is defined in d dimensions by:

$$K_b(\mathbf{x}) = \begin{cases} \prod_{i=1}^{d} \frac{(x_i)^{a-1}(1-x_i)^{a-1}}{\mathrm{B}(a,a)} & \text{if } \mathbf{x} \in [0,1]^d \\ 0 & \text{otherwise} \end{cases} \tag{E.7}$$

where B is the Beta function. Applying the α and β parameters definition (2.29) yields:

$$\alpha = \int_{\mathbb{R}^d} x_1^2 K_b(\mathbf{x}) \mathrm{d}\mathbf{x} = \frac{\Gamma(a)\Gamma(2+a)}{\mathrm{B}(a,a)\Gamma(2a+2)} \tag{E.8}$$

and

$$\beta = \int_{\mathbb{R}^d} K_b(\mathbf{x})^2 \mathrm{d}\mathbf{x} = \left(\frac{\Gamma(2a-1)^2}{\mathrm{B}(a,a)^2\Gamma(4a-2)} \right)^d \tag{E.9}$$

The beta kernel K_b tends to a uniform density when $a \to 1^+$.

F

RÉSUMÉ EN FRANÇAIS

L'autonomie d'un engin aérospatial requiert de disposer d'une boucle de navigation-guidage-pilotage efficace et sûre. Cette boucle intègre des filtres estimateurs et des lois de commande qui doivent dans certains cas s'accommoder de non-linéarités sévères et être capables d'exploiter des mesures ambiguës. De nombreuses approches ont été développées à cet effet et parmi celles-ci, les approches particulaires présentent l'avantage de pouvoir traiter de façon unifiée des problèmes dans lesquels les incertitudes d'évolution du système et d'observation peuvent être soumises à des lois statistiques quelconques. Cependant, ces approches ne sont pas exemptes de défauts dont le plus important est celui du coût de calcul élevé. D'autre part, dans certains cas, ces méthodes ne permettent pas non plus de converger vers une solution acceptable. Des adaptations récentes de ces approches, combinant les avantages du particulaire tel que la possibilité d'extraire la recherche d'une solution d'un domaine local de description et la robustesse des approches ensemblistes, ont été à l'origine du travail présenté dans cette thèse.

Cette thèse présente le développement d'un algorithme d'estimation d'état, nommé le *Box Regularised Particle Filter* (BRPF), ainsi qu'un algorithme de commande, le *Box Particle Control* (BPC). Ces algorithmes se basent tous deux sur l'utilisation de mixtures de noyaux bornés par des boites (i.e. des vecteurs d'intervalles) pour décrire l'état du système sous la forme d'une densité de probabilité multimodale. Cette modélisation permet un meilleur recouvrement de l'espace d'état et apporte une meilleure cohérence entre la prédite et la vraisemblance. L'hypothèse est faite que les incertitudes incriminées sont bornées. L'exemple d'application choisi est la navigation par corrélation de terrain qui constitue une application exigeante en termes d'estimation d'état.

Pour traiter des problèmes d'estimation ambiguë, c'est-à-dire lorsqu'une valeur de mesure peut correspondre à plusieurs valeurs possibles de l'état, le *Box Regularised Particle Filter* (BRPF) est introduit. Le BRPF est une évolution de l'algorithme de *Box Particle Filter* (BPF) et est doté d'une étape de ré-échantillonnage garantie et d'une stratégie de lissage par noyau (Kernel Regularisation). Le BRPF assure théoriquement une meilleure estimation que le BPF en termes de *Mean Integrated Square Error* (MISE). La précision d'estimation du BRPF est empiriquement évaluée sur plusieurs cas de navigation par corrélation de terrain. L'algorithme permet une réduction significative du coût de calcul par rapport aux approches précédentes pour des performances similaires (75% de réduction par rapport au BPF, 97% par rapport au filtre particulaire). Le BRPF est également étudié dans le cadre d'une intégration dans des architectures fédérées et distribuées, ce qui démontre son efficacité dans des cas multi-capteurs et multi-agents.

Un autre aspect de la boucle de navigation–guidage-pilotage est le guidage qui nécessite de planifier la future trajectoire du système. Pour tenir compte de l'incertitude sur l'état et des contraintes potentielles de façon versatile, une approche nommé *Box Particle Control* (BPC) est introduite. Comme pour le BRPF, le BPC se base sur des mixtures de noyaux bornés par des boites et consiste en la propagation de la densité d'état sur une trajectoire jusqu'à un certain horizon de prédiction. Ceci permet d'estimer la probabilité de satisfaire les contraintes d'état au cours de la trajectoire et de déterminer la séquence de futures commandes qui maintient cette probabilité au-delà d'un certain seuil, tout en minimisant un coût. Pour des performances similaires, le BPC permet de réduire la charge de calcul de 30% par rapport à une approche stochastique de référence (le Particle Control).

Cette thèse tente de répondre aux questions de recherche suivantes:

1. *Le BPF peut-il être formalisé pour n'importe quel type de noyau borné et de densité de mesure, et cela a-t-il un intérêt pratique?*
   Les précédents travaux ayant portés sur le BPF (Gning [Gni+13]) n'ont considéré que des mixtures de boites portant des noyaux uniformes, ainsi que des bruits de mesure uniformes. Dans la Section 4.1, nous proposons une formalisation générale du BPF en termes de type de noyaux utilisés pour la mixture servant à approximer la densité d'état et de densité de bruit de mesure (Proposition 1). Nous avons mis expérimentalement en évidence que la prise en compte d'hypothèses probabilistes précises sur le bruit de mesure permet d'améliorer les performances du filtre en terme de précision (*Root Mean Square Error*, RMSE) et de robustesse (nombre de non-convergences). Des simulations ont été réalisées avec un modèle d'observation linéaire Gaussien puis linéaire uniforme (Section 4.1.3). Cependant, la formulation générale du BPF devient rapidement non résoluble pour des cas non-linéaires, comme par exemple la navigation par corrélation de terrain (*Terrain Aided Navigation*, TAN). De plus, les propriétés statistiques du bruit de mesure ne sont pas toujours connues. En pratique, la formulation uniforme du BPF de Gning [Gni+13] est suffisamment robuste pour traiter les cas ambigus et non-linéaires (par exemple, l'application de TAN). Il est donc possible de formaliser le BPF sous forme d'une mixture de noyaux quelconques et pour n'importe quel type de bruit de mesure, mais nous n'y avons pas décelé d'intérêt pratique pour les cas de mesure non-linéaire et ambigüe. Des simulations ont permis de mettre en évidence la robustesse de la formulation uniforme du filtre à des bruits de mesure dont la densité est *inconnue-mais-bornée* (Section 4.5).

2. *L'algorithme original du BPF peut-il être modifié pour garantir que l'état réel soit nécessairement contenu dans au moins une box particle du nuage de boites?*
   L'algorithme de BPF (Gning [Gni+13]) se compose de trois étapes réalisée périodiquement: prédiction (propagation des box particles), correction (mise à jour des poids des box particles, quantifiant leur probabilité de contenir l'état réel sachant la nouvelle mesure capteur, si celle-ci est disponible), et ré-échantillonnage (remplacement des box particles ayant une faible probabilité de contenir l'état par des subdivisions des box particles ayant un poids fort). Les étapes de prédiction et de correction assurent que si l'état réel est contenu par au moins une boite à un pas de temps donné, il le restera au pas de temps suivant. Cependant, l'étape de ré-échantillonnage n'assure pas cette propriété et peut en pratique faire diverger le filtre. La technique de ré-échantillonnage utilisée le plus souvent dans la littérature est le ré-échantillonnage multinomial (Algorithme 3). Dans la Section 4.2, nous développons une version garantie du ré-échantillonnage multinomial, nommée *Guaranteed Resampling* (GR, algorithme 6). Le BPF, une fois modifié dans son étape de ré-échantillonnage, garantit qu'au moins une box particle contient nécessairement l'état réel à chaque instant (Proposition 2). En pratique, cela se traduit par un taux de non-convergence égal à zéro (Section 4.2.3). L'intérêt pratique est démontré à travers l'application de navigation par corrélation de terrain.

3. *Quel est l'impact du choix de la dimension de subdivision des box particles lors de l'étape de ré-échantillonnage? La prise en compte de l'observabilité a-t-elle un impact positif sur les performances du filtre?*
   L'étape de ré-échantillonnage du BPF consiste à remplacer les box particles ayant un poids faible par des subdivisions de celles ayant un poids fort. Le nombre de nouvelles particules nées de la subdivision d'une particule donnée est déterminé par l'algorithme de ré-échantillonnage, et est compris entre zéro (box particle supprimée) et N (nombre total de box particles). La méthode

utilisée dans la littérature pour l'étape de subdivision consiste à choisir au hasard une dimension de subdivision pour chaque boite (Gning [Gni+13]). Bien qu'elle soit indépendante de la modélisation du système étudié, cette approche introduit un phénomène de dégénérescence des box particles, aboutissant à des boites dont certaines arrêtes sont de longueur tendant vers zéro. Dans la Section 4.2, nous introduisons deux méthodes de subdivision. La première, basée sur une partition du vecteur d'état en une collection de sous-vecteurs de taille égale, permet de normaliser les longueurs des arrêtes de chaque boite afin de la subdiviser le long de l'arrête normalisée la plus longue (Proposition 3). Cette méthode, nommée *Geometrical Subdivision* (GS), ne prend pas en compte l'observabilité du système et permet de conserver des box particles proportionnées. La seconde méthode consiste à subdiviser chaque box particle selon la dimension qui contient le plus d'information apportée par les mesures. (voir l'annexe A qui définit le concept d'information). Comme l'hypothèse est faite que l'incertitude de mesure est bornée, l'information ne peut pas être calculée au sens de Fisher. Elle peut cependant être approximée par la variance de l'estimateur de maximum de vraisemblance. Ceci permet de définir la méthode de *Maximum Likelihood Covariance Subdivision* (MLCS, Proposition 4), qui prend en compte l'observabilité (ici définie comme l'information apportée à l'état par la mesure). Nous mettons expérimentalement en évidence que les deux méthodes améliorent notablement la précision d'estimation du filtre par rapport à la méthode de subdivision aléatoire (Gning [Gni+13]). La méthode MLCS améliore légèrement la précision du filtre par rapport à la méthode GS dans le cadre du ré-échantillonnage multinomial. Cependant, dans le contexte du *Guaranteed Resampling*, la méthode GS offre des résultats significativement meilleurs, ce qui tend à montrer que la prise en compte de l'observabilité dans la subdivision des boites n'est pas nécessaire lorsque le filtre assure sa robustesse par construction (grâce à l'algorithme de ré-échantillonnage garanti).

4. *Les techniques de régularisation par noyaux peuvent-elles améliorer la précision d'estimation du BPF?*
   Quelle que soit la méthode employée pour subdiviser les box particles lors du ré-échantillonnage, cette opération résulte en un nuage de particules dont beaucoup se superposent exactement sur certaines dimensions. Ceci produit une estimation de densité d'état très échelonnée qui gagnerait à être lissée afin de mieux approximer la densité vraie. Ce phénomène peut être interprété comme un manque d'indépendance statistique entre les paramètres des boites (définies par leurs centres et leurs diamètres). Une méthode possible pour lisser la distribution de ces paramètres est la régularisation (*Kernel Regularisation*). Développée à l'origine pour le filtre particulaire (Musso [MOLG01]) sur la base des travaux de Silverman [Sil86], cette approche permet de bruiter les paramètres des box particles selon une distribution optimale au sens du critère de MISE (*Mean Integrated Square Error*), qui quantifie la précision de l'approximation de la densité vraie par la mixture de box particles. Ceci permet de définir le *Box Regularised Particle Filter* (BRPF), qui assure une MISE asymptotique plus faible que le BPF original (Section 4.4, Proposition 5). En pratique, cela se traduit par une meilleure précision d'estimation du filtre (Section 4.4.3). Cependant, en cas de multimodalités dues aux ambiguïtés de mesure (voir Annexe B), un paramètre de réglage est introduit pour limiter l'impact de la régularisation qui peut rendre le filtre instable.

5. *Les algorithmes d'estimation de type BPF peuvent-ils être intégrés dans des architectures fédérées?*
   Les filtres de type BPF (et en particulier le BRPF) permettent de résoudre des problèmes d'estimation ambigus, non-linéaires et d'incertitude de mesure inconnue-mais-bornée. Cependant, le BPF apparait peu précis dans le cas de mesures linéaires et/ou Gaussiennes, comparée aux filtres de type Kalman aux hypothèses plus restrictives. Une approche possible pour traiter des mesures aux caractéristiques disparates est l'utilisation d'une architecture d'estimation fédérée (Carlson [Car88]). Dans ce type d'architecture, chaque mesure est pré-traitée indépendamment dans un filtre dédié, produisant une première estimation locale de l'état. Les filtres locaux forment

une première couche de filtres. Les estimations locales d'état sont ensuite fusionnées dans un second filtre appelé *Master Filter* (voir Section 2.2.3). Ceci permet de traiter les ambiguïtés et les non-linéarités dans la première couche, puis de fusionner de façon linéaire les estimations provenant de l'ensemble des mesures pré-traitée. Nous étudions plus particulièrement le cas de *Master Filter* Gaussian (ex, filtre de Kalman ou sa formulation d'information). Afin d'intégrer le BPF à une telle architecture en temps que filtre local (traitant par exemple les mesures de corrélation de terrain), l'incertitude sur son estimation doit être proche d'une Gaussienne. Nous discutons les hypothèses permettant cette approximation dans la Section 5.1. Dans le cadre de l'utilisation du ré-échantillonnage multinomial (Algorithme 3), la densité d'état *a postériori* du filtre tend asymptotiquement vers une densité unimodale (i.e. n'ayant qu'un seul maximum, King [KF00]), qui peut être approximée par une Gaussienne. Dans le cadre de l'utilisation du ré-échantillonnage garanti (Algorithme 6), cette propriété asymptotique n'est plus assurée. Nous avons néanmoins établi que (a) l'utilisation du ré-échantillonnage garanti assure que la densité *a posteriori* est non-nulle au voisinage de l'état vrai (Proposition 6) et (b) l'utilisation de la régularisation (BRPF, Section 4.4) permet de lisser asymptotiquement la densité *a posteriori* pour les états scalaires, aboutissant à une densité unimodale (Proposition 7), assimilable à une Gaussienne. Cette propriété est admise pour les états vectoriels et vérifiée expérimentalement dans le cas de navigation par corrélation de terrain. Ceci justifie la possibilité d'intégration du BRPF dans une architecture fédérée. Plusieurs types d'architectures fédérées sont testées en simulation: architecture centralisée pour des applications multi-capteurs (Section 5.2.1) et architecture distribuée pour de la navigation coopérative au sein d'une flotte de véhicules (Section 5.3). Les résultats montrent plusieurs intérêts des architectures fédérées par rapport à une architecture en une seule couche où un BRPF traiterait toutes les mesures: meilleure précision d'estimation, moins d'incertitude sur l'estimation et un coût de calcul plus faible (Sections 5.2.2 et 5.3.2).

6. *L'utilisation de mixture de noyaux bornés par des boites est-elle avantageuse en termes de coût de calcul pour l'estimation d'une probabilité d'échec, comparée à l'utilisation plus classique d'une mixture de Diracs?* Nous nous plaçons maintenant dans le cadre d'une optimisation sous contrainte de trajectoire future (jusqu'à un horizon de prédiction $K = k + n$), prédite à partir de la connaissance courante de l'état du système (instant k). Par exemple, un véhicule se trouvant en un point A (état courant) souhaite se rendre en un point B (objectif) en évitant un obstacle situé sur son chemin (contrainte). Dans le cas déterministe, l'optimisation sous contrainte peut être réalisée à partir des méthodes d'*optimisation différentiable* (Section 2.3.1). Lorsque la trajectoire est incertaine, elle peut être modélisée comme un vecteur aléatoire, ce qui rend beaucoup plus difficile l'optimisation. Afin de ramener le problème dans un cadre déterministe, les approches ensemblistes considèrent l'ensemble des trajectoires possibles comme un domaine borné qu'il est possible de forcer à rester en dehors de l'ensemble des contraintes d'état (ex, Bemporad [BM99]). Cependant, ces méthodes sont parfois trop conservatives et peuvent ne pas trouver de trajectoire solution. Afin de relaxer les contraintes, des approches d'optimisation stochastique proposent de contraindre la probabilité que la trajectoire d'état ne satisfasse pas les contraintes au lieu des contraintes elles-mêmes. Cette approche, appelée *chance constrained optimisation* (Charnes [CCS58]), permet de maitriser le risque de ne pas satisfaire les contraintes (probabilité d'échec, par exemple la collision avec un obstacle). La probabilité d'échec maximale admissible est choisie *a priori*. L'implémentation de l'optimisation sous contrainte de probabilité d'échec n'est cependant pas aisée. Des travaux ont proposé des formulations pour des états Gaussiens et uniformes ([BLW06; CEG06]) dans le cadre d'espaces admissibles convexes (espace des trajectoires satisfaisant les contraintes). Pour traiter des cas plus complexes dans lesquels la densité d'état n'est pas analytique et/ou l'espace admissible est non-convexe, une méthode d'optimisation de type particulaire a été proposée (*Sample*

Average Approximation for chance constrained optimisation, Pagnoncelli [PAS09]). Cette approche a été appliquée dans le cadre du guidage de véhicules par Blackmore [Bla+10] (*Particle Control*). Ces approches d'optimisation contrainte sous incertitude d'état peuvent être utilisées dans des cas de commande avec retour d'état. Elles ont été largement appliquées à la commande prédictive (*Model Predictive Control*), par exemple par Calafiore [CF13]. Dans cette thèse, nous nous limiterons cependant à l'étude de l'optimisation d'une trajectoire à un instant donné. L'intégration des méthodes d'optimisation sous contrainte de probabilité d'échec dans un contrôleur de type MPC est donc au delà du périmètre que nous avons défini. Dans la Section 6.2, nous introduisons une nouvelle formulation de la probabilité d'échec construite à partir de la propagation d'une mixture de noyaux bornés par des boites (Proposition 8). Nous montrons par des simulations que la précision de l'estimation de probabilité d'échec est supérieure à celle obtenue avec les méthodes particulaires classiques (Blackmore [Bla+10]) pour une même charge de calcul. Ceci permet, à performances égales, de diminuer le coût de calcul de façon significative.

7. *La formulation d'estimateur de probabilité d'échec par mixture de noyaux peut-elle être utilisée dans des algorithmes classiques d'optimisation différentiable?*
La formulation de l'estimation de probabilité d'échec évoquée ci-dessus n'est malheureusement pas différentiable par rapport aux variables d'optimisation (ici, la séquence des commandes déterminant la future trajectoire). Cette propriété rend le problème non différentiable et empêche l'utilisation de méthodes d'optimisation classiques (par exemple les méthodes SQP, voir Boggs [BT95]). Nous proposons donc une méthode de lissage du gradient de la probabilité d'échec afin de la rendre différentiable, au moyen d'une fonction sigmoïde approximant la fonction indicatrice dans les opérations de calcul d'intervalles. Des simulations mettent en évidence l'efficacité de la méthode sur plusieurs scénarios, impliquant des non-linéarités de dynamique (modèle d'avion sous forme d'*unicycle*) et de fortes multimodalités (estimation de l'état initial issue du BRPF pour la navigation par corrélation de terrain), en Section 6.3.3.

L'algorithme *Box Regularised Particle Filter* (BRPF) présenté dans cette thèse permet une réduction significative du coût de calcul de l'estimation d'état à partir de mesures non-linéaires et ambigües par rapport aux approches précédentes étudiées (*Box Particle Filter* [Gni+13] et Filtre Particulaire [GSS93]). De futurs axes de recherches peuvent être dégagés de ce travail. Premièrement, nous n'avons étudié que la méthode de ré-échantillonnage multinomial et sa formulation garantie. L'impact d'autres méthodes de ré-échantillonnage sur les performances du BPF et la faisabilité d'une formulation garantie seraient des points intéressants à étudier. L'étude de la subdivision des boites lors du ré-échantillonnage pourrait elle aussi être poursuivie. En effet, les résultats présentés dans cette thèse tendent à montrer que la prise en compte de l'observabilité n'est pas nécessaire, mais d'autres méthodes pourraient être envisagées pour l'étudier, impliquant éventuellement un coût de calcul moindre que la méthode de *Maximum Likelihood Covariance Subdivision*. Enfin, l'intérêt de la régularisation dans le BRPF a été montré théoriquement en termes de MISE et empiriquement en termes de RMSE. Une preuve théorique de convergence de l'erreur d'estimation pourrait être envisagée.

La méthode de *Box Particle Control* (BPC) permet de réaliser une optimisation de trajectoire contrainte sous incertitude multimodale. Le BPC permet la prise en compte de non-linéarités de dynamique et présente un coût de calcul inférieur aux approches précedentes (*Particle Control* [Bla+10]). L'intégration du BPC dans le contexte de commande prédictive (MPC) semble être un point important pour son implémentation dans des véhicules autonomes. De nombreuses études de ce type ont déjà été menées dans le cadre des méthodes particulaires et pourraient servir de base à l'intégration du BPC dans une commande de type MPC. Les méthodes d'optimisation particulaire sont également capables de traiter des problèmes de dynamique incertaine (ex: modèles hybrides, modèles à sauts Markoviens [Bla+10]). L'extension du BPC à de telles applications est un axe important pour de futures recherches.

En conclusion, les algorithmes d'estimation et de guidage développés dans cette thèse permettent de réduire significativement le coût de calcul par rapport aux approches précédentes. L'utilisation des mixtures de noyaux bornés par des boites permet de développer des algorithmes d'estimation et de commande versatiles, efficaces et robustes, pouvant être embarqués sur des engins aux performances de calcul limitées, ce qui bénéficiera au domaine aérospatial, et plus généralement à la robotique autonome.

BIBLIOGRAPHY

[AGB07] Fahed Abdallah, Amadou Gning and Philippe Bonnifait. 'Box Particle Filtering for Non linear State Estimation using Interval Analysis'. In: *Automatica* (2007) (cit. on pp. 2, 3, 7, 22, 32, 33, 72, 112, 171).

[Art15] Emil Artin. *The gamma function*. Courier Dover Publications, 2015 (cit. on p. 185).

[Ath71] Michael Athans. 'The role and use of the stochastic linear-quadratic-Gaussian problem in control system design'. In: *IEEE transactions on automatic control* 16.6 (1971), pp. 529–552 (cit. on p. 42).

[BB17] Axel Barrau and Silvère Bonnabel. 'The invariant extended Kalman filter as a stable observer'. In: *IEEE Transactions on Automatic Control* 62.4 (2017), pp. 1797–1812 (cit. on p. 17).

[BG00] Alberto Bemporad and Andrea Garulli. 'Output-feedback predictive control of constrained linear systems via set-membership state estimation'. In: *International Journal of Control* 73.8 (2000), pp. 655–665 (cit. on p. 46).

[BM99] Alberto Bemporad and Manfred Morari. 'Robust model predictive control: A survey'. In: *Robustness in identification and control* (1999), pp. 207–226 (cit. on pp. 46, 190).

[Ber99] Niclas Bergman. 'Recursive bayesian estimation'. In: *Department of Electrical Engineering, Linköping University, Linköping Studies in Science and Technology. Doctoral dissertation* 579 (1999), p. 11 (cit. on pp. 164, 177).

[Ber+05] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas and Dimitri P Bertsekas. *Dynamic programming and optimal control*. Vol. 1. 3. Athena scientific Belmont, MA, 2005 (cit. on p. 42).

[BJ13] Aymeric Bethencourt and Luc Jaulin. 'Cooperative localization of underwater robots with unsynchronized clocks'. In: *Paladyn, Journal of Behavioral Robotics* 4.4 (2013), pp. 233–244 (cit. on p. 41).

[BLW06] Lars Blackmore, Hui Li and Brian Williams. 'A probabilistic approach to optimal robust path planning with obstacles'. In: *American Control Conference, 2006*. IEEE. 2006, 7–pp (cit. on pp. 47, 190).

[BOW11] Lars Blackmore, Masahiro Ono and Brian C Williams. 'Chance-constrained optimal path planning with obstacles'. In: *IEEE Transactions on Robotics* 27.6 (2011), pp. 1080–1094 (cit. on p. 47).

[Bla+10] Lars Blackmore, Masahiro Ono, Askar Bektassov and Brian C Williams. 'A probabilistic particle-control approximation of chance-constrained stochastic predictive control'. In: *IEEE transactions on Robotics* 26.3 (2010), pp. 502–517 (cit. on pp. 3, 5, 47, 48, 149, 152, 156, 161, 163–165, 167, 170–172, 191).

[BI+15] Joaquim Blesa Izquierdo, Françoise Le Gall, Carine Jauberthie and Louise Travé-Massuyès. 'State estimation and fault detection using box particle filtering with stochastic measurements'. In: *DX 2015-26th International Workshop on Principles of Diagnosis, 31 August-1 Septembrer, Paris (France)*. 2015, pp. 67–73 (cit. on p. 68).

[BT95] Paul T Boggs and Jon W Tolle. 'Sequential quadratic programming'. In: *Acta numerica* 4 (1995), pp. 1–51 (cit. on pp. 45, 191).

[Bri71] Kenneth R Britting. 'Inertial navigation systems analysis'. In: (1971) (cit. on p. 51).

[CF13] Giuseppe C Calafiore and Lorenzo Fagiano. 'Robust model predictive control via scenario optimization'. In: *IEEE Transactions on Automatic Control* 58.1 (2013), pp. 219–224 (cit. on pp. 47, 48, 191).

[CEG06] Giuseppe Carlo Calafiore and Laurent El Ghaoui. 'On distributionally robust chance-constrained linear programs'. In: *Journal of Optimization Theory and Applications* 130.1 (2006), pp. 1–22 (cit. on pp. 47, 190).

[CM87a] Paul H Calamai and Jorge J Moré. 'Projected gradient methods for linearly constrained problems'. In: *Mathematical programming* 39.1 (1987), pp. 93–116 (cit. on p. 45).

[CR09] Fabien Campillo and Vivien Rossi. 'Convolution particle filter for parameter estimation in general state-space models'. In: *IEEE Transactions on Aerospace and Electronic Systems* 45.3 (2009) (cit. on p. 24).

[CM87b] Peter J Campo and Manfred Morari. 'Robust model predictive control'. In: *American control conference, 1987*. IEEE. 1987, pp. 1021–1026 (cit. on pp. 42, 46).

[Car88] Neal A Carlson. 'Federated filter for fault-tolerant integrated navigation systems'. In: *Position Location and Navigation Symposium, 1988. Record. Navigation into the 21st Century. IEEE PLANS'88., IEEE*. IEEE. 1988, pp. 110–119 (cit. on pp. 2, 41, 42, 171, 189).

[CR96] George Casella and Christian P Robert. 'Rao-Blackwellisation of sampling schemes'. In: *Biometrika* 83.1 (1996), pp. 81–94 (cit. on p. 21).

[CCS58] Abraham Charnes, William W Cooper and Gifford H Symonds. 'Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil'. In: *Management Science* 4.3 (1958), pp. 235–263 (cit. on pp. 3, 46, 190).

[CWS97] Guanrong Chen, Jianrong Wang and Leang S Shieh. 'Interval Kalman Filtering'. In: *IEEE Transactions on Aerospace and electronic Systems* 33.1 (1997), pp. 250–259 (cit. on p. 32).

[Che95] Yizong Cheng. 'Mean shift, mode seeking, and clustering'. In: *IEEE transactions on pattern analysis and machine intelligence* 17.8 (1995), pp. 790–799 (cit. on p. 19).

[CGT91] Andrew R Conn, Nicholas IM Gould and Philippe Toint. 'A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds'. In: *SIAM Journal on Numerical Analysis* 28.2 (1991), pp. 545–572 (cit. on p. 45).

[Coo12] Michael V Cook. *Flight dynamics principles: a linear systems approach to aircraft stability and control*. Butterworth-Heinemann, 2012 (cit. on pp. 58, 163, 173).

[CH99] Tobe Corazzini and Jonathan P How. 'Onboard pseudolite augmentation system for relative navigation'. In: *Proceedings of US Institute of Navigation GPS-99, Nashville, TN* (1999), pp. 1559–1568 (cit. on p. 165).

[CM03] John L Crassidis and F Landis Markley. 'Unscented filtering for spacecraft attitude estimation'. In: *Journal of guidance, control, and dynamics* 26.4 (2003), pp. 536–542 (cit. on p. 17).

[CD02] Dan Crisan and Arnaud Doucet. 'A survey of convergence results on particle filtering methods for practitioners'. In: *IEEE Transactions on signal processing* 50.3 (2002), pp. 736–746 (cit. on p. 20).

[CO12] Frank E Curtis and Michael L Overton. 'A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization'. In: *SIAM Journal on Optimization* 22.2 (2012), pp. 474–500 (cit. on p. 45).

[Dah05] Karim Dahia. 'Nouvelles méthodes en filtrage particulaire-Application au recalage de navigation inertielle par mesures altimétriques'. PhD thesis. Université Joseph-Fourier-Grenoble I, 2005 (cit. on pp. 15, 24, 51).

[DF+16] Allan De Freitas, Lyudmila Mihaylova, Amadou Gning, Donka Angelova and Visakan Kadirkamanathan. 'Autonomous crowds tracking with box particle filtering and convolution particle filtering'. In: *Automatica* 69 (2016), pp. 380–394 (cit. on pp. 35, 72).

[DRM07] Alessio Dore, Carlo S Regazzoni and Mirko Musso. 'Map particle selection in shape-based object tracking'. In: *Image Processing, 2007. ICIP 2007. IEEE International Conference on*. Vol. 5. IEEE. 2007, pp. V–341 (cit. on p. 19).

[DC05] Randal Douc and Olivier Cappé. 'Comparison of resampling schemes for particle filtering'. In: *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*. IEEE. 2005, pp. 64–69 (cit. on pp. 19, 173).

[Fox02] Dieter Fox. 'KLD-sampling: Adaptive particle filters'. In: *Advances in neural information processing systems*. 2002, pp. 713–720 (cit. on p. 19).

[Gal80] Jl Galdos. 'A Cramer-Rao bound for multidimensional discrete-time dynamical systems'. In: *IEEE Transactions on Automatic Control* 25.1 (1980), pp. 117–119 (cit. on p. 176).

[GL94] J-L Gauvain and Chin-Hui Lee. 'Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains'. In: *IEEE transactions on speech and audio processing* 2.2 (1994), pp. 291–298 (cit. on p. 11).

[GMS05] Philip E Gill, Walter Murray and Michael A Saunders. 'SNOPT: An SQP algorithm for large-scale constrained optimization'. In: *SIAM review* 47.1 (2005), pp. 99–131 (cit. on p. 45).

[Gir14] Christophe Giraud. *Introduction to high-dimensional statistics*. Vol. 138. CRC Press, 2014 (cit. on p. 21).

[Gni+13] A. Gning, B. Ristic, L. Mihaylova and F. Abdallah. 'An Introduction to Box Particle Filtering'. In: *IEEE Signal Processing Magazine* (2013), pp. 166–171 (cit. on pp. 32, 61, 62, 68, 69, 72, 78, 84, 93, 100, 109, 111, 119, 188, 189, 191).

[GMA10] Amadou Gning, Lyudmila Mihaylova and Fahed Abdallah. 'Mixture of uniform probability density functions for non linear state estimation using interval analysis'. In: *Information fusion (FUSION), 2010 13th conference on*. IEEE. 2010, pp. 1–8 (cit. on pp. 32, 68, 72).

[GMA12] Amadou Gning, Lyudmila Mihaylova and Fahed Abdallah. 'Particle filtering combined with interval methods for tracking applications'. In: *Integrated Tracking, Classification, and Sensor Management: Theory and Applications* (2012) (cit. on pp. 35, 72).

[GRM12] Amadou Gning, Branko Ristic and Lyudmila Mihaylova. 'Bernoulli particle/box-particle filters for detection and tracking in the presence of triple measurement uncertainty'. In: *IEEE Transactions on Signal Processing* 60.5 (2012), pp. 2138–2151 (cit. on pp. 33, 61, 72).

[GSS93] Neil J Gordon, David J Salmond and Adrian FM Smith. 'Novel approach to nonlinear/non-Gaussian Bayesian state estimation'. In: *IEE Proceedings F (Radar and Signal Processing)*. Vol. 140. 2. IET. 1993, pp. 107–113 (cit. on pp. 2, 7, 17, 19, 67, 109, 113, 119, 171, 191).

[Gro02] Ben Grocholsky. 'Information-theoretic control of multiple sensor platforms'. In: (2002) (cit. on pp. 40, 175).

[Hei+18] Tor Aksel N Heirung, Joel A Paulson, Jared O'Leary and Ali Mesbah. 'Stochastic model predictive control—how does it work?' In: *Computers & Chemical Engineering* 114 (2018), pp. 158–170 (cit. on pp. 47, 48).

[HBA96] Valery Heymann and Joseph Z Ben-Asher. 'Aircraft Trajectory Optimization in the Horizontal Plane'. In: *IFAC Proceedings Volumes* 29.8 (1996), pp. 69–74 (cit. on p. 42).

[Hig88] Nicholas J Higham. 'Computing a nearest symmetric positive semidefinite matrix'. In: *Linear algebra and its applications* 103 (1988), pp. 103–118 (cit. on p. 92).

[Hua+11] Minh-Duc Hua, Mohammad Zamani, Jochen Trumpf, Robert Mahony and Tarek Hamel. 'Observer design on the special euclidean group SE (3)'. In: *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE. 2011, pp. 8169–8175 (cit. on p. 17).

[IGN18] IGN. *IGN BD ALTI*. www.professionnels.ign.fr/bdalti/. 2018. (Visited on 20/04/2018) (cit. on p. 53).

[Jau01] Luc Jaulin. *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*. Vol. 1. Springer Science & Business Media, 2001 (cit. on pp. 27, 30, 31, 125).

[Jau09] Luc Jaulin. 'Robust set-membership state estimation'. In: *Automatica* 45 (2009), pp. 202–206 (cit. on pp. 7, 26, 31, 61).

[Jaz70] AA Jazwinski. *Stochastic and Filtering Theory. Mathematics in Sciences and Engineering Series, Series 64*. 1970 (cit. on p. 15).

[Jaz07] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007 (cit. on p. 8).

[JZZ16] Chen Jiang, Shu-bi Zhang and Qiu-zhao Zhang. 'A Novel Robust Interval Kalman Filter Algorithm for GPS/INS Integrated Navigation'. In: *Journal of Sensors* 2016 (2016) (cit. on p. 32).

[JT08] Dongwon Jung and Panagiotis Tsiotras. 'Bank-to-turn control for a small UAV using backstepping and parameter adaptation'. In: (2008) (cit. on p. 57).

[Kal+60] Rudolph Emil Kalman et al. 'A new approach to linear filtering and prediction problems'. In: *Journal of basic Engineering* 82.1 (1960), pp. 35–45 (cit. on pp. 2, 13).

[KSG05] Rickard Karlsson, Thomas Schon and Fredrik Gustafsson. 'Complexity analysis of the marginalized particle filter'. In: *IEEE Transactions on Signal Processing* 53.11 (2005), pp. 4408–4411 (cit. on pp. 112, 181, 182).

[Kar+14] Toni Karvonen et al. 'Stability of linear and non-linear Kalman filters'. In: (2014) (cit. on p. 17).

[Kat+08] M Kato, S Sasaki, K Tanaka, Y Iijima and Y Takizawa. 'The Japanese lunar mission SELENE: Science goals and present status'. In: *Advances in Space Research* 42.2 (2008), pp. 294–300 (cit. on p. 53).

[KF00] O King and David A Forsyth. 'How does CONDENSATION behave with a finite number of samples?' In: *European Conference on Computer Vision*. Springer. 2000, pp. 695–709 (cit. on pp. 121, 190).

[KLW94] Augustine Kong, Jun S Liu and Wing Hung Wong. 'Sequential imputations and Bayesian missing data problems'. In: *Journal of the American statistical association* 89.425 (1994), pp. 278–288 (cit. on pp. 20, 80).

[KD03] Jayesh H Kotecha and Petar M Djuric. 'Gaussian sum particle filtering'. In: *IEEE Transactions on signal processing* 51.10 (2003), pp. 2602–2612 (cit. on p. 22).

[KRS00] Basil Kouvaritakis, J Anthony Rossiter and Jan Schuurmans. 'Efficient robust predictive control'. In: *IEEE Transactions on automatic control* 45.8 (2000), pp. 1545–1549 (cit. on p. 46).

[KT14] Harold W Kuhn and Albert W Tucker. 'Nonlinear programming'. In: *Traces and emergence of nonlinear programming*. Springer, 2014, pp. 247–258 (cit. on p. 45).

[Lee05] Deok-Jin Lee. 'Nonlinear Bayesian filtering with applications to estimation and navigation'. PhD thesis. Texas A&M University, 2005 (cit. on p. 13).

[LN13] Hao Li and Fawzi Nashashibi. 'Cooperative multi-vehicle localization using split covariance intersection filter'. In: *IEEE Intelligent transportation systems magazine* 5.2 (2013), pp. 33–44 (cit. on p. 41).

[LBD15] Tiancheng Li, Miodrag Bolic and Petar M Djuric. 'Resampling methods for particle filtering: classification, implementation, and strategies'. In: *IEEE Signal Processing Magazine* 32.3 (2015), pp. 70–86 (cit. on pp. 19, 173).

[Lim+10] D Limon, I Alvarado, T Alamo and EF Camacho. 'Robust tube-based MPC for tracking of constrained linear systems with additive disturbances'. In: *Journal of Process Control* 20.3 (2010), pp. 248–260 (cit. on p. 42).

[LG94] Lennart Ljung and Torkel Glad. 'On global identifiability for arbitrary model parametrizations'. In: *Automatica* 30.2 (1994), pp. 265–276 (cit. on p. 129).

[Lu+17] Qin Lu, Yaakov Bar-Shalom, Peter Willett, Francesco Palmieri and Fred Daum. 'The Multidimensional Cramér–Rao–Leibniz Lower Bound for Likelihood Functions With Parameter-Dependent Support'. In: *IEEE Transactions on Aerospace and Electronic Systems* 53.5 (2017), pp. 2331–2343 (cit. on pp. 175, 176).

[LQ18a] Jingwen Luo and Shiyin Qin. 'A Fast Algorithm of SLAM Based on Combinatorial Interval Filters'. In: *IEEE Access* 6 (2018), pp. 28174–28192 (cit. on pp. 35, 173).

[LQ18b] Jingwen Luo and Shiyin Qin. 'A Fast Algorithm of Simultaneous Localization and Mapping for Mobile Robot Based on Ball Particle Filter'. In: *IEEE Access* (2018) (cit. on p. 22).

[LP03] Fabien Lydoire and Philippe Poignet. 'Nonlinear predictive control using constraints satisfaction'. In: *International Workshop on Global Optimization and Constraint Satisfaction*. Springer. 2003, pp. 142–153 (cit. on p. 46).

[MN96] DG Maksarov and JP Norton. 'State bounding with ellipsoidal set description of the uncertainty'. In: *International Journal of Control* 65.5 (1996), pp. 847–866 (cit. on p. 26).

[Man93] James Manyika. 'An information-theoretic approach to data fusion and sensor management'. PhD thesis. University of Oxford, 1993 (cit. on pp. 40, 41, 120, 176).

[Mar03] F Landis Markley. 'Attitude error representations for Kalman filtering'. In: *Journal of guidance, control, and dynamics* 26.2 (2003), pp. 311–317 (cit. on p. 17).

[May82] Peter S Maybeck. *Stochastic models, estimation, and control*. Vol. 3. Academic press, 1982 (cit. on pp. 40, 176).

[MDPL16] Nicolas Merlinge, Karim Dahia and Hélène Piet-Lahanier. 'A Box Regularized Particle Filter for terrain navigation with highly non-linear measurements'. In: *IFAC-PapersOnLine* 49.17 (2016), pp. 361–366 (cit. on p. 72).

[Mes+14] Ali Mesbah, Stefan Streif, Rolf Findeisen and Richard D Braatz. 'Stochastic nonlinear model predictive control with probabilistic constraints'. In: *American Control Conference (ACC), 2014*. IEEE. 2014, pp. 2413–2419 (cit. on p. 47).

[MM89] Hanna Michalska and David Q Mayne. 'Receding horizon control of nonlinear systems'. In: *Decision and Control, 1989., Proceedings of the 28th IEEE Conference on*. IEEE. 1989, pp. 107–108 (cit. on p. 44).

[Mil64] Kenneth S Miller. *Multidimensional gaussian distributions*. John Wiley & Sons Inc, 1964 (cit. on p. 185).

[ML99] Manfred Morari and Jay H Lee. 'Model predictive control: past, present and future'. In: *Computers and Chemical Engineering* 23.4-5 (1999), pp. 667–682 (cit. on pp. 42, 44).

[Mur14] Achille Murangira. 'Nouvelles approches en filtrage particulaire. Application au recalage de la navigation inertielle'. PhD thesis. Université de Technologie de Troyes-UTT, 2014 (cit. on p. 53).

[MMD16] Achille Murangira, Christian Musso and Karim Dahia. 'A mixture regularized rao-blackwellized particle filter for terrain positioning'. In: *IEEE Transactions on Aerospace and Electronic Systems* 52.4 (2016), pp. 1967–1985 (cit. on pp. 19, 24).

[Mur+11] Achille Murangira, Christian Musso, Karim Dahia and Jean-Michel Allard. 'Robust regularized particle filter for terrain navigation'. In: *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*. IEEE. 2011, pp. 1–8 (cit. on pp. 23, 53, 106, 179).

[MOLG01] Christian Musso, Nadia Oudjane and François Le Gland. 'Improving Regularized Particle Filters'. In: *Sequential Monte Carlo Methods in Practice* 12 (2001), pp. 247–271 (cit. on pp. 2, 3, 17, 21–23, 62, 101, 171, 189).

[NFM16] Christopher Nemeth, Paul Fearnhead and Lyudmila Mihaylova. 'Particle approximations of the score and observed information matrix for parameter estimation in state–space models with linear computational cost'. In: *Journal of Computational and Graphical Statistics* 25.4 (2016), pp. 1138–1157 (cit. on p. 22).

[NRM09] Esha D Nerurkar, Stergios I Roumeliotis and Agostino Martinelli. 'Distributed maximum a posteriori estimation for multi-robot cooperative localization'. In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 1402–1409 (cit. on p. 40).

[Neu98] Arnold Neumaier. 'Solving ill-conditioned and singular linear systems: A tutorial on regularization'. In: *SIAM review* 40.3 (1998), pp. 636–666 (cit. on p. 91).

[Nic86] Karl Nickel. 'How to fight the wrapping effect'. In: *Interval Mathematics 1985*. Springer, 1986, pp. 121–132 (cit. on pp. 150, 151).

[Nor02] PJ Nordlund. 'Sequential Monte Carlo filters and integrated navigation Thesis no. 945'. In: *Linköping University, Linköping, Sweden* (2002) (cit. on p. 134).

[Nor98] James R Norris. *Markov chains*. 2. Cambridge university press, 1998 (cit. on p. 10).

[OJM08] Frauke Oldewurtel, Colin N Jones and Manfred Morari. 'A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback'. In: *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE. 2008, pp. 4731–4736 (cit. on p. 48).

[OS09] Reza Olfati-Saber. 'Kalman-consensus filter: Optimality, stability, and performance'. In: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE. 2009, pp. 7036–7042 (cit. on p. 120).

[Olv00] Peter J Olver. *Applications of Lie groups to differential equations*. Vol. 107. Springer Science & Business Media, 2000 (cit. on p. 17).

[Oud00] Nadia Oudjane. 'Stabilité et approximations particulaires en filtrage non linéaire application au pistage'. PhD thesis. Rennes 1, 2000 (cit. on pp. 23, 126).

[PAS09] BK Pagnoncelli, Shapiro Ahmed and Alexander Shapiro. 'Sample average approximation method for chance constrained programming: theory and applications'. In: *Journal of optimization theory and applications* 142.2 (2009), pp. 399–416 (cit. on pp. 3, 47, 149, 191).

[PP91] Athanasios Papoulis and Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes*. 1991 (cit. on p. 175).

[PDM03] D Pham, Karim Dahia and Christian Musso. 'A kalman-particle kernel filter and its application to terrain navigation'. In: *Proceedings 6th ICIF* (2003) (cit. on p. 24).

[Pha01] Dinh Tuan Pham. 'Stochastic methods for sequential data assimilation in strongly nonlinear systems'. In: *Monthly weather review* 129.5 (2001), pp. 1194–1207 (cit. on p. 20).

[PLW94] Hélène Piet-Lahanier and Eric Walter. 'Exact description of feasible parameter sets and minimax estimation'. In: *International journal of adaptive control and signal processing* 8.1 (1994), pp. 5–14 (cit. on p. 26).

[PW00] Florian A Potra and Stephen J Wright. 'Interior-point methods'. In: *Journal of Computational and Applied Mathematics* 124.1-2 (2000), pp. 281–302 (cit. on p. 45).

[PB17] Stefano Primatesta and Basilio Bona. 'Motion control of mobile robots with Particle Filter Model Predictive Equilibrium Point Control'. In: *Autonomous Robot Systems and Competitions (ICARSC), 2017 IEEE International Conference on*. IEEE. 2017, pp. 11–16 (cit. on p. 48).

[RB08] Wei Ren and Randal W Beard. *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008 (cit. on p. 37).

[Röb03] K Röbenack. 'Computation of Controllability and Observability Matrices–Duality and Automatic Differentiation'. In: *PAMM* 2.1 (2003), pp. 98–99 (cit. on p. 119).

[SN99] Alexander T Schwarm and Michael Nikolaou. 'Chance-constrained model predictive control'. In: *AIChE Journal* 45.8 (1999), pp. 1743–1752 (cit. on p. 48).

[Sch68] F.C. Schweppe. 'Recursive state estimation: unknown but bounded errors and system inputs'. In: *IEEE Trans. on Autom. Contr.* 13 (1968), pp. 22–28 (cit. on p. 26).

[ST08] Rajnikant Sharma and Clark Taylor. 'Cooperative navigation of MAVs in GPS denied areas'. In: *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*. IEEE. 2008, pp. 481–486 (cit. on pp. 40, 41).

[Shn98] Neryahu A Shneydor. *Missile guidance and pursuit: kinematics, dynamics and control*. Elsevier, 1998 (cit. on p. 42).

[Sil86] Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall, 1986 (cit. on pp. 17, 21–23, 101, 103, 189).

[Sim63] George F Simmons. *Introduction to topology and modern analysis*. Tokyo, 1963 (cit. on p. 183).

[Sun04] Shu-Li Sun. 'Multi-sensor information fusion white noise filter weighted by scalars based on Kalman predictor'. In: *Automatica* 40.8 (2004), pp. 1447–1453 (cit. on p. 42).

[TBF05] Sebastian Thrun, Wolfram Burgard and Dieter Fox. *Probabilistic robotics*. MIT press, 2005 (cit. on p. 8).

[Tic+98] Petr Tichavsky, Carlos H Muravchik, Nehorai and Arye. 'Posterior Cramér-Rao bounds for discrete-time nonlinear filtering'. In: *IEEE Transactions on signal processing* 46.5 (1998), pp. 1386–1396 (cit. on pp. 175, 176).

[USG18] USGS. *Map a Planet*. www.mapaplanet.org/. 2018. (Visited on 30/03/2018) (cit. on p. 53).

[Val97] David A Vallado. *Fundamentals of Astrodynamics and Applications*. Vol. 1. 1997 (cit. on p. 58).

[VM17] Manuel A Vázquez and Joaquín Míguez. 'A robust scheme for distributed particle filtering in wireless sensors networks'. In: *Signal Processing* 131 (2017), pp. 190–201 (cit. on p. 40).

[VVD86] Michel Verhaegen and Paul Van Dooren. 'Numerical aspects of different Kalman filter implementations'. In: *IEEE Transactions on Automatic Control* 31.10 (1986), pp. 907–917 (cit. on pp. 40, 112, 181, 182).

[Vie+17] Christophe Viel, Sylvain Bertrand, Michel Kieffer and Hélène Piet-Lahanier. 'Distributed event-triggered control for multi-agent formation stabilization'. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 8025–8030 (cit. on p. 142).

[WVDM00] Eric A Wan and Rudolph Van Der Merwe. 'The unscented Kalman filter for nonlinear estimation'. In: *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. Ieee. 2000, pp. 153–158 (cit. on p. 17).

[WCG13] Xiaogang Wang, Naigang Cui and Jifeng Guo. 'INS/VisNav/GPS relative navigation system for UAV'. In: *Aerospace Science and Technology* 28.1 (2013), pp. 242–248 (cit. on p. 42).

[Was13] Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013 (cit. on pp. 13, 88, 176).

[Zha+14] Hao Zhang, Gang Feng, Huaicheng Yan and Qijun Chen. 'Observer-Based Output Feedback Event-Triggered Control for Consensus of Multi-Agent Systems.' In: *IEEE Trans. Industrial Electronics* 61.9 (2014), pp. 4885–4894 (cit. on p. 142).

[ZL08] Huilong Zhang and Dann Laneuville. 'Grid based solution of Zakai equation with adaptive local refinement for bearings-only tracking'. In: *Aerospace Conference, 2008 IEEE*. IEEE. 2008, pp. 1–8 (cit. on p. 57).

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies
de l'information
et de la communication (STIC)

Titre : Estimation d'état et planification de trajectoire par mixtures de noyaux bornés

Mots clés : Filtrage particulaire, Estimation ensembliste, Filtre Particulaire à Boîtes, Probabilité d'échec, Commande sous incertitude, Navigation

Résumé : L'autonomie d'un engin aérospatial requière de disposer d'une boucle de navigation-guidage-pilotage efficace et sûre. Cette boucle intègre des filtres estimateurs et des lois de commande qui doivent dans certains cas s'accommoder de non-linéarités sévères et être capables d'exploiter des mesures ambiguës. De nombreuses approches ont été développées à cet effet et parmi celles-ci, les approches particulaires présentent l'avantage de pouvoir traiter de façon unifiée des problèmes dans lesquels les incertitudes d'évolution du système et d'observation peuvent être soumises à des lois statistiques quelconques. Cependant, ces approches ne sont pas exemptes de défauts dont le plus important est celui du coût de calcul élevé. D'autre part, dans certains cas, ces méthodes ne permettent pas non plus de converger vers une solution acceptable. Des adaptations récentes de ces approches, combinant les avantages du particulaire tel que la possibilité d'extraire la recherche d'une solution d'un domaine local de description et la robustesse des approches ensemblistes, ont été à l'origine du travail présenté dans cette thèse.

Cette thèse présente le développement d'un algorithme d'estimation d'état, nommé le Box Regularised Particle Filter (BRPF), ainsi qu'un algorithme de commande, le Box Particle Control (BPC). Ces algorithmes se basent tous deux sur l'utilisation de mixtures de noyaux bornés par des boîtes (i.e., des vecteurs d'intervalles) pour décrire l'état du système sous la forme d'une densité de probabilité multimodale. Cette modélisation permet un meilleur recouvrement de l'espace d'état et apporte une meilleure cohérence entre la prédite et la vraisemblance. L'hypothèse est faite que les incertitudes incriminées sont bornées. L'exemple d'application choisi est la navigation par corrélation de terrain qui constitue une application exigeante en termes d'estimation d'état.

Pour traiter des problèmes d'estimation ambiguë, c'est-à-dire lorsqu'une valeur de mesure peut correspondre à plusieurs valeurs possibles de l'état, le Box Regularised Particle Filter (BRPF) est introduit. Le BRPF est une évolution de l'algorithme de Box Particle Filter (BPF) et est doté d'une étape de ré-échantillonnage garantie et d'une stratégie de lissage par noyau (Kernel Regularisation). Le BRPF assure théoriquement une meilleure estimation que le BPF en termes de Mean Integrated Square Error (MISE). L'algorithme permet une réduction significative du coût de calcul par rapport aux approches précédentes (BPF, PF). Le BRPF est également étudié dans le cadre d'une intégration dans des architectures fédérées et distribuées, ce qui démontre son efficacité dans des cas multi-capteurs et multi-agents.

Un autre aspect de la boucle de navigation–guidage-pilotage est le guidage qui nécessite de planifier la future trajectoire du système. Pour tenir compte de l'incertitude sur l'état et des contraintes potentielles de façon versatile, une approche nommé Box Particle Control (BPC) est introduite. Comme pour le BRPF, le BPC se base sur des mixtures de noyaux bornés par des boîtes et consiste en la propagation de la densité d'état sur une trajectoire jusqu'à un certain horizon de prédiction. Ceci permet d'estimer la probabilité de satisfaire les contraintes d'état au cours de la trajectoire et de déterminer la séquence de futures commandes qui maintient cette probabilité au-delà d'un certain seuil, tout en minimisant un coût. Le BPC permet de réduire significativement la charge de calcul.

Title : State estimation and trajectory planning using box particle kernels

Keywords : Particle filter, Set-based estimation, Box Particle Filter, Chance constrained, Control under uncertainty, Navigation

Abstract : State estimation and trajectory planning are two crucial functions for autonomous systems, and in particular for aerospace vehicles. Particle filters and sample-based trajectory planning have been widely considered to tackle non-linearities and non-Gaussian uncertainties. However, these approaches may produce erratic results due to the sampled approximation of the state density. In addition, they have a high computational cost which limits their practical interest.

This thesis investigates the use of box kernel mixtures to describe multimodal probability density functions. A box kernel mixture is a weighted sum of basic functions (e.g., uniform kernels) that integrate to unity and whose supports are bounded by boxes, i.e., vectors of intervals. This modelling yields a more extensive description of the state density while requiring a lower computational load. New algorithms are developed, based on a derivation of the Box Particle Filter (BPF) for state estimation, and of a particle based chance constrained optimisation (Particle Control) for trajectory planning under uncertainty.

In order to tackle ambiguous state estimation problems, a Box Regularised Particle Filter (BRPF) is introduced. The BRPF consists of an improved BPF with a guaranteed resampling step and a smoothing strategy based on kernel regularisation. The proposed strategy is theoretically proved to outperform the original BPF in terms of Mean Integrated Square Error (MISE), and empirically shown to reduce the Root Mean Square Error (RMSE) of estimation. BRPF reduces the computation load in a significant way and is robust to measurement ambiguity. BRPF is also integrated to federated and distributed architectures to demonstrate its efficiency in multi-sensors and multi-agents systems.

In order to tackle constrained trajectory planning under non-Gaussian uncertainty, a Box Particle Control (BPC) is introduced. BPC relies on an interval bounded kernel mixture state density description, and consists of propagating the state density along a state trajectory at a given horizon. It yields a more accurate description of the state uncertainty than previous particle based algorithms. A chance constrained optimisation is performed, which consists of finding the sequence of future control inputs that minimises a cost function while ensuring that the probability of constraint violation (failure probability) remains below a given threshold. For similar performance, BPC yields a significant computation load reduction with respect to previous approaches.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France