



HAL
open science

Detection and tracking in Track-Before-Detect context with particle filter

Alexandre Lepoutre

► **To cite this version:**

Alexandre Lepoutre. Detection and tracking in Track-Before-Detect context with particle filter. Signal and Image processing. Université de Rennes, 2016. English. NNT : 2016REN1S101 . tel-01495993

HAL Id: tel-01495993

<https://theses.hal.science/tel-01495993v1>

Submitted on 27 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du signal et télécommunications

Ecole doctorale MATISSE

présentée par

Alexandre Lepoutre

préparée à l'ONERA centre de Palaiseau
INRIA Rennes – Bretagne Atlantique

**Détection et pistage
en contexte
Track-Before-Detect
par
filtrage particulière**

**Thèse soutenue à Palaiseau
le 5 octobre 2016**

devant le jury composé de :

Hans DRIESSEN

Ingénieur de recherche à Thales Nederland + Pro-
fesseur associé à Delft University of Technology / rap-
porteur

Eric GRIVEL

Professeur à l'ENSEIRB / rapporteur

Jean-Yves TOURNERET

Professeur à l'ENSEEIHTE / rapporteur

François LE CHEVALIER

Professeur émérite à Delft University of Technology
/ examinateur

Sylvie MARCOS

Directrice de recherche CNRS à Supélec / examina-
trice

François LE GLAND

Directeur de recherche INRIA / directeur de thèse

Olivier RABASTE

Ingénieur de recherche à l'ONERA / co-directeur de
thèse

Remerciements

Merci !

Contents

Résumé étendu	1
1 Radar signal processing and Bayesian filtering tools	11
1.1 Radar signal processing	12
1.1.1 General principle	12
1.1.2 Radar signal	13
1.1.3 The matched filter	15
1.1.3.1 Matched Filter definition and properties	15
1.1.3.2 The Matched Filter in the Detection Theory framework	16
1.1.3.3 The Matched Filter in radar	17
1.1.4 The ambiguity function	17
1.1.5 Pulse compression and linear frequency-modulated pulse	18
1.1.6 Coherent pulse train and Range-Doppler processing	22
1.1.7 Phase array processing	23
1.1.8 Measurement model	25
1.1.9 Detection and "hit" extraction	26
1.1.10 Radar tracking algorithms	28
1.1.10.1 Radar tracking objectives	28
1.1.10.2 Classic radar tracking algorithms	29
1.2 Bayesian filtering	29
1.2.1 Hidden Markov Models	30
1.2.2 Theoretical Bayesian filter	31
1.2.3 Linear Gaussian models: Kalman filter	32
1.2.4 Particle filter	33
1.2.4.1 Monte Carlo principle	34
1.2.4.2 Importance Sampling	35
1.2.4.3 Sequential Importance Sampling particle filter	36
1.2.4.4 Degeneracy problem	37
1.2.4.5 Instrumental density	39
1.2.4.6 Resampling	39
1.3 Conclusion	40
2 Monotarget Track-Before-Detect particle filters	43
2.1 Introduction	43
2.2 State model	44
2.2.1 General TBD model	44

2.2.2	Model used in this work	45
2.3	Measurement model	47
2.3.1	Raw radar data	47
2.3.2	Target Signal to Noise Ratio	48
2.4	A particle filter solution for the Track-Before-Detect problem	49
2.4.1	The TBD particle filter	49
2.4.2	Measurement likelihood	51
2.4.2.1	Eliminating the random phase	52
2.4.2.2	Dealing with the unknown parameter ρ	53
2.4.2.3	Truncating the ambiguity function	53
2.5	Instrumental density	54
2.5.1	Instrumental density for the initialization of the position	55
2.5.1.1	The optimal instrumental density	55
2.5.1.2	Approximating the instrumental density as a mixture	57
2.5.1.3	Calculation of the mixture probability and choice of the threshold	59
2.5.2	Instrumental density for the amplitude parameter	60
2.5.2.1	The optimal instrumental density	60
2.5.2.2	An instrumental density based on an estimator of the amplitude	62
2.5.3	Instrumental density for the velocity	63
2.5.4	Instrumental density for the presence variable	64
2.6	Marginalized TBD particle filter	65
2.7	Simulations and results	69
2.7.1	Scenarios	69
2.7.2	Methodology for the performance evaluation	71
2.7.2.1	Detection procedure	71
2.7.2.2	Evaluation of the detection performance	72
2.7.2.3	Evaluation of the estimation performance	73
2.7.3	Influence of the instrumental density	73
2.7.3.1	Influence of the Instrumental density for the position	74
2.7.3.2	Influence of the Instrumental density for the amplitude parameter	75
2.7.3.3	Influence of the Instrumental density for the velocity variable	76
2.7.3.4	Influence of the Instrumental density for the presence variable	77
2.7.4	Choice of the instrumental density	80
2.7.5	Influence of the target SNR	81
2.8	Conclusion	83
3	A novel approach for monotarget Track-Before-Detect	85
3.1	Introduction	85
3.2	A Bayesian solution for time appearance detection in TBD	86
3.2.1	State model	86
3.2.1.1	Time appearance model	86
3.2.1.2	Target state model	87

3.2.2	Measurement model	88
3.2.3	Theoretical Bayesian solution	88
3.2.3.1	Calculation of the posterior state density	89
3.2.3.2	Calculation of the mixture components	90
3.2.3.3	Calculation of the probabilities of appearance	90
3.2.4	Particle filter approximation	91
3.2.4.1	Approximation of the mixture components	92
3.2.4.2	Calculation of the probabilities of appearance	93
3.2.4.3	Dealing with the increasing number of particles and re- sampling strategies	94
3.3	Particle filter for target disappearance time detection	101
3.3.1	State model	102
3.3.2	Measurement model	103
3.3.3	Bayesian filter and particle filter approximation	103
3.4	Combination of particle filters for target appearance and disappearance detection	105
3.5	Simulations and results	106
3.5.1	Scenario	107
3.5.2	Methodology for the performance evaluation	107
3.5.3	Comparison between the ADD particle filter and the marginalized particle filter	108
3.6	Conclusion	111
4	Measurement equation and likelihood calculation for Track-Before-Detect applications	113
4.1	Introduction	113
4.2	Problem Formulation	114
4.2.1	Temporal coherence versus spatial coherence	115
4.2.2	State of the art	115
4.3	Likelihood calculation for Track-Before-Detect applications	117
4.3.1	Likelihood computation with complex measurements	117
4.3.1.1	Likelihood from the measurement equation	117
4.3.1.2	Marginalizing over the phase and modulus parameters	118
4.3.1.3	Dealing with the unknown static parameters of the mod- ulus fluctuation densities	120
4.3.2	Likelihood computation with squared modulus	120
4.3.2.1	The monotarget case	122
4.3.2.2	The multitarget case	124
4.4	Likelihood computation for Swerling models	124
4.4.1	Complex measurements	124
4.4.1.1	Swerling 0 case	124
4.4.1.2	Swerling 1 case	126
4.4.1.3	Swerling 3 case	127
4.4.2	Squared modulus measurements	128
4.4.2.1	The coherent case	129
4.4.2.2	The non coherent case	129

4.4.3	Summary	130
4.5	Simulation and Results	130
4.5.1	Single target simulation and results	131
4.5.1.1	Scenario of the simulation	131
4.5.1.2	Single target particle filter and performance evaluation	132
4.5.1.3	Simulations	133
4.5.2	Multitarget simulation and results	137
4.5.2.1	Multitarget scenario	137
4.5.2.2	Multitarget particle filter	137
4.5.2.3	Calculation of probability of track loss	138
4.5.2.4	Calculation of the Root Mean Square Error (RMSE)	139
4.5.2.5	Simulations	139
4.6	Conclusion	142
5	Multitarget Bayesian filter in Track-Before-Detect	145
5.1	Introduction	145
5.2	Classic Multitarget Bayesian Filter	145
5.2.1	Multitarget State Model	146
5.2.2	Theoretical Bayesian Filter	147
5.2.3	Particle filter approximation	148
5.2.4	The invariant permutation problem	148
5.2.5	Instrumental densities for the multitarget particle filter	150
5.2.6	Drawbacks of the existing solutions	151
5.3	A new approach for the multitarget Track-Before-Detect problem	152
5.3.1	A new Multitarget State Model	153
5.3.2	Measurement equation and likelihood for distant target	154
5.3.3	Theoretical Bayesian filter for non-interacting targets	155
5.3.4	Theoretical Bayesian filter for interacting targets	158
5.4	Particle filter approximations	159
5.4.1	Disappearance multitarget detection particle filter	159
5.4.1.1	Single and interacting targets particle filters	159
5.4.1.2	Outline of the proposed particle filter solution	161
5.4.2	Appearance Multitarget particle filter	161
5.4.2.1	Outline of the proposed solution	162
5.4.2.2	Managing the interaction between particles	163
5.4.2.3	Proposed instrumental density	163
5.4.3	Overall TBD multitarget particle filter	165
5.5	Simulation and Results	166
5.5.1	Non-interacting targets	166
5.5.2	Interacting targets	170
5.6	Conclusion	172
	Conclusion	173
	A Properties of time of appearance τ_b with a geometric prior	177

B Particle filter for time appearance detection in TBD	179
C Multitarget Bayesian filter and particle filters	183
C.1 Theoretical Bayesian filter for interacting targets	183
C.2 Disappearance multitarget detection particle filter	185
C.2.1 Calculating the sets I_{sing} and $I_{int,1:N_g}$ over time	185
C.2.2 Reorganization of the particle posterior density at previous step for the sets I_{sing} and $I_{int,1:N_g}$	187
C.2.3 Proposed solution for Disappearance multitarget particle filter . . .	187
Bibliography	196

Résumé étendu

Cette thèse a pour objectifs d'étudier et de développer de nouvelles méthodes de pistage radar d'une ou plusieurs cibles radars en contexte Track-Before-Detect par filtrage particulaire.

Brève définition du pistage dans un cadre bayésien

La problématique du filtrage et plus particulièrement du pistage consiste à estimer, à partir de mesures successives, l'état d'une variable non observée – par exemple, dans le cas du radar, la position et la vitesse des cibles – dont on a néanmoins une idée *a priori* de l'évolution au cours du temps – par exemple, on peut supposer qu'une cible est animée d'un mouvement rectiligne. C'est un problème extrêmement général et qui se retrouve dans de nombreux domaines tels que la finance, les télécommunications, la télémétrie, *etc.*

Il existe plusieurs approches pour résoudre ce problème, néanmoins nous nous limiterons dans cette thèse à l'approche bayésienne et plus particulièrement aux modèles de Markov cachés à temps discret. Ces modèles peuvent être globalement définis de la manière suivante¹ : l'état caché est défini par une variable aléatoire, notée \mathbf{x}_k ; l'évolution temporelle de l'état caché est modélisée par un processus de Markov qui est entièrement déterminé par sa densité à l'instant initial $p(\mathbf{x}_0)$ et sa densité de transition $p(\mathbf{x}_k | \mathbf{x}_{k-1})$. Cette dernière peut être définie par l'équation :

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_k, \mathbf{v}_k), \quad (1)$$

où $\mathbf{f}_k(\cdot)$ est une fonction non-linéaire connue et \mathbf{v}_k un bruit blanc. D'autre part, l'observation (ou la mesure), notée \mathbf{z}_k , est reliée à l'état caché par l'équation suivante (appelée équation de mesure) :

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{n}_k, \quad (2)$$

où $\mathbf{h}_k(\cdot)$ est une fonction non-linéaire connue et \mathbf{n}_k un bruit blanc. Cette équation permet notamment de calculer la vraisemblance de l'observation sachant l'état caché $p(\mathbf{z}_k | \mathbf{x}_k)$.

L'objectif est alors de calculer à chaque instant la densité $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ (appelée densité *a posteriori*) afin de calculer des estimateurs de l'état caché, tel que l'estimateur du Maximum *A Posteriori* ou encore l'estimateur MMSE (Minimum Mean Square Error). En règle générale on cherche à calculer cette densité de manière séquentielle ; en effet,

¹Ici, nous considérons un cadre simple où les processus étudiés peuvent être décrits à partir de densité ; il existe néanmoins des modélisations plus générales dont nous ne parlerons pas ici.

dans le cadre des modèles de Markov cachés la densité $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ s'écrit à partir de la densité à l'étape précédente à partir l'équation suivante :

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | \mathbf{x}_k)}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})}, \quad (3)$$

où :

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}. \quad (4)$$

De manière générale, l'Eq. (3) ne permet pas de calculer la densité $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ de manière analytique, excepté dans le cas du modèle linéaire et gaussien où la solution exacte est fournie par le filtre de Kalman. Quand le modèle est toujours gaussien et que les fonctions \mathbf{f}_k et \mathbf{h}_k ne présentent pas de fortes non-linéarités, des approximations du type EKF (Extended Kalman Filter) et UKF (Unscented Kalman Filter) peuvent être utilisées [AMGC02].

Dans les autres situations – fortes non-linéarités et/ou bruits non-gaussiens –, il est souvent nécessaire d'avoir recours à d'autres approximations pour obtenir des performances acceptables. L'une des solutions couramment utilisée aujourd'hui est le filtre particulaire. L'idée sous-jacente est d'approximer la densité continue $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ par une densité discrète, *i.e.* :

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta_{\mathbf{x}_k^i}(\mathbf{x}_k), \quad (5)$$

où $\delta_{\mathbf{x}_k^i}(\cdot)$ est la fonction de Dirac centrée en \mathbf{x}_k^i et les variables \mathbf{x}_k^i sont appelées particules avec w_k^i leur poids associé. L'avantage d'une telle approximation est qu'elle permet un fonctionnement séquentiel, ainsi l'approximation particulaire de la densité $p(\mathbf{x}_{k+1} | \mathbf{z}_{1:k+1})$ peut être obtenue par le mécanisme – relativement simple à mettre en oeuvre – suivant :

- Chaque particule \mathbf{x}_{k+1}^i est tirée à partir de la particule à l'instant précédent \mathbf{x}_k^i suivant une densité $q(\mathbf{x}_{k+1} | \mathbf{x}_k^i, \mathbf{z}_k)$, appelée densité instrumentale dont le choix est laissé à l'utilisateur. En pratique, la densité *a priori* $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$ issue du modèle d'état est souvent utilisée car la densité instrumentale optimale fournie par $p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ ne permet que rarement de tirer facilement des échantillons à partir de celle-ci.
- Ensuite les poids des particules sont mis à jour au moyen de l'équation suivante :

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{x}_{k+1}^i | \mathbf{x}_k^i) p(\mathbf{z}_k | \mathbf{x}_k^i)}{q(\mathbf{x}_{k+1}^i | \mathbf{x}_k^i, \mathbf{z}_k)}, \quad (6)$$

qui fait intervenir la densité *a priori* $p(\mathbf{x}_{k+1} | \mathbf{x}_k^i)$ et la vraisemblance de la mesure conditionnellement à l'état caché $p(\mathbf{z}_k | \mathbf{x}_k)$.

Le pistage radar classique

Le pistage radar consiste à créer, à partir d'une succession de mesures temporelles, des chaînes d'états successifs cohérents de la cinématique d'une cible. Pour un traitement

radar classique, ces mesures temporelles correspondent aux plots issus d'une étape préalable de détection et d'extraction appliquée au signal radar brut. Au delà du chaînage proprement dit, l'étape de pistage permet également d'améliorer l'estimation des différents paramètres des cibles (tels que la position, la vitesse, etc.), estimation limitée lors de l'étape préliminaire de détection/extraction par les caractéristiques du radar.

L'une des difficultés majeures du pistage radar tient tout d'abord à la structure de la mesure brute \mathbf{z}_k délivrée par le radar. En effet, celle-ci peut être vue comme un tableau multidimensionnel – dont chaque axe représente un des paramètres mesurés, tels que distance, angle, Doppler, etc. – pouvant contenir un nombre de cases (ou cellules) beaucoup plus important que le nombre de cibles d'intérêt et qui par conséquent peut être potentiellement difficile de traiter dans sa totalité, notamment dans le cadre d'une application temps réel. De ce fait, la première étape d'un traitement radar classique consiste à seuiliser la mesure \mathbf{z}_k et à ne garder qu'un nombre limité de cases radar susceptibles de contenir les différentes cibles présentes. A partir de ces cases radar, des plots de détection – qui sont une mesure bruitée des paramètres d'une cible potentielle – sont alors formés et permettent ainsi de réaliser l'étape de pistage proprement dite, qui est notamment vouée à créer, à partir des plots fournis au cours du temps, des chaînes d'états successifs cohérents de la cinématique d'une cible, et dans le même temps à améliorer l'estimation des différents paramètres d'état.

Dans l'approche classique, la méconnaissance de l'origine des différents plots de détection, qui peuvent être générés aussi bien par une des cibles présentes dans la fenêtre de veille que par des fausses alarmes, conduit à la nécessité d'associer chaque plot mesuré à une piste (existante ou nouvelle). Ce problème d'association plots/pistes ne présente pas de difficultés lorsque l'on cherche à pister des cibles à fort Rapport Signal à Bruit (RSB) ; en effet dans ce cas il suffit de choisir un seuil de détection élevé qui permet de limiter très fortement le nombre de fausses alarmes et la complexité du problème. Par contre, si l'on cherche à pister des cibles à faible RSB, il devient nécessaire de baisser le seuil de détection pour permettre la détection des cibles. Cela conduit à augmenter sensiblement le nombre de fausses alarmes et le problème d'association peut alors devenir beaucoup plus difficile à résoudre.

L'approche Track-Before-Detect

Comme nous venons de le voir, l'approche classique n'est pas forcément la plus adaptée pour détecter et pister des cibles à faible RSB – bien qu'elle permette par ailleurs une réduction importante de la taille de la mesure \mathbf{z}_k . Par conséquent, une nouvelle approche, connue sous le nom de Track-Before-Detect (TBD), a été proposée dont l'idée est simple : il s'agit de ne plus travailler à partir des données seuillées comme dans l'approche classique mais directement à partir des données radar brutes $\mathbf{z}_1, \dots, \mathbf{z}_k$. Le premier avantage est la suppression du problème d'association. D'autre part toute l'information présente dans les données est conservée laissant penser qu'il sera ainsi plus facile de pister des cibles à faible RSB dans ce cadre que dans le cadre classique.

Néanmoins, cette nouvelle approche n'est pas sans difficulté l'exploitation directe de la mesure \mathbf{z}_k conduit à un modèle de mesure plus difficile à appréhender que dans le cas

classique, modèle qui peut être défini par l'équation suivante :

$$\mathbf{z}_k = \sum_{i=1}^{N_k} \rho_{k,i} e^{j\varphi_{k,i}} \mathbf{h}(\mathbf{x}_{k,i}) + \mathbf{n}_k, \quad (7)$$

où :

- N_k est le nombre de cibles présentes dans la mesure.
- $\mathbf{x}_{k,i}$ est l'état de la $i^{\text{ième}}$ cible.
- les paramètres $\rho_{k,i}$ et $e^{j\varphi_{k,i}}$ définissent l'amplitude complexe de la cible i , inconnue et possiblement fluctuante (de manière aléatoire) au cours du temps.
- $\mathbf{h}(\cdot)$ est la fonction d'ambiguïté de la forme d'onde radar qui est connue.

Clairement, il s'avère que les contributions des différentes cibles sont sommées et donc mélangées dans le vecteur de mesure. Dans l'approche classique ce n'est généralement pas le cas dès lors que les cibles sont suffisamment distantes pour être résolues en sortie de filtre adapté. On peut alors faire l'hypothèse qu'à un plot ne peut être associé qu'une seule piste. De plus, la fonction $\mathbf{h}(\cdot)$ est souvent fortement non-linéaire, ce qui rend difficile l'utilisation de solutions telles que l'EKF ou l'UKF. Enfin, la présence des paramètres inconnus et fluctuants $\rho_{k,i}$ et $\varphi_{k,i}$ ne permet généralement pas de calculer directement la vraisemblance de la mesure conditionnellement aux états des cibles $p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k})$.

Au vu de ces difficultés, les premières solutions au problème de pistage dans le cadre du Track-Before-Detect ont d'abord été proposées dans le cas plus simple monocible – c'est-à-dire que l'on cherche à détecter l'apparition et/ou la disparition d'une et une seule cible. Parmi ces solutions on peut citer les solutions basées sur la transformée de Hough [CEW94], celles basées sur la programmation dynamique [Bar85] ou encore celles utilisant le filtre particulaire [SB01]. Suite à ces premières solutions du problème monocible, d'autres solutions ont été proposées dans le cadre plus général du pistage multicible [KKH05]. Dans cette thèse, nous nous intéresserons uniquement aux solutions particulières (mono comme multicibles), sans perdre d'esprit que d'autres travaux devraient être entrepris par la suite pour comparer les différentes solutions au problème TBD. L'objectif est dans ce travail de développer et d'étendre les solutions particulières existantes dans le cadre général du pistage mono ou multicibles en contexte TBD. En pratique, le problème monocible a d'abord été considéré.

Filtres particuliers monocibles en Track-Before-Detect

Filtre classique et lois instrumentales

En TBD, la présence ou l'absence de la cible n'est pas connue *a priori* et il est donc nécessaire de modéliser cette méconnaissance. Dans le cadre bayésien des modèles de markov cachés étudiés dans cette thèse, la méthode classique pour modéliser la présence ou l'absence de la cible consiste à utiliser une variable binaire s_k qui prend la valeur 1 quand la cible est présente et 0 sinon [SB01]. Ainsi, si on note \mathbf{x}_k l'état de la cible

(représentant sa position et sa vitesse, etc.), l'objectif du pistage est alors d'estimer l'état hybride (s_k, \mathbf{x}_k) au cours du temps. Pour ce faire, il est nécessaire de définir la densité de transition $p(s_k, \mathbf{x}_k | s_{k-1}, \mathbf{x}_{k-1})$ du modèle *a priori*. En règle générale, cette densité est factorisée de la manière suivante :

$$p(s_k, \mathbf{x}_k | s_{k-1}, \mathbf{x}_{k-1}) = p(s_k | s_{k-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}, s_k, s_{k-1}), \quad (8)$$

ce qui permet de modéliser le processus $(s_k)_{k \in \mathbb{N}}$ comme une chaîne de Markov à deux états, indépendante de l'état \mathbf{x}_k . Ensuite, il reste à modéliser la densité $p(\mathbf{x}_k | \mathbf{x}_{k-1}, s_k, s_{k-1})$. Bien qu'il y ait quatre cas de figure, dans les faits seuls les deux cas suivants, qui correspondent à une présence de la cible à l'instant k , sont nécessaires:

- le cas $s_k = 1$ et $s_{k-1} = 0$ qui correspond à l'apparition ou à la naissance de la cible. La variable \mathbf{x}_k est généralement initialisée uniformément dans l'espace d'état, pour modéliser l'absence de connaissance sur l'état de la cible.
- le cas $s_k = 1$ et $s_{k-1} = 1$ où la cible est déjà présente et qui modélise donc son évolution au cours du temps (par exemple, un mouvement rectiligne).

A partir du modèle d'état ainsi défini, un premier filtre particulaire a été proposé par Salmond *et al.* [SB01] afin d'approximer le filtre bayésien théorique qui n'est pas calculable en pratique. Dans le cadre du filtre particulaire, la densité instrumentale utilisée pour échantillonner les particules peut être choisie par l'utilisateur. Même si la loi souvent retenue est la loi *a priori* correspondant au modèle d'état, qui ne prend pas en compte l'information fournie par l'observation courante \mathbf{z}_k , il est tout à fait possible de la prendre en compte, notamment pour améliorer la performance du filtre en propageant les particules de manière plus efficace. En Track-Before-Detect, le cas réellement critique pour l'échantillonnage des particules est l'initialisation (ou la naissance). En effet, à cause de l'*a priori* uniforme sur la densité $p(\mathbf{x}_k | \mathbf{x}_{k-1}, s_k = 1, s_{k-1} = 0)$, il est nécessaire d'échantillonner l'ensemble de l'espace d'état, ce qui peut nécessiter un nombre très important de particules, généralement proportionnel au nombre de cases de résolution. Des approches heuristiques ont été proposées dans la littérature afin de résoudre ce problème en exploitant l'information fournie par l'observation courante, notamment par Salmond *et al.* Toutefois les solutions proposées n'étaient pas nécessairement justifiées théoriquement. Ainsi, nous proposons au chapitre 2 de nouvelles lois instrumentales dérivées à partir d'approximations de la densité instrumentale optimale $p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ – qui n'est pas calculable en pratique. Par exemple, le filtre particulaire développé par Salmond *et al.* échantillonne la variable s_k à partir de la loi *a priori* qui ne tient pas compte de l'observation \mathbf{z}_k . Nous montrons qu'il est en fait possible de prendre en compte l'observation en échantillonnant la variable s_k à partir de la loi *a posteriori* $p(s_k | s_{k-1}, \mathbf{z}_k)$.

Finalement, nous comparons sur simulation les lois instrumentales proposées avec celles classiquement utilisées dans la littérature. Ces simulations illustrent l'importance de l'initialisation des particules (notamment la position) et montrent qu'il peut être plus intéressant d'utiliser une loi instrumentale différente de la loi *a priori* fournie par le modèle d'état que de simplement augmenter le nombre de particules.

Modélisation alternative du problème TBD monocible

Dans certains cas de figure, et notamment à faible RSB, le fait de continuer à initialiser des particules alors que le filtre a déjà convergé sur la cible peut biaiser l'estimation. Il est donc légitime de remettre en cause la nécessité d'initialiser des particules quand le filtre a déjà convergé, et ce d'autant plus que l'initialisation des particules est relativement coûteuse en temps de calcul. Partant de ce constat, nous avons considéré une stratégie alternative permettant d'effectuer la détection de l'apparition de la cible et sa disparition avec des filtres différents. Les prémices d'une telle solution se trouvent dans les travaux de Kligys *et al.* [KRT98] qui proposent une modélisation du problème TBD comme un problème de détection de changement : il s'agit alors d'estimer le plus rapidement possible un changement de densité de probabilité tout en minimisant la probabilité d'erreur. Dans le cas du TBD, le changement survient quand la cible apparaît : on passe alors de la densité de probabilité du bruit seul à une densité de probabilité décentrée par la contribution de la cible (voir Eq. 7). Néanmoins, la solution de Kligys *et al.* n'est pas développée dans le cadre des Modèles de Markov cachés. Nous proposons donc dans cette thèse une solution originale adoptant cette modélisation du problème TBD comme un problème de détection de changement et dérivons le filtre particulaire correspondant. Ainsi, au chapitre 3 le modèle d'état considéré modélise non plus l'évolution du couple (s_k, \mathbf{x}_k) au cours du temps mais l'évolution du couple (τ_b, \mathbf{x}_k) où τ_b est l'instant d'apparition de la cible. Un modèle d'état similaire peut être considéré pour la disparition de la cible. Dans les deux cas, nous dérivons les équations des filtres bayésiens correspondants ainsi que des approximations particulières pour chacun d'eux. Enfin, nous proposons un filtre particulaire combinant ces deux filtres afin de gérer à la fois l'apparition et la disparition de la cible. Les simulations effectuées permettent de montrer l'intérêt de séparer la détection de l'apparition et de la disparition notamment au niveau de temps de calcul du filtre mais également en matière d'estimation (surtout à faible RSB).

Calcul de la vraisemblance en Track-Before-Detect

Un autre problème important qui se pose en TBD concerne le calcul de la vraisemblance de la mesure conditionnellement à l'état des cibles $p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k})$, qui est nécessaire pour la mise en oeuvre du filtre bayésien. Or cette vraisemblance ne peut pas être calculée directement à partir de l'équation de mesure (7) du fait de la présence des paramètres d'amplitudes $\rho_{k,i}$, $\varphi_{k,i}$ qui sont inconnus et peuvent fluctuer d'itération à itération. En radar, les fluctuations du module $\rho_{k,i}$ sont généralement modélisées par un des modèles *Swerling* : pour le modèle Swerling 0, le module est supposé constant et donc non fluctuant, tandis que les modèles Swerling 1 et 3 modélisent des fluctuations lentes (de rafale à rafale) de l'amplitude cible, et les modèles de Swerling 2 et 4 modélisent des fluctuations rapides (d'impulsion à impulsion). La phase $\varphi_{k,i}$ est quant à elle supposée uniformément distribuée sur l'intervalle $[0, 2\pi[$. Dans la mesure où le modèle de mesure considéré est développé au niveau de la rafale, nous ne considérons dans cette thèse que les modèles de Swerling 0, 1 et 3, soit l'absence de fluctuation ou une fluctuation lente de la cible.

Plusieurs solutions ont été proposées dans la littérature pour s'affranchir de ces paramètres d'amplitude et ainsi permettre de calculer la vraisemblance des observations

$p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k})$. La première solution proposée [RRG05, DRC08, BDV⁺03] consiste à travailler sur les modules des échantillons complexes $|z_k^l|^2$. En effet, cette solution permet de calculer la vraisemblance de manière simple dans le cas monocible. Par contre, cette solution conduit à perdre l'information de cohérence spatiale de la phase de la cible, *i.e.* le fait que la phase de la cible est la même pour tous les échantillons de la mesure \mathbf{z}_k . Cette perte d'information peut être préjudiciable pour les performances comme démontré par Davey *et al.* [DRC12]. D'autre part, nous avons montré que l'extension de cette solution au cas multicible est loin d'être simple, sauf dans le cas *Swerling 1* où une expression analytique de la vraisemblance peut être obtenue.

Afin de palier la perte de la cohérence spatiale sur le module, Davey *et al.* [DRC12] ont proposé dans le cas monocible une autre approche qui consiste à travailler directement à partir de la mesure complexe \mathbf{z}_k et à marginaliser la densité $p(\mathbf{z}_k | \mathbf{x}_k, \rho_k, \varphi_k)$ (qui peut être obtenue facilement à partir de l'Eq. (7)) par rapport à la variable φ_k , soit :

$$p(\mathbf{z}_k | \mathbf{x}_k, \rho_k) = \int p(\mathbf{z}_k | \mathbf{x}_k, \rho_k, \varphi_k) p(\varphi_k) d\varphi_k, \quad (9)$$

où $p(\varphi_k)$ est la densité uniforme sur $[0, 2\pi[$. Contrairement à l'approche précédente, la cohérence spatiale de la phase est ici conservée. Davey *et al.* montrent alors que l'Eq. (9) est calculable de manière analytique. Dans le cas *Swerling 0*, la vraisemblance $p(\mathbf{z}_k | \mathbf{x}_k)$ nécessaire pour le filtrage particulière est alors simplement obtenue en remplaçant la variable ρ_k par la valeur du paramètre. Pour les modèles *Swerling 1* et *3*, il est nécessaire de marginaliser également la densité $p(\mathbf{z}_k | \mathbf{x}_k, \rho_k)$ par rapport au module ; dans ce cas, aucune formule analytique n'a jusqu'alors été fournie. Suite à cette constatation, nous avons tout d'abord étendu l'approche proposée par Davey *et al.* pour les modèles de fluctuations *Swerling 1* et *3* dans le cas monocible. Nous montrons, dans le chapitre 4 cette thèse, que la marginalisation de la densité $p(\mathbf{z}_k | \mathbf{x}_k, \rho_k)$ suivant le paramètre ρ_k est calculable de manière exacte pour les modèles *Swerling 1* et *3*. Dans un second temps, nous avons considéré le problème de la marginalisation des paramètres d'amplitude dans le cas multicible. Nous obtenons une expression analytique uniquement dans le cas *Swerling 1* ; pour les autres modèles de fluctuations, nous proposons néanmoins des approximations permettant le calcul des vraisemblances en un temps raisonnable. Enfin, nous montrons par simulation l'intérêt d'utiliser la mesure complexe \mathbf{z}_k au lieu des modules carrés dans le cas monocible pour les fluctuations *Swerling 1* et *3*, et dans le cas multicibles pour les fluctuations *Swerling 0, 1* et *3*.

Filtres particuliers multicibles en Track-Before-Detect

Précédemment nous avons donné un bref aperçu de la modélisation du problème Track-Before-Detect en monocible avec l'utilisation de la variable discrète s_k . La modélisation généralement utilisée dans le cadre multicible suit une idée similaire avec l'introduction d'une variable aléatoire supplémentaire modélisant le nombre de cible présent, sauf que dans le cas multicible ce nombre n'est plus limité par 1. En notant N_k le nombre de cibles à l'instant k , le but du pistage est alors d'estimer la densité $p(N_k, \mathbf{x}_{k,1:N_k} | \mathbf{z}_{1:k})$ au cours du temps.

A partir de cette modélisation du problème multicible, Kreucher *et al.* ont proposé un filtre particulière permettant d'approximer le filtre bayésien théorique. Celui-ci échantillonne pour chaque particule un nombre de cibles N_k^i et les états des cibles associés $\mathbf{x}_{k,1:N_k^i}^i$, ce qui permet d'écrire l'approximation particulière de la manière suivante :

$$p(\mathbf{x}_{k,1:N_k}, N_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta_{\mathbf{x}_{k,1:N_k^i}^i}(\mathbf{x}_{k,1:N_k}). \quad (10)$$

Malgré cette approximation particulière de la densité *a posteriori*, l'estimation du nombre de cibles ainsi que de leur état reste difficile en pratique. En effet, la densité *a posteriori* est invariante par permutation des états de cibles – par exemple, pour deux cibles, $p(\mathbf{x}_{k,1}, \mathbf{x}_{k,2} | \mathbf{z}_{1:k}) = p(\mathbf{x}_{k,2}, \mathbf{x}_{k,1} | \mathbf{z}_{1:k})$. Par conséquent, si les états des cibles $\mathbf{x}_{k,1:N_k^i}^i$ pour chaque particule ne sont pas ordonnés, il n'est pas possible d'estimer correctement l'état des cibles correspondantes. C'est pourquoi Kreucher *et al.* préconise une étape supplémentaire de clustering afin d'ordonner les différents états des particules en partition représentant chacune une cible. D'autre part, le fait de considérer des particules multicibles implique que le poids de la particule, obtenu par le produit des vraisemblances des différents états échantillonnés par cette particule, représente uniquement un comportement global de la particule pour l'ensemble des états, mais ne reflète pas la qualité des différents états en particulier. En pratique, on peut alors obtenir des particules échantillonnant correctement un certain nombre d'états et incorrectement d'autres états ; les poids de ces particules ne permettront pas de distinguer les états correctement échantillonnés des autres états, ce qui pourra conduire à une détérioration de la qualité d'estimation du filtre. Suite à ces constatations, nous proposons dans le chapitre 5 une modélisation permettant de découpler les différentes cibles quand celles-ci sont éloignées les unes des autres ; on utilise alors simplement des filtres différents et indépendants pour pister les différentes cibles. Ainsi, l'étape de clusterisation n'est plus nécessaire et par construction les cibles sont indépendantes les unes des autres. Pour ce faire, nous ne considérons plus la variable N_k modélisant le nombre de cibles et pouvant varier au cours du temps, mais plutôt un nombre constant N_t de couples $(s_{k,l}, \mathbf{x}_{k,l})$ correspondant à la modélisation monocible du problème TBD ; N_t représente le nombre maximum de cibles que le filtre particulière peut gérer conjointement. Nous montrons qu'avec cette modélisation, lorsque les cibles n'interagissent pas entre elles, le filtre bayésien peut être factorisé comme suit :

$$p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t}) = \prod_{l=1}^{N_t} p(s_{k,l}, \mathbf{x}_{k,l}), \quad (11)$$

ce qui permet effectivement l'emploi d'un filtre par cible. Par contre, lorsque des cibles sont proches, elles doivent être traitées conjointement. Il reste toutefois possible de traiter séparément les groupes de cibles proches et les cibles isolées.

De manière similaire au chapitre 3, nous proposons alors trois filtres particuliers, l'un pour effectuer la détection de l'apparition de plusieurs cibles, le second pour gérer la disparition, et le dernier qui combine ces deux premières solutions pour gérer à la fois l'apparition et la disparition. Cette approche est validée sur simulation en considérant deux scénarios simples, l'un où trois cibles à faible RSB sont présentes mais n'interagissent pas entre elles et un autre où deux cibles à fort RSB se croisent.

Contributions

Dans cette thèse nous nous sommes intéressés au problème du pistage monocible et multicible en contexte Track-Before-Detect par filtrage particulaire. Concernant le pistage monocible, nous avons tout d'abord proposé de nouvelles lois instrumentales pour l'initialisation des particules et montré par simulation qu'elles apportaient un gain significatif tant au niveau de la détection que de l'estimation. Ces travaux ont fait l'objet d'une communication [LRLG12a]. Par ailleurs, nous avons proposé une modélisation alternative originale du problème TBD monocible basée sur l'instant d'apparition ou de disparition de la cible. Ainsi, nous avons proposé trois filtres particulaires, le premier pour détecter l'apparition de la cible, le second pour détecter sa disparition, et le dernier qui combine les deux filtres précédents pour gérer conjointement l'apparition et la disparition. Finalement, nous montrons par simulation l'intérêt de séparer la détection de l'apparition et de la détection notamment en matière de temps de calcul mais également en ce qui concerne l'estimation (surtout à faible RSB). Cette solution originale a été partiellement présentée dans la communication [LRLG12b].

Ensuite, nous nous sommes intéressés au calcul de la vraisemblance en contexte Track-Before-Detect. Nous avons étendu les travaux de Davey *et al.* permettant le calcul de la vraisemblance en tenant compte de la cohérence spatiale des paramètres d'amplitude pour des fluctuations *Swerling 0*, à d'autres modèles de fluctuations (*Swerling 1* et *3*) et aux trois modèles de fluctuation dans le cas multicible. Ainsi, nous avons montré que dans le cas monocible, des expressions analytiques de la vraisemblance pouvaient être obtenues pour les fluctuations *Swerling 1* et *3* ; dans le cas multicible, nous obtenons une expression analytique uniquement dans le cas *Swerling 1* ; néanmoins pour les autres modèles nous proposons des approximations permettant de calculer la vraisemblance en un temps raisonnable. Ces travaux ont fait l'objet d'une première communication en conférence [LRG13] puis d'une publication plus avancée acceptée dans la revue *IEEE Transactions on Aerospace and Electronic Systems* [LRLG16].

Finalement dans la dernière partie de cette thèse, nous nous sommes intéressés au pistage multicible. Notre démarche a consisté à mettre en place une solution permettant d'une part d'exploiter au maximum l'indépendance des cibles entre elles afin d'utiliser autant que possible un filtre par cible plutôt que des filtres multicibles, et d'autre part, comme pour le cas monocible, de séparer la détection de l'apparition et de la disparition. Ainsi, nous avons montré qu'il était possible d'étendre la modélisation du problème monocible au cas multicible et que le filtre multicible résultant pouvait être factorisé par un produit de filtres monocibles dès lors que les cibles sont suffisamment éloignées les unes des autres. Nous avons alors proposé comme dans le cas monocible trois filtres particuliers : un pour la détection des apparitions, un second pour la gestion des croisements et des disparitions et enfin un dernier filtre réunissant les deux filtres précédents.

Liste des publications

Publications dans des conférences internationales et nationales avec actes :

1. M. Bocquel, A. Lepoutre, O. Rabaste, and F. Le Gland. Optimisation d'un filtre particulaire en contexte track-before-detect. In *Actes du 23ème Colloque sur le Traitement du Signal et des Images*, Bordeaux, France, September 2011
2. A. Lepoutre, O. Rabaste, and F. Le Gland. Optimized instrumental density for particle filter in track-before-detect. In *Data Fusion Target Tracking Conference : Algorithms Applications, 9th IET*, pages 1–6, London, United Kingdom, May 2012
3. O. Rabaste, C. Riché, and A. Lepoutre. Long-time coherent integration for low SNR target via particle filter in track-before-detect. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 127–134, July 2012
4. A. Lepoutre, O. Rabaste, and F. Le Gland. A particle filter for target arrival detection and tracking in track-before-detect. In *Sensor Data Fusion: Trends, Solutions, Applications (SDF), Workshop on*, pages 13–18, Bonn, Germany, Sept. 2012
5. A. Lepoutre, O. Rabaste, and F. Le Gland. Exploiting amplitude spatial coherence for multi-target particle filter in track-before-detect. In *Information Fusion (FUSION), 16th International Conference on*, pages 319–326, Istanbul, Turkey, July 2013
6. A. Lepoutre, O. Rabaste, and F. Le Gland. Filtres particuliers en contexte track-before-detect en présence de fluctuations d'amplitude de type Swerling 1 et 3. In *Actes du 24ème Colloque sur le Traitement du Signal et des Images*, Brest, France, September 2013. GRETSI

Publication dans un journal international à comité de lecture :

1. A. Lepoutre, O. Rabaste, and F. Le Gland. Multitarget likelihood computation for track-before-detect applications with amplitude fluctuations of type Swerling 0, 1, and 3. *Aerospace and Electronic Systems, IEEE Transactions on*, June 2016

Chapter 1

Radar signal processing and Bayesian filtering tools

The whole classic radar chain from the signal reception to the tracking stage can be decomposed into three different steps, as illustrated in Figure 1.1.

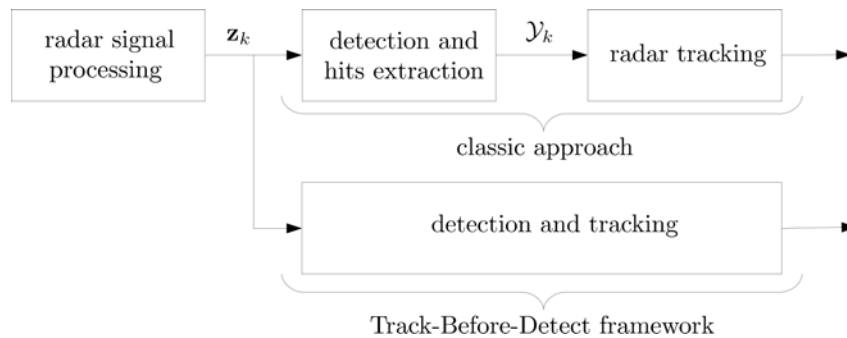


Figure 1.1 – Block diagram of the classic radar chain from the signal processing stage to the tracking stage. The Track-Before-Detect processing takes place after the signal processing stage.

The first stage, denoted here "radar signal processing", is performed in order to improve the target Signal to Noise Ratio (SNR), thus allowing to detect and estimate the target parameters (such as range, radial velocity, azimuth,...). In the classic radar chain, the "radar signal processing" stage provides a measurement \mathbf{z}_k as an input to the "detection and hit extraction". This next step consists first in thresholding the radar measurement \mathbf{z}_k and then in extracting the potential target parameters from any signal sample (called "hit") that passed the detection threshold. At the end of this step, a set of detection hits \mathcal{Y}_k is provided to the tracking step. This last stage takes advantage of some target motion information (*e.g.* a linear trajectory) to enhance the estimation of the target parameters over time. Moreover it enables to discriminate over time the "hits" that come from the targets from the ones that are due to false alarms in order to form tracks.

In practice, "the detection and hit extraction" stage allows to dramatically reduce the amount of data to process – indeed the size of the measurement \mathbf{z}_k may be very large (it is a multidimensional array that may contain several tens of thousands of cells) whereas, if the threshold is conveniently chosen to limit the false alarms, the set \mathcal{Y}_k is

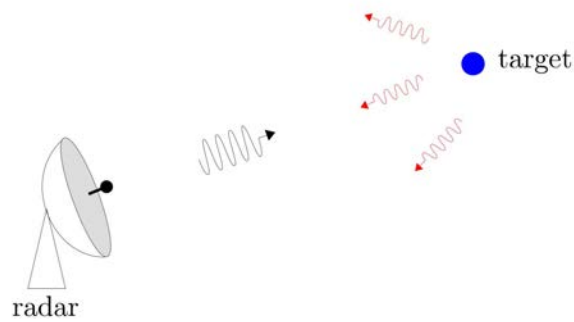


Figure 1.2 – General radar principle. In black the transmitted signal and red the signal reflected in all the directions.

much smaller – but in return some information is lost that may be detrimental, especially if some targets have a low SNR. A new framework, known as Track-Before-Detect, has therefore been proposed and consists in jointly performing detection and tracking from the measurement \mathbf{z}_k rather than from the set of detection "hits", as illustrated in Figure 1.1. This framework will be at the heart of this thesis.

Before going further into the details of the Track-Before-Detect strategy, we propose first in this chapter to present some aspects of the radar signal processing theory and of the Bayesian filtering theory that will be useful along this manuscript. In particular, in section 1.1, we present the main signal processing tools used to transform the received signal into the output measurements \mathbf{z}_k and the detection hits \mathcal{Y}_k while in section 1.2 we outline the Bayesian filtering tools that are used in the radar tracking stage.

1.1 Radar signal processing

1.1.1 General principle

A RADAR (RADio Detection and Ranging) is an electromagnetic system consisting of an antenna that transmits a signal with a particular waveform and then receives and detects the signal backscattered by any scatterer present in the scene, among which possibly one or several targets (such as aircrafts, vessels, *etc.*). This principle is illustrated in Figure 1.2. Then by measuring the duration τ of the round trip between the radar and the target, it is possible to calculate the corresponding range R with the following relationship:

$$R = \frac{c\tau}{2}, \quad (1.1)$$

where c is the speed of the electromagnetic wave. Furthermore, due to the motion of the target, the signal received by the radar may be shifted in frequency compared to the transmitted one: this is the so called Doppler effect. The frequency shift between the transmitted signal and the received one is approximately equal to $\frac{2\dot{R}}{c}f_0$ where f_0 is the frequency of the transmitted signal and \dot{R} the radial velocity. Therefore by measuring this Doppler shift, it is possible to extract the radial velocity of the target.

This is basically the very general radar principle. However, in practice measuring the delay and Doppler is not as simple as it looks. Indeed, the transmitted signal will be

attenuated, and only a portion of backscattered energy will be reflected by the target in the radar direction. Therefore, at the reception side, the received signal is passed through a complete reception chain that allows to recover the backscattered signal with some additive noise. Two questions can then be raised:

- Does the received signal contain one (or several) target contribution(s) or only noise ?
- How to accurately estimate the delay and Doppler parameters while the received signal is corrupted by noise ?

The first question corresponds to a detection problem; the detection theory [Kay98] provides a convenient framework to solve this problem in a radar context, in the form of the Neyman-Pearson criteria; that is to say maximizing the probability of detecting the signal (if present) while ensuring a given false alarm probability. In many application, and in particular in radar, this detection procedure involves the matched filter [Tur60, Woo53], that will be presented in section 1.1.3.

The second question corresponds to an estimation problem which is often solved using the Maximum Likelihood criteria, *i.e.* finding the value of the parameter maximizing the likelihood that the signal occurs with the corresponding parameter value. In practice, this maximization often leads to find the maximum output of the matched filter and is highly related to the characteristics of the transmitted signal (in particular the duration and the frequency bandwidth). Thus, in paragraph 1.1.4 and 1.1.5, we expose very shortly the tools used to study their properties and detail a very common signal used in radar.

Lastly, we would like to highlight that the purpose of this section is not to extensively study all the aspects of the radar theory¹ but rather to provide a realistic but simple model for the input data used to perform the radar tracking stage, and in particular the Track-Before-Detect methods that represent the heart of this work.

1.1.2 Radar signal

A radar signal is constituted of two parts, first a baseband signal with band B and duration T_p and then a carrier f_0 (usually such that $B \ll f_0$) allowing to carry the signal through the air. The transmitted signal $s(t)$ can be written in a complex formalism as

$$s(t) = Eu(t)e^{j2\pi f_0 t}, \quad (1.2)$$

where $u(\cdot)$ is the complex envelop of the baseband signal with energy equal to one and E is the energy of signal $s(\cdot)$. At the reception side, if the transmitted signal has been reflected by a target (or any backscatter), the radar receives a signal $s_r(\cdot)$ which is an attenuated replica of the transmitted signal delayed by the time $\tau(t)$ taken by the electromagnetic wave to make the round trip between the radar and the target:

$$s_r(t) = \rho' e^{j\varphi'} u(t - \tau(t)) e^{j2\pi f_0(t - \tau(t))}, \quad (1.3)$$

where $\rho' e^{j\varphi'}$ is a complex coefficient of attenuation that is unknown and random:

¹Readers wishing to deepen the radar theory may refer to [Sko80, Rih69, Dar94, LC89]

- the phase φ' is assumed to be uniformly drawn over $[0, 2\pi)$;
- the modulus ρ' is subject to random fluctuations usually modeled in radar processing by a *Swerling* model that will be detailed in chapter 4.

$\tau(\cdot)$ is a function of time t . Noticing that the signal received at a given time t was reflected by the target at time $t - \frac{\tau(t)}{2}$, the function $\tau(\cdot)$ verifies the following relationship [Rih69]:

$$c\tau(t) = 2R\left(t - \frac{\tau(t)}{2}\right), \quad (1.4)$$

with $R(\cdot)$ the range between the target and the radar with respect to time t . In practice, the function $\tau(\cdot)$ may be difficult to calculate. Thus, it is generally approximated by its Taylor expansion [Rih69]. We consider here an approximation of order one which is the common hypothesis made in radar – Note however that higher orders may be required for highly manoeuvring targets. The Taylor polynomial of order 1 of $\tau(\cdot)$ around time t_0 such that $\tau_0 = \tau(t_0)$ is given by [Rih69]:

$$\tau(t) = \tau_0 + \dot{\tau}_0(t - \tau_0), \quad (1.5)$$

where:

- $\tau_0 = \frac{2R_0}{c}$, with $R_0 = R(\frac{\tau_0}{2})$.
- $\dot{\tau}_0 = \frac{2\dot{R}_0}{c}(1 + \frac{\dot{R}_0}{c})^{-1} \approx \frac{2\dot{R}_0}{c}$, with $\dot{R}_0 = \dot{R}(\frac{\tau_0}{2})$ the relative radial velocity between the target and the radar. Note that the approximation of $\dot{\tau}_0$ is valid for usual target velocity verifying $\dot{R}_0 \ll c$.

Then, by replacing $\tau(t)$ by its polynomial approximation, the received signal $s_r(t)$ can be rewritten as follows:

$$s_r(t) = \rho' e^{j\varphi'} u((t - \tau_0)(1 - \beta)) e^{2\pi f_0(1-\beta)(t-\tau_0)}, \quad (1.6)$$

with $\beta = \frac{2\dot{R}_0}{c}$. The target motion induces a compression/dilatation effect on the baseband signal and a Doppler shift both on the carrier. Fortunately, the time compression dilatation induced by the factor $1 - \beta$ over the baseband signal can be neglected as long as $\frac{2\dot{R}_0}{c} \ll 1$ and the only effect to take into account on the complex envelop is then the delay τ_0 . On the contrary, the Doppler shift on the carrier must be taken into account since the multiplication by f_0 induces a fast phase rotation equal to $-2\pi f_0 \beta t$. For instance, with $f_0 = 3\text{GHz}$, $T_p = 100\mu\text{s}$ and $\dot{R}_0 = -300\text{m.s}^{-1}$, the phase rotation after a duration T_p is equal to $-2\pi f_0 \beta T_p = 216^\circ$, which may not be negligible depending on the signal considered.

Finally, the received signal $s_r(t)$ is passed through the reception chain that consists, in particular, in demodulating – an intermediate step that consists in removing the carrier $e^{j2\pi f_0 t}$ – and in amplifying the received signal, and becomes:

$$s_r(t) = \rho e^{j\varphi} u(t - \tau_0) e^{j2\pi\nu_0 t} + n(t), \quad (1.7)$$

where $\nu_0 = -f_0\beta$ is the Doppler shift, $\varphi = \varphi' + 2\pi f_0(1 - \beta)\tau_0$ a random phase, ρ the amplified modulus, and $n(t)$ a stationary complex Gaussian noise with autocorrelation function

$$\gamma_n(s) = \mathbb{E}[n(t)n^*(t-s)] = 2\sigma^2\delta(s) \quad (1.8)$$

due to the reception chain, where $\delta(s)$ is the delta mass Dirac function at point zero. Finally, the baseband signal in Eq. (1.7) is processed by the radar processing chain in order to perform target detection and parameter estimation. The basic tool of this processing chain is the matched filter.

1.1.3 The matched filter

The matched filter is widely used in many applications, for instance radar, sonar, telecommunication, in order to detect a signal with a known waveform corrupted by noise. Roughly speaking, the matched filter consists in calculating the correlation between the received signal and the known waveform; the detection is then performed by comparing the output signal with a given threshold γ .

1.1.3.1 Matched Filter definition and properties

A filter is called a matched filter for a physical waveform $u(t)$ with energy E if its impulse response $h(t)$ has the form [Tur60]

$$h_u(t) = Ku^*(t_a - t), \quad (1.9)$$

where K and t_a are arbitrary constants. The matched filter impulse response is a conjugate time-reversed version of the physical waveform $u(t)$.

Then, for a received signal of the form

$$r(t) = u(t - \tau_0) + n(t), \quad (1.10)$$

where τ_0 is here assumed to be known and $n(t)$ is a stationary Gaussian complex noise with autocorrelation function defined in Eq. (1.8), the output $r_{MF,h_u}(\cdot)$ of the matched filter is obtained by convolving the received signal $r(t)$ with the impulse response $h_u(t)$. By setting $t_a = 0$ and $K = 1$, this leads to

$$\begin{aligned} r_{MF,h_u}(\tau) &= (h_u \star r)(\tau) \\ &= \underbrace{\int_{-\infty}^{\infty} u(t - \tau_0) u^*(t - \tau) dt}_{r_{MF,u}(\tau)} + \underbrace{\int_{-\infty}^{\infty} n(t) u^*(t - \tau) dt}_{n_{MF}(\tau)}, \end{aligned} \quad (1.11)$$

which consists of two terms $r_{MF,u}(\tau)$ and $n_{MF}(\tau)$. $r_{MF,u}(\tau)$ is the autocorrelation function of the deterministic signal $u(t)$ delayed by τ_0 , *i.e.* $r_{MF,u}(\tau) = R_u(\tau - \tau_0)$, where

$$R_u(\tau') = \int_{-\infty}^{\infty} u(t) u^*(t - \tau') dt.$$

Therefore $r_{MF,u}(\tau)$ is maximum for $\tau = \tau_0$ and $r_{MF,u}(\tau_0) = E$. The second term $n_{MF}(\tau)$ is still a stationary Gaussian complex noise with autocorrelation function:

$$\begin{aligned}\gamma_{n_{MF}}(\tau_s) &= \mathbb{E}[n_{MF}(\tau)n_{MF}^*(\tau - \tau_s)] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbb{E}[n(t)n^*(s)]u^*(t - \tau)u(s - \tau + \tau_s) dt ds \\ &= 2\sigma^2 \int_{-\infty}^{\infty} u(t)u^*(t - \tau_s) dt = 2\sigma^2 R_u(\tau_s).\end{aligned}\tag{1.12}$$

This last equation means that even though the input noise is white, the output noise is, in general, not white since it depends on the signal autocorrelation $R_u(\tau_s)$. Finally, by defining the Signal to Noise Ratio (SNR) output as

$$\text{SNR}(\tau) = \frac{|r_{MF,u}(\tau)|^2}{\mathbb{E}[|n_{MF}(\tau)|^2]},\tag{1.13}$$

it can be easily shown that the matched filter is the linear filter that maximizes the SNR output for $\tau = \tau_0$ [Tur60, LM04], given by

$$\text{SNR}(\tau_0) = \frac{E}{2\sigma^2}.\tag{1.14}$$

1.1.3.2 The Matched Filter in the Detection Theory framework

In order to illustrate the fundamental role played by the matched filter in the detection theory, let us consider the following statistical hypothesis-testing problem [Tur60, Kay98]

$$\begin{cases} \mathcal{H}_0 : s_r(t) = n(t), t \in [0, T_r] \\ \mathcal{H}_1 : s_r(t) = u(t - \tau_0) + n(t), t \in [0, T_r]. \end{cases}\tag{1.15}$$

where $u(t)$ is any signal waveform assumed to be known, τ_0 a delay also assumed to be known and T_r is the time during which the received signal has been observed. The decision over the hypotheses \mathcal{H}_0 and \mathcal{H}_1 can lead to two types of errors:

- Either decide hypothesis \mathcal{H}_1 whereas hypothesis \mathcal{H}_0 is true. Such an error is called a false alarm and we denote by P_{fa} the corresponding probability of false alarm.
- Or decide hypothesis \mathcal{H}_0 whereas a target is present. This a miss detection and its corresponding miss detection probability is denoted by P_{md} . Lastly, the probability of detection P_D is defined by $P_D = 1 - P_{md}$.

These two decision errors behave in an opposite manner: trying to decrease the P_{fa} will lead to increase P_{md} and reciprocally. Therefore, a trade-off must be found and the classic criteria, called the Neyman-Pearson criteria, consists in maximizing the probability of detection P_D while ensuring a given P_{fa} . The optimal detector, for this criteria, is provided by the Neyman-Pearson theorem [Tur60, Kay98]; it consists in comparing the ratio between the likelihood of the signal $s_r(t)$ under hypothesis \mathcal{H}_1 and the likelihood

of the same signal under hypothesis \mathcal{H}_0 . Under a white Gaussian noise assumption, the optimal detector is provided by the following procedure [LC89]

$$\text{accept } \mathcal{H}_1 \text{ if } \operatorname{Re} \left(\int_0^{T_r} s_r(t) u^*(t - \tau_0) dt \right) > \gamma, \quad (1.16)$$

where $\operatorname{Re}(\cdot)$ stands for the real part. Thus, the detection scheme consists in comparing the output of the matched filter sampled at $\tau = \tau_0$ with a threshold γ calculated in order to ensure the given P_{fa} .

1.1.3.3 The Matched Filter in radar

In radar, the received signal depends on unknown parameters (delay τ_0 , Doppler shift ν_0 , complex amplitude $\rho e^{j\varphi}$). As a consequence, the decision problem becomes a composite hypothesis-testing problem [Kay98], and procedure (1.16) cannot be applied directly. An heuristic procedure, called GLRT (Generalized Likelihood Ratio Test), was then proposed: it consists in estimating these parameters in the maximum likelihood sense and injecting them in the likelihood ratio test. From Eq.(1.7), the radar composite hypothesis testing problem has the form

$$\begin{cases} \mathcal{H}_0 : s_r(t) = n(t), t \in [0, T_r] \\ \mathcal{H}_1 : s_r(t) = \rho e^{j\varphi} u(t - \tau) e^{j2\pi\nu t} + n(t), t \in [0, T_r], \end{cases} \quad (1.17)$$

where $(\varphi, \rho, \tau, \nu)$ are the unknown parameters. Using the GLRT heuristic, and since the maximization over parameters (ρ, φ) can be easily obtained and does not depend on parameter (τ, ν) , the detection test becomes [LC89]:

$$\text{accept } \mathcal{H}_1 \text{ if } \max_{(\tau, \nu)} \frac{\left| \int_0^{T_r} s_r(t) u^*(t - \tau) e^{-j2\pi\nu t} dt \right|^2}{\int_0^{T_r} |u(t - \tau)|^2 dt} > \gamma. \quad (1.18)$$

Furthermore, if we define by $h_{u,\nu}(\cdot)$ the impulse response of the filter matching the signal $u(t)e^{j2\pi\nu t}$, *i.e.*,

$$h_{u,\nu}(t) = u^*(-t) e^{j2\pi\nu t}, \quad (1.19)$$

the detection procedure can be finally rewritten as

$$\text{accept } \mathcal{H}_1 \text{ if } \max_{(\tau, \nu)} \frac{|s_{r, MF, h_{u,\nu}}(\tau)|^2}{\int_0^{T_r} |u(t - \tau)|^2 dt} > \gamma, \quad (1.20)$$

which consists in comparing the maximum output of the matched filter in range and Doppler with a given threshold. In practice, the maximum is rarely available in closed form; search for the maximum may be then performed by applying several matched filters adapted to different Doppler hypotheses ν_i .

1.1.4 The ambiguity function

In the previous paragraph, the matched filter has been presented from a detection point of view. Nevertheless, in radar applications, retrieving information on the target parameters

τ_0 and ν_0 is also of interest. Estimating these parameters in the maximum likelihood sense is equivalent to find the values (τ, ν) maximizing the matched filter output. Intuitively, in order to obtain good estimation performance, the energy of the matched filter should concentrate in a narrow peak around (τ_0, ν_0) . Of course, the output of the matched-filter is dependent on the choice of the waveform $u(t)$ and, as a consequence, the choice of the waveform impacts the estimation performance. It is thus of importance to study the behaviour of the matched filter output for a particular waveform $u(t)$ with respect to parameters τ and ν ; this is provided by the ambiguity function.

In order to introduce the ambiguity function, let us rewrite the received signal $s_r(t)$ defined in Eq. (1.7) after a matched filter operation with impulse response $h_{u,\nu}(\cdot)$:

$$\begin{aligned} s_{r,MF,\nu}(\tau) &= (h_{u,\nu} \star s_r)(\tau) \\ &= \rho e^{j\varphi} e^{j2\pi\nu\tau} \chi_u(\tau - \tau_0, \nu - \nu_0) + n_{MF,\nu}(\tau) \end{aligned} \quad (1.21)$$

where $n_{MF,\nu}(\tau)$ is the noise component after the matched filtering step and

$$\chi_u(\tau, \nu) = \int_{-\infty}^{+\infty} u(t) u^*(t - \tau) \exp(-j2\pi\nu t) dt, \quad (1.22)$$

The function $\chi_u(\tau, \nu)$ is called the ambiguity function²[LM04, LC89]. It corresponds to the output of the matched filter in absence of noise. Its maximum is obtained at the origin (*i.e.* $\tau = 0$ and $\nu = 0$) and corresponds to the energy of the signal $u(t)$. Therefore, if we want to accurately estimate these parameters in presence of noise the waveform $u(t)$ has to be chosen such that it ensures the narrowest peak around the origin of the ambiguity function.

Another important requirement for the radar is its capability to resolve close targets. This capability of a radar to resolve two close targets is often measured with the delay Δ_τ and Doppler Δ_ν resolutions defined as follows:

$$|\chi_u(\Delta_\tau, 0)|^2 = \frac{1}{2}, \quad |\chi_u(0, \Delta_\nu)|^2 = \frac{1}{2}, \quad (1.23)$$

that correspond to 3 dB losses along the range or along the Doppler axis. Note that the resolution, both in delay and Doppler, is often approximated by the first null of the ambiguity function since it is easier to calculate and provides values quite close to the ones obtained by the actual definition. Finally, the range resolution Δ_r and the range rate (radial velocity) resolution $\Delta_{\dot{r}}$ are related to the delay and Doppler resolutions by the following relationships:

$$\Delta_r = \frac{c}{2} \Delta_\tau \quad \text{and} \quad \Delta_{\dot{r}} = \frac{c}{2f_0} \Delta_\nu. \quad (1.24)$$

1.1.5 Pulse compression and linear frequency-modulated pulse

Clearly two different waveforms will provide two different ambiguity functions, as well as their corresponding delay and Doppler resolutions. Delay and Doppler resolutions

²Note that other definitions are possible, in particular, using $+\tau$ and $+\nu$ rather than $-\tau$ and $-\nu$ in the integral (1.22). However, it is only a convention and it does not change the results on the ambiguity function, in particular the ones provided by Levanon *et al.* [LM04] which will be used in the sequel.

often behave in an opposite manner, *i.e.* a better resolution in delay will lead to a poorer resolution in Doppler and reciprocally. To illustrate this, let us consider the simple following pulse:

$$u_{UP}(t) = \begin{cases} \frac{1}{\sqrt{T_p}}, & \text{if } |t| \leq \frac{T_p}{2}, \\ 0, & \text{otherwise,} \end{cases} \quad (1.25)$$

denoted as unmodulated pulse (or constant pulse). The ambiguity function for this signal is provided by [LM04]

$$\chi_{UP}(\tau, \nu) = \begin{cases} \left(1 - \frac{|\tau|}{T_p}\right) \frac{\sin(\pi T_p \nu)}{\pi T_p \nu}, & \text{if } |\tau| \leq T_p, \\ 0, & \text{otherwise.} \end{cases} \quad (1.26)$$

The zero-Doppler cut and the zero-delay cut are then obtained respectively by setting $\nu = 0$ and $\tau = 0$ in Eq. (1.26), which gives:

$$\chi_{UP}(\tau, 0) = \left(1 - \frac{|\tau|}{T_p}\right), \text{ if } |\tau| \leq T_p, \text{ zero elsewhere,} \quad (1.27)$$

$$\chi_{UP}(0, \nu) = \frac{\sin(\pi T_p \nu)}{\pi T_p \nu}. \quad (1.28)$$

The delay and Doppler resolution for the unmodulated pulse are respectively equal to:

$$\Delta_{\tau,UP} \approx T_p \text{ and } \Delta_{\nu,UP} \approx \frac{1}{T_p}, \quad (1.29)$$

leading to the corresponding range and range rate resolution,

$$\Delta_{r,UP} \approx \frac{cT_p}{2} \text{ and } \Delta_{\dot{r},UP} \approx \frac{c}{2f_0 T_p}, \quad (1.30)$$

Thus one cannot obtain simultaneously a good delay and a good Doppler resolution with this single pulse. In addition, in a more general perspective, for most of the signals used in radar the delay resolution is related to the inverse signal bandwidth³ $1/B$, *i.e.* higher the bandwidth, smaller the delay resolution; on the contrary the Doppler resolution is related to the inverse of the integration duration, *i.e.* $1/T_p$ in the case of the constant pulse.

Pulse compression is a technique widely used in radar and sonar in order to improve the range resolution. The main idea is to increase the bandwidth of the unmodulated transmitted signal. In the sequel, we outline this technique for a common signal used in radar, that is the Linearly Frequency Modulated (LFM) pulse signal (commonly known as a chirp pulse) that consists in sweeping linearly the frequency bandwidth B during the pulse duration T_p [LM04]:

$$u_C(t) = \begin{cases} \frac{1}{\sqrt{T_p}} \exp(j\pi k t^2), & \text{if } |t| \leq \frac{T_p}{2}, \\ 0, & \text{otherwise,} \end{cases} \quad (1.31)$$

³Note that the delay resolution of the constant pulse seems to depend only on the pulse duration, however it can be shown that for this signal the bandwidth is approximately equal to $1/T_p$ leading to the corresponding delay resolution.

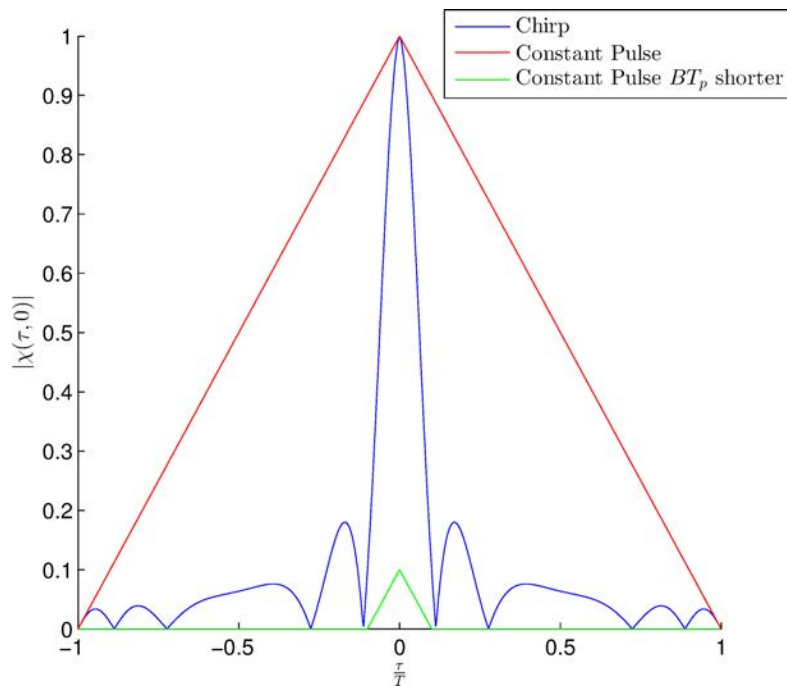


Figure 1.3 – Comparison of the zero-Doppler cut $|\chi_C(\tau, 0)|$ for a chirp signal (in blue) with a time-bandwidth product of $BT_p = 10$, an unmodulated pulse (in red) with duration T_p and an unmodulated pulse (in green) with duration BT_p shorter than the two others.

with $k = \frac{B}{T_p}$ (k can also be negative). The ambiguity of the chirp signal is given by [LM04]

$$\chi_C(\tau, \nu) = \begin{cases} \left(1 - \frac{|\tau|}{T_p}\right) \frac{\sin\left[\pi T_p \left(\nu + B \frac{\tau}{T_p}\right) \left(1 - \frac{|\tau|}{T_p}\right)\right]}{\pi T_p \left(\nu + B \frac{\tau}{T_p}\right) \left(1 - \frac{|\tau|}{T_p}\right)}, & \text{if } |\tau| \leq T_p, \\ 0, & \text{otherwise,} \end{cases} \quad (1.32)$$

The zero-Doppler cut is obtained by setting $\nu = 0$ in Eq. (1.32), *i. e.*

$$\chi_C(\tau, 0) = \frac{\sin\left[\pi B \tau \left(1 - \frac{|\tau|}{T_p}\right)\right]}{\pi B \tau}, \text{ if } |\tau| \leq T_p, \text{ zero elsewhere.} \quad (1.33)$$

while the zero-delay cut is the same as the unmodulated pulse (see Eq. (1.28)). In figure 1.3, the zero-Doppler cut of the ambiguity function of the LFM pulse is presented and compared first to an unmodulated pulse of same duration, and second to an unmodulated pulse of smaller duration enabling the same range resolution. It appears clearly from this figure that the use of the frequency modulation allows to dramatically improve the delay resolution and therefore the range resolution, when considering only the zero-Doppler cut. It also illustrates the gain in energy enabled by the chirp compared to an unmodulated pulse of the same maximum power but with a duration BT_p shorter and thus providing the same range resolution as the chirp.

The delay resolution for the chirp is approximately

$$\Delta_{\tau, C} \approx \frac{1}{B}, \quad (1.34)$$

which corresponds in range to

$$\Delta_{r,C} \approx \frac{c}{2B}. \quad (1.35)$$

Let us illustrate the gain between the chirp and the unmodulated pulse for typical radar parameter values. For an unmodulated pulse of duration $T_p = 100 \mu\text{s}$ and a chirp with the same duration and a bandwidth $B = 1 \text{ MHz}$, the widths of the zero-Doppler cuts are respectively equal to

$$\Delta_{r,UP} \approx \frac{cT_p}{2} = 15000 \text{ m}, \text{ and } \Delta_{r,C} \approx \frac{c}{2B} = 150 \text{ m}.$$

The chirp thus provides an improvement of a factor BT_p compared to the unmodulated pulse (here indeed $BT_p = 100$).

Note that until now, the delay and the Doppler has been studied independently. In particular, the cut for $\nu = 0$ has been considered while in practice this Doppler may be different from zero. Indeed if a matched filter is performed with the null Doppler hypothesis, from equation (1.21) the ambiguity function will be shifted by the target Doppler ν_0 . For the unmodulated pulse it has no consequence since the location of the maximum in delay is $\tau = 0$ whatever the value of ν_0 (see Eq. (1.26)). On the contrary, for the chirp signal, a coupling is induced between parameters τ and ν , so that the maximum in delay does not occur at $\tau = 0$ anymore but is shifted (for reasonable value of ν) by the quantity [LM04]

$$\tau_{shift} = \frac{\nu}{k} = \frac{\nu T_p}{B}. \quad (1.36)$$

This coupling phenomenon is illustrated in Figure 1.4 where the maximum peak in delay is shifted along the diagonal $\tau = \frac{\nu}{k}$. Let us make the correspondence in term of range shift, *i.e.*

$$R_{shift} = \frac{c\tau_{shift}}{2} = \frac{c}{2B} \times \nu T_p = -\Delta_r \frac{2\dot{R}_0 f_0 T_p}{c}, \quad (1.37)$$

and illustrate it with a numerical example. For a target with radial velocity (or range rate) $\dot{R}_0 = -300 \text{ m.s}^{-1}$ and the following radar parameters: $B = 1 \text{ MHz}$ (*i.e.* $\Delta_r = 150 \text{ m}$), $T_p = 100 \mu\text{s}$ (leading to $BT_p = 100$) and $f_0 = 3 \text{ GHz}$, the maximum of the matched-filter (with hypothesis $\nu = 0$) is shifted by

$$R_{shift} = 90 \text{ m},$$

i.e. 60% of the range resolution Δ_r . Note that a small decrease in energy is observed along the diagonal $\tau = \nu/k$ that is equal, near the origin, to [LM04]

$$|\chi_C(\tau_{peak}, \nu)| = 1 - \left| \frac{\nu}{B} \right|, \quad (1.38)$$

providing a negligible loss of 0.052 dB for the same numerical values as previously. Finally, note also that for the same radial velocity and pulse duration, the loss observed with the constant pulse or any classic phase code would be greater than 3 dB. This means that, when considering the chirp signal, a single matched filter at Doppler hypothesis $\nu = 0$ is sufficient to get a high output energy, even for large target radial velocities. The price to pay for this cheap processing is a possible non negligible range bias.

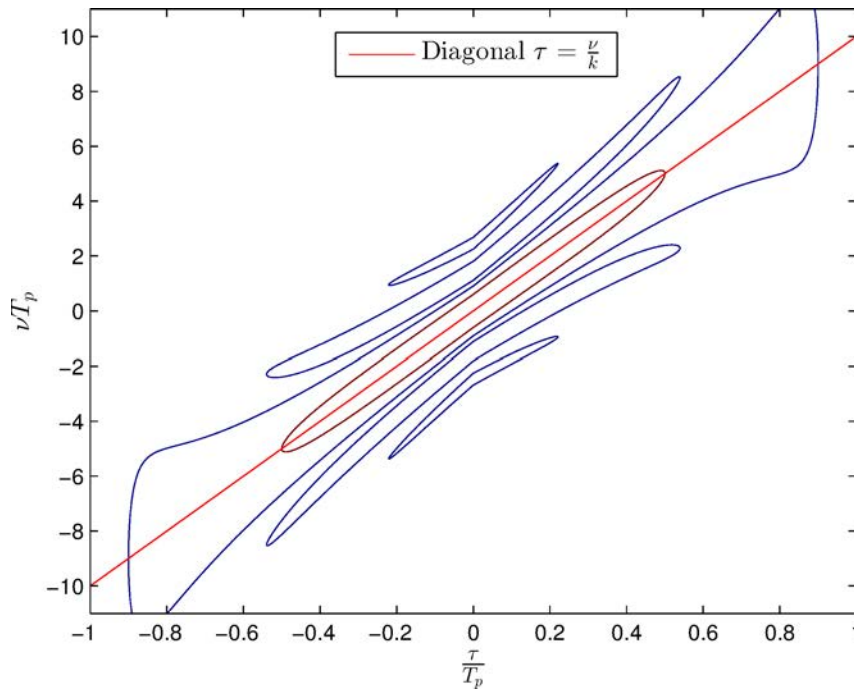


Figure 1.4 – 0.1 and 0.707 contours of the chirp ambiguity function with $BT_p = 10$. For a mismatch Doppler ν the maximum in delay is shifted along the diagonal $\tau = \frac{\nu}{k}$.

In summary, the chirp pulse allows to improve the resolution by a factor BT_p compared to the unmodulated pulse while ensuring the same amount of energy, at the price of a coupling between the delay τ and the Doppler ν . This coupling provides advantages and drawbacks: on one hand it induces an ambiguity between delay and Doppler parameters that remains acceptable for most applications. On the other hand, it results in a good tolerance to Doppler shift, *i.e.* the loss induced by a Doppler mismatch when applying a filter matched to hypothesis $\nu = 0$ is small even for large Doppler shifts, allowing to use a low cost processing.

1.1.6 Coherent pulse train and Range-Doppler processing

The Doppler resolution (and thus the velocity resolution) depends on the integration time. For the parameters used in previous section, the velocity resolution is approximately equal to $500 \text{ m}\cdot\text{s}^{-1}$, which is clearly not acceptable. A possible solution to get a good Doppler resolution is then to transmit a long pulse. However, this leads to an unacceptable blind range – indeed, during the transmission of the signal, the radar does not receive any signal and therefore cannot detect a target with a delay lower than T_p . A better solution consists in using a coherent pulse train, *i.e.* several identical pulses are transmitted at a given repetition period T_r . For a coherent pulse train of length N , the complex envelop of the band limited signal is given by [LM04]

$$u_N(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} u(t - kT_r), \quad (1.39)$$

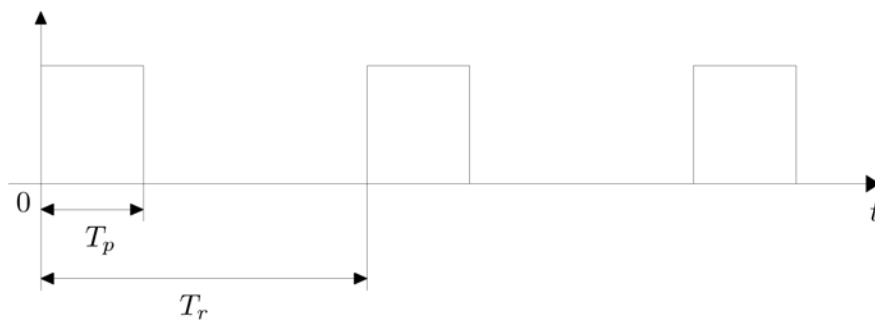


Figure 1.5 – Pulse train of 3 pulses with pulse duration T_p and repetition period T_r .

where $u(t)$ is any waveform with duration T_p . An example of a coherent pulse train is presented in Figure 1.5. For such a pulse train, the range and Doppler processing can be decoupled. Indeed as seen before, the range matched filter to the chirp will compress the signal whatever the Doppler. The first step of the Range-Doppler processing consists then in performing a range matched-filter with the transmitted elementary pulse. The signal after the range matched-filter can be expressed as follows [LC89]:

$$s_{r,MF}(\tau) = \sum_{k=1}^{N-1} e^{j2\pi\nu kT_r} \int_0^{T_p} s_r(t + kT_r + \tau) u^*(t) \exp(j2\pi\nu t) dt. \quad (1.40)$$

However, the use of a coherent pulse train is not without consequences since this creates an ambiguity in delay every T_r , due to the periodicity of the transmitted signal: it is impossible to know if the detected target return comes from a target delayed by $0 \leq \tau \leq T_r$ or by a target delayed by $mT_r \leq \tau \leq (m+1)T_r$ where m is any integer greater than one. Therefore the delay is measured modulo T_r .

In a second step, for each delay τ , a Fast Fourier transform is performed in order to coherently integrate the phases $e^{j2\pi\nu kT_r}$ in (1.40) and thus provide an estimate of the Doppler parameter ν . Since the overall integration time considered by this range-Doppler processing is equal to the total duration of the pulse train NT_r , the Doppler resolution becomes equal to $\frac{1}{NT_r}$. However, since the phases $e^{j2\pi\nu kT_r}$ are ambiguous modulo $\frac{1}{T_r}$, the Doppler measurement provided by the pulse train also becomes ambiguous.

1.1.7 Phase array processing

Range and Doppler parameters are not sufficient to fully locate a target: it is also necessary to estimate its angular direction. Phase array processing [VT02] is a convenient framework to estimate the target azimuth and/or elevation. In the following, the principle of the latter for the case of a linear array with N_a isotropic elements uniformly spaced by distance d is briefly recalled. Let us define by p_{x_m} the position of the m^{th} element along the x -axis, given by (assuming that the center of the array is located at the origin)

$$p_{x_m} = \left(m - \frac{N-1}{2} \right) d, \quad m = 0, 1, \dots, N_a - 1. \quad (1.41)$$

This linear array is presented in Figure 1.6. For a target located at angle θ_T , range R_0

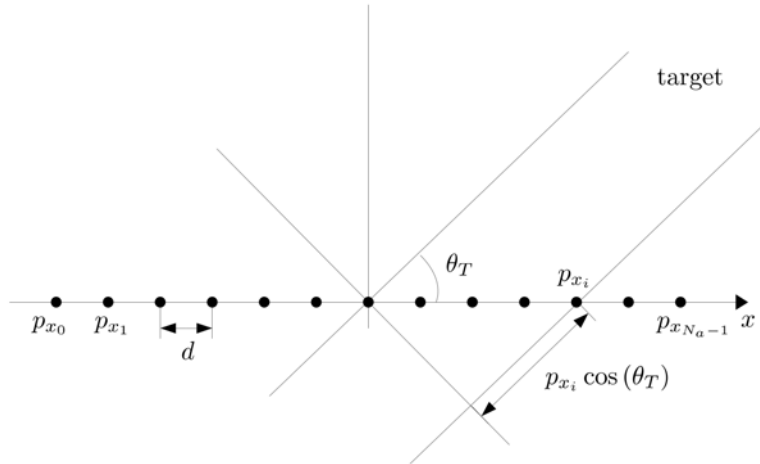


Figure 1.6 – Linear array along the x -axis with a target in the direction θ_T .

from the radar with a Doppler shift ν_0 , the phase of the signal received by each elementary antenna will differ due to the different travel time of the wave (as illustrated in Figure 1.6). Thus the signal received by antenna m can be written as

$$s_{r,m}(t) = \rho e^{j\varphi} u(t - \tau_0) e^{j2\pi\nu_0 t} e^{j\frac{2\pi}{\lambda} p_{x_m} \cos(\theta_T)} + n_m(t), \quad m = 0, \dots, N_a - 1, \quad (1.42)$$

where $n_m(t)$ is a stationary complex white Gaussian noise, and, for $m \neq q$, the noise processes $n_m(t)$ and $n_q(t)$ are assumed to be independent. Finally λ denotes the wavelength of the transmitted wave (*i.e.* $\lambda = \frac{c}{f_0}$).

The differential phase $e^{j\frac{2\pi}{\lambda} p_{x_m} \cos(\theta_T)}$ depends on the target direction θ_T and on the position of the elementary antenna. For instance, if a target is located in a direction $\theta_T = \pi/2$, the differential phase will be the same on all antennas, while in a direction $\theta_T = 0$, the differential phase between two consecutive antennas will be equal to π .

The aim of array processing is to recover the target direction from the phase difference measured on each receiving antenna. This can be done by applying a digital beamforming at the reception that consists in correlating the antenna outputs with the steering vector corresponding to the direction θ under test. This steering vector consists of the differential phases for this direction and is thus given by $\mathbf{v}_\theta = \frac{1}{N_a} \left[e^{j\frac{2\pi}{\lambda} p_{x_0} \cos(\theta)}, \dots, e^{j\frac{2\pi}{\lambda} p_{x_{N_a-1}} \cos(\theta)} \right]^T$. In practice, the direction θ_T is unknown and the radar will form the beam for some directions $\theta_1, \dots, \theta_{N_\theta}$. Let us define $\mathbf{s}_r(t) = [s_{r,0}(t), \dots, s_{r,N_a-1}(t)]^T$, and $\mathbf{n}(t) = [n_0(t), \dots, n_{N_a-1}(t)]^T$. The signal after beamforming in direction θ_i is then obtained by

$$s_{r,\theta_i}(t) = \mathbf{v}_{\theta_i}^H \mathbf{s}_r(t). \quad (1.43)$$

After some calculations, it comes [VT02]

$$s_{r,\theta_i}(t) = \rho e^{j\varphi} u(t - \tau_0) e^{j2\pi\nu_0 t} \psi_{\theta_T}(\theta_i) + n_{\theta_i}(t), \quad (1.44)$$

where $\psi_{\theta_T}(\theta_i) = \frac{2\pi d}{\lambda} (\cos(\theta_T) - \cos(\theta_i))$, $n_{\theta_i}(t) = \mathbf{v}_{\theta_i}^H \mathbf{n}(t)$ is a complex white Gaussian noise and

$$\Psi(\psi_{\theta_T}(\theta_i)) = \frac{1}{N_a} \frac{\sin\left(N_a \frac{\psi_{\theta_T}(\theta_i)}{2}\right)}{\sin\left(\frac{\psi_{\theta_T}(\theta_i)}{2}\right)}. \quad (1.45)$$

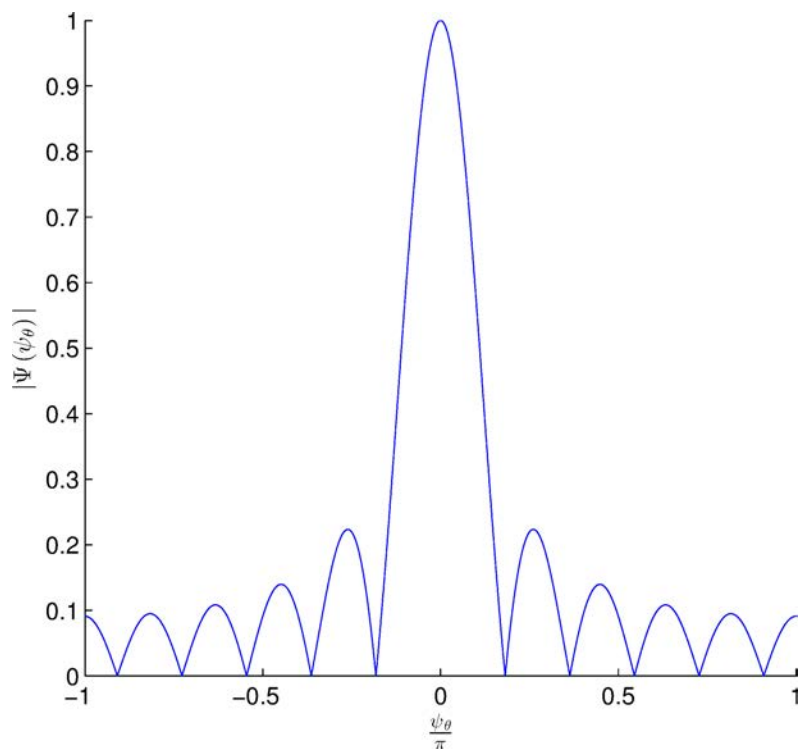


Figure 1.7 – $\Psi(\psi_\theta) : \psi_{\theta_T}(\theta) = \frac{2\pi d}{\lambda}(\cos(\theta_T) - \cos(\theta))$, $N_a = 11$, $\theta_T = \frac{\pi}{2}$.

This function is shown in Figure 1.7. The half beam-width is defined as $|\Psi(\psi_{\theta_{3dB}})|^2 = 1/2$ and is given, for $N_a > 30$, by $\theta_{3dB} \approx 0.886 \frac{\lambda}{Nd}$. Finally, note that whereas the noise processes on each elementary antenna are independent, this is not, in general, the case for the noise processes $n_{\theta_i}(t)$ and $n_{\theta_j}(t)$ where the covariance is equal to

$$\text{cov}(n_{\theta_i}(t), n_{\theta_j}(s)) = 2\sigma^2 \mathbf{v}_{\theta_i}^H \mathbf{v}_{\theta_j} \delta(t - s). \quad (1.46)$$

1.1.8 Measurement model

Now that the radar processing has been briefly described, we can present the measurement model (before the detection stage) that will be used in the following of this document. Let us denote by T_S the radar cycle duration, *i.e.* the duration during which the radar transmits the signal, receives it, and performs the signal processing stage. Therefore, denoting by k the time index, the radar provides a measurement \mathbf{z}_k every kT_S . At the k -th iteration if N_k targets have reflected the transmitted signal, then from the previous section, it follows that the output signal after the radar processing chain (reception beamforming, range and Doppler matched-filters) can be expressed as

$$s_{r, MF, k}(\tau, \nu, \theta) = \sum_{i=1}^{N_k} \rho_{k,i} e^{j\varphi_{k,i}} \chi_u(\tau_{k,i} - \tau, \nu_{k,i} - \nu) \Psi(\psi_{\theta_{k,i}}(\theta)) + n_k(\tau, \theta, \nu), \quad (1.47)$$

where $\rho_{k,i}$ and $\varphi_{k,i}$ are the amplitude and the phase defined in paragraph 1.1.2, and $\tau_{k,i}$, $\nu_{k,i}$ and $\theta_{k,i}$ represent respectively the delay, Doppler and azimuth of the i -th target. Obviously, parameters $\tau_{k,i}$ and $\nu_{k,i}$ are respectively related to the target range $r_{k,i}$ and

the target range rate $\dot{r}_{k,i}$. In tracking these unknown parameters will correspond to the hidden state⁴ $\mathbf{x}_{k,i} = [r_{k,i}, \dot{r}_{k,i}, \theta_{k,i}]^T$. Here parameters (τ, ν, θ) are continuous. However, in practice, the reception processing is performed for several values of the parameter $(\tau^l, \nu^l, \theta^l)$, $l = 1, \dots, N_c$ where N_c is the number of test cells. Thus, denoting by z_k^l the signal in cell l , it can be rewritten as

$$z_k^l = \sum_{i=1}^{N_k} \rho_{k,i} e^{j\varphi_{k,i}} h^l(\mathbf{x}_{k,i}) + n_k^l, \quad (1.48)$$

where

$$h^l(\mathbf{x}_{k,i}) = \chi_u \left(\frac{2}{c} (r_{k,i} - r^l), \frac{c}{2f_0} (\nu_{k,i} - \nu^l) \right) \Psi(\psi_{\theta_{k,i}}(\theta^l)). \quad (1.49)$$

Finally by concatenating the signal samples z_k^l , the ambiguity function samples $h^l(\mathbf{x}_{k,i})$ and the noise samples n_k^l into vectors $\mathbf{z}_k = [z_k^1, \dots, z_k^{N_c}]^T$, $\mathbf{h}(\mathbf{x}_{k,i}) = [h_{k,i}^1, \dots, h_{k,i}^{N_c}]^T$ (where $h_{k,i}^l = h^l(\mathbf{x}_{k,i})$) and $\mathbf{n}_k = [n_k^1, \dots, n_k^{N_c}]^T$ respectively, the measurement equation can be rewritten in a compact form as

$$\mathbf{z}_k = \sum_{i=1}^{N_k} \rho_{k,i} e^{j\varphi_{k,i}} \mathbf{h}(\mathbf{x}_{k,i}) + \mathbf{n}_k. \quad (1.50)$$

Here \mathbf{n}_k is a circular Gaussian complex noise with a covariance matrix $\mathbf{\Gamma}$ assumed to be known and often equal to $\mathbf{\Gamma} = 2\sigma^2 \mathbf{I}_{N_c}$, *i.e.* signal samples are independent. The Equation (1.50) defines the raw radar measurement \mathbf{z}_k that will be used as the input of the detection and extraction stage (as illustrated in Figure 1.1 and detailed in the next paragraph) for the classic radar tracking applications and as the input of Track-Before-Detect applications, that are at the heart of this work.

1.1.9 Detection and "hit" extraction

The aim of the detection and extraction stage is to detect potential targets and extract their parameters from the raw radar data \mathbf{z}_k . This process is performed in two steps. First the detection stage that provides detection "hits" and then the extraction stage that aggregates detection "hits" and extracts target parameters.

In all this document, we will consider a simple case where the radar measurements are only composed of target signals and homogeneous additive noise with known variance. More realistic cases with heterogeneous noise and clutter will thus be out of our scope. Under this restriction, the first detection step simply consists in comparing each sample $|z_k^l|^2$, $l = 1, \dots, N_c$ with a threshold γ as in the detection procedure defined in Eq. (1.20).

⁴Note that this hidden state may possibly include other hidden parameters. Moreover, these parameters may be expressed in another coordinate system (*e.g.* Cartesian coordinates). Indeed, the radar measurements are intrinsically defined in polar coordinates that do not allow to easily model the evolution of the target parameters over time, for instance a rectilinear target motion is quite difficult to model in polar coordinates while this kind of trajectory is modeled by a linear equation in Cartesian coordinates. This will be detailed in paragraphs 2.2 and 2.3.1.

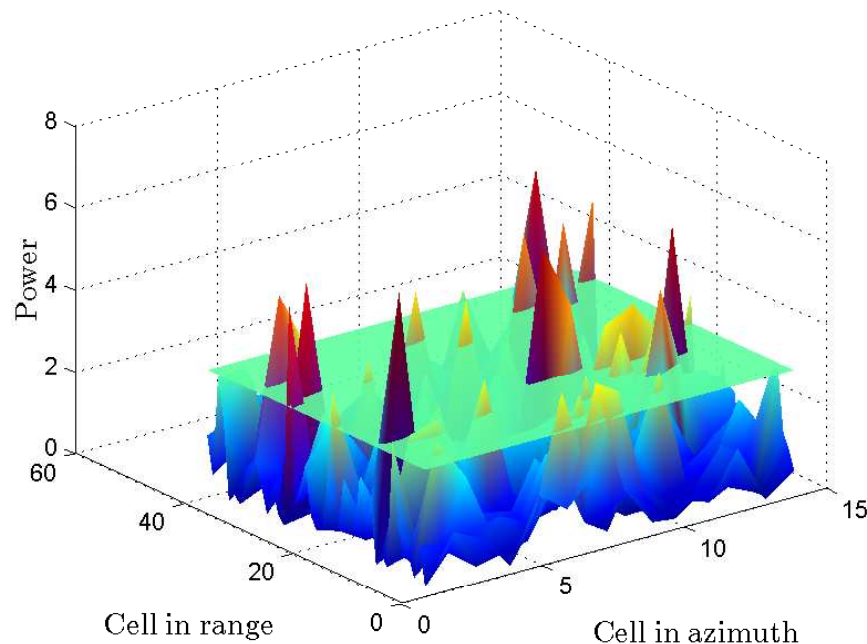


Figure 1.8 – Detection procedure for radar measurement in range and azimuth.

Since the noise is a circular Gaussian noise with variance $2\sigma^2$, the threshold γ is simply provided by [Kay98]

$$\gamma = -2\sigma^2 \ln(P_{fa}). \quad (1.51)$$

The probability of target detection P_D depends on the target SNR. The detection procedure is illustrated in Figure 1.8. Of course, in the presence of noise of unknown variance of clutter, this simple detection procedure would be replaced by an adaptive one, for instance a classic Constant False Alarm Rate (CFAR) detector.

Then a simple procedure to extract the parameter would be to consider as parameter estimate the corresponding value (for instance $(r^l, \dot{r}^l, \theta^l)$ with the example defined in the previous paragraph) for any cell that exceeds the threshold. However, in practice further developments are required; indeed, recall that if the target SNR is high, due to the ambiguity function sidelobes, one single target may produce several contiguous detection "hits". Thus, a clustering step is generally added in order to aggregate the detection "hits" that are likely to be generated by the same target. This aggregation step is often based on an heuristic procedure. Then an estimation procedure is applied to each extracted hit in order to retrieve the corresponding parameter value (for instance, range, Doppler, azimuth...).

Finally at the end of the detection and "hits" extraction stage a set of detection "hits" is provided:

$$\mathcal{Y}_k = \{\mathbf{y}_{k,1}, \dots, \mathbf{y}_{k,N_D}\} \quad (1.52)$$

where each "hit" $\mathbf{y}_{k,l}$ is possibly⁵ related to a target state \mathbf{x}_k by the following equation:

$$\mathbf{y}_{k,l} = \mathbf{H}(\mathbf{x}_k) + \mathbf{w}_k, \quad (1.53)$$

with \mathbf{H} a known function (possibly linear) and \mathbf{w}_k a Gaussian noise with covariance matrix \mathbf{R}_k . Finally, as presented in Figure 1.1, the set \mathcal{Y}_k is provided to the tracking stage in order to form tracks and enhance the estimation of the target parameters.

1.1.10 Radar tracking algorithms

1.1.10.1 Radar tracking objectives

Each measurement $\mathbf{y}_{k,i}$ in the set of detection \mathcal{Y}_k may either correspond to an actual target or to a false alarm. Therefore, one objective of radar tracking algorithms is to be able to retrieve from the sets of detection \mathcal{Y}_k the measurements that come from the same target in order to create a track, while discarding the false alarms. Moreover, the accuracy of the target parameter estimation is limited by the radar characteristics, for instance the range resolution Δ_r (see paragraph 1.1.5). Therefore, a filtering step is added to estimate the target parameters from all measurements until k (*i.e.* $\mathcal{Y}_1, \dots, \mathcal{Y}_k$), and thus improve the parameter accuracy. More precisely, this step consists in estimating the state of a dynamic system (that is unobserved and denoted as hidden state) from a sequence of noisy measurements. In radar applications, the hidden states are the target parameters (*e.g.*, position, velocity, *etc.*) and their temporal evolution can often be modelled by a dynamic equation where the state at current step depends to the ones at previous iterations. The noisy measurement is the set of detection hits \mathcal{Y}_k provided by the radar or the measurement \mathbf{z}_k . Thus, by taking advantages of some prior knowledge on the target motion, the filtering step allows to aggregate the information provided by all the noisy measurements until the current step (*i.e.* $\mathcal{Y}_1, \dots, \mathcal{Y}_k$) and then to enhance the estimation of the target parameters. Finally, since the measurements are provided at each radar cycle (*i.e.* every T_S), solutions proposed to perform the tracking stage are often sequential or, in other words, the previous estimated parameters are updated with the new measurement instead of calculating again the estimation at each iteration from all the available measurements. A convenient way to do so is the Bayesian framework, and more precisely the Hidden Markov Models (HMM) that allow to sequentially estimate hidden parameters from a measurement related to the hidden state. This framework will be detailed in the next section.

To sum up, the aim of the tracking stage may be viewed as fulfilling the two following tasks:

- creating or deleting tracks, either from the sets of detection hits $\mathcal{Y}_1, \dots, \mathcal{Y}_k$ in classic radar tracking or from the raw radar measurements $\mathbf{z}_1, \dots, \mathbf{z}_k$ in the TBD framework.
- estimating the track parameters from the sets of detection hits or from the raw radar measurements.

⁵Note that we use the term "possibly" since a detection hit may not come from an actual target but may rather be a false alarm. This uncertainty on the measurement origin (actual target or not) may be solved by the tracking stage.

1.1.10.2 Classic radar tracking algorithms

In the radar tracking community these two problems refer to the Multiple-Target Tracking (MTT) problem [Bla86]. The first proposed solutions used sequential analysis in order to initialize or delete track. Tracks were associated via the nearest-neighbour association rule that consists in assigning detection hits to existing tracks in a way that minimizes a certain distance criterion. However, this approach may lead to wrong associations, especially when there are a lot of false alarms, and, as a consequence, to poor tracking performance. Then, new algorithms were proposed in a Bayesian framework that are able to deal with such situations. The first one was proposed by Singer *et al.* [SSH74] and is denoted as Multiple Hypothesis Tracking (MHT). It is a measurement oriented algorithm (*i.e.* hypotheses are calculated from the measurements) where the key idea is to consider all the possible hypotheses in order to initialize, to maintain or to delete tracks, *i.e.* at a given instant k , any considered hit can be either allocated to an existing track, can initialize a new track, or can be associated to a false alarm. The solution would then be provided by the most likely hypothesis. This approach leads to a number of hypotheses that increases extremely rapidly with time, so that this approach leads to a complexity that may be difficult to handle in a reasonable time. Therefore, a suboptimal approximation has been proposed by Reid [Rei79] in 1979 which allows to make the MHT feasible by pruning hypotheses with low probabilities.

An alternative approach was proposed by Bar-Shalom *et al.* [BST75] in 1975, known as the Probabilistic Data association Filter (PDAF). Contrary to the MHT which manages the whole MTT problem (*i.e.* track life stages and association problem), the PDAF is only devoted to the association problem. As a consequence, it assumes the number of targets known (this is a target-track oriented algorithm) and does not provide track initialization and termination. Note that the PDAF may fail when multiple tracks are close since it does not consider the possible interaction between them. To handle this situation, the Joint PDAF (JPDAF) was then proposed [FBSS83]. For such filters, the track initialization and termination is often done by using the "M out of N" rule that consists in initializing a track if a detection is present in the validation gate [BS87] of a given initializing track at least M times over N iterations [Cas76, BSCS89]. A similar rule is applied for the track termination.

Both the MHT and the PDAF solutions perform the tracking itself for a given track/hit association thanks to a Bayesian filter, usually the well-known Kalman filter.

1.2 Bayesian filtering

Most radar tracking algorithms are derived from the Bayesian filtering theory, and among them the particle filter that will thoroughly be used in this work. Thus, we present in the sequel some aspects of the general Bayesian filtering theory. We will restrict our attention here to the discrete-time formulation of the filtering problem. Let us denote by $(\mathbf{x}_k)_{k \in \mathbb{N}}$ the random state process that is hidden (or unobserved) and by $(\mathbf{z}_k)_{k \in \mathbb{N}^*}$ the measurement process (that is observed). We adopt the state-space approach in a particular class of models called Hidden Markov Models (HMM) which is based on a dynamic system modelled by a set of two equations [AM79]:

- one equation for the temporal evolution of the current hidden state \mathbf{x}_k from the hidden state at the previous iteration \mathbf{x}_{k-1} (state model).
- A second equation that relates the noisy measurements \mathbf{z}_k to the current state \mathbf{x}_k (measurement model).

Moreover, we shall assume that both models are available in a probabilistic form.

Then, in a Bayesian perspective, the aim is to calculate some estimators of the state \mathbf{x}_k [BS09]. Most of the time, estimation is performed by one of the following well-known estimators:

- the Minimum Mean Square Error Estimator (MMSE) $\hat{\mathbf{x}}_k^{MMSE} = \mathbb{E}[\mathbf{x}_k | \mathbf{z}_{1:k}]$,
- the Maximum *a posteriori* (MAP) estimator $\hat{\mathbf{x}}_k^{MAP} = \arg \max_{\mathbf{x}_k} p(\mathbf{x}_k | \mathbf{z}_{1:k})$,

where the notation $\mathbf{z}_{1:k}$ refers to the sequence $(\mathbf{z}_1, \dots, \mathbf{z}_i, \dots, \mathbf{z}_k)$. Both approaches require the knowledge of the posterior density⁶ $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ which is obtained for the HMM by the Bayesian filter.

1.2.1 Hidden Markov Models

Hidden Markov Models are a particular class of state-space models where the density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ can be computed recursively from the density at previous step $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$. First, let us assume that the process $(\mathbf{x}_k)_{k \in \mathbb{N}}$ takes its values in \mathbb{R}^{n_x} and evolves according to the following equation:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_k), \quad (1.54)$$

where \mathbf{f}_k is a known and possibly non-linear function and $(\mathbf{v}_k)_{k \in \mathbb{N}^*}$ is an independent and identically distributed (i.i.d.) noise sequence. \mathbf{x}_0 is assumed to be distributed according to a density $p_0(\cdot)$. Under these conditions, the process $(\mathbf{x}_k)_{k \in \mathbb{N}}$ is a Markov process of order one, *i.e.*

$$p(\mathbf{x}_k | \mathbf{x}_{0:k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}), \text{ for any } k \geq 1. \quad (1.55)$$

In other words, the density of \mathbf{x}_k conditionally to $\mathbf{x}_{0:k-1}$ only depends on the state at previous step \mathbf{x}_{k-1} . The measurement \mathbf{z}_k is related to the state \mathbf{x}_k by the following measurement equation⁷:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{n}_k, \quad (1.56)$$

where $\mathbf{h}_k(\cdot)$ is a possibly non-linear function of the state \mathbf{x}_k at value in \mathbb{R}^{n_z} (or in \mathbb{C}^{n_z}), and $(\mathbf{n}_k)_{k \in \mathbb{N}^*}$ an i.i.d noise sequence. Moreover, it is assumed that noise samples \mathbf{n}_k and \mathbf{v}_k are mutually independent. Then, the measurement \mathbf{z}_k conditionally to \mathbf{x}_k is independent of $\mathbf{z}_{1:k-1}$, *i.e.*

$$p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{z}_{1:k-1}) = p(\mathbf{z}_k | \mathbf{x}_k). \quad (1.57)$$

⁶Note that here, we adopt the formalism of density with respect to some measure (in general, the Lebesgue measure or the counting measure). However, in some cases this density may not exist and one must consider the probability distribution. In order to avoid unnecessary complexity, this latter will not be treated here. A more general approach is presented in [DM04].

⁷Note that we restrict ourselves to additive noise in the measurement equation since this latter is generally used in radar tracking and also because non-additive models can lead to theoretical issues which are beyond the scope of this manuscript.

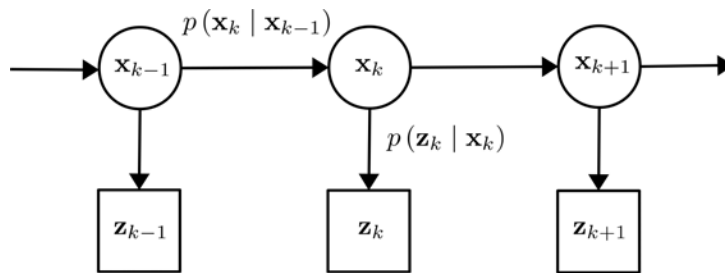


Figure 1.9 – Block diagram of the Hidden Markov Model.

$p(\mathbf{z}_k | \mathbf{x}_k)$ is called the likelihood function and is entirely defined by the measurement equation (1.56) and the statistics of \mathbf{n}_k . Moreover it is generally assumed to be easily computable. This may not always be the case: the chapter 4 will be precisely dedicated to the calculation of this latter in the Track-Before-Detect framework. The diagram of the Hidden Markov Model is shown in Figure 1.9.

1.2.2 Theoretical Bayesian filter

For the HMMs defined in previous paragraph, it is possible to calculate recursively the density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ from $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$. Indeed, using the Bayes rule and the properties of the HMM, $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ can be rewritten as follows:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | \mathbf{x}_k)}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})}, \quad (1.58)$$

where $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ is obtained by the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) p(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}, \quad (1.59)$$

and corresponds to the prediction step, where the density of \mathbf{x}_k conditionally to the previous measurements $\mathbf{z}_{1:k-1}$ is evaluated. Then, this density is updated with the new observation \mathbf{z}_k via Eq. (1.58) where the normalized constant is given by

$$p(\mathbf{z}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k. \quad (1.60)$$

The recursion to obtain $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ from $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$ can be summarized as follows

$$p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) \xrightarrow[\text{Eq.(1.59)}]{\text{prediction}} p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) \xrightarrow[\text{Eq.(1.58)}]{\text{update}} p(\mathbf{x}_k | \mathbf{z}_{1:k}). \quad (1.61)$$

In general, Eq. (1.59) and Eq. (1.60) cannot be computed analytically and, as a consequence, neither is the Bayesian filter. However, the exact solution can be obtained when the state and measurement models are linear and Gaussian – the solution being the very well-known Kalman filter [Kal60] – or when the state space is discrete with a finite number of states [AMGC02].

1.2.3 Linear Gaussian models: Kalman filter

Linear Gaussian models are a particular class of HMM where the Bayesian filter can be solved exactly. For these models, the hidden process $(\mathbf{x}_k)_{k \in \mathbb{N}}$ verifies

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_k, \quad (1.62)$$

where \mathbf{F}_k is a matrix of size $n_x \times n_x$, and \mathbf{v}_k is a Gaussian noise with covariance matrix \mathbf{Q}_k . The initial state \mathbf{x}_0 is also assumed to be Gaussian with mean \mathbf{m}_0 and covariance matrix \mathbf{Q}_0 . The observed process $(\mathbf{z}_k)_{k \in \mathbb{N}}$ is related to the state \mathbf{x}_k according to the following equation:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k, \quad (1.63)$$

where \mathbf{H}_k is a matrix of size $n_z \times n_x$, and \mathbf{w}_k is a Gaussian noise with covariance matrix \mathbf{R}_k . Lastly, it is also assumed that \mathbf{x}_0 , $(\mathbf{v}_k)_{k \in \mathbb{N}^*}$, $(\mathbf{w}_k)_{k \in \mathbb{N}^*}$ are mutually independent.

Under these conditions, all the densities at each step of the Bayesian recursion defined in Eq. (1.61) are Gaussian, *i.e.*

$$p(\mathbf{x}_{k-1} \mid \mathbf{z}_{:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \mathbf{x}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}), \quad (1.64)$$

$$p(\mathbf{x}_k \mid \mathbf{z}_{:k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}), \quad (1.65)$$

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k}, \mathbf{P}_{k|k}), \quad (1.66)$$

where $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{P})$ represents here the standard Gaussian density with mean \mathbf{m} and covariance matrix \mathbf{P} evaluated at point \mathbf{x} . Then, the parameters of the aforementioned densities (the mean and covariance) can be computed by applying the following set of equations:

$$\mathbf{x}_{k|k-1} = \mathbf{F}_k \mathbf{x}_{k-1|k-1}, \quad (1.67)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_k, \quad (1.68)$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{z}}_k, \quad (1.69)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}, \quad (1.70)$$

where

$$\begin{aligned} \tilde{\mathbf{z}}_k &= \mathbf{z}_k - \mathbf{H}_k \mathbf{x}_{k|k-1}, \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k, \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}, \end{aligned} \quad (1.71)$$

are respectively the innovation, the covariance of the innovation and the Kalman gain. Equations (1.67)-(1.70) define the Kalman filter [Kal60] which is the optimal solution for the Linear Gaussian models. Furthermore note that the parameters $\mathbf{x}_{k|k}$ and $\mathbf{P}_{k|k}$ provide directly the MMS estimator $\hat{\mathbf{x}}_k^{MMSE} = \mathbf{x}_{k|k} = \mathbb{E}[\mathbf{x}_k \mid \mathbf{z}_{1:k}]$ and its covariance matrix

$$\mathbb{E} \left[(\hat{\mathbf{x}}_k^{MMSE} - \mathbf{x}_k) (\hat{\mathbf{x}}_k^{MMSE} - \mathbf{x}_k)^T \mid \mathbf{z}_{1:k} \right] = \mathbf{P}_{k|k},$$

where both are calculated via the classic Bayesian scheme:

$$\begin{aligned} \mathbf{x}_{k-1|k-1} &\xrightarrow[\text{Eq.(1.67)}]{\text{prediction}} \mathbf{x}_{k|k-1} \xrightarrow[\text{Eq.(1.69)}]{\text{update}} \mathbf{x}_{k|k}, \\ \mathbf{P}_{k-1|k-1} &\xrightarrow[\text{Eq.(1.68)}]{\text{prediction}} \mathbf{P}_{k|k-1} \xrightarrow[\text{Eq.(1.70)}]{\text{update}} \mathbf{P}_{k|k}. \end{aligned} \quad (1.72)$$

Lastly, as a remark, note also that the calculation of the posterior covariance matrix $\mathbf{P}_{k|k}$ and the Kalman gain do not depend on the measurement \mathbf{z}_k and may therefore be calculated off-line.

The Kalman filter is very popular and is extensively used since it is very simple to implement, has a very low complexity and is quite robust. However, whereas the Kalman filter is optimal for the very specific case of the Linear Gaussian model, it is not optimal anymore when the Gaussian hypothesis and/or the linear assumption are violated. It appears that the raw radar measurement equation (1.50) is not linear according to the target state. Since Track-Before-Detect methods seek precisely to track targets from this kind of measurements, this prevents to use the Kalman filter in this case (other reasons exist and will be detailed in Chapter 2). Thus, other methods must be considered in the TBD framework and we propose in the sequel to outline the particle filter method that will be extensively used throughout this thesis and can handle such non-linear and/or non-Gaussian models.

1.2.4 Particle filter

When the HMM is non-linear and/or non-Gaussian, the Bayesian filter cannot be computed analytically (see paragraph 1.2.2) and we must therefore resort to some approximations. When the noises (state and measurement) are still assumed Gaussian but the functions $\mathbf{f}_k(\cdot)$ and/or $\mathbf{h}_k(\cdot)$, in Eq. (1.54) and Eq. (1.56), are non-linear, extensions of the Kalman filter can be considered:

- the first extension, known as EKF (Extended Kalman Filter) [AMGC02], consists in locally linearizing the functions $\mathbf{f}_k(\cdot)$ and $\mathbf{h}_k(\cdot)$ and then applying the Kalman recursion with the linearized equations.
- the second extension, known as UKF (Unscented Kalman Filter) [WVdM00], uses a set of points that are propagated deterministically through the non-linear equations and allow to estimate the parameters of the Gaussian approximation of $p(\mathbf{x}_k | \mathbf{z}_{1:k})$.

However, as for the Kalman filter, these solutions may also fail for highly non-linear function and/or non-Gaussian noise, then other solutions must be proposed to handle such difficulties.

Another approach consists in transforming the continuous state into a discrete state. In such a strategy, the continuous density $p(\mathbf{x})$ is approximated by a discrete measure using a set of samples $\{\mathbf{x}^i\}_{i=1}^{N_p}$, often called particles, and associated weights $\{w^i\}_{i=1}^{N_p}$, as follows:

$$p(\mathbf{x}) \approx \sum_{i=1}^{N_p} w^i \delta_{\mathbf{x}^i}(\mathbf{x}), \quad (1.73)$$

where $\delta_{\mathbf{x}^i}(\mathbf{x})$ is the delta mass Dirac function at point \mathbf{x}^i . This is the main idea behind the Monte Carlo methods and in particular in the particle filter: approximate a continuous density by a discrete density which is simpler to manipulate and in particular from which quantities, like mathematical expectations, can be easily calculated.

Following this idea, grid-based methods were proposed [AMGC02] in order to approximate the posterior density with a fix and deterministic set $\{\mathbf{x}^i\}_{i=1}^{N_p}$ (called grid), for

which the Bayesian filter can be exactly solved (see paragraph 1.2.2). However, when the state-space is large, such a method may require to use a lot of grid points \mathbf{x}^i to properly discretize the whole state space, and, as a consequence, may lead to a prohibitive computational time. Thus, a new solution, called particle filter, was proposed in the early 90s by Gordon *et al.* [GSS93], that consists in using an adaptive and random grid rather than a fix and deterministic grid. In the particle filter, particles are adaptively drawn with higher probability, thanks to a technique called Importance Sampling, in the areas where the posterior density takes high values, which prevents to discretize the whole state space.

The principle of Monte Carlo methods and particularly of Importance Sampling strategy will be briefly explained in the sequel. Then, a particular attention is given to a Monte Carlo technique, called Sequential Importance Sampling, that allows to approximate the density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ by a discrete density in a sequential manner. Finally, we provide with Algorithm 1.1 the scheme of the generic particle filter that will be used throughout this thesis.

1.2.4.1 Monte Carlo principle

Many applications require the computation of integrals of the form

$$I(\Phi) = \mathbb{E}_{p(\cdot)}[\Phi(\mathbf{x})] = \int \Phi(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \quad (1.74)$$

where Φ is a measurable bounded function and $p(\mathbf{x})$ is a given probability density function. Such integrals can seldom be calculated analytically. Then Monte Carlo methods propose to construct an empirical estimator of the quantity $I(\Phi)$ from N_p samples $(\mathbf{x}^1, \dots, \mathbf{x}^{N_p})$ independently drawn from $p(\mathbf{x})$.

First, an empirical estimator of $p(\mathbf{x})$ is provided by

$$\hat{p}_{N_p}(\mathbf{x}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta_{\mathbf{x}^i}(\mathbf{x}). \quad (1.75)$$

Then, by replacing the density $p(\mathbf{x})$ by its empirical estimator $\hat{p}_{N_p}(\mathbf{x})$ in Eq. (1.74), an estimator of $I(\Phi)$ is

$$\hat{I}_{N_p}(\Phi) = \frac{1}{N_p} \sum_{i=1}^{N_p} \Phi(\mathbf{x}^i), \quad (1.76)$$

This estimator is unbiased with variance

$$\text{var}(\hat{I}_{N_p}(\Phi)) = \mathbb{E}[|\hat{I}_{N_p}(\Phi) - I(\Phi)|^2] = \text{var}_{p(\cdot)}(\Phi) / N_p \quad (1.77)$$

where

$$\text{var}_{p(\cdot)}(\Phi) = \int |\Phi(\mathbf{x})|^2 p(\mathbf{x}) d\mathbf{x} - |I(\Phi)|^2 < +\infty. \quad (1.78)$$

However, in many cases, it might be difficult to directly draw samples according to the density $p(\mathbf{x})$. In particular, in the Bayesian framework, if we want to approximate the classic MMSE estimator $\mathbb{E}[\mathbf{x}_k | \mathbf{z}_{1:k}]$ directly via the Monte Carlo principle, this requires

to be able to sample from the density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$; this is often difficult to do. Thus a method known as Importance Sampling was proposed in order to estimate the quantity $I(\Phi)$ with a set of N_p samples using a different probability density function that allows to draw samples easily.

1.2.4.2 Importance Sampling

The key idea of Importance Sampling consists in rewriting Eq. (1.74) as a mathematical expectation under another density $q(\cdot)$ called the importance density or instrumental density for which samples can be easily drawn⁸. It requires as only condition that the support of $p(\cdot)$ must be included in the support of $q(\cdot)$, *i.e.* if $p(\mathbf{x}) > 0$ then $q(\mathbf{x}) > 0$.

First, let us rewrite equation (1.74) as follows

$$I(\Phi) = \int \Phi(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \int \Phi(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} = \mathbb{E}_{q(\cdot)} [\Phi(\mathbf{x}) \tilde{w}(\mathbf{x})], \quad (1.79)$$

where

$$\tilde{w}(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})}. \quad (1.80)$$

The integral (1.74) has been rewritten as an expectation from another density $q(\cdot)$ rather than $p(\cdot)$; then, for any N_p samples $(\mathbf{x}^1, \dots, \mathbf{x}^{N_p})$ independently drawn from $q(\cdot)$, $I(\Phi)$ can be estimated by

$$\hat{I}_{N_p, IS}(\Phi) = \frac{1}{N_p} \sum_{i=1}^{N_p} \tilde{w}^i \Phi(\mathbf{x}^i) \quad (1.81)$$

where

$$\tilde{w}^i = \frac{p(\mathbf{x}^i)}{q(\mathbf{x}^i)}, \quad i = 1, \dots, N_p, \quad (1.82)$$

are called the importance weights. The estimator $\hat{I}_{N_p, IS}(\Phi)$ is unbiased with variance

$$\text{var}(\hat{I}_{N_p, IS}(\Phi)) = \mathbb{E}[|\hat{I}_{N_p, IS}(\Phi) - I(\Phi)|^2] = \text{var}_{q(\cdot)} \left(\frac{p}{q} \Phi \right) / N_p \quad (1.83)$$

where

$$\text{var}_{q(\cdot)} \left(\frac{p}{q} \Phi \right) = \int |\Phi(\mathbf{x})|^2 \left(\frac{p(\mathbf{x})}{q(\mathbf{x})} \right)^2 q(\mathbf{x}) d\mathbf{x} - |I(\Phi)|^2. \quad (1.84)$$

Alternatively, $I(\Phi)$ can also be estimated by

$$\hat{I}_{N_p, IS, SN}(\Phi) = \sum_{i=1}^{N_p} w^i \Phi(\mathbf{x}^i), \quad (1.85)$$

where the importance weights have been normalized, *i.e.*

$$w^i = \frac{\tilde{w}^i}{\sum_{j=1}^{N_p} \tilde{w}^j}. \quad (1.86)$$

⁸However note that this choice is, in fact, not trivial since the variance of the estimator directly depends on the instrumental density $q(\cdot)$ and has therefore to be carefully made [DdFG01].

Note that the weights w^i can be computed up to a constant, *i.e.*

$$w^i \propto \frac{p(\mathbf{x}^i)}{q(\mathbf{x}^i)}, \quad i = 1, \dots, N_p. \quad (1.87)$$

Indeed, if these weights w^i share a common constant, it will be discarded through the normalization. Note that in general $\hat{I}_{N_p, IS, SN}(\Phi)$ is a biased estimator. An approximation of the density $p(\cdot)$ as an empirical approximation $\hat{p}_{N_p}(\cdot)$ is then obtained by

$$\hat{p}_{N_p}(\mathbf{x}) = \sum_{i=1}^{N_p} w^i \delta_{\mathbf{x}^i}(\mathbf{x}). \quad (1.88)$$

Importance sampling with this additional normalization step is called self-normalized importance sampling in the literature [Owe13].

1.2.4.3 Sequential Importance Sampling particle filter

Importance sampling can be applied in order to approximate the density $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$ when it cannot be computed analytically. However, recall that the Bayesian filter presents a recursive structure. Thus it is interesting to take advantage of this property of the HMM to compute the density $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$ recursively. This is the purpose of the Sequential Importance Sampling technique that allows to sequentially approximate the posterior density of all the previous states⁹ $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$.

Let $q(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$ be an instrumental density from which it is easy to draw samples and let also assume that this latter factorizes as follows

$$q(\mathbf{x}_{0:k} | \mathbf{z}_{1:k}) = q(\mathbf{x}_0) \prod_{l=1}^k q(\mathbf{x}_l | \mathbf{x}_{0:l-1}, \mathbf{z}_{1:l}). \quad (1.89)$$

This factorization ensures that the importance weight of the i^{th} particle

$$w_k^i = \frac{p(\mathbf{x}_{0:k}^i | \mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^i | \mathbf{z}_{1:k})} \quad (1.90)$$

can be computed recursively. Indeed, w_k^i can then be rewritten as follows

$$\begin{aligned} w_k^i &= \frac{p(\mathbf{z}_k | \mathbf{x}_{0:k}^i, \mathbf{z}_{1:k-1}) p(\mathbf{x}_{0:k}^i | \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1}) q(\mathbf{x}_k^i | \mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1}^i | \mathbf{z}_{1:k-1})} \\ &= \frac{p(\mathbf{x}_{0:k-1}^i | \mathbf{z}_{1:k-1})}{q(\mathbf{x}_{0:k-1}^i | \mathbf{z}_{1:k-1})} \times \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1}) q(\mathbf{x}_k^i | \mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})} \\ &= w_{k-1}^i \times \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1}) q(\mathbf{x}_k^i | \mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})}. \end{aligned} \quad (1.91)$$

⁹Note that we present the method for the whole sequence $\mathbf{x}_{0:k}$ since the posterior density of the state \mathbf{x}_k can be simply obtained through a marginalization.

Finally, since $p(\mathbf{z}_k | \mathbf{z}_{1:k-1})$ is a constant independent of the particle sequence $\mathbf{x}_{0:k}^i$, the weights are proportional to

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})}. \quad (1.92)$$

In practice, we are mainly interested by the posterior density of the state $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ rather than the density of all the states $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$. Therefore, in order to avoid storing all the history of the particles $\{\mathbf{x}_{0:k}^i\}_{i=1}^{N_p}$, it is convenient to choose an instrumental density that depends only on the previous state and the current measurement:

$$q(\mathbf{x}_k^i | \mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k}) = q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k). \quad (1.93)$$

In that case, the only variables to store for time step k are \mathbf{x}_{k-1}^i and \mathbf{z}_k , while all the previous particle states and past measurements can be discarded. In the following, we will always consider instrumental densities that verify this condition. Then, under this condition, the weights are finally provided by

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)}. \quad (1.94)$$

After the normalization, the posterior density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ can be approximated by

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta_{\mathbf{x}_k^i}(\mathbf{x}_k). \quad (1.95)$$

The Sequential Importance Sampling (SIS) particle filter follows the two steps of the Bayesian filter defined in Eq. (1.61): first particles are propagated in the state space via the instrumental density defined in Eq. (1.93); then particles are updated according to Eq. (1.94). The mechanism of the SIS particle filter is illustrated in Figure 1.10. Finally, using the estimated density, the classic MMSE is simply obtained as

$$\hat{\mathbf{x}}_{k|k} = \sum_{i=1}^{N_p} w_k^i \mathbf{x}_k^i, \quad (1.96)$$

and the covariance matrix $\mathbf{P}_{k|k} = \text{var}(\mathbf{x}_k | \mathbf{z}_{1:k})$ estimator as

$$\hat{\mathbf{P}}_{k|k} = \sum_{i=1}^{N_p} w_k^i (\mathbf{x}_k^i - \hat{\mathbf{x}}_{k|k}) (\mathbf{x}_k^i - \hat{\mathbf{x}}_{k|k})^T. \quad (1.97)$$

1.2.4.4 Degeneracy problem

Whereas theoretical results ensure that the approximated posterior density (1.95) converges to the posterior density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ as $N_p \rightarrow +\infty$ [CD02], in practice, the number of particles N_p is always finite. In that case, the SIS particle filter suffers from a degeneracy phenomenon: after some iterations, one particle will present a weight very close to one while other particles will present negligible weights. This phenomenon cannot be avoided; indeed it has been proven that the variance of the weights can only increase over time [DGA00]. In practice, this leads to two major problems:

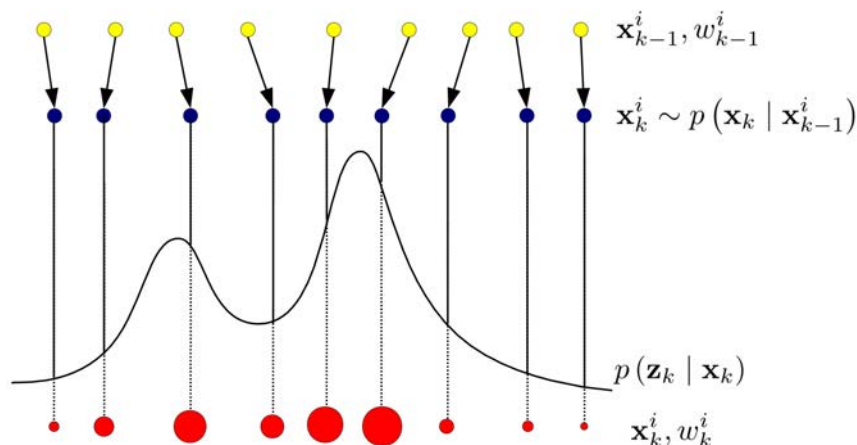


Figure 1.10 – Mechanism of the SIS particle filter with the two steps: propagation of the particles with the prior density $p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)$ and then update with the measurement \mathbf{z}_k .

- First, after some iterations, the particle approximation will be a poor estimate of the objective posterior density, and therefore, the corresponding estimators will not be accurate.
- Computing resources are devoted to update the weights of a possibly large number of particles whereas most of them have a negligible contribution to the approximation of the posterior density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$.

In order to have an idea of the quality of the particle approximation of the posterior density, it can be interesting to measure this degeneracy phenomenon. Several indicators have been proposed in the literature, among which the most popular is probably the effective sample size N_{eff} proposed in [LR98], based on the calculation of the variance of the weights. In general, it cannot be computed exactly but an estimate is given by

$$N_{\text{eff}} \approx \left(\sum_{i=1}^{N_p} (w_k^i)^2 \right)^{-1}. \quad (1.98)$$

This indicator provides a good estimation of the number of particles that effectively participate in the estimation of the posterior density. For instance, when particles share the same weights $w_k^i = 1/N_p$ (that corresponds to a weight variance equal to zero), then $N_{\text{eff}} = N_p$ since all particles contribute equally to the estimation. On the contrary, when only one particle concentrates all the weight (*i.e.* the particle has a weight equal to one which corresponds to a maximum variance), then $N_{\text{eff}} = 1$.

However, although the indicator N_{eff} allows to measure the degeneracy phenomenon, it does not prevent from this issue. Thus, several solutions have been proposed to minimize the degeneracy phenomenon among which the most common is certainly the addition of a resampling procedure in the SIS particle filter and, to a lesser extent, a careful choice of the instrumental density which may sensibly reduce the degeneracy phenomenon.

1.2.4.5 Instrumental density

In the SIS particle filter, the choice of the importance density is left to the user. In general, simply choosing the prior density $p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)$ (from which it is generally easy to sample) as the importance density is enough to ensure acceptable performance. In that case, the weight update equation (1.94) simply becomes

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_k^i), \quad (1.99)$$

and only requires to calculate $p(\mathbf{z}_k | \mathbf{x}_k^i)$, *i.e.* the likelihood of the observation conditionally to the state \mathbf{x}_k^i .

However, in some applications this simple choice may lead to poor performance with a severe degeneracy phenomenon. This is the case for instance in Track-Before-Detect applications, as will be shown in Chapter 2. Therefore, a more suitable choice that takes into account the current measurement \mathbf{z}_k must be made. The optimal one, in the sense that it minimizes the variance of the importance weights (and thus maximizes N_{eff}), is given by [DGA00]

$$q_{\text{opt}}(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k), \quad (1.100)$$

for which the variance of the weights is zero. This density can be rewritten as follows:

$$q_{\text{opt}}(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)}, \quad (1.101)$$

and requires the calculation of the density $p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)$ provided by:

$$p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) = \int p(\mathbf{z}_k | \mathbf{x}') p(\mathbf{x}' | \mathbf{x}_{k-1}^i) d\mathbf{x}'. \quad (1.102)$$

In practice, except for very specific cases, this integral is intractable and, as a consequence, so is the optimal density. Moreover, it might be difficult to draw samples from this optimal importance density. Therefore, suboptimal approximations of the optimal importance density have been proposed [AMGC02]. However, the possible gain of using such suboptimal approximations is not always justified since an additional computational cost is induced by using such suboptimal approximations. Besides, in some applications, using more particles sampled with the prior is equivalent to using a more sophisticated density with less particles [AMGC02].

1.2.4.6 Resampling

The use of a convenient instrumental density may slow the degeneracy phenomenon, but it cannot avoid it totally. As a consequence, other solutions must be used to prevent the degeneracy phenomenon. A common technique consists in adding a resampling step in the SIS particle filter before any strong degeneracy occurs, for instance when the effective sample size N_{eff} falls below a given threshold $N_T = \beta N_p$ with $0 < \beta \leq 1$. The principle of resampling consists in selecting particles with large weights and discarding particles with small weights. In practice, this is done by drawing independently N_p particles from the particle representation of the posterior density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ given by Eq. (1.95). As

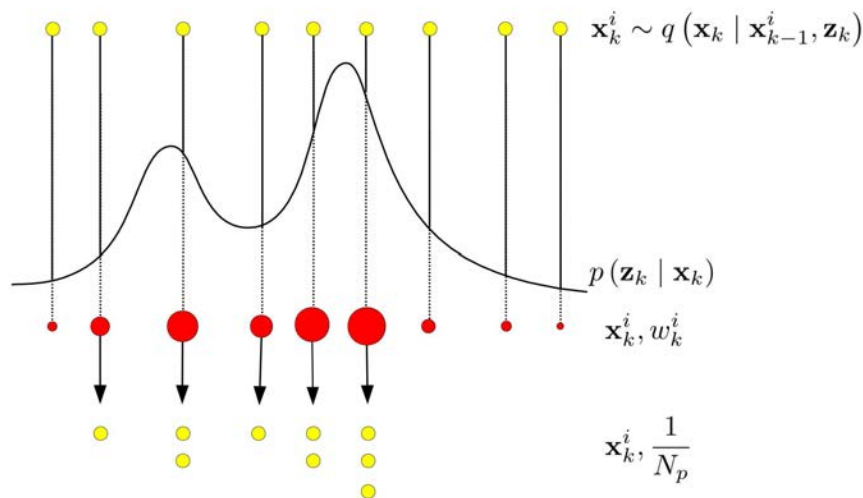


Figure 1.11 – Mechanism of SIS particle filter with resampling step: particles with large weights are selected while particles with small weights are discarded.

these new particles are sampled independently from the same density, they share the same weight equal to $1/N_p$. The particle filter with a resampling procedure is illustrated in Figure 1.11. In practice, several methods can be used to perform the resampling step, including multinomial resampling, residual resampling [LR98] and systematic resampling [Kit96]. The latter is one of the most popular since it is easy to implement and requires to draw only one single uniform variable [AMGC02]. Note however that in some situations, especially when the variance of the importance density is small (or even equal to zero) – this may be the case for instance when the prior density provided by the state equation is used as instrumental density and the variance of the state noise is very small – the resampling step can induce a severe loss of diversity among the particles. Indeed, in that case, many drawn particles will share the exact same state, and no diversity will be generated afterwards by the instrumental density, thus leading to an impoverishment of the particle cloud. Nevertheless, this effect can be corrected by adding a regularization step [MOLG01], where the key idea is to sample particles from a continuous density rather than a discrete density in order to obtain a better exploration of the state space. In practice, this is achieved by convolving the discrete density with a continuous kernel. This regularization step will not be considered in this thesis.

Finally, a description of the generic particle filter is given by Algorithm 1.1. This algorithm will be used throughout this thesis.

1.3 Conclusion

In this chapter, a brief overview of the radar chain from the signal processing stage to the tracking stage has been first presented. In particular, the fundamental role of the matched-filter both in detection and in estimation has been highlighted. Finally, at the end of this section, we specify the measurements \mathbf{z}_k and \mathcal{Y}_k that are respectively provided as an input to the tracking stage in the TBD framework and in the classic approach (see

Algorithm 1.1 Generic particle filter algorithm

Require: Particle cloud $\{w_{k-1}^i, \mathbf{x}_{k-1}^i\}_{i=1}^{N_p}$ at step $k-1$,

- 1: **for** $i = 1$ to N_p **do**
- 2: Propagation: draw particle \mathbf{x}_k^i according to $q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$.
- 3: Update: compute weight according to $w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)}$
- 4: **end for**
- 5: Normalize weights: $w_k^i \leftarrow \frac{w_k^i}{\sum_{l=1}^{N_p} w_k^l}$, $i = 1, \dots, N_p$
- 6: Compute N_{eff} according to Eq. (1.98).
- 7: **if** $N_{\text{eff}} < \beta N_p$ **then**
- 8: Resample N_p particles from $\sum_{i=1}^{N_p} w_k^i \delta_{\mathbf{x}_k^i}(\mathbf{x}_k)$
- 9: Reset weights: $w_k^i \leftarrow \frac{1}{N_p}$, $i = 1, \dots, N_p$
- 10: **end if**
- 11: **return** $\{w_k^i, \mathbf{x}_k^i\}_{i=1}^{N_p}$

Figure 1.1).

In a second step, the Bayesian filtering framework has been detailed and a special attention has been given to the Hidden Markov Models that allow to recursively solve the filtering problem. For this model, we detailed more specifically two solutions:

- The first one, known as Kalman filter, that allows to exactly solve the Bayesian filter when the model is Gaussian and linear. It has been extensively used in a wide range of applications and in particular in classic radar tracking applications.
- And the second one, known as particle filter, that allows to handle more general models than the Kalman filter (*i.e.* non-linear or/and non-Gaussian models). The latter will be intensively used and studied in the next chapters as a possible solution of the Track-Before-Detect problem.

This chapter has provided the main ingredients that will be used throughout the rest of this document, *i.e.* the measurement equation used in the TBD framework, based on the radar signal processing chain, and the particle filter.

Chapter 2

Monotarget Track-Before-Detect particle filters

2.1 Introduction

In the previous chapter, we have briefly outlined the whole radar chain from the reception of the signal to the track management (*i.e.* formation, update, deletion). In particular, we have highlighted that the classic tracking stage is not performed from the raw data \mathbf{z}_k but from a set of detection hits \mathcal{Y}_k which correspond either to noisy measurements of the actual target parameters or to false alarm measurements as illustrated in Figure 1.1. When the target Signal to Noise Ratio (SNR) is high, this pre-detection step has no consequence and allows to dramatically reduce the amount of data to process. Indeed, in such a situation, the detection threshold γ may be chosen relatively high in order to strongly limit the false alarm rate while guaranteeing to detect the targets almost at each iteration, thus making the Multiple Target Tracking (MTT) problem "easy" to solve. However, when the application seeks to detect and track low SNR targets, the MTT problem may become much more tricky. Indeed, maintaining a high threshold will not ensure anymore to detect the target at each iteration since, in this case, the detection probability P_D may be pretty small (low SNR). This is illustrated in Figure 2.1, where a

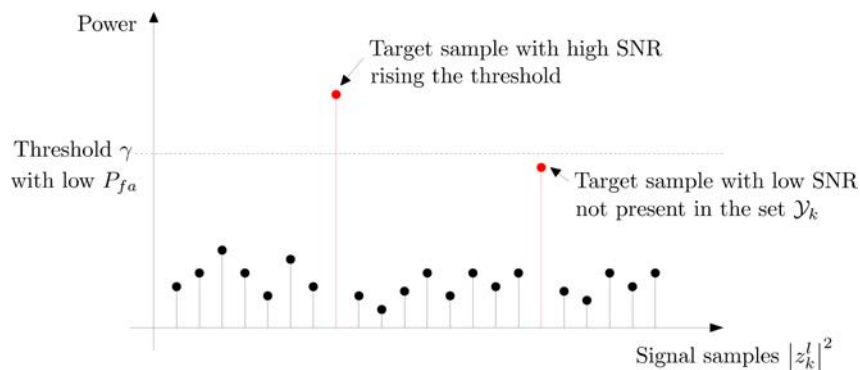


Figure 2.1 – Scheme of the pre-detection step: where the target signal sample with a high SNR target is kept, while the low SNR target sample is discarded.

signal sample due to a target with a low SNR is discarded, since it does not exceed the

threshold. As a consequence, all the information provided by this target signal sample is lost in the tracking stage. Furthermore, the solution that consists in decreasing the detection threshold will make both the initialization and the association problem much more arduous to solve since the set of detection \mathcal{Y}_k would be of much larger cardinality and mostly constituted, at each iteration, of false alarm measurements.

As a result, a new framework, known as Track-Before-Detect, was proposed to overcome the initialization and the association problem. The key concept of the Track-Before-Detect framework consist in jointly performing the detection and tracking from the whole raw measurements $\mathbf{z}_{1:k}$ rather than the sets $\mathcal{Y}_{1:k}$ in order to keep all the information provided by the measurement (since no pre-detection has been made). As a result, it allows to postpone and then to enhance the detection decision by exploiting all the information provided by the raw data.

The first methods proposed to solve the Track-Before-Detect in a monotarget setting were based on the Hough transform [CEW94] or Dynamic programming [Bar85]. However, although these methods are effective, they are not recursive and must process blocks of data, therefore leading to an intensive computational burden. Moreover, since the scope of this thesis is to study particle filter solutions to the Track-Before-Detect problem, we do not consider Hough transform and Dynamic programming throughout this thesis.

In this chapter, we consider the particle solution to the monotarget Track-Before-Detect problem proposed by Salmond *et al.* [SB01]. First, we define the state model and the measurement model in section 2.2 and 2.3. Then, we consider the particle solution for this model in section 2.4. In Section 2.5 we propose some contribution on the instrumental density in order to improve the filter performance. Finally in section 2.6 a modified particle filter is presented and in section 2.7 performances of the different filters are evaluated via Monte Carlo simulations.

2.2 State model

2.2.1 General TBD model

Track-Before-Detect solutions work on raw data $\mathbf{z}_{1:k}$ where no pre-detection step has been made. At each iteration step k , the presence of a target in the data \mathbf{z}_k is not *a priori* known. In a Bayesian framework, the classic method to deal with this ignorance consists in modelling *a priori* the presence or absence of the target by a variable s_k that takes value 1 if the target is present at step k , and 0 otherwise, and then considering as hidden state the hybrid state (s_k, \mathbf{x}_k) (where \mathbf{x}_k is the classic target state, *e.g.* position, velocity, *etc.*) [SB01].

Hence, the new hidden process $(s_k, \mathbf{x}_k)_{k \in \mathbb{N}}$ is Markovian and entirely defined by its transition density

$$p(s_k, \mathbf{x}_k \mid s_{k-1}, \mathbf{x}_{k-1}), \quad (2.1)$$

and its density $p_0(s_0, \mathbf{x}_0)$ at step $k = 0$. In practice, the transition density is often chosen to factorize as follows:

$$p(s_k, \mathbf{x}_k \mid s_{k-1}, \mathbf{x}_{k-1}) = p(s_k \mid s_{k-1}) p(\mathbf{x}_k \mid s_{k-1}, s_k, \mathbf{x}_{k-1}), \quad (2.2)$$

in order to simplify the implementation of the Bayesian Track-Before-Detect solutions. In this case, the process $(s_k)_{k \in \mathbb{N}}$ is a two-state Markov chain with transition probabilities

$$P_b = p(s_k = 1 \mid s_{k-1} = 0), \quad (2.3)$$

$$P_d = p(s_k = 0 \mid s_{k-1} = 1), \quad (2.4)$$

where P_b is the probability of target "birth" (or appearance) and P_d is the probability of target "death" (or disappearance), leading to the following transition matrix

$$\Pi = \begin{bmatrix} 1 - P_b & P_b \\ P_d & 1 - P_d \end{bmatrix}. \quad (2.5)$$

Finally, at step $k = 0$, let us define by $P_0 = p(s_0 = 1)$. On the other hand, two transition densities have to be specified for the state \mathbf{x}_k :

- $p(\mathbf{x}_k \mid s_k = 1, s_{k-1} = 1, \mathbf{x}_{k-1})$ the continuing density that models the target dynamic. In order to alleviate the notation, it will be denoted as $p_c(\mathbf{x}_k \mid \mathbf{x}_{k-1})$.
- $p(\mathbf{x}_k \mid s_k = 1, s_{k-1} = 0, \mathbf{x}_{k-1})$ the birth density that models how the target appears in the radar surveillance area. The dependence in \mathbf{x}_{k-1} can be always removed in that case since \mathbf{x}_{k-1} has no physical meaning. This density will be referred as $p_b(\mathbf{x}_k)$ in the following.

Note that the densities $p(\mathbf{x}_k \mid s_k = 0, s_{k-1} = 1, \mathbf{x}_{k-1})$ and $p(\mathbf{x}_k \mid s_k = 0, s_{k-1} = 0, \mathbf{x}_{k-1})$ that represent the state \mathbf{x}_k when it is absent from the radar surveillance area do not need to be defined since the state \mathbf{x}_k has no physical meaning when $s_k = 0$.

In summary, the state model defined in Eq. (2.2) requires the knowledge of the two transition probabilities P_b and P_d , the initial probability P_0 and the two states densities : the birth density and the prior target dynamical density. This model is very general and can handle non-linear target motion (in particular for the target dynamics). Note that, throughout this thesis, for the sake of simplicity, a linear model for the target dynamic will be used.

2.2.2 Model used in this work

The performance of the Track-Before-Detect algorithms proposed in this thesis will be evaluated via Monte Carlo simulations. Therefore, in order to avoid prohibitive computational time, we will restrict our study to a target moving in a two dimensional space. The extension of the Track-Before-Detect solutions to a target state \mathbf{x}_k with one or two additional dimensions is of course straightforward and does not lead to any theoretical issue but will rather increase the computational time required to evaluate the performance.

Thus, let us consider a target evolving (when present) in the area defined in polar coordinates by $\mathcal{D} = [r_{min}, r_{max}] \times [\theta_{min}, \theta_{max}]$ which corresponds to the surveillance area covered by the radar under consideration. The area \mathcal{D} is illustrated in Figure 2.2. Then, let us define by $\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$ the target state vector where (x_k, y_k) and (\dot{x}_k, \dot{y}_k) represent respectively its position and its velocity in Cartesian coordinates. Note that here, two systems of coordinates are used, polar and Cartesian, for the sake of convenience.

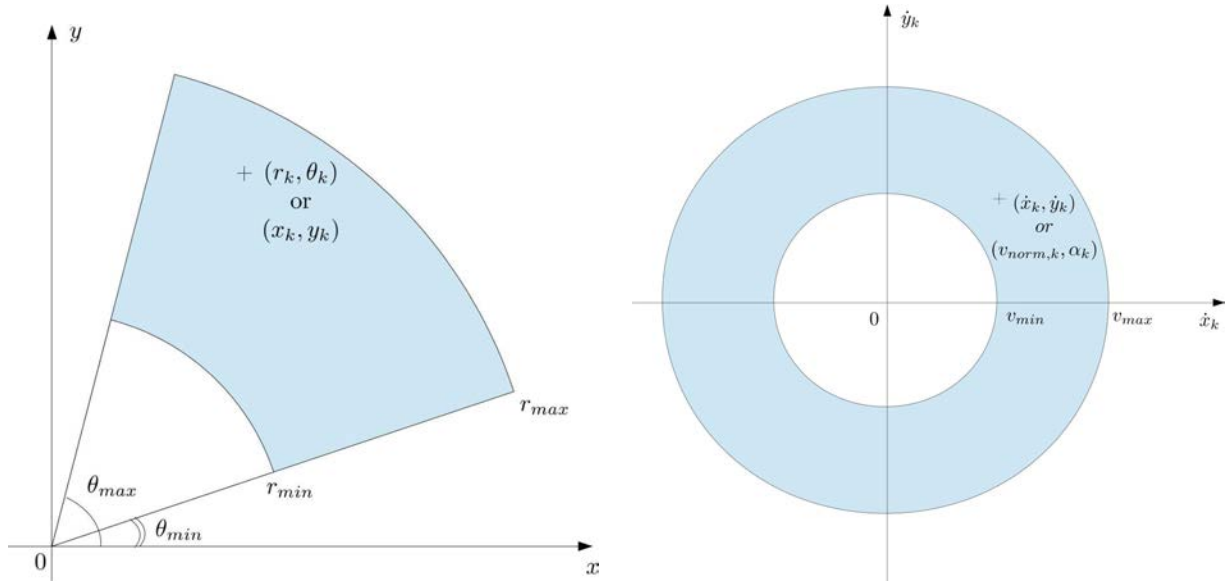


Figure 2.2 – Left: Surveillance area covered by the radar. Right: Area to initialize the velocity components (\dot{x}_k, \dot{y}_k) .

Indeed, the radar signal processing stage is well suited to polar coordinates (the radar naturally provides range and angle information, see section 1.1) while linear motion is easier to handle in Cartesian coordinates than in polar coordinates. Therefore, assuming that the radar is located at the origin, we also define by $r_k = \sqrt{x_k^2 + y_k^2}$ the target range with respect to the radar and by $\theta_k = \arctan(\frac{y_k}{x_k})$ the target azimuth. The inversion formulas are classic and simply given by $x_k = r_k \cos(\theta_k)$ and $y_k = r_k \sin(\theta_k)$. In the same manner, we define by $v_{norm,k} = \sqrt{\dot{x}_k^2 + \dot{y}_k^2}$ the velocity norm and $\alpha_k = \arctan(\frac{\dot{y}_k}{\dot{x}_k})$ the velocity direction in polar coordinates. Finally, in the following, the two representations will be used depending on the situation where they are the best suited.

The linear target dynamical model is chosen as follows [BSLK01]:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k, \quad (2.6)$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_S & 0 \\ 0 & \mathbf{F}_S \end{bmatrix} \text{ with } \mathbf{F}_S = \begin{bmatrix} 1 & T_S \\ 0 & 1 \end{bmatrix},$$

and T_S represents the sampling period of the measurements (or the duration of a radar cycle). The noise \mathbf{v}_k is assumed white and Gaussian with covariance matrix [BSLK01]

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_S & 0 \\ 0 & \mathbf{Q}_S \end{bmatrix}, \text{ where } \mathbf{Q}_S = q_S \begin{bmatrix} T_S^3/3 & T_S^2/2 \\ T_S^2/2 & T_S \end{bmatrix}. \quad (2.7)$$

Concerning the birth density $p_b(\cdot)$, the position (r_k, θ_k) and the velocity (v_k, α_k) are assumed to be distributed independently as follows:

- $p_b(r_k, \theta_k) = \mathcal{U}(r_{min}, r_{max}) \times \mathcal{U}(\theta_{min}, \theta_{max})$.

- $p_b(v_{norm,k}, \alpha_k) = \mathcal{U}(v_{min}, v_{max}) \times \mathcal{U}(0, 2\pi)$ (where v_{min} and v_{max} are the minimum and the maximum velocity reachable by the target). The domain used to initialize the velocity components is illustrated in Figure 2.2, of course v_{min} can be set to 0 if desired.

The choice of the uniform distribution both for the position and the velocity corresponds to the least possible informative prior. It means that when a target appears in the radar window, it can be located anywhere in the area \mathcal{D} with a velocity vector in any direction. The only *a priori* information used is the constants v_{min} and v_{max} which can be easily obtained by physical considerations (*e.g.* an aircraft has a limit velocity). However, if some other informations are available about the target appearance area (*e.g.* an airport) or direction, they should be taken into account in the prior birth density. Here, in order to keep the model as general as possible, this case will not be considered.

2.3 Measurement model

2.3.1 Raw radar data

We consider here a measurement model based on the presentation detailed in chapter 1 paragraph 1.1.8 with only some slight differences. It is provided by:

$$\mathbf{z}_k = s_k \rho e^{j\varphi_k} \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k. \quad (2.8)$$

The phase φ_k is assumed to be uniformly drawn over the interval $[0, 2\pi)$ while the noise \mathbf{n}_k is a zero-mean circular complex Gaussian vector with a known covariance matrix $\mathbf{\Gamma}$. The first difference introduced here concerns the modulus ρ which is assumed constant and unknown. This corresponds in the radar terminology to the *Swerling 0* fluctuation model [Sko80] – the other fluctuation models will be considered in chapter 4. The second difference concerns the introduction of the variable s_k in the measurement equation (2.8) in order to take into account the presence or the absence of the target in the measurement \mathbf{z}_k . Remark that when the target is absent, the measurement \mathbf{z}_k consists of noise only.

The function $\mathbf{h}(\cdot)$ depends on the application considered: for instance, in optics, it is often chosen with a Gaussian shape [TBS98]. Nevertheless, as we are here concerned by a radar tracking application, this measurement function $\mathbf{h}(\cdot)$ corresponds to the radar ambiguity function. For the sake of simplicity and also for computational cost reasons, we will restrict ourselves in this manuscript to the range and azimuth parameters. Of course other parameters (*e.g.* Doppler) may be easily added to the model.

Thus, let us consider a radar transmitting a chirp signal with bandwidth B and pulse duration T_p (see paragraph 1.1.5) and receiving the backscattered signal via a linear array with N_a antennas spaced by d . In a first step, a beamforming operation is realized for different directions

$$\theta_v = \theta_{min} + (v + \frac{1}{2})\Delta_\theta, \quad v = 0, \dots, N_\theta - 1, \quad (2.9)$$

where $\Delta_\theta = 0.886 \frac{\lambda}{N_a d}$ is the half-power beam-width (see paragraph 1.1.7) and $N_\theta = \left\lceil \frac{\theta_{max} - \theta_{min}}{\Delta_\theta} \right\rceil$ is the number of azimuth cells (here $\lceil \cdot \rceil$ is the ceiling function). The corresponding ambiguity function along the azimuth axis is then given by (see paragraph

1.1.7):

$$h_{\theta}^v(\theta_k) = \frac{\sin\left(\frac{N_a \psi_{\theta_v}}{2}\right)}{N_a \sin\left(\frac{\psi_{\theta_v}}{2}\right)}, \quad (2.10)$$

where $\psi_{\theta_v} = \frac{2\pi d}{\lambda} (\cos(\theta_k) - \cos(\theta_v))$.

Then, for each direction θ_v a range matched-filter is performed to the corresponding received signal. The output signal is sampled at period $1/B$ leading to the following ambiguity function along the range direction (see paragraph 1.1.5):

$$h_r^u(r_k) = \frac{\sin\left(\pi B \tau_u \left(1 - \frac{|\tau_u|}{T_p}\right)\right)}{\pi B \tau_u} \text{ for } |\tau_u| \leq T_p, \quad (2.11)$$

where $\tau_u = 2(r_k - r_u)/c$ and

$$r_u = r_{min} + \left(u + \frac{1}{2}\right)\Delta_r, \quad u = 0, \dots, N_r - 1, \quad (2.12)$$

are the range cells corresponding to the sampling instants, with Δ_r the range resolution equal to $\frac{c}{2B}$ and $N_r = \left\lceil \frac{r_{max} - r_{min}}{\Delta_r} \right\rceil$ the number of range cells.

Finally, the overall ambiguity function in range and azimuth $\mathbf{h}(\cdot)$ is a two dimensional image consisting of $N_c = N_r \times N_{\theta}$ cells where the value in the cell (u, v) is simply provided by the product $h_r^u(r_k)h_{\theta}^v(\theta_k)$. For mathematical considerations, we rewrite the function $\mathbf{h}(\cdot)$ as a vector of size N_c by using the following mapping: $l = u + (v - 1) \times N_r$, *i.e.* the value of the l -th component is given by $h^l(\mathbf{x}_k) = h_r^u(r_k)h_{\theta}^v(\theta_k)$. A scheme of the proposed mapping is given in Figure 2.3. Furthermore, for the sake of compactness, the vector $\mathbf{h}(\mathbf{x}_k)$ will be denoted by \mathbf{h}_k in the sequel.

Note that here, since no Doppler measurement is considered, the ambiguity function does not depend on the velocity parameters (\dot{x}_k, \dot{y}_k) and as a consequence neither does the equation of the measurement (2.8). However, there is no difficulty to handle such a situation where the measurement depends only on a subset of the state parameters. The connection is ensured by the prior model (*i.e.* the target dynamical model) that links velocity components with the position components, themselves related to the measurement \mathbf{z}_k . Note also that an additional Doppler shift measurement introduced in \mathbf{z}_k would provide partial information on this target velocity, and thus could be exploited to enhance the tracking filter.

2.3.2 Target Signal to Noise Ratio

An important notion that must be clearly defined is the target SNR (Signal to Noise Ratio). A possible definition, from Eq. (1.14), is $\text{SNR} = 10 \log_{10} \left(\frac{\rho^2}{2\sigma^2} \right)$. First, note that we implicitly made the hypothesis that the noise covariance matrix is $\mathbf{\Gamma} = 2\sigma^2 \mathbf{I}_{N_c}$, *i.e.* noise samples are independent with the same variance. Second, remark that this definition represents the maximum SNR reachable by the processing and is obtained when the target is exactly located at the center of the cell, *i.e.* when $h^l(r_u, \theta_v) = 1$; for other target positions, the energy extracted by the processing will be lower due to a target located outside the sampling grid. Clearly, performance of the Track-Before-Detect algorithms will highly depend on the target SNR.

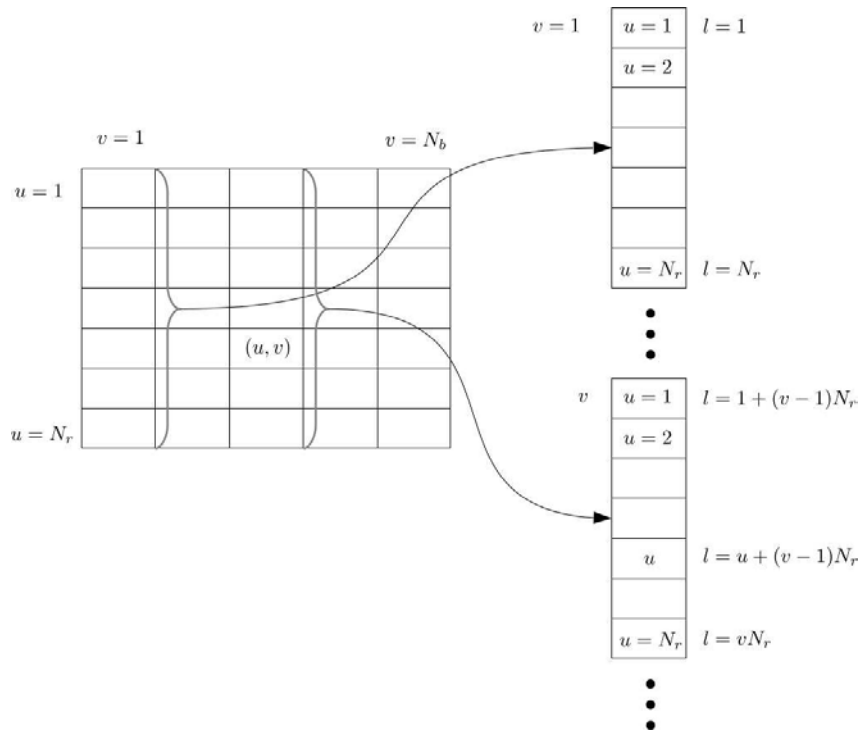


Figure 2.3 – Mapping between indices (u, v) and l .

2.4 A particle filter solution for the Track-Before-Detect problem

Previously, a state-space model has been set up in order to model the Track-Before-Detect problem in the HMM framework. The aim is now to estimate recursively the posterior density $p(\mathbf{x}_k, s_k | \mathbf{z}_{1:k})$. Since here the hidden state is hybrid (continuous variable \mathbf{x}_k and discrete variable s_k), it is convenient to reorganize the posterior density as follows:

$$p(\mathbf{x}_k, s_k | \mathbf{z}_{1:k}) = p(s_k | \mathbf{z}_{1:k}) p(\mathbf{x}_k | s_k, \mathbf{z}_{1:k}). \quad (2.13)$$

When $s_k = 0$, the state \mathbf{x}_k is meaningless and independent from the measurements $\mathbf{z}_{1:k}$ so that the density $p(\mathbf{x}_k | s_k = 0, \mathbf{z}_{1:k})$ does not need to be evaluated. On the contrary, when $s_k = 1$, the posterior density $p(\mathbf{x}_k | s_k = 1, \mathbf{z}_{1:k})$ allows to calculate estimators $\hat{\mathbf{x}}_{k|k}$ and $\hat{\mathbf{P}}_{k|k}$ defined respectively in Eq. (1.96) and (1.97) while the posterior probability of target existence $P_{e,k} = p(s_k = 1 | \mathbf{z}_{1:k})$ provides some information about the presence or the absence of the target in the radar window.

2.4.1 The TBD particle filter

In practice, the conceptual Bayesian filter defined in paragraph 1.2.2 can be derived for the proposed model but the exact solution is intractable. Therefore we must resort to some approximations. Methods based on the EKF and the UKF would be inoperative, essentially because the measurement equation (2.8) is highly non-linear and the birth density $p_b(\mathbf{x}_k)$ dramatically differs from a Gaussian density. In the other hand, due to

the large size of the state space (essentially the space \mathcal{D}), a grid-based approach seems unrealistic to implement for real-time applications. In order to overcome these difficulties, a solution based on the particle filter was proposed by Salmond *et al.* [SB01] and is detailed in the sequel.

Let us assume that a set of particles $\{(s_{k-1}^i, \mathbf{x}_{k-1}^i), w_{k-1}^i\}_{i=1}^{N_p}$, approximating the posterior density $p(s_{k-1}, \mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})$ at step $k-1$, is available:

$$p(s_{k-1}, \mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) \approx \sum_{i=1}^{N_p} w_{k-1}^i \delta_{(s_{k-1}^i, \mathbf{x}_{k-1}^i)}(s_{k-1}, \mathbf{x}_{k-1}). \quad (2.14)$$

The first step of the particle filter consists in drawing new particles (s_k^i, \mathbf{x}_k^i) from the particles at previous step. In [SB01], this is done first by drawing variables s_k^i according to the transition matrix Π defined in Eq. (2.5). Then states \mathbf{x}_k^i can be drawn conditionally to s_k^i and s_{k-1}^i . When $s_k^i = 0$, the state \mathbf{x}_k^i is meaningless and therefore does not need to be sampled. On the contrary, when $s_k^i = 1$, two cases must be considered:

1. Birth case (*i.e.* $s_{k-1}^i = 0$): the particle state \mathbf{x}_k^i is initialized with an instrumental density $q_b(\mathbf{x}_k \mid \mathbf{z}_k)$. As will be seen in this chapter, the choice of the instrumental density for the state initialization is crucial for the performance of the filter and is a key point of the TBD particle filter solutions.
2. Continuing case (*i.e.* $s_{k-1}^i = 1$): the particle was already present at step $k-1$ and is propagated with an instrumental density $q_c(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k)$.

The different cases considered when sampling particle states \mathbf{x}_k^i according to s_{k-1}^i and s_k^i are summarized in Table 2.1.

	$s_{k-1}^i = 0$	$s_{k-1}^i = 1$
$s_k^i = 0$	nothing to do	nothing to do
$s_k^i = 1$	$q_b(\mathbf{x}_k \mid \mathbf{z}_k)$	$q_c(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k)$

Table 2.1 – Instrumental densities to sample \mathbf{x}_k^i depending on s_k^i and s_{k-1}^i .

The second step of the particle filter consists in calculating the particle weights w_k^i , provided by Eq. (1.94), that differ according to the values of s_k^i and s_{k-1}^i . Considering the different possible cases, the weight expression is given by:

$$w_k^i \propto w_{k-1}^i \times \begin{cases} p(\mathbf{z}_k \mid s_k^i = 0), & \text{if } s_k^i = 0, \\ \frac{p_b(\mathbf{x}_k^i)}{q_b(\mathbf{x}_k^i \mid \mathbf{z}_k)} p(\mathbf{z}_k \mid s_k^i = 1, \mathbf{x}_k^i), & \text{if } s_k^i = 1 \text{ and } s_{k-1}^i = 0, \\ \frac{p_c(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)}{q_c(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k)} p(\mathbf{z}_k \mid s_k^i = 1, \mathbf{x}_k^i), & \text{if } s_k^i = 1 \text{ and } s_{k-1}^i = 1. \end{cases} \quad (2.15)$$

Then, weights are normalized and a resampling procedure is performed, if required, as in the generic particle filter (see Chapter 1, Algorithm 1.1). A pseudocode of a single cycle of the current particle filter, denoted here by Classic TBD Particle Filter, is described in Algorithm 2.1.

Algorithm 2.1 Classic TBD Particle Filter

Require: Particle cloud $\{(s_{k-1}^i, \mathbf{x}_{k-1}^i), w_{k-1}^i\}_{i=1}^{N_p}$ at step $k-1$,

- 1: **for** $i = 1$ to N_p **do**
- 2: Draw s_k^i according to the transition matrix defined in Eq. (2.5)
- 3: **if** $s_k^i = 1$ **then**
- 4: **if** $s_{k-1}^i = 1$ **then**
- 5: Draw $\mathbf{x}_k^i \sim q_c(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$
- 6: **else**
- 7: Draw $\mathbf{x}_k^i \sim q_b(\mathbf{x}_k | \mathbf{z}_k)$
- 8: **end if**
- 9: **end if**
- 10: Update particle weight w_k^i according to Eq. (2.15)
- 11: **end for**
- 12: Normalize weights: $w_k^i \leftarrow \frac{w_k^i}{\sum_{l=1}^{N_p} w_k^l}$, $i = 1, \dots, N_p$
- 13: Compute N_{eff} according to Eq. (1.98).
- 14: **if** $N_{\text{eff}} < \beta N_p$ **then**
- 15: Resample N_p particles
- 16: Reset weights: $w_k^i \leftarrow \frac{1}{N_p}$, $i = 1, \dots, N_p$
- 17: **end if**
- 18: **return** $\{(s_k^i, \mathbf{x}_k^i), w_k^i\}_{i=1}^{N_p}$

Finally, the probability of presence $P_{e,k}$ can be estimated from the set of particles $\{(s_k^i, \mathbf{x}_k^i), w_k^i\}_{i=1}^{N_p}$ by:

$$\hat{P}_{e,k} = \sum_{i=1}^{N_p} s_k^i w_k^i, \quad (2.16)$$

while the target state \mathbf{x}_k can be estimated by:

$$\hat{\mathbf{x}}_{k|k} = \frac{1}{\hat{P}_{e,k}} \sum_{i=1}^{N_p} s_k^i w_k^i \mathbf{x}_k^i, \quad (2.17)$$

and the posterior covariance matrix by:

$$\hat{\mathbf{P}}_{k|k} = \frac{1}{\hat{P}_{e,k}} \sum_{i=1}^{N_p} s_k^i w_k^i (\mathbf{x}_k^i - \hat{\mathbf{x}}_{k|k}) (\mathbf{x}_k^i - \hat{\mathbf{x}}_{k|k})^T. \quad (2.18)$$

2.4.2 Measurement likelihood

The calculation of the weights in Eq. (2.15) requires the likelihood function $p(\mathbf{z}_k | s_k, \mathbf{x}_k)$. However, in the particular case of the TBD particle filter, this density is not directly available since the measurement equation (2.8) depends on the unknown parameters ρ and φ_k which correspond to the target amplitude parameters. In fact, only the density $p(\mathbf{z}_k | s_k = 1, \mathbf{x}_k, \varphi_k, \rho)$ is directly provided from Eq. (2.8). This is a complex Gaussian

density with mean $s_k \rho e^{j\varphi_k} \mathbf{h}_k$ and covariance matrix $\mathbf{\Gamma}$, *i.e.*

$$p(\mathbf{z}_k | s_k = 1, \mathbf{x}_k, \varphi_k, \rho) = \frac{1}{\pi_c^N \det(\mathbf{\Gamma})} \exp \left\{ -(\mathbf{z}_k - s_k \rho e^{j\varphi_k} \mathbf{h}_k)^H \mathbf{\Gamma}^{-1} (\mathbf{z}_k - s_k \rho e^{j\varphi_k} \mathbf{h}_k) \right\}, \quad (2.19)$$

that expands as follows:

$$p(\mathbf{z}_k | s_k, \mathbf{x}_k, \rho, \varphi_k) = \frac{1}{\pi^{N_c} \det(\mathbf{\Gamma})} \exp \left\{ -\mathbf{z}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k \right\} \times \exp \left\{ -s_k \rho^2 \mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k + 2s_k \rho |\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k| \cos(\varphi_k - \zeta_k) \right\}, \quad (2.20)$$

where $\zeta_k = \arg(\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k)$. First note that when $s_k = 0$ the likelihood in Eq. (2.20) is independent from the parameters \mathbf{x}_k , ρ and φ_k . Therefore the likelihood $p(\mathbf{z}_k | s_k = 0, \mathbf{x}_k)$ is a constant provided by

$$p(\mathbf{z}_k | s_k = 0) = \frac{1}{\pi^{N_c} \det(\mathbf{\Gamma})} \exp \left\{ -\mathbf{z}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k \right\}. \quad (2.21)$$

On the contrary, when $s_k = 1$, the likelihood in Eq. (2.20) is, of course, dependent on the parameters ρ , φ_k , \mathbf{x}_k . Additional developments must then be performed in order to evaluate the likelihood $p(\mathbf{z}_k | s_k = 1, \mathbf{x}_k)$. Several strategies have been proposed in the literature to deal with the phase φ_k . As this chapter focuses on the TBD particle filter, we will consider here only the best solution detailed in paragraph 2.4.2.1. Further developments and details will be provided in chapter 4. Concerning the modulus ρ , we will use the approach proposed by Kitagawa [Kit98] which is detailed in paragraph 2.4.2.2.

Let us finally note that in order to alleviate the notation, the likelihood $p(\mathbf{z}_k | s_k = 1, \mathbf{x}_k)$ will be denoted by $p(\mathbf{z}_k | \mathbf{x}_k)$ in the rest of the chapter since it depends on \mathbf{x}_k only when $s_k = 1$. Moreover, as the particle filter requires the calculation of the likelihood $p(\mathbf{z}_k | s_k, \mathbf{x}_k)$ only up to a constant, it is convenient to divide the expression in Eq. (2.20) by the likelihood term $p(\mathbf{z}_k | s_k = 0)$ defined in Eq. (2.21). In the sequel, the likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$ will be always calculated up to this constant term. Thus, in that case, the weight equation (2.15) becomes

$$w_k^i \propto w_{k-1}^i \times \begin{cases} 1, & \text{if } s_k^i = 0, \\ \frac{p_b(\mathbf{x}_k^i)}{q_b(\mathbf{x}_k^i | \mathbf{z}_k)} p(\mathbf{z}_k | \mathbf{x}_k^i), & \text{if } s_k^i = 1 \text{ and } s_{k-1}^i = 0, \\ \frac{p_c(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i)}{q_c(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} p(\mathbf{z}_k | \mathbf{x}_k^i), & \text{if } s_k^i = 1 \text{ and } s_{k-1}^i = 1. \end{cases} \quad (2.22)$$

2.4.2.1 Eliminating the random phase

The best way to eliminate the random phase φ_k consists in marginalizing it in the likelihood defined in Eq. (2.20). This method was first proposed in [DRC12]. It leads to:

$$p(\mathbf{z}_k | \mathbf{x}_k) \propto \exp \left\{ -\rho^2 \mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k \right\} I_0 \left(2\rho |\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k| \right), \quad (2.23)$$

where $I_0(\cdot)$ is the modified Bessel function of the first kind, *i.e.*

$$I_0(x) = \sum_{l=0}^{+\infty} \frac{\left(\frac{x}{2}\right)^{2l}}{(l!)^2}. \quad (2.24)$$

2.4.2.2 Dealing with the unknown parameter ρ

Contrary to the phase φ_k that randomly fluctuates from step to step, the parameter ρ is in this chapter assumed constant. Thus it might be preferable to estimate it rather than marginalize it (which leads to an intractable integral). The problem of state-space models with unknown static parameter has been widely studied in the literature [Kit98, Sto02, ADST04]. A convenient solution, easy to implement, consists in introducing an artificial Markovian dynamic on the static parameter ρ and adding it to the state vector \mathbf{x}_k , *i.e.* $\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k, \rho_k]^T$. As the parameter ρ_k has been appended to the hidden state, its evolution must be specified *a priori*. As for the position and the velocity, two cases must be considered:

- The continuing case (*i.e.* $s_k = 1$ and $s_{k-1} = 1$) where the parameter ρ_k evolves according to the following equation [ADST04],

$$\rho_k = \rho_{k-1} + \varepsilon_k, \quad (2.25)$$

where ε_k is a white Gaussian noise with a "small" variance σ_ρ^2 independent of \mathbf{v}_k .

- The birth case (*i.e.* $s_k = 1$ and $s_{k-1} = 0$), where the parameter ρ_k is assumed uniformly drawn over the interval $[\rho_{min}, \rho_{max}]$, *i.e.* $p_b(\rho_k) = \mathcal{U}(\rho_{min}, \rho_{max})$ – note that we may sometimes replace ρ_{min} and ρ_{max} by their corresponding SNR value (see paragraph 2.3.2), that is SNR_{min} and SNR_{max} .

Moreover, as variable ρ_k has been added to the state vector \mathbf{x}_k , the "new" likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$ can simply be calculated by replacing ρ by ρ_k in Eq. (2.23).

Finally, note that in most of the articles dealing with Track-Before-Detect particle filters, the parameter ρ_k is not assumed to be constant but rather directly a component of the state vector with dynamic model (2.25). Here, we prefer to assume that ρ is an unknown constant parameter, following the *Swerling 0* model. We then use the proposed method to estimate it but do not model it *a priori* that way. Obviously the difference between the two approaches is just conceptual and in practice they are completely equivalent.

2.4.2.3 Truncating the ambiguity function

The ambiguity function presents significant values only in a small subset of cells around the target location while being negligible elsewhere. Therefore, in order to avoid unnecessary computations, Salmond *et al.* [SB01] have proposed to keep only a subset of cells $\mathcal{V}_{\mathbf{x}_k}$ where the ambiguity function remains significant. For a state \mathbf{x}_k located in cell (u_k, v_k) , the set $\mathcal{V}_{\mathbf{x}_k}$ may be defined as

$$\mathcal{V}_{\mathbf{x}_k} = \{(u, v) \mid |u_k - u| \leq \delta_{h_r}, \text{ and } |v_k - v| \leq \delta_{h_\theta}\}. \quad (2.26)$$

From this definition, the ambiguity function will be calculated over $N_{\delta_{h_r}} \times N_{\delta_{h_\theta}}$ cells – where $N_{\delta_{h_r}} = 2\delta_{h_r} + 1$ and $N_{\delta_{h_\theta}} = 2\delta_{h_\theta} + 1$ – rather than N_c cells. In Figure 2.4, an illustration of the subset $\mathcal{V}_{\mathbf{x}_k}$ is proposed. A problem arising from the direct calculation of the likelihood (2.23) is the prohibitive computational cost induced by the large number

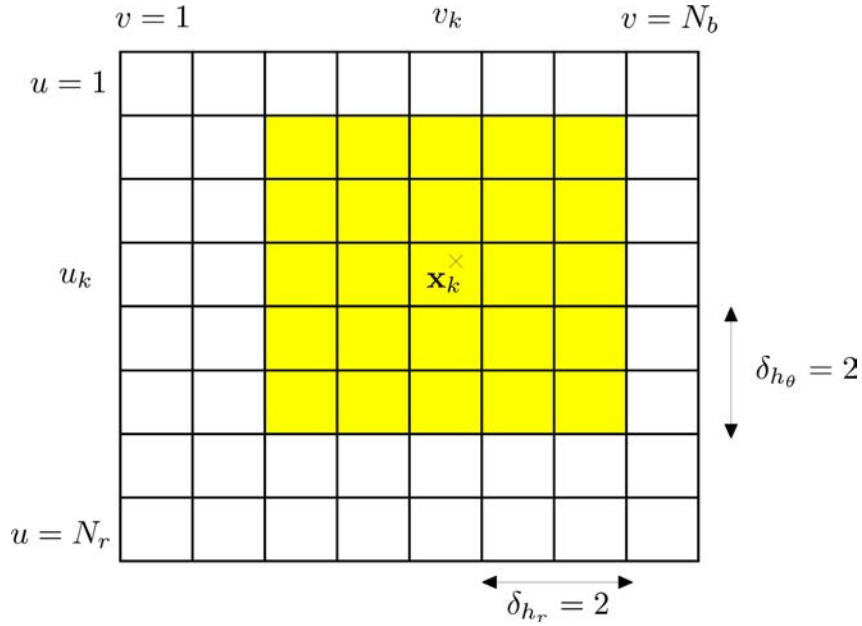


Figure 2.4 – An example of the subset $\mathcal{V}_{\mathbf{x}_k}$ (in yellow), for a target located in cell (u_k, v_k) , with $\delta_{h_r} = 2$ and $\delta_{h_\theta} = 2$.

of cells N_c included in the measurement \mathbf{z}_k . Indeed, from a theoretical point of view, the scalar product quantities $\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k$ and $\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k$ in Eq. (2.23) must be evaluated over the N_c cells, *i.e.*

$$\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k = \sum_{l=1}^{N_c} \text{conj}(h_k^l) \tilde{h}_k^l \quad \text{and} \quad \mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k = \sum_{l=1}^{N_c} \text{conj}(h_k^l) \tilde{z}_k^l, \quad (2.27)$$

where the samples \tilde{h}_k^l and \tilde{z}_k^l are respectively the components of vectors $\mathbf{\Gamma}^{-1} \mathbf{h}_k$ and $\mathbf{\Gamma}^{-1} \mathbf{z}_k$ and $\text{conj}(\cdot)$ is the complex conjugate operator. Fortunately, by truncating the ambiguity function, the previous quantities are simply evaluated over the small subset $\mathcal{V}_{\mathbf{x}_k}$, *i.e.*

$$\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k = \sum_{l \in \mathcal{V}_{\mathbf{x}_k}} \text{conj}(h_k^l) \tilde{h}_k^l \quad \text{and} \quad \mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k = \sum_{l \in \mathcal{V}_{\mathbf{x}_k}} \text{conj}(h_k^l) \tilde{z}_k^l, \quad (2.28)$$

Note that here index l refers to the index (u, v) (as explained in paragraph 2.3.1 and Figure 2.3 for details). Thus, for instance, with $\delta_{h_r} = 2$ and $\delta_{h_\theta} = 2$, the previous quantities are computed over only 25 cells, which is much smaller than the N_c cells. Lastly, in the following, \mathbf{h}_k will refer indifferently to the full ambiguity function or the truncated ambiguity function as it does not change the presented algorithms.

2.5 Instrumental density

As outlined in paragraph 1.2.4.5, the instrumental density may impact dramatically the performance of the particle filter. This is especially true in the TBD application for the birth density which samples uniformly the position in the very large space \mathcal{D} (see Figure 2.2).

A first contribution of this work consists in studying the instrumental densities for all the components of the state vector (s_k, \mathbf{x}_k) . For each of them, we derive the optimal instrumental density¹ $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k)$ and then propose some approximations that still take into account the measurement \mathbf{z}_k for sampling the particle state (s_k^i, \mathbf{x}_k^i) . Finally, in section 2.7, the influence of the different instrumental densities is studied via Monte Carlo simulations.

Let us first consider rapidly the instrumental density for the continuing case. The continuing particles are propagated via Eq. (2.6), which corresponds to a very classic model. As we stressed in paragraph 1.2.4.5, taking the prior as instrumental density in such a situation is enough to ensure good performance; using a more "sophisticated" instrumental density will induce an additional computational cost for only a small gain [AMGC02]. Therefore, in the following, we choose as instrumental density for the continuing case the prior, *i.e.* $q_c(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k) = p_c(\mathbf{x}_k | \mathbf{x}_{k-1})$.

2.5.1 Instrumental density for the initialization of the position

2.5.1.1 The optimal instrumental density

The initialization of the particle position is the key point of the Track-Before-Detect particle filter. Indeed, the likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$ in Eq. (2.23) highly depends on the vector \mathbf{h}_k and, consequently, on the position (r_k, θ_k) . Thus, simply using the prior, *i.e.* the uniform distribution over the set \mathcal{D} , as instrumental density will in one hand require to use a lot of particles to properly sample the set \mathcal{D} and, in the other hand, lead to a large variance of the importance weights (*i.e.* small N_{eff}) since the particles will be set indifferently in the area whatever the value of the likelihood (high or low). Therefore, another instrumental density should be proposed in order to "carefully" initialize the particle positions.

To do so, we propose to start from the (intractable) optimal instrumental density and then resort to some approximations. From paragraph 1.2.4.5 the optimal instrumental density is given by $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k)$. In the birth case considered here, this density does not depend on the previous state \mathbf{x}_{k-1} . Moreover, in the sequel, we will consider the target position in polar coordinates (*i.e.* (r_k, θ_k)). Indeed, since the radar ambiguity function is defined with these coordinates (see Eq. (1.49)) it simplifies the definition of the instrumental density for initialization of the position. Thus, the instrumental density for the position will be denoted as $p_b(r_k, \theta_k | \mathbf{z}_k)$ while the prior density $p_b(x_k, y_k)$ will be denoted as $p_b(r_k, \theta_k)$. In a similar way, in this section the likelihood will be defined with the polar coordinates rather than with the Cartesian coordinates, and thus $p(\mathbf{z}_k | \mathbf{x}_k)$ will be denoted by $p(\mathbf{z}_k | r_k, \theta_k)$. Note that these two likelihoods represent the same quantity, even if the velocity components are not considered in the expression $p(\mathbf{z}_k | r_k, \theta_k)$. Indeed, recall that the measurement equation (2.8) does not depend on the velocity components (\dot{x}_k, \dot{y}_k) (see section 2.3.1).

Using Bayes rule or Eq. (1.101), the optimal instrumental density in polar coordinates

¹Recall from paragraph 1.2.4.5 that this density is often intractable and cannot be used in practice.

can be simply rewritten as follows:

$$p_b(r_k, \theta_k | \mathbf{z}_k) = \frac{p_b(r_k, \theta_k) p(\mathbf{z}_k | r_k, \theta_k)}{p(\mathbf{z}_k)}, \quad (2.29)$$

where,

$$p(\mathbf{z}_k) = \int_{r_{min}}^{r_{max}} \int_{\theta_{min}}^{\theta_{max}} p_b(r_k, \theta_k) p(\mathbf{z}_k | r_k, \theta_k) dr_k d\theta_k. \quad (2.30)$$

This last term is, according to us, intractable and therefore so is the optimal instrumental density. However, here, the independence with \mathbf{x}_{k-1} leads to the same optimal instrumental density for all the birth particles. It might then be still interesting to devote some computational resources in order to approximate it. Thus, we propose here to use a grid-based approach [AMGC02].

To this purpose, let us first discretize the space for the position \mathcal{D} . We propose to discretize each cell l as follows:

$$\begin{aligned} r^{(l,p)} &= r_l + p \frac{\Delta_r}{2(\delta_r+1)}, & p &= -\delta_r, -\delta_r + 1, \dots, 0, \dots, \delta_r - 1, \delta_r, \\ \theta^{(l,q)} &= \theta_l + q \frac{\Delta_\theta}{2(\delta_\theta+1)}, & q &= -\delta_\theta, -\delta_\theta + 1, \dots, 0, \dots, \delta_\theta - 1, \delta_\theta, \end{aligned} \quad (2.31)$$

thus oversampling, in polar coordinates, the cell l , where δ_r and δ_θ are some positive integers. A scheme of the discretization for the cell l is proposed in Figure 2.5.

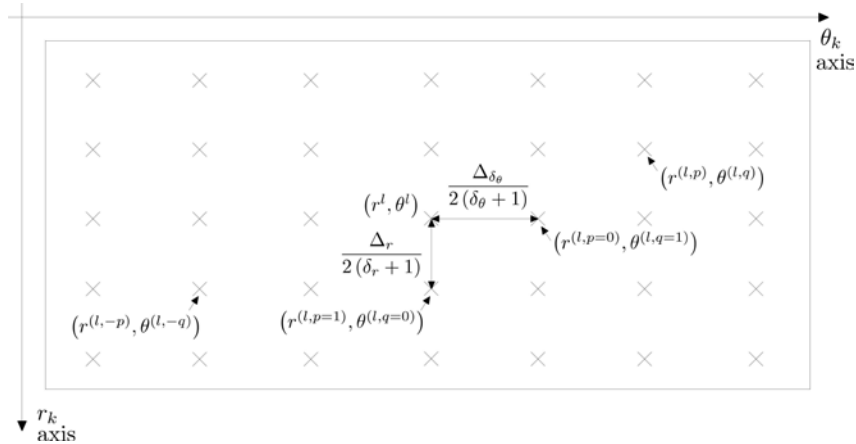


Figure 2.5 – Scheme of the discretization for the cell l .

$(r^{(l,p)}, \theta^{(l,q)})$ represents the points on the discrete grid, where the components of (l, p, q) take values respectively in $\{1, \dots, N_c\}$, $\{-\delta_r, \dots, +\delta_r\}$ and $\{-\delta_\theta, \dots, +\delta_\theta\}$. Thus, from the definition of the proposed grid, each cell l is approximated with $N_{\delta_r} = 2\delta_r + 1$ samples along r_k axis and with $N_{\delta_\theta} = 2\delta_\theta + 1$ along θ_k axis, so that the grid used to discretize the space \mathcal{D} is composed of $N_c \times N_{\delta_r} \times N_{\delta_\theta}$ points.

Then, let us approximate the density $p_b(r_k, \theta_k)$ over the proposed grid. Since the prior birth density for position is uniform, each point in the grid has the weight $\frac{1}{N_c N_{\delta_r} N_{\delta_\theta}}$, leading to the following approximation:

$$p_b(r_k, \theta_k) \approx \frac{1}{N_c N_{\delta_r} N_{\delta_\theta}} \sum_{l=1}^{N_c} \sum_{p=-\delta_r}^{\delta_r} \sum_{q=-\delta_\theta}^{\delta_\theta} \delta_{(r^{(l,p)}, \theta^{(l,q)})}(r_k, \theta_k), \quad (2.32)$$

Finally, using Eq. (2.29), the approximation of the instrumental density $p_b(r_k, \theta_k | \mathbf{z}_k)$ is obtained by

$$p_b(r_k, \theta_k | \mathbf{z}_k) \approx \sum_{l=1}^{N_c} \sum_{p=-\delta_r}^{\delta_r} \sum_{q=-\delta_\theta}^{\delta_\theta} \zeta_{k,l}^{(p,q)} \delta_{(r^{(l,p)}, \theta^{(l,q)})}(r_k, \theta_k) \quad (2.33)$$

where

$$\zeta_{k,l}^{(p,q)} \propto p(\mathbf{z}_k | r^{(l,p)}, \theta^{(l,q)}). \quad (2.34)$$

These weights cannot be computed directly since the likelihood $p(\mathbf{z}_k | r^{(l,p)}, \theta^{(l,q)})$ cannot be calculated directly as explained in paragraph 2.4.2. Indeed, a marginalization must be performed over the parameter ρ_k in the likelihood equation (2.23), leading to

$$p(\mathbf{z}_k | r^{(l,p)}, \theta^{(l,q)}) = \int p_b(\rho) p(\mathbf{z}_k | \rho, r^{(l,p)}, \theta^{(l,q)}) d\rho. \quad (2.35)$$

Again the integral in Eq. (2.35) is intractable. However, it can be simply approximated by a numerical integration, *i.e.*

$$p(\mathbf{z}_k | r^{(l,p)}, \theta^{(l,q)}) \approx \frac{1}{N_\rho} \sum_{s=0}^{N_\rho-1} p(\mathbf{z}_k | \rho^s, r^{(l,p)}, \theta^{(l,q)}), \quad (2.36)$$

with N_ρ a positive integer and $\rho^s = \rho_{min} + \frac{s}{N_\rho}(\rho_{max} - \rho_{min})$, $s = 0, \dots, N_\rho - 1$.

Although this method allows to approximate the instrumental density, in practice, it is unrealistic to use such a density since it requires to calculate $N_c \times N_{\delta_r} \times N_{\delta_\theta} \times N_\rho$ likelihoods $p(\mathbf{z}_k | \rho^s, r^{(l,p)}, \theta^{(l,q)})$ where N_c may be very large. However, this approach can be kept in mind to initialize the particles only in the interesting areas of the state space and, for instance in the cells exceeding a given threshold γ [RAG04]. This approach is developed in the next paragraph.

2.5.1.2 Approximating the instrumental density as a mixture

Let us define by

$$\mathcal{D}_{k,\gamma} = \{(r_k, \theta_k) | (r_k, \theta_k) \in \text{cell } l \text{ and } |z_k^l|^2 > \gamma\}, \quad (2.37)$$

the set of positions (r_k, θ_k) where the measurement $|z_k^l|^2$ exceeds a given threshold γ , and $\mathcal{D}_{k,\gamma}^c$ its complement (*i.e.* $\mathcal{D} = \mathcal{D}_{k,\gamma} \cap \mathcal{D}_{k,\gamma}^c$ and $\mathcal{D}_{k,\gamma} \cup \mathcal{D}_{k,\gamma}^c = \emptyset$). Let us also define $P_{\mathcal{D}_{k,\gamma}}$ the probability that the position (r_k, θ_k) belongs to the set $\mathcal{D}_{k,\gamma}$ (*i.e.* $P_{\mathcal{D}_{k,\gamma}} = p((r_k, \theta_k) \in \mathcal{D}_{k,\gamma})$), $\mathcal{I}_{k,\gamma} = \{l | |z_k^l|^2 > \gamma\}$ the set of indexes where the signal exceeds the threshold γ and $N_{\mathcal{I}_{k,\gamma}} = \text{card}(\mathcal{I}_{k,\gamma})$. Then the optimal instrumental density in Eq. (2.29) can be rewritten as a mixture with two components:

$$p_b(r_k, \theta_k | \mathbf{z}_k) = P_{\mathcal{D}_{k,\gamma}} p_b(r_k, \theta_k | \mathbf{z}_k, (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}) + (1 - P_{\mathcal{D}_{k,\gamma}}) p_b(r_k, \theta_k | \mathbf{z}_k, (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}^c), \quad (2.38)$$

where each density $p_b(r_k, \theta_k | \mathbf{z}_k, (r_k, \theta_k) \in \mathcal{D}_{k,\gamma})$ and $p_b(r_k, \theta_k | \mathbf{z}_k, (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}^c)$ can be approximated exactly in the same manner as $p_b(r_k, \theta_k | \mathbf{z}_k)$ with the only difference that

the former is calculated over the cells in $\mathcal{I}_{k,\gamma}$ while the latter is evaluated over the remaining cells. This reorganization of the optimal instrumental density as a mixture can be exploited to avoid calculating likelihoods $p(\mathbf{z}_k | r^{(l,p)}, \theta^{(l,q)})$ in the non-interesting areas of the measurement \mathbf{z}_k . To this purpose, we propose to remove the dependence on \mathbf{z}_k for the remaining cells, leading to the following instrumental density:

$$q_b(r_k, \theta_k | \mathbf{z}_k) = P_{\mathcal{D}_{k,\gamma}} p_b(r_k, \theta_k | \mathbf{z}_k, (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}) + (1 - P_{\mathcal{D}_{k,\gamma}}) p_b(r_k, \theta_k | (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}^c), \quad (2.39)$$

where $p_b(r_k, \theta_k | (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}^c)$ is simply the uniform distribution over the set $\mathcal{D}_{k,\gamma}^c$. As the proposed instrumental density differs now from the prior, the particle weight requires the calculation of the weighting term, provided by

$$\frac{p_b(r_k, \theta_k)}{q_b(r_k, \theta_k | \mathbf{z}_k)} = \begin{cases} \frac{N_{\mathcal{I}_{k,\gamma}}}{N_c N_{\delta_r} N_{\delta_\theta} P_{\mathcal{D}_{k,\gamma}} p_b(r_k, \theta_k | \mathbf{z}_k, (r_k, \theta_k) \in \mathcal{D}_{k,\gamma})}, & \text{if } (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}, \\ \left(1 - \frac{N_{\mathcal{I}_{k,\gamma}}}{N_c}\right) \frac{1}{1 - P_{\mathcal{D}_{k,\gamma}}}, & \text{if } (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}^c. \end{cases} \quad (2.40)$$

Note that the proposed instrumental density $q_b(r_k, \theta_k | \mathbf{z}_k)$ can be further simplified by also removing the dependence on \mathbf{z}_k in the first mixture component. This approach leads to the following instrumental density

$$q_b^{\mathcal{U}}(r_k, \theta_k | \mathbf{z}_k) = P_{\mathcal{D}_{k,\gamma}} p_b(r_k, \theta_k | (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}) + (1 - P_{\mathcal{D}_{k,\gamma}}) p_b(r_k, \theta_k | (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}^c), \quad (2.41)$$

with the corresponding weighting term

$$\frac{p_b(r_k, \theta_k)}{q_b^{\mathcal{U}}(r_k, \theta_k | \mathbf{z}_k)} = \begin{cases} \frac{N_{\mathcal{I}_{k,\gamma}}}{N_c P_{\mathcal{D}_{k,\gamma}}}, & \text{if } (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}, \\ \left(1 - \frac{N_{\mathcal{I}_{k,\gamma}}}{N_c}\right) \frac{1}{1 - P_{\mathcal{D}_{k,\gamma}}}, & \text{if } (r_k, \theta_k) \in \mathcal{D}_{k,\gamma}^c. \end{cases} \quad (2.42)$$

This expression leads to the heuristic solution proposed by Rutten *et al.* [RRG05] (except that they take a fix number of highest cells rather than the cells exceeding the threshold) or in [RAG04] where only the cells that exceed a given threshold are considered.

Note that in the solution proposed by Rutten *et al.* in [RRG05], or if $P_{\mathcal{D}_{k,\gamma}}$ is set to 1 in the instrumental density in Eq. (2.38), only a fix number of cells, or only the cells exceeding the threshold, will be considered to initialize the position of the particles. However, we have seen in paragraph 1.2.4.2 that the support of the prior $p_b(r_k, \theta_k)$ must be included in the support of the instrumental density $p(r_k, \theta_k | \mathbf{z}_k)$. Therefore, *stricto sensu*, from a theoretical point of view, the above instrumental densities should not be used to sample the particle positions. Nevertheless, in practice, using such densities has no noticeable consequence. Indeed, when a particle filter is implemented, the number of particles N_p is always finite. Therefore, even if the support of the prior is included in the support of the instrumental density, it may be possible that some cells will not contain any particle as for the densities that do not respect the condition on the support. Such an instrumental density with $P_{\mathcal{D}_{k,\gamma}} = 1$ will be used in the section "Simulation and Results" (*i.e.* section 2.7) with $P_{\mathcal{D}_{k,\gamma}} = 1$.

2.5.1.3 Calculation of the mixture probability and choice of the threshold

In the literature, the detection probability $P_{\mathcal{D}_{k,\gamma}}$ in the mixture (2.41) is often chosen to be equal to one, leading in practice to a mixture with only one component so that the particle positions are initialized only in the cells exceeding γ [RAG04]. However, the probability $P_{\mathcal{D}_{k,\gamma}}$ is strictly lower than 1 for any $\gamma > 0$. Thus, it might be interesting to evaluate its actual value in order to be as close as possible to the optimal instrumental density defined in Eq. (2.38).

First notice that the event $\{(r_k, \theta_k) \in \mathcal{D}_{k,\gamma}\}$ can be decomposed as

$$\{(r_k, \theta_k) \in \mathcal{D}_{k,\gamma}\} = \bigcup_{l=1}^{N_c} \left\{ \{(r_k, \theta_k) \in \text{cell } l\} \cap \{|z_k^l|^2 > \gamma\} \right\} \quad (2.43)$$

where all the events in the decomposition are disjoint, *i. e.*

$$\left\{ \{(x_k, y_k) \in \text{cell } l\} \cap \{|z_k^l|^2 > \gamma\} \right\} \cap \left\{ \{(x_k, y_k) \in \text{cell } m\} \cap \{|z_k^m|^2 > \gamma\} \right\} = \emptyset,$$

for $l \neq m$, since the target cannot be located in the cells l and m simultaneously. Moreover, by using a grid-based approach as in paragraph 2.5.1.1, the event $\{(r_k, \theta_k) \in \text{cell } l\}$ can be approximated as follows:

$$\{(r_k, \theta_k) \in \text{cell } l\} = \bigcup_{p=-\delta_r}^{\delta_r} \bigcup_{q=-\delta_\theta}^{\delta_\theta} \{(r_k, \theta_k) = (r^{(l,p)}, \theta^{(l,q)})\}, \quad (2.44)$$

where all the events $\{(r_k, \theta_k) = (r^{(l,p)}, \theta^{(l,q)})\}$ do not intersect and present the same probability $\frac{1}{N_c N_{\delta_r} N_{\delta_\theta}}$ (uniform prior). Then, it comes

$$P_{\mathcal{D}_{k,\gamma}} = \frac{1}{N_c N_{\delta_r} N_{\delta_\theta}} \sum_{l=1}^{N_c} \sum_{p=-\delta_r}^{\delta_r} \sum_{q=-\delta_\theta}^{\delta_\theta} p(|z_k^l|^2 > \gamma | r^{(l,p)}, \theta^{(l,q)}), \quad (2.45)$$

and if $h^l(\mathbf{x}_k)$ does not depend on l (*i. e.*, the calculation of the ambiguity function does not depend on the cell index l), it simplifies as follows:

$$P_{\mathcal{D}_{k,\gamma}} = \frac{1}{N_{\delta_r} N_{\delta_\theta}} \sum_{p=-\delta_r}^{\delta_r} \sum_{q=-\delta_\theta}^{\delta_\theta} p(|z_k^l|^2 > \gamma | r^{(l,p)}, \theta^{(l,q)}). \quad (2.46)$$

The probability $p(|z_k^l|^2 > \gamma | r^{(l,p)}, \theta^{(l,q)})$ can be obtained as in Eq. (2.36), by marginalization over the amplitude parameter, *i. e.*

$$p(|z_k^l|^2 > \gamma | r^{(l,p)}, \theta^{(l,q)}) \approx \frac{1}{N_\rho} \sum_{s=0}^{N_\rho-1} p(|z_k^l|^2 > \gamma | \rho^s, r^{(l,p)}, \theta^{(l,q)}). \quad (2.47)$$

The probability $p(|z_k^l|^2 > \gamma | \rho^s, r^{(l,p)}, \theta^{(l,q)})$ can be easily computed since conditionally to $(r^{(l,p)}, \theta^{(l,q)})$ and ρ^s , $\frac{|z_k^l|^2}{\sigma^2}$ follows a non-central chi-square distribution with two degrees

of freedom and non centrality parameter $\lambda_{nc}^{(p,q,s)} = \frac{(\rho^s)^2 |h^l(r_l^q, \theta_l^q)|^2}{\sigma^2}$. Then, denoting by $F_{\chi^2}^{-1}(\cdot | \lambda_{nc}^{(p,q,s)})$ the inverse cumulative distribution function of this non central chi-square distribution, the probability $p(|z_k^l|^2 > \gamma | r^{(l,p)}, \theta^{(l,q)})$ can be expressed as

$$p(|z_k^l|^2 > \gamma | r^{(l,p)}, \theta^{(l,q)}) \approx 1 - \frac{1}{N_\rho} \sum_{s=0}^{N_\rho-1} F_{\chi^2}^{-1}\left(\frac{\gamma}{\sigma^2} \middle| \lambda_{nc}^{(p,q,s)}\right). \quad (2.48)$$

Finally,

$$P_{\mathcal{D}_{k,\gamma}} \approx 1 - \frac{1}{N_{\delta_r} N_{\delta_\theta} N_\rho} \sum_{p=-\delta_r}^{\delta_r} \sum_{q=-\delta_\theta}^{\delta_\theta} \sum_{s=0}^{N_\rho-1} F_{\chi^2}^{-1}\left(\frac{\gamma}{\sigma^2} \middle| \lambda_{nc}^{(p,q,s)}\right). \quad (2.49)$$

Concerning the choice of the threshold γ , the classic detection threshold defined in Eq. (1.51) (*i.e.* $\gamma = -2\sigma^2 \log P_{fa}$) can be used in order to design the instrumental density. Indeed, using such a threshold will lead to properly sample approximatively $P_{fa} N_c$ cells (*i.e.* cells in $N_{\mathcal{I}_{k,\gamma}}$) while ensuring, if a target appears, that its position will be in $\mathcal{D}_{k,\gamma}$ with probability $P_{\mathcal{D}_{k,\gamma}}$. Obviously, the lower the P_{fa} , the smaller the set $\mathcal{D}_{k,\gamma}$ and accordingly the computational time to calculate the instrumental density; but in return, the smaller the probability $P_{\mathcal{D}_{k,\gamma}}$ will be. Furthermore, note that the probability $P_{\mathcal{D}_{k,\gamma}}$ is highly dependent on the target SNR prior. Indeed, for instance if a prior interval $[\rho_{min}, \rho_{max}]$ is chosen such that the corresponding SNR_{min} and SNR_{max} values are small, the probability in Eq. (2.48) will be small, and, as a consequence, so will be the probability $P_{\mathcal{D}_{k,\gamma}}$; on the contrary, if the prior interval corresponds to high values of SNR_{min} and SNR_{max} values are high, the probability $P_{\mathcal{D}_{k,\gamma}}$ will be much greater. Therefore, in Figure 2.6, we show the evolution of the probability

$$p(|z_k^l|^2 > \gamma | \rho) = 1 - \sum_{p=-\delta_r}^{\delta_r} \sum_{q=-\delta_\theta}^{\delta_\theta} p(|z_k^l|^2 > \gamma | \rho, r^{(l,p)}, \theta^{(l,q)}) \quad (2.50)$$

according to the $\text{SNR} = 10 \log_{10} \left(\frac{\rho^2}{2\sigma^2} \right)$ for different P_{fa} rather than the evolution of the probability $P_{\mathcal{D}_{k,\gamma}}$ that depends on the choice of the prior interval $[\rho_{min}, \rho_{max}]$. We can remark that for small SNR the probability $p(|z_k^l|^2 > \gamma | \rho)$ may become pretty small for small P_{fa} . Therefore it is then preferable to use a large enough P_{fa} in order to ensure that some particles are initialized in $\mathcal{D}_{k,\gamma}^c$.

2.5.2 Instrumental density for the amplitude parameter

2.5.2.1 The optimal instrumental density

In the literature, the parameter ρ_k is usually sampled according to the prior density [RRG05], *i.e.*:

- uniformly sampled in $[\rho_{min}, \rho_{max}]$ for the newborn particles;
- propagated according to equation (2.25) for continuing particles.

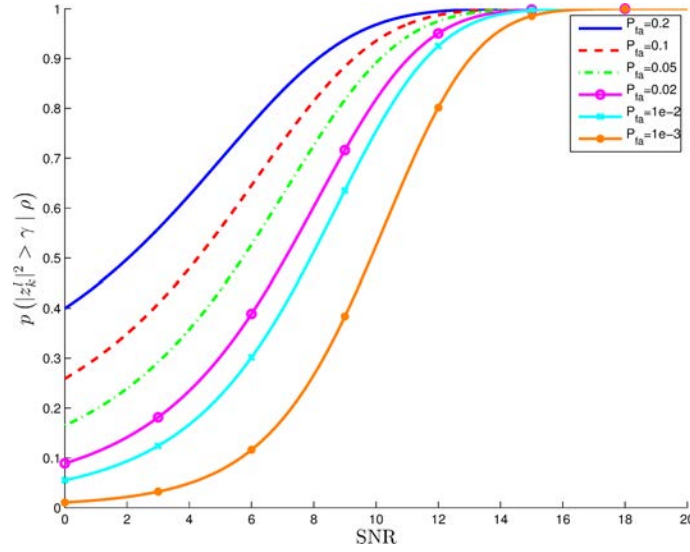


Figure 2.6 – Probability $p(|z_k^l|^2 > \gamma | \rho)$ provided in Eq. (2.50) according to the target SNR for different probabilities of false alarm. This probability takes into account the target position inside the resolution cell, and the corresponding losses. Therefore, this probability is lower than the classic probability of detection P_D in radar, where the target is assumed to be at the center of the resolution cell (*i.e.* no loss).

However, in practice, it may be inefficient because the interval $[\rho_{min}, \rho_{max}]$ may be large and the noise variance σ_ρ^2 in Eq. (2.25) is often chosen to be small. Another instrumental density may then be considered to initialize and/or propagate the amplitude parameter.

Concerning the birth amplitude parameter, the optimal instrumental density is given by $p_b(\rho_k | \mathbf{z}_k)$ and can be approximated using a grid-based approach as for the position parameters (r_k, θ_k) . However, the weight calculation will require a marginalization over the variables r_k and θ_k leading, as in Eq. (2.35), to calculate $N_c \times N_{\delta_r} \times N_{\delta_\theta} \times N_\rho$ likelihoods. This cannot be used in practice. We rather propose to factorize the instrumental density for position and amplitude as follows

$$q_b(r_k, \theta_k, \rho_k | \mathbf{z}_k) = q_b(r_k, \theta_k | \mathbf{z}_k) q_b(\rho_k | r_k, \theta_k, \mathbf{z}_k), \quad (2.51)$$

where the parameter ρ_k is now sampled according to the position (r_k, θ_k) . Using the same reasoning as in paragraph 2.5.1.1, the optimal instrumental density for amplitude parameter is then obtained by

$$q_b(\rho_k | r_k, \theta_k, \mathbf{z}_k) = p_b(\rho_k | r_k, \theta_k, \mathbf{z}_k) \propto p_b(\rho_k) p(\mathbf{z}_k | \rho_k, r_k, \theta_k), \quad (2.52)$$

and can be approximated by

$$p_b(\rho_k | r_k, \theta_k, \mathbf{z}_k) \approx \sum_{s=0}^{N_\rho-1} \zeta_{k,\rho}^s \delta_{\rho^s}(\rho_k), \quad (2.53)$$

where

$$\zeta_{k,\rho}^s \propto p(\mathbf{z}_k | \rho^s, r_k, \theta_k), \quad (2.54)$$

N_ρ is a positive integer and $\rho^s = \rho_{min} + \frac{s}{N_\rho} (\rho_{max} - \rho_{min})$, $s = 0, \dots, N_\rho - 1$. Note that N_ρ can be different from the one defined in Eq. (2.36) but we have kept here the same notation for the sake of simplicity. Furthermore, if the position of the particles are sampled with the instrumental density defined in Eq. (2.39), then N_ρ likelihoods for the positions $(r^{(l,p)}, \theta^{(l,q)})$ belonging to $\mathcal{D}_{k,\gamma}$ have been already calculated. Therefore, by taking the same N_ρ and storing the likelihoods $p(\mathbf{z}_k | \rho^s, r^{(l,p)}, \theta^{(l,q)})$ for $(r^{(l,p)}, \theta^{(l,q)}) \in \mathcal{D}_{k,\gamma}$, no extra calculation is needed. Of course, N_ρ likelihood calculations would still be required for particles belonging to $\mathcal{D}_{k,\gamma}^c$ unless another instrumental density is used instead (*e.g.*, a prior distribution). On the other hand, the weighting term induced by the (possibly different) instrumental density must be carefully calculated. If the amplitude parameter is sampled from $p_b(\rho_k | r_k, \theta_k, \mathbf{z}_k)$, the corresponding weighting term is then given by

$$\frac{p_b(\rho_k | r_k, \theta_k)}{q_b(\rho_k | r_k, \theta_k, \mathbf{z}_k)} = \frac{1}{N_\rho p_b(\rho_k | r_k, \theta_k, \mathbf{z}_k)}. \quad (2.55)$$

2.5.2.2 An instrumental density based on an estimator of the amplitude

If the instrumental density $q_b(r_k, b_k | \mathbf{z}_k)$ defined in Eq.(2.39) is not used, then it may be preferable not to use the instrumental density $p_b(\rho_k | r_k, \theta_k, \mathbf{z}_k)$ that requires to calculate N_ρ extra likelihoods per particle. Thus, we propose a different instrumental density that exploits the measurement \mathbf{z}_k at a lower computational cost. This instrumental density is composed of the two following densities for the birth and continuing cases:

$$q_b^{est}(\rho_k | r_k, \theta_k, \mathbf{z}_k) = \mathcal{N}(\rho_k; \hat{\rho}_b, \sigma_{b,\rho}^2), \quad (2.56)$$

$$q_c^{est}(\rho_k | \rho_{k-1}, r_k, \theta_k, \mathbf{z}_k) = \mathcal{N}(\rho_k; \hat{\rho}_c, \sigma_{c,\rho}^2), \quad (2.57)$$

where $\hat{\rho}_b$ and $\hat{\rho}_c$ are some estimators of ρ_k calculated from $(r_k, \theta_k, \mathbf{z}_k)$ for the birth case and from $(\rho_{k-1}, r_k, \theta_k, \mathbf{z}_k)$ for the continuing case, and $\sigma_{b,\rho}^2, \sigma_{c,\rho}^2$ are some variances defined by the user. The weighting terms induced by these instrumental densities are given by

$$\frac{p_b(\rho_k)}{q_b^{est}(\rho_k | r_k, \theta_k, \mathbf{z}_k)} = \frac{\sqrt{2\pi}\sigma_{b,\rho} \exp\left\{-\frac{(\rho_k - \hat{\rho}_b)^2}{2\sigma_{b,\rho}^2}\right\}}{\rho_{max} - \rho_{min}}, \quad (2.58)$$

$$\frac{p_c(\rho_k | \rho_{k-1})}{q_c^{est}(\rho_k | \rho_{k-1}, r_k, \theta_k, \mathbf{z}_k)} = \frac{\sigma_{c,\rho}}{\sigma_\rho} \exp\left\{-\frac{(\rho_k - \hat{\rho}_c)^2}{2\sigma_{c,\rho}^2} - \frac{(\rho_k - \rho_{k-1})^2}{2\sigma_\rho^2}\right\}. \quad (2.59)$$

Concerning the estimators, we choose a MAP approach leading to calculate $\hat{\rho}_b$ and $\hat{\rho}_c$ as

$$\hat{\rho}_b = \arg \max_{\rho_k} \left(\max_{\varphi_k} p_b(\rho_k) p(\mathbf{z}_k | r_k, \theta_k, \rho_k, \varphi_k) \right), \quad (2.60)$$

$$\hat{\rho}_c = \arg \max_{\rho_k} \left(\max_{\varphi_k} p_c(\rho_k | \rho_{k-1}) p(\mathbf{z}_k | r_k, \theta_k, \rho_k, \varphi_k) \right). \quad (2.61)$$

Note that we choose to maximize first the likelihood $p(\mathbf{z}_k | r_k, \theta_k, \rho_k, \varphi_k)$ over the phase φ_k (see Eq. (2.19)) since the corresponding expressions (for birth and continuing cases) allow to obtain a closed-form for the estimators, while using the likelihood expression

defined in Eq. (2.23) does not (because of the Bessel function). The estimator $\hat{\rho}_b$ is then obtained by

$$\hat{\rho}_b = \begin{cases} \frac{|\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k|}{\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k}, & \text{if } \rho_{min} < \frac{|\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k|}{\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k} < \rho_{max}, \\ \rho_{min}, & \text{if } \frac{|\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k|}{\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k} \leq \rho_{min}, \\ \rho_{max}, & \text{if } \frac{|\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k|}{\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k} \geq \rho_{max}, \end{cases} \quad (2.62)$$

and $\hat{\rho}_c$ by

$$\hat{\rho}_c = \frac{\rho_{k-1} + 2\sigma_\rho^2 |\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k|}{1 + 2\sigma_\rho^2 \mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k}. \quad (2.63)$$

Note that quantities $|\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k|$ and $\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k$ can be stored for each particle since they are required to compute the particle weight via the likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$ (see Eq. (2.23)).

2.5.3 Instrumental density for the velocity

As seen in paragraph 2.3, the measurement equation (2.8) does not directly depend on the velocity component (\dot{x}_k, \dot{y}_k) . Therefore, when the particle velocity components are initialized at time step k , the measurement \mathbf{z}_k does not provide any information about them and the prior must be used. This may be problematic in some cases: for instance, if a target appears in the radar window with a high SNR and a particle is initialized very close to the actual target position, the corresponding weight will be very high. Then the resampling step will tend to select this particle more often than others, and the children particles will share the same velocity components. However, this velocity sampled from the prior may tend to propagate the particles in a wrong direction. In order to avoid this last drawback, we propose a very simple strategy: instead of sampling the velocity components at step k when the particle is initialized (birth case), we propose to sample it at the next step $k + 1$. Then, if many particles have been resampled from the same birth particle at step k , their velocity components at step $k + 1$ will be different and therefore they will better explore the state space. Although there is no theoretical justification for such a choice, the state model can be changed in order to allow the velocity component of birth particles at step k to be initialized at step $k + 1$. Thus, we propose to add to the state model a variable

$$t_k = \begin{cases} t_{k-1} + 1, & \text{if } s_k = 1, \\ 0, & \text{if } s_k = 0, \end{cases} \quad (2.64)$$

that counts the number of iterations when the particle is alive, and we define the transition density as follows:

$$p(t_k, s_k, \mathbf{x}_k | t_{k-1}, s_{k-1}, \mathbf{x}_{k-1}) = p(s_k | s_{k-1}) p(t_k | t_{k-1}, s_k) p(\mathbf{x}_k | t_k, \mathbf{x}_{k-1}). \quad (2.65)$$

First note that $p(t_k | t_{k-1}, s_k)$ does not need to be sampled since the variable t_k conditionally to variable t_{k-1} and s_k is completely determined. Consequently, the transition of the state \mathbf{x}_k now depends on the variable t_k as follows

- $t_k = 0$ corresponds to the death case (*i.e.* the state is meaningless),

- $t_k = 1$ corresponds to the birth case with the density $p_b(\mathbf{x}_k)$ (except for the velocity components),
- the case $t_k = 2$ corresponds to the particles born at previous step,
- $t_k > 2$ corresponds to the continuing case with the density $p_c(\mathbf{x}_k | \mathbf{x}_{k-1})$.

Then it is now possible to initialize the velocity components at step $t_k = 2$. This can be done, for instance, by choosing as prior density

$$p(\mathbf{x}_k | t_k = 2, \mathbf{x}_{k-1}) = p_c(\rho_k | \rho_{k-1}) p_b(\dot{x}_k, \dot{y}_k) \times \mathcal{N}\left(x_k; x_{k-1} + T_S \dot{x}_k, q_S \frac{T_S^3}{3}\right) \mathcal{N}\left(y_k; y_{k-1} + T_S \dot{y}_k, q_S \frac{T_S^3}{3}\right). \quad (2.66)$$

Note that the position (x_k, y_k) is almost sampled according to the state equation (2.6) except that the variances and the covariances for the velocity components in the matrix \mathbf{Q} are set to zero. Therefore, in order to avoid unnecessary complications, the same notation $p_c(\mathbf{x}_k | \mathbf{x}_{k-1})$ is kept for both state models since they only differ by the initialization of the velocity components (*i.e.* when $t_k = 2$). Moreover, if the corresponding prior is taken to propagate the position and the velocity of the particles, no additional weighting term is induced in both cases.

2.5.4 Instrumental density for the presence variable

In the literature, the instrumental density is usually factorized in the same manner as the prior density defined in Eq. (2.2):

$$q(s_k, \mathbf{x}_k | s_{k-1}, \mathbf{x}_{k-1}, \mathbf{z}_k) = p(s_k | s_{k-1}) \times \begin{cases} q_c(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k), & \text{if } s_k = 1 \text{ and } s_{k-1} = 1, \\ q_b(\mathbf{x}_k | \mathbf{z}_k), & \text{if } s_k = 1 \text{ and } s_{k-1} = 0, \end{cases} \quad (2.67)$$

leading to sample the variable s_k from the prior transition matrix and, as a consequence, independently from the particle state \mathbf{x}_k and the measurement \mathbf{z}_k . In this case, some particles may be "killed" whereas they were located in informative areas of the state space, while some others may be drawn in non informative areas. To avoid these drawbacks, we propose to factorize the proposal density, taking into account the state \mathbf{x}_k and the measurement \mathbf{z}_k , as follows:

$$q(s_k, \mathbf{x}_k | s_{k-1}, \mathbf{x}_{k-1}, \mathbf{z}_k) = p(s_k | s_{k-1}, \mathbf{x}_k, \mathbf{z}_k) \times \begin{cases} q_c(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k), & \text{if } s_{k-1} = 1, \\ q_b(\mathbf{x}_k | \mathbf{z}_k), & \text{if } s_{k-1} = 0, \end{cases} \quad (2.68)$$

where $p(s_k | s_{k-1}, \mathbf{x}_k, \mathbf{z}_k)$ is the posterior transition probability, proportional to:

$$\begin{aligned} p(s_k = 1 | s_{k-1}, \mathbf{x}_k, \mathbf{z}_k) &\propto p(\mathbf{z}_k | \mathbf{x}_k) p(s_k = 1 | s_{k-1}), \\ p(s_k = 0 | s_{k-1}, \mathbf{x}_k, \mathbf{z}_k) &\propto p(s_k = 0 | s_{k-1}). \end{aligned} \quad (2.69)$$

For a particle i , the posterior transition probabilities are then given by:

$$p(s_k^i = 1 | s_{k-1}^i = 0, \mathbf{x}_k^i, \mathbf{z}_k) = \frac{P_b p(\mathbf{z}_k | \mathbf{x}_k^i)}{P_b p(\mathbf{z}_k | \mathbf{x}_k^i) + 1 - P_b}, \text{ if } s_{k-1}^i = 0, \quad (2.70)$$

$$p(s_k^i = 1 | s_{k-1}^i = 1, \mathbf{x}_k^i, \mathbf{z}_k) = \frac{(1 - P_d) p(\mathbf{z}_k | \mathbf{x}_k^i)}{(1 - P_d) p(\mathbf{z}_k | \mathbf{x}_k^i) + P_d}, \text{ if } s_{k-1}^i = 1. \quad (2.71)$$

Note that in Eq. (2.67) the birth and the target dynamical densities depend on variables s_k and s_{k-1} while they only depend on s_{k-1} in Eq. (2.68). With a slight abuse of notation, we have kept the same notation for the instrumental density in both cases since in practice it does not change the way to sample the particles.

It resorts from this proposed strategy that the state \mathbf{x}_k is sampled first, *i.e.* before drawing the variable s_k . Then, if \mathbf{x}_k is drawn in an area of the state space presenting a high likelihood, the corresponding posterior probability defined in Eq. (2.70) or (2.71) will be high, leading to sample the variable s_k in a more efficient manner than with the prior. As the prior is not used here as instrumental density, different weighting terms are induced, leading to the following expression of the particle weights:

$$w_k^i \propto w_{k-1}^i \times \begin{cases} \frac{p(s_k^i=0|s_{k-1})}{p(s_k^i=0|s_{k-1}^i, \mathbf{x}_k^i, \mathbf{z}_k)}, & \text{if } s_k^i = 0, \\ \frac{P_b}{p(s_k^i=1|s_{k-1}^i=0, \mathbf{x}_k^i, \mathbf{z}_k)} \times \frac{p_b(\mathbf{x}_k^i)}{q_b(\mathbf{x}_k^i|\mathbf{z}_k)} p(\mathbf{z}_k | \mathbf{x}_k^i), & \text{if } s_k^i = 1 \text{ and } s_{k-1}^i = 0, \\ \frac{1-P_d}{p(s_k^i=1|s_{k-1}^i=1, \mathbf{x}_k^i, \mathbf{z}_k)} \times \frac{p_c(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q_c(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)} p(\mathbf{z}_k | \mathbf{x}_k^i), & \text{if } s_k^i = 1 \text{ and } s_{k-1}^i = 1. \end{cases} \quad (2.72)$$

Note that we did not take into account the weighting term for state \mathbf{x}_k^i when $s_k^i = 0$ since it is meaningless. Furthermore, contrary to the prior proposal that needs to calculate the likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ only for the particles with $s_k = 1$, this new strategy requires the likelihood computation for each particle in order to draw its state parameter s_k according to Eq. (2.70) and (2.71). Therefore an additional cost is induced which occurs mainly when most of the particles share the state $s_k = 0$. On the bright side, it should be stressed that the proposed strategy computes the same number of likelihoods at each iteration, leading to constant computational time per iteration, while for the classic approach this cost depends on the number of particles in state $s_k = 1$ and may thus highly vary.

Finally, a single cycle of the proposed particle filter, that we call the Posterior TBD Particle Filter, is described in Algorithm 2.2.

2.6 Marginalized TBD particle filter

The Classic TBD particle filter samples the whole augmented state (s_k, \mathbf{x}_k) whereas the only particles that effectively participate to the estimation of \mathbf{x}_k are particles with state $s_k^i = 1$. Particles with state $s_k^i = 0$ just allow to calculate the probability of presence $\hat{P}_{e,k}$. However, we are mainly interested by the density $p(\mathbf{x}_k | s_k = 1, \mathbf{z}_{1:k})$ and the probability $P_{e,k}$ rather than the whole posterior $p(s_k, \mathbf{x}_k | \mathbf{z}_{1:k})$. Thus, following that idea, Rutten *et al.* [RGM04] developed an approach where only the quantities of interest are calculated, leading to a more efficient use of the particles.

To this purpose, the density $p(s_k, \mathbf{x}_k | \mathbf{z}_{1:k})$ is first rewritten as follows:

$$p(s_k, \mathbf{x}_k | \mathbf{z}_{1:k}) = p(s_k | \mathbf{z}_{1:k}) p(\mathbf{x}_k | s_k, \mathbf{z}_{1:k}). \quad (2.73)$$

By definition of s_k ,

$$p(s_k = 1 | \mathbf{z}_{1:k}) + p(s_k = 0 | \mathbf{z}_{1:k}) = 1. \quad (2.74)$$

Therefore, only one of the two probabilities must be computed and $p(s_k = 1 | \mathbf{z}_{1:k})$ will be considered in the sequel (*i.e.* the probability of existence $P_{e,k}$). Moreover, in Eq. (2.73),

Algorithm 2.2 Posterior TBD Particle Filter

Require: Particle cloud $\{(s_{k-1}^i, \mathbf{x}_{k-1}^i), w_{k-1}^i\}_{i=1}^{N_p}$ at step $k-1$.

- 1: **for** $i = 1$ to N_p **do**
 - 2: **if** $s_{k-1}^i = 1$ **then**
 - 3: Draw $\mathbf{x}_k^i \sim q_c(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$
 - 4: Draw s_k^i according to Eq. (2.71)
 - 5: **else**
 - 6: Draw $\mathbf{x}_k^i \sim q_b(\mathbf{x}_k | \mathbf{z}_k)$
 - 7: Draw s_k^i according to Eq. (2.70)
 - 8: **end if**
 - 9: Update particle weight w_k^i according to Eq. (2.72)
 - 10: **end for**
 - 11: Normalize weights: $w_k^i \leftarrow \frac{w_k^i}{\sum_{l=1}^{N_p} w_k^l}$, $i = 1, \dots, N_p$
 - 12: Compute N_{eff} according to Eq. (1.98).
 - 13: **if** $N_{\text{eff}} < \beta N_p$ **then**
 - 14: Resample N_p particles
 - 15: Reset weights: $w_k^i \leftarrow \frac{1}{N_p}$ $i = 1, \dots, N_p$
 - 16: **end if**
 - 17: **return** $\{(s_k^i, \mathbf{x}_k^i), w_k^i\}_{i=1}^{N_p}$
-

the density $p(\mathbf{x}_k | s_k = 1, \mathbf{z}_{1:k})$ can simply be decomposed as:

$$\begin{aligned}
 p(\mathbf{x}_k | s_k = 1, \mathbf{z}_{1:k}) &= p(s_{k-1} = 1 | s_k = 1, \mathbf{z}_{1:k}) \underbrace{p(\mathbf{x}_k | s_k = 1, s_{k-1} = 1, \mathbf{z}_{1:k})}_{\text{posterior continuing density}} + \\
 &\quad p(s_{k-1} = 0 | s_k = 1, \mathbf{z}_{1:k}) \underbrace{p(\mathbf{x}_k | s_k = 1, s_{k-1} = 0, \mathbf{z}_{1:k})}_{\text{posterior birth density}},
 \end{aligned} \tag{2.75}$$

which is a mixture with two components where:

- the first component $p(\mathbf{x}_k | s_k = 1, s_{k-1} = 1, \mathbf{z}_{1:k})$, that we call the *posterior continuing density*, considers the case where the target is present at step $k-1$. In order to alleviate the notation, it will be denoted as $p_c(\mathbf{x}_k | \mathbf{z}_{1:k})$ in the sequel.
- the second component $p(\mathbf{x}_k | s_k = 1, s_{k-1} = 0, \mathbf{z}_{1:k})$, that we call the *posterior birth density*, considers the case where the target shows up in the radar surveillance area between steps $k-1$ and k . It will be denoted as $p_b(\mathbf{x}_k | \mathbf{z}_{1:k})$ in the following.

Each of these two components can be computed using the classic recursion of the Bayesian filter. For the first component, it comes:

$$p_c(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p_c(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{p_c(\mathbf{z}_k | \mathbf{z}_{1:k-1})}, \tag{2.76}$$

where $p_c(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ is the classic predicted density obtained via the Chapman-Kolmogorov equation (1.59):

$$p_c(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_{k-1} | s_k = 1, s_{k-1} = 1, \mathbf{z}_{1:k-1}) p_c(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}. \tag{2.77}$$

The density at previous step $p(\mathbf{x}_{k-1} | s_k = 1, s_{k-1} = 1, \mathbf{z}_{1:k-1})$ is still conditioned by $s_k = 1$, but, in practice, it is easy to show, using the definition of the prior model, that the dependence with $s_k = 1$ can be removed. Indeed, from a simple Bayes rule, it comes

$$p(\mathbf{x}_{k-1} | s_k = 1, s_{k-1} = 1, \mathbf{z}_{1:k-1}) = \frac{p(\mathbf{x}_{k-1} | s_{k-1} = 1, \mathbf{z}_{1:k-1}) p(s_k = 1 | s_{k-1} = 1, \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1})}{p(s_k = 1 | s_{k-1} = 1, \mathbf{z}_{1:k-1})}. \quad (2.78)$$

Since the process $(s_k)_{k \in \mathbb{N}}$ is Markovian, the probabilities $p(s_k = 1 | s_{k-1} = 1, \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1})$ and $p(s_k = 1 | s_{k-1} = 1, \mathbf{z}_{1:k-1})$ do not depend on \mathbf{x}_{k-1} and $\mathbf{z}_{1:k-1}$. Therefore, they simplify in the last equation and it only remains the density $p(\mathbf{x}_{k-1} | s_{k-1} = 1, \mathbf{z}_{1:k-1})$ where the dependency with $s_k = 1$ has been removed. Finally, the Chapman-Kolmogorov equation (2.77) becomes

$$p_c(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_{k-1} | s_{k-1} = 1, \mathbf{z}_{1:k-1}) p_c(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}, \quad (2.79)$$

which only depends on the density at previous step and the transition probability $p_c(\mathbf{x}_k | \mathbf{x}_{k-1})$ while the normalization term $p_c(\mathbf{z}_k | \mathbf{z}_{1:k-1})$ in Eq. (2.76) is obtained by

$$p_c(\mathbf{z}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k | \mathbf{x}_k) p_c(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k. \quad (2.80)$$

In the same manner, the second component $p_b(\mathbf{x}_k | \mathbf{z}_{1:k})$ can be expressed as follows

$$p_b(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p_b(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{p_b(\mathbf{z}_k | \mathbf{z}_{1:k-1})} \quad (2.81)$$

where

$$p_b(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p_b(\mathbf{x}_{k-1} | s_k = 1, s_{k-1} = 1, \mathbf{z}_{1:k-1}) p_b(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}. \quad (2.82)$$

Here, since the density $p_b(\mathbf{x}_k | \mathbf{x}_{k-1})$ does not depend on \mathbf{x}_{k-1} , it directly comes that

$$p_b(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = p_b(\mathbf{x}_k). \quad (2.83)$$

Finally the normalization term $p_b(\mathbf{z}_k | \mathbf{z}_{1:k-1})$ is equal to

$$p_b(\mathbf{z}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k | \mathbf{x}_k) p_b(\mathbf{x}_k) d\mathbf{x}_k. \quad (2.84)$$

In practice, each density (continuing or birth) can be approximated by a particle filter. Let us assume that at step $k-1$ a set of $N_{p,c}$ particles $\{w_{k-1}^i, \mathbf{x}_{k-1}^i\}_{i=1}^{N_{p,c}}$ approximates the posterior density $p(\mathbf{x}_k | s_k = 1, \mathbf{z}_k)$. By using Eq. (2.76), the *posterior continuing density* can be approximated by a set of particles $\{\mathbf{x}_{k,c}^i, w_{k,c}^i\}_{i=1}^{N_{p,c}}$ sampled from an instrumental density $q_c(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{z}_k)$ where the unnormalized weights are equal to

$$\tilde{w}_{k,c}^i = w_{k-1}^i \frac{p_c(\mathbf{x}_{k,c}^i | \mathbf{x}_{k-1}^i)}{q_c(\mathbf{x}_{k,c}^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} p(\mathbf{z}_k | \mathbf{x}_{k,c}^i). \quad (2.85)$$

Concerning the *birth posterior density*, since alive particles \mathbf{x}_{k-1}^i do not provide any information on the newborn particles, it can be directly estimated by a set of $N_{p,b}$ particles $\{\mathbf{x}_{k,b}^i, w_{k,b}^i\}_{i=1}^{N_{p,b}}$ sampled from $q_b(\mathbf{x}_k | \mathbf{z}_k)$ where the unnormalized weights are calculated from the following equation

$$\tilde{w}_{k,b}^i = \frac{p_b(\mathbf{x}_{k,b}^i)}{q_b(\mathbf{x}_{k,b}^i | \mathbf{z}_k)} p(\mathbf{z}_k | \mathbf{x}_{k,b}^i). \quad (2.86)$$

Note that in Eq. (2.85) and in Eq. (2.86) we use the sign $=$ rather than \propto , indeed, the unnormalized weights are required to calculate other quantities that will be detailed in the sequel. Obviously the normalized weights $w_{k,c}^i$ and $w_{k,b}^i$ are simply obtained through a normalization.

In order to approximate the posterior mixture density defined in Eq. (2.75), both probabilities $p(s_{k-1} = 1 | s_k = 1, \mathbf{z}_{1:k})$ and $p(s_{k-1} = 0 | s_k = 1, \mathbf{z}_{1:k})$ must also be calculated. Again, using Bayes rule, it is easy to show that

$$\begin{aligned} p(s_{k-1} = 1 | s_k = 1, \mathbf{z}_{1:k}) &\propto (1 - P_d) P_{e,k-1} p_c(\mathbf{z}_k | \mathbf{z}_{1:k-1}), \\ p(s_{k-1} = 0 | s_k = 1, \mathbf{z}_{1:k}) &\propto P_b(1 - P_{e,k-1}) p_b(\mathbf{z}_k | \mathbf{z}_{1:k-1}). \end{aligned} \quad (2.87)$$

The calculation of the terms $p_b(\mathbf{z}_k | \mathbf{z}_{1:k-1})$ and $p_c(\mathbf{z}_k | \mathbf{z}_{1:k-1})$ is intractable. However they can be approximated via a Monte Carlo integration [VGP05] leading to the following unnormalized probabilities,

$$\begin{aligned} \hat{p}_u(s_{k-1} = 1) &= \frac{(1 - P_d) \hat{P}_{e,k-1}}{C_k} \sum_{i=1}^{N_{p,c}} \tilde{w}_{k,c}^i, \\ \hat{p}_u(s_{k-1} = 0) &= \frac{P_b(1 - \hat{P}_{e,k-1})}{N_{p,b}} \sum_{i=1}^{N_{p,b}} \tilde{w}_{k,b}^i, \end{aligned} \quad (2.88)$$

where $\hat{P}_{e,k-1}$ is the approximated probability of existence at step $k-1$ while C_k is a normalization constant given by

$$C_k = \sum_{i=1}^{N_{p,c}} w_{k-1}^i \frac{p_c(\mathbf{x}_{k,c}^i | \mathbf{x}_{k-1}^i)}{q_c(\mathbf{x}_{k,c}^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)}. \quad (2.89)$$

Note that when the instrumental density is the prior, the constant C_k does not need to be calculated since it is equal to 1. Finally the two considered probabilities can be approximated by

$$\begin{aligned} \hat{p}(s_{k-1} = 1 | s_k = 1, \mathbf{z}_{1:k}) &= \frac{\hat{p}_u(s_{k-1} = 1)}{\hat{p}_u(s_{k-1} = 1) + \hat{p}_u(s_{k-1} = 0)}, \\ \hat{p}(s_{k-1} = 0 | s_k = 1, \mathbf{z}_{1:k}) &= 1 - \hat{p}(s_{k-1} = 1 | s_k = 1, \mathbf{z}_{1:k}). \end{aligned} \quad (2.90)$$

The probability of presence $P_{e,k}$ at step k can be decomposed as follows [RGM04]:

$$P_{e,k} = \frac{(1 - P_d) P_{e,k-1} p_c(\mathbf{z}_k | \mathbf{z}_{1:k-1}) + P_b(1 - P_{e,k-1}) p_b(\mathbf{z}_k | \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})}, \quad (2.91)$$

where

$$p(\mathbf{z}_k | \mathbf{z}_{1:k-1}) \propto (1 - P_d) P_{e,k-1} p_c(\mathbf{z}_k | \mathbf{z}_{1:k-1}) + P_b(1 - P_{e,k-1}) p_b(\mathbf{z}_k | \mathbf{z}_{1:k-1}) + P_{e,k-1} P_d + (1 - P_{e,k-1})(1 - P_b). \quad (2.92)$$

Here the used of \propto means that $p(\mathbf{z}_k | \mathbf{z}_{1:k-1})$ is provided up to the constant $p(\mathbf{z}_k | s_k = 0)$ (see paragraph 2.4.2). Finally, the probability of presence can be approximated by [RGM04]:

$$\hat{P}_{e,k} = \frac{\hat{p}_u(s_{k-1} = 1) + \hat{p}_u(s_{k-1} = 0)}{\hat{p}_u(s_{k-1} = 1) + \hat{p}_u(s_{k-1} = 0) + \hat{P}_{e,k-1} P_d + (1 - \hat{P}_{e,k-1})(1 - P_b)} \quad (2.93)$$

Lastly, a single cycle of this particle filter, denoted by Marginalized TBD Particle Filter, is described in Algorithm 2.3. Note that, as the strategy proposed in paragraph 2.5.4, this algorithm calculates always the same number of likelihoods $p(\mathbf{z}_k | \mathbf{x}_k^i)$ and initializes always the same number of particle $N_{p,b}$. Therefore, its computational cost is constant at each iteration.

2.7 Simulations and results

In this section, we propose to illustrate the performance of the different TBD algorithms proposed in this chapter via Monte Carlo simulation. As we have seen, the TBD particle filters depend on many parameters. For the sake of clarity, we will focus here on the key points of the TBD particle filters and in particular on the different instrumental densities proposed in section 2.5. For each of them, we will study the impact on the filter performance and the eventual gain compared to the instrumental densities proposed in the literature. Moreover, as computational time may sensibly vary between the different instrumental densities for a given number of particles, we will try to provide as much as possible as a fairly evaluation of the possible gain in terms of performance with the eventual additional computational time required to reach it.

2.7.1 Scenarios

For the simulation scenarios, we consider two scenarios with a number of iterations $N_{it} = 100$. The first scenario considers that the target is absent during all the experiment: this will allow to evaluate the probability that the filter declares a detection whereas no target is present (*i.e.* false alarm). The second scenario considers a target appearing at step $k_b = 15$ and disappearing at step $k_d = 75$ in order to measure both the ability of the different filters to truly detect the target and the accuracy of the corresponding estimator. For each Monte Carlo run, the initialization of the target state for the position and the velocity is done according to the birth density $p_b(\cdot)$ defined in section 2.2 (*i.e.* uniform prior over $\mathcal{D} = [r_{min}, r_{max}] \times [\theta_{min}, \theta_{max}]$ for the position and over $[v_{min}, v_{max}] \times [0, 2\pi]$ for the velocity), with the following parameters:

- $r_{min} = 30$ km, $r_{max} = 36$ km, $\theta_{min} = 35^\circ$ and $\theta_{max} = 55^\circ$,
- $v_{min} = 100$ m.s⁻¹ and $v_{max} = 300$ m.s⁻¹.

Algorithm 2.3 Marginalized TBD Particle Filter

Require: Particle cloud $\{(s_{k-1}^i, \mathbf{x}_{k-1}^i), w_{k-1}^i\}_{i=1}^{N_{p,c}}$ and probability $\hat{P}_{e,k-1}$ at step $k-1$.

- 1: Reset $C_k \leftarrow 0$,
 - 2: **for** $i = 1$ to $N_{p,c}$ **do**
 - 3: Draw $\mathbf{x}_{k,c}^i \sim q_c(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$
 - 4: Calculate the unnormalized weights $\tilde{w}_{k,c}^i$ with Eq. (2.85)
 - 5: $C_k \leftarrow C_k + w_{k-1}^i \frac{p_c(\mathbf{x}_{k,c}^i | \mathbf{x}_{k-1}^i)}{q_c(\mathbf{x}_{k,c}^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)}$
 - 6: **end for**
 - 7: **for** $i = 1$ to $N_{p,b}$ **do**
 - 8: Draw $\mathbf{x}_{k,b}^i \sim q_b(\mathbf{x}_k | \mathbf{z}_k)$
 - 9: Calculate the unnormalized weights $\tilde{w}_{k,b}^i$ with Eq. (2.86)
 - 10: **end for**
 - 11: Calculate the unnormalized probabilities $\hat{p}_u(s_{k-1} = 1)$ and $\hat{p}_u(s_{k-1} = 0)$ with Eq. (2.88)
 - 12: Calculate the probabilities $\hat{p}(s_{k-1} = 1 | s_k = 1, \mathbf{z}_{1:k})$ and $\hat{p}(s_{k-1} = 0 | s_k = 1, \mathbf{z}_{1:k})$ with Eq. (2.90)
 - 13: Calculate the probability of existence $\hat{P}_{e,k}$ with Eq. (2.93)
 - 14: Normalize weights of the continuing particles: $w_{k,c}^i \leftarrow \frac{\tilde{w}_{k,c}^i}{\sum_{l=1}^{N_{p,c}} \tilde{w}_{k,c}^l}$, $i = 1, \dots, N_{p,c}$
 - 15: Normalize weights of the birth particles: $w_{k,b}^i \leftarrow \frac{\tilde{w}_{k,b}^i}{\sum_{l=1}^{N_{p,b}} \tilde{w}_{k,b}^l}$, $i = 1, \dots, N_{p,b}$
 - 16: Mix the birth and continuing particles to create a set of $N_{p,c} + N_{p,b}$ particles $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_{p,c}+N_{p,b}}$:
 - 17: **for** $i = 1$ to $N_{p,c} + N_{p,b}$ **do**
 - 18: **if** $i \leq N_{p,c}$ **then**
 - 19: $\mathbf{x}_k^i \leftarrow \mathbf{x}_{k,c}^i$
 - 20: $w_k^i \leftarrow \hat{p}(s_{k-1} = 1 | s_k = 1, \mathbf{z}_{1:k}) w_{k,c}^i$
 - 21: **else**
 - 22: $\mathbf{x}_k^i \leftarrow \mathbf{x}_{b,k}^{(i-N_{p,c})}$
 - 23: $w_k^i \leftarrow \hat{p}(s_{k-1} = 0 | s_k = 1, \mathbf{z}_{1:k}) w_{k,b}^{(i-N_{p,c})}$
 - 24: **end if**
 - 25: **end for**
 - 26: Resample $N_{p,c}$ particles from $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_{p,c}+N_{p,b}}$
 - 27: Reset weights: $w_k^i \leftarrow \frac{1}{N_{p,c}}$ $i = 1, \dots, N_{p,c}$
 - 28: **return** $\left\{ \mathbf{x}_k^i, \frac{1}{N_{p,c}} \right\}_{i=1}^{N_{p,c}}$.
-

Note that a small radar window has been taken in order to limit the computational time. Indeed, the number of particles required is directly proportional to the overall number of radar cells. Between the iterations $k_b + 1$ and $k_d - 1$, the target state \mathbf{x}_k (for the position and the velocity) evolves according to Eq. (2.6) with no noise process (*i.e.* uniform linear motion) and $T_S = 0.3$ s (the time between two consecutive measurements).

The generation of the raw radar data is done according to Eq. (2.8) with $\mathbf{\Gamma} = \mathbf{I}_{N_c}$ (*i.e.* noise samples are assumed independent with noise $2\sigma^2 = 1$). The function $\mathbf{h}(\cdot)$, defined

in paragraph 2.3, is used with the following parameters:

- For the range axis, a chirp signal is considered with $B = 1$ MHz, corresponding to a range resolution $\Delta_r = 150$ m and $T_p = 66.7 \mu\text{s}$.
- For the azimuth axis, an antenna array is considered, composed of $N_a = 70$ antennas linearly spaced by $d = \lambda/2$, corresponding to a resolution (that does not depend on the value of λ) in azimuth $\Delta_\theta = 1.45^\circ$. Note that the maximum of the ambiguity function in azimuth arises normally for such an array for the direction $\pi/2$ whereas here the interval $[\theta_{min}, \theta_{max}]$ is centered around $\pi/4$. Therefore, in order to set the maximum at $\pi/4$, quantities θ_k and θ_v are just shifted from $\pi/4$ in Eq. (2.10).

Finally, different SNR values (following the SNR definition provided in paragraph 2.3.2) will be considered in the simulations.

2.7.2 Methodology for the performance evaluation

All the proposed particle filters provide information about the target presence or absence via the probability of presence $P_{e,k}$ but do not take any decision about it. However, the ability of the particle filter to provide useful information to take such a decision is interesting to evaluate. We propose here to evaluate the performance in two steps:

- First in terms of detection, *i.e.* measuring the ability of the filter to effectively detect the target.
- Second in terms of estimation in order to evaluate the accuracy of the estimator when the TBD particle filter has converged on the true target state.

2.7.2.1 Detection procedure

In order to perform the detection stage, let us call $d_{k,i}^T$ the decision variable at each iteration k of the i -th Monte Carlo run, that takes value 1 if a target is declared present by the filter, and 0 otherwise. A simple procedure to set the variable $d_{k,i}^T$ consists in comparing the probability of presence $P_{e,k}$ with a given probability P_T [RAG04], leading to

$$d_{k,i}^T = \begin{cases} 1, & \text{if } \hat{P}_{e,k} > P_T, \\ 0, & \text{otherwise.} \end{cases} \quad (2.94)$$

In practice, especially when the target SNR is low, the variable $\hat{P}_{e,k}$ can present large fluctuations leading to a situation where most of the particles may be located near the actual target position whereas $\hat{P}_{e,k}$ decreases below the threshold and as a consequence no detection is declared (*i.e.* $d_{k,i}^T = 0$).

To avoid such a situation, we propose a detection scheme that is based on an adaptive threshold that depends on the previous detection $d_{k-1,i}^T$ (we call this procedure the adaptive TBD target detection):

$$d_{k,i}^{T_{ad}} = \begin{cases} 1, & \text{if } \hat{P}_{e,k} > P_T \left(d_{k-1,i}^T \right), \\ 0, & \text{otherwise.} \end{cases} \quad (2.95)$$

In practice, $P_T(d_{k-1,i}^{T_{ad}} = 0)$ is chosen relatively high (*e.g.* 0.9) as it corresponds to the case where the target has not been detected yet. Choosing a high threshold ensures that the filter has converged on a true target with a good probability. On the contrary, when the filter has already detected a track (*i.e.* $d_{k-1,i}^{T_{ad}} = 1$), the probability threshold can be taken lower (*e.g.* 0.2) in order to deal with the possible fluctuations of the estimated probability of presence.

2.7.2.2 Evaluation of the detection performance

We propose first to evaluate the detection performance by averaging the probability of presence $P_{e,k}$ at each iteration over N_{MC} runs. This allows to evaluate the behavior of the different filters without using a particular detection scheme. Note that averaging the detection variable $d_{k,i}^{T_{ad}}$ provides performance with very similar behavior as the probability of presence. Thus, in order to avoid unnecessary redundant results, we do not present them here.

We also propose to measure the detection performance by providing the percentage of time t_D in which the target has been detected during time step k_b and $k_d - 1$. A first solution would be to compute the average of the variable $d_{k,i}^{T_{ad}}$ from k_b to $k_d - 1$. However, as it was stressed at the beginning of the section, such a method does not take into account the possible divergence between the estimate state $\mathbf{x}_{k|k}$ and the actual state \mathbf{x}_k ; if the variable $d_{k,i}^{T_{ad}} = 1$ whereas the estimator $\hat{\mathbf{x}}_{k|k}$ is far away from the actual state, it does not seem reasonable to count it as a detection. Thus, we define, for the i th Monte Carlo run, an indicator of good estimate (for $k \in \{k_b, \dots, k_d - 1\}$) by

$$e_{k,i} = \begin{cases} 1, & \hat{\mathbf{x}}_{k|k} \in \mathcal{V}_{\mathbf{x}_k}, \\ 0, & \text{otherwise.} \end{cases} \quad (2.96)$$

where the target is effectively considered as a detection (*i.e.* $e_{k,i} = 1$) if the estimated state is located in the subset $\mathcal{V}_{\mathbf{x}_k}$ defined in Eq. (2.26) with $\delta_{h_r} = \delta_{h_\theta} = 2$ (*i.e.* the estimated target state $\hat{\mathbf{x}}_{k|k}$ is located in a vicinity of two range azimuth cells from the actual target state \mathbf{x}_k). Finally, t_D is simply obtained by

$$t_D = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \frac{1}{k_d - k_b} \sum_{k=k_b}^{k_d-1} d_{k,i}^{T_{ad}} e_{k,i}. \quad (2.97)$$

In the same manner, we define the average time of bad-detection t_{bD} , *i.e.* when the filter declares a detection but the estimate is not relevant, by

$$t_{bD} = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \frac{1}{k_d - k_b} \sum_{k=k_b}^{k_d-1} d_{k,i}^{T_{ad}} (1 - e_{k,i}). \quad (2.98)$$

Concerning the false alarm probability of the particle filter P_{fa}^{PF} , it is computed by making the average of variable d_k for the scenario where the target is assumed absent, that is to say

$$P_{fa}^{PF} = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \frac{1}{N_{it}} \sum_{k=1}^{N_{it}} d_{k,i}^{T_{ad}}. \quad (2.99)$$

2.7.2.3 Evaluation of the estimation performance

For the evaluation of estimation performance, the RMSE (Root Mean Square Error) in position $\text{RMSE}_{k,pos}$ and in velocity $\text{RMSE}_{k,vel}$ are computed between step k_b and $k_d - 1$ from the following formulas

$$\text{RMSE}_{k,pos} = \sqrt{\frac{1}{\sum_{i=1}^{N_{MC}} d_{k,i}^{T_{ad}} e_{k,i}} \sum_{i=1}^{N_{MC}} d_{k,i}^{T_{ad}} e_{k,i} \left[\frac{(\hat{x}_{k|k} - x_k)^2 + (\hat{y}_{k|k} - y_k)^2}{2} \right]}, \quad (2.100)$$

$$\text{RMSE}_{k,vel} = \sqrt{\frac{1}{\sum_{i=1}^{N_{MC}} d_{k,i}^{T_{ad}} e_{k,i}} \sum_{i=1}^{N_{MC}} d_{k,i}^{T_{ad}} e_{k,i} \left[\frac{(\hat{\dot{x}}_{k|k} - \dot{x}_k)^2 + (\hat{\dot{y}}_{k|k} - \dot{y}_k)^2}{2} \right]} \quad (2.101)$$

Note that here the RMSE represents an error over a single component (*e.g.* x_k or y_k), hence the presence of the factor 1/2 in Eq. (2.100) and (2.101) in order to make the average over the two components. The choice of this definition is arbitrary and, of course, other definitions are possible. The most important is to be coherent with the definition, in particular when the RMSE is compared to any theoretical bound (*e.g.* radar resolution, Cramer-Rao bound, *etc.*). However, in the sequel, the RMSE of the different filters are compared relatively with each other, therefore the factor 1/2 does not impact the conclusions that can be made from the simulation results.

2.7.3 Influence of the instrumental density

We propose in this section to measure the impact of the different instrumental densities proposed in section 2.5 for the initialization of the particle state. The classic Track-Before-Detect particle filter described in section 2.4 is considered with the following parameters: $N_p = 1500$, $\beta = 1$, $P_b = P_d = 0.1$, $q_S = 0.01$, $v_{min} = 100 \text{ m.s}^{-1}$, $v_{max} = 300 \text{ m.s}^{-1}$, $\text{SNR}_{min} = 3 \text{ dB}$, $\text{SNR}_{max} = 13 \text{ dB}$ and $\delta_{h_r} = \delta_{h_\theta} = 2$ (for the truncation of the ambiguity function). Then, for each component of the state vector (*i.e.* position, velocity, amplitude, presence), we compare the performance in detection and estimation for the different instrumental densities outlined in section 2.5, for the initialization case, while assuming that the other parameters are initialized according to the prior density. As already stated, the prior $p_c(\mathbf{x}_k | \mathbf{x}_{k-1})$ is chosen to sample the continuing case.

Moreover, it is also important to compare the performance of the different instrumental densities with respect to the computational time required to reach such performance. To this purpose, the averaged Monte Carlo run duration is calculated over the N_{MC} simulations for all the instrumental densities, and normalized by the fastest one. Note that this quantity should be subject to cautious interpretation since it clearly depends on the scenario considered. Indeed, the instrumental densities for the initialization are principally used when the filter has not converged yet to a target and many particles must then be initialized. On the contrary, when the filter has converged to a target, most of the particles are in tracking stage and the initialization densities only concern a few particles. Therefore, the duration of the MC run will partly depend on the proportion of time when the target is present. However, it still gives a good idea of the impact of the instrumental density on the averaged MC run duration.

2.7.3.1 Influence of the Instrumental density for the position

The position (x_k, y_k) is probably the most important parameter to carefully initialize and the performance is evaluated for the following instrumental densities derived from section 2.5.1:

1. The prior case where the particle position is simply drawn from the prior; it is labelled as "Prior".
2. A second case, where the particle position is initialized uniformly over the cells exceeding the threshold γ corresponding to a probability of false alarm of 0.1. Note that it corresponds, as we stated in paragraph 2.5.1.3, to choose $P_{\mathcal{D}_{k,\gamma}} = 1$ for the instrumental density $q^{\mathcal{U}}(\cdot)$ defined in Eq. (2.41). This instrumental density is labelled as "Threshold".
3. A third case where the particle position is sampled according to $q^{\mathcal{U}}(\cdot)$ with $P_{fa} = 0.1$ while $P_{\mathcal{D}_{k,\gamma}} = 0.79$ has been calculated from Eq. (2.46) with $\delta_r = 2$, $\delta_\theta = 3$ and $N_\rho = 5$. This density is labelled as "Mix \mathcal{U} ".
4. And lastly, the optimal mixture importance density $q(\cdot | \mathbf{z}_k)$ specified in Eq. (2.39) with the following parameters: $P_{fa} = 0.1$, $\delta_r = 2$, $\delta_\theta = 3$, $N_\rho = 5$ and $P_{\mathcal{D}_{k,\gamma}} = 0.79$. This density is labelled as "Mix Opt".

In Figure 2.7 the averaged probability of presence is shown for target SNR of 7 dB, while detection performance is presented in Table 2.2. Clearly, the density "Mix Opt" outperforms the other instrumental densities in terms of detection although it induces a slight increase of the probability of presence when the target is absent. Nevertheless, the increase in terms of probability of false alarm is not significant, as demonstrated in Table 2.2.

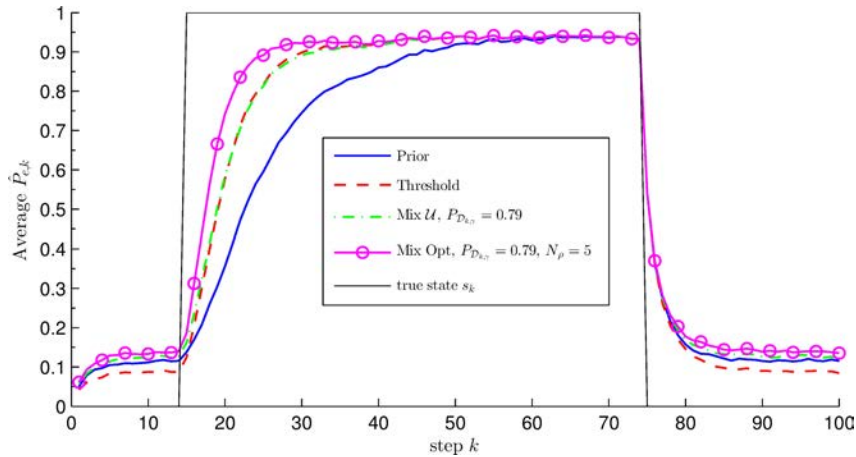


Figure 2.7 – Averaged probability of presence $P_{e,k}$ for different instrumental densities in position. SNR = 7dB, $N_p = 1500$ and $P_{fa} = 0.1$.

The performance reached in terms of RMSE in position and velocity is shown in Figure 2.8. Again, the instrumental density "Mix Opt" provides better performance than the other instrumental densities during convergence. When the filter has converged, all the

	Prior	Threshold	Mix \mathcal{U}	Mix Opt
P_{fa}^{PF}	3.67×10^{-3}	2.69×10^{-3}	4.35×10^{-3}	4.23×10^{-3}
t_D	70.6%	85.4%	84.2%	90%
t_{bD}	0.41%	0.26%	0.36%	0.34%
relative MC run duration	1	1.06	1.09	2.61

Table 2.2 – Detection performance and relative averaged MC run duration for different instrumental densities in position. SNR = 7dB, with $N_p = 1500$ and $P_{fa} = 0.1$

instrumental densities provide similar results. This demonstrates the requirement to use a relevant instrumental densities to ensure a faster convergence of the filter on the target.

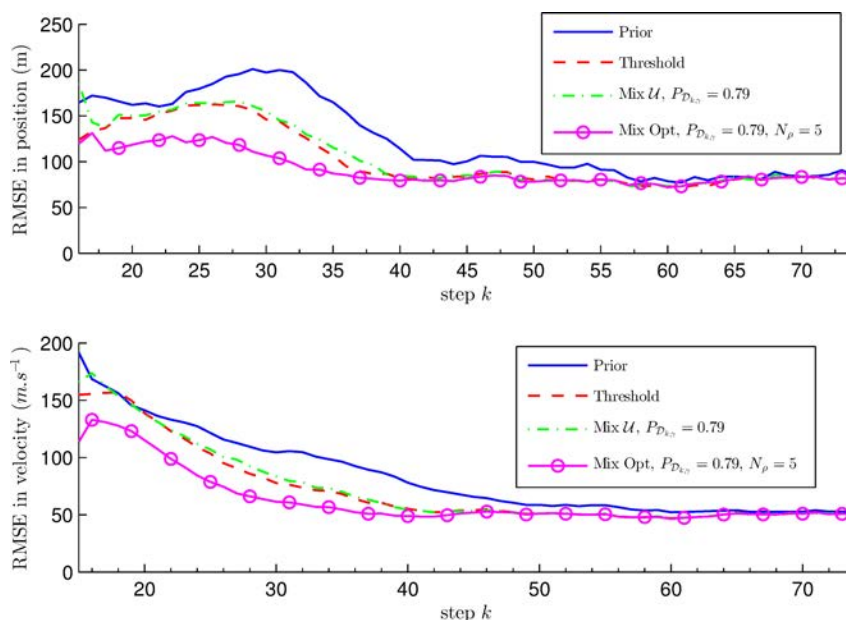


Figure 2.8 – Performance in estimation for different instrumental densities in position. Top: RMSE in position. Bottom: RMSE in velocity. SNR = 7dB, $N_p = 1500$ and $P_{fa} = 0.1$.

Lastly, the relative averaged MC run durations for the different instrumental densities are presented in the last row of Table 2.2. On the contrary, the cost induced by the "Threshold" and "Mix \mathcal{U} " instrumental densities, is relatively small compared to the gain in performance. Note that this conclusion should be moderated, as will be shown in section 2.7.4.

2.7.3.2 Influence of the Instrumental density for the amplitude parameter

In this paragraph, the influence of the instrumental density for the initialization of the amplitude parameter is evaluated. The following instrumental densities are considered:

1. The prior case where the particle amplitude is simply drawn from the prior. It is labelled as "Prior". Moreover, we consider two different intervals $[\rho_{min}, \rho_{max}]$, a

first one where the parameter interval for parameter ρ corresponds to an interval $\text{SNR} = [3, 20]$ (in dB) and a second one that corresponds to an interval $\text{SNR} = [3, 13]$ (in dB).

2. A second case where the amplitude is drawn according to the density based on the MAP estimator and provided by Eq. (2.56). It is labelled as "MAP Init". Again we consider two different intervals $[\rho_{min}, \rho_{max}]$ with the same values as previously.
3. Lastly, the approximation of the Optimal instrumental density defined by Eq. (2.53), with $N_\rho = 10$ and $\text{SNR} = [3, 13]$. It is labelled as "Discrete Init"

In Figure 2.9 the averaged probability of presence is shown for a target SNR of 7 dB. Note first that the choice of the prior values SNR_{min} and SNR_{max} dramatically impacts

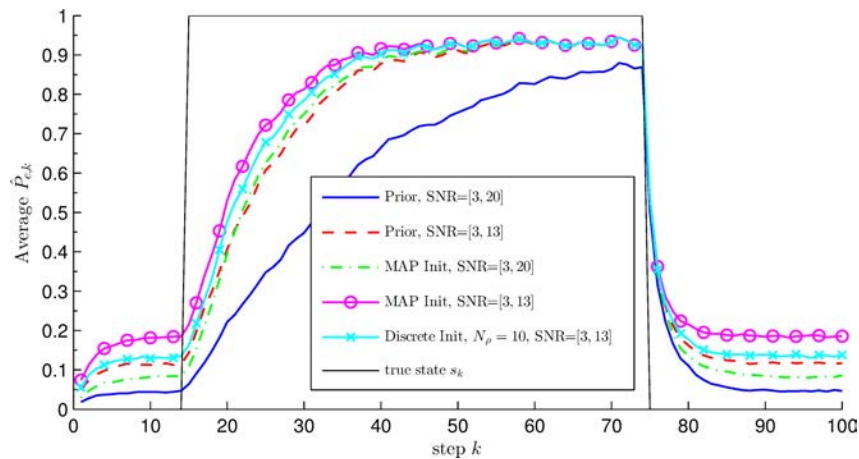


Figure 2.9 – Averaged probability of presence $P_{e,k}$ for different instrumental densities in amplitude. $\text{SNR} = 7\text{dB}$, $N_p = 1500$.

the performance of the "Prior" instrumental density, as well as the proposed instrumental densities to a lesser extent. Globally, it seems that the gain in probability of presence when the target is indeed present is obtained at the cost of an increase of the same probability when the target is absent. This in turns can be observed on the false alarm probabilities provided in Table 2.3.

Concerning the performance in terms of RMSE in position and velocity, it turns out to be very similar in all cases. Thus, we do not provide a figure here.

Lastly, the relative averaged MC run durations for the different instrumental densities are presented in the last row of Table 2.3. The extra computational time required for "MAP Init" instrumental densities compare to the "Prior" densities is relatively small with respect to the gain induced in terms of detection. However, this is not the case for the "Discrete Init" where this gain appears small compared to the extra time required.

2.7.3.3 Influence of the Instrumental density for the velocity variable

Two instrumental densities have been proposed in paragraph 2.5.3 in order to sample the velocity:

	Prior	Prior	MAP Init	MAP Init	Discrete Init
SNR	[3, 20]	[3, 13]	[3, 20]	[3, 13]	[3, 13]
P_{fa}^{PF}	1.45×10^{-3}	3.78×10^{-3}	1.61×10^{-3}	6.08×10^{-3}	4.04×10^{-3}
t_D	64.1%	81.3%	82%	85.4%	84.3%
t_{bD}	0.16%	0.28%	0.19%	0.43%	0.34%
relative MC run duration	1	1.08	1.12	1.16	1.30

Table 2.3 – Detection performance and relative averaged MC run duration for different instrumental densities in amplitude. SNR = 7dB and $N_p = 1500$.

1. The first density that uniformly samples the velocity components of a newborn particle. It is labelled as "Prior".
2. And the second density that samples the velocity component uniformly at the next step after the birth event. It is labelled as "Next step".

Results are shown in Figures 2.10 and 2.11. The density "Next step" provides a small

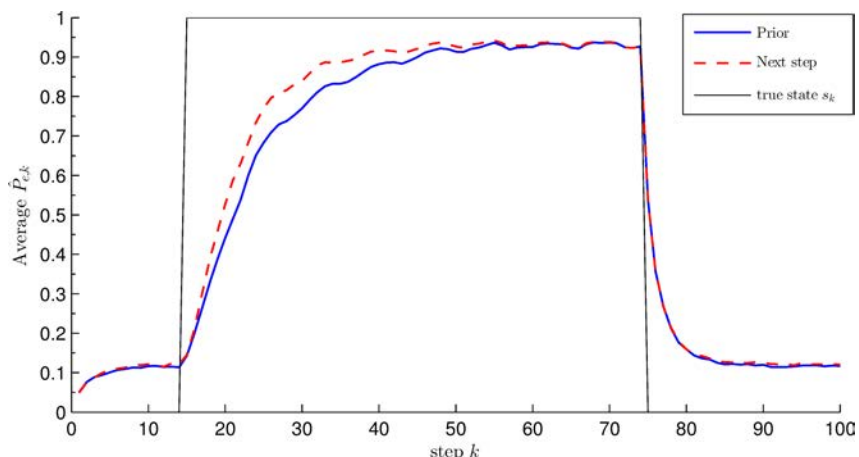


Figure 2.10 – Averaged probability of presence $P_{e,k}$ for different instrumental densities in velocity. SNR = 7dB, $N_p = 1500$.

improvement compared to the density "Prior" both in terms of averaged probability of presence and in estimation.

2.7.3.4 Influence of the Instrumental density for the presence variable

In this paragraph, the performance for three filters that use different strategies to sample the variable s_k are evaluated:

1. The first one, denoted by "Prior", that corresponds to the classic TBD particle filter defined by Algorithm 2.1 where the variable s_k is sampled according to the transition probability matrix defined in Eq. (2.5).

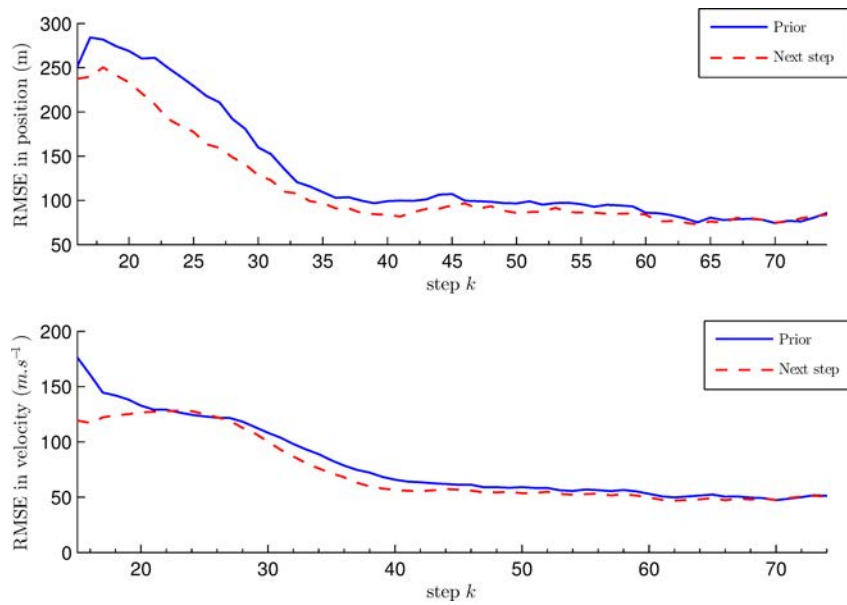


Figure 2.11 – Performance in estimation for different instrumental densities in velocity. SNR = 7dB, $N_p = 1500$.

2. The second one, labelled as " s_k a posteriori" is defined by Algorithm 2.2 where the variable s_k is drawn according to the *a posteriori* transition probabilities defined in Eq. (2.70) and (2.71).
3. The last one, denoted by " s_k marginalized" is detailed in Algorithm 2.3 which considers only particles with the state $s_k = 1$. For this particular TBD filter, the parameter $N_{p,c}$ is set to 1000 particles.

In Figure 2.12 the averaged probability of presence is shown for a target SNR of 5 dB – Note that here a smaller SNR has been taken in order to highlight the importance of the choice of the sampling strategy for the variable s_k . Clearly, the filters " s_k a posteriori"

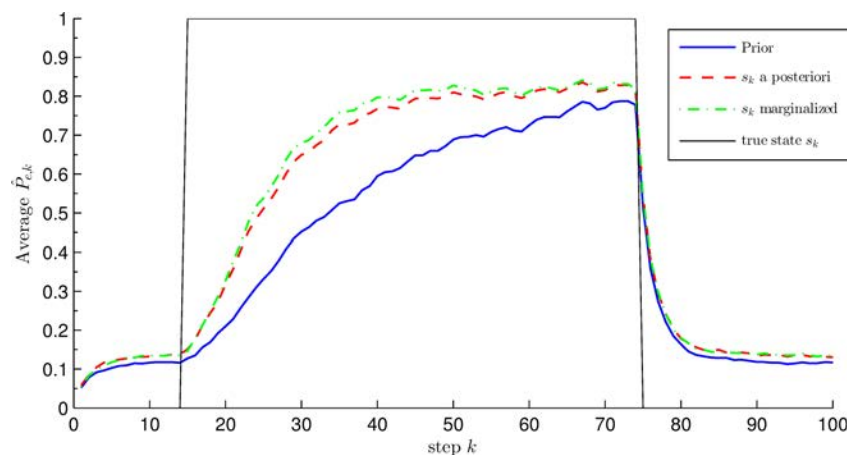


Figure 2.12 – Averaged probability of presence $P_{e,k}$ for different sampling strategies of the variable s_k . SNR = 5dB, $N_p = 1500$.

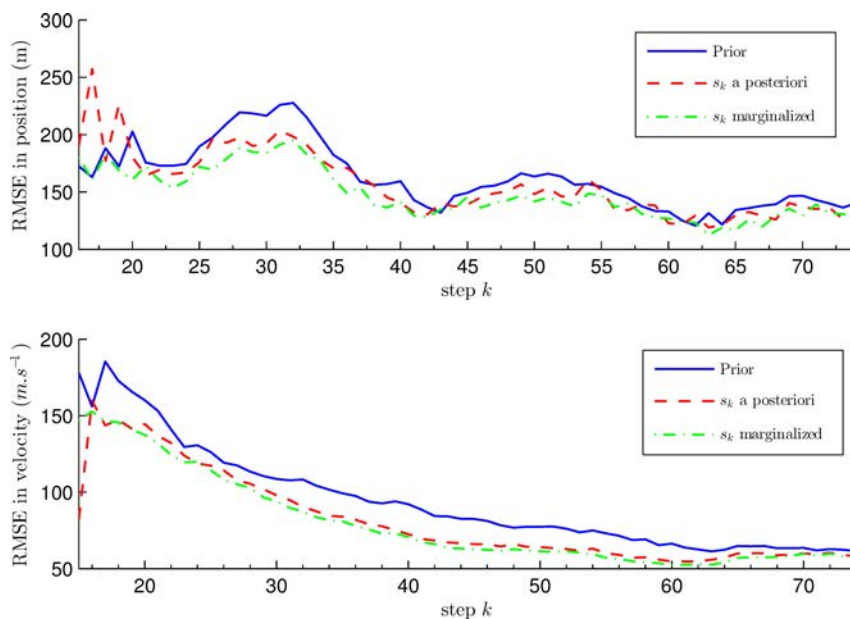


Figure 2.13 – Performance in estimation for different sampling strategies of the variable s_k . SNR = 5dB, $N_p = 1500$.

	Prior	s_k a posteriori	s_k marginalized
P_{fa}^{PF}	3.9×10^{-3}	3.53×10^{-3}	3.43×10^{-3}
t_D	56.6%	73%	75.1%
t_{bD}	1.22%	1.67%	1.62%
relative MC run duration	1	1.48	1.34

Table 2.4 – Detection performance and relative averaged MC run duration for different sampling strategies of the variable s_k . $N_p = 1500$ for a target SNR of 5dB

and " s_k marginalized" provide much better performance than the classic particle filter with a small advantage to the " s_k marginalized" filter over the " s_k a posteriori" filter. Moreover, as stated in Table 2.4, the use of the two proposed densities slightly decreases the probability of false alarm.

The performance in terms of RMSE in position and velocity is shown in Figure 2.13. Whereas there is a gain by using the filters " s_k a posteriori" and " s_k marginalized", the latter is not as important as for the detection.

Lastly, the relative averaged MC run durations for the different sampling strategies of the variable s_k are presented in the last row of Table 2.4. Obviously, the strategies " s_k a posteriori" and " s_k marginalized" are more costly. However as it was said in paragraph 2.5.4 and in section 2.6, the two methods calculated the same number of likelihood at each iteration (which is the most demanding part of the algorithm) and therefore better utilize the computer resources than the prior instrumental density.

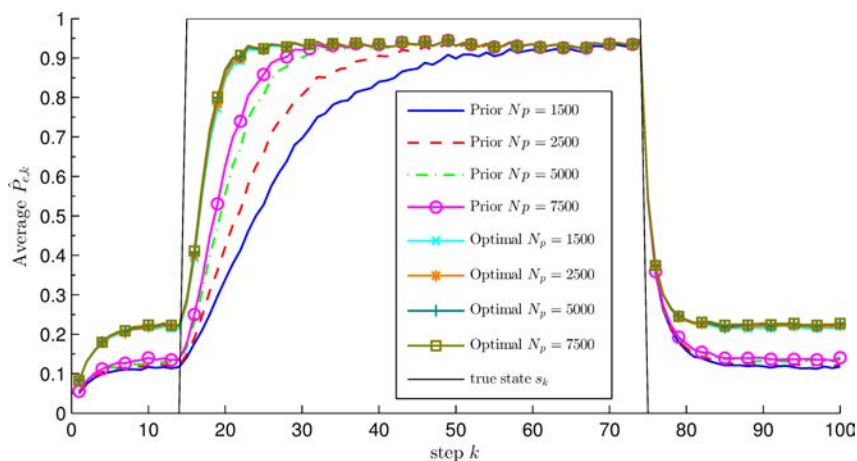


Figure 2.14 – Comparison of the Averaged probability of presence $P_{e,k}$ between the prior instrumental density and the optimal one for several number of particles N_p . SNR = 7dB.

2.7.4 Choice of the instrumental density

The aim of this paragraph is to demonstrate the benefit of using a suitable instrumental density to initialize the particles compared to a one using the prior with a higher number of particles. Performance are compared for the two following filters:

1. The first one uses the classic TBD particle filter defined by Algorithm 2.1 with the filter parameters defined at the beginning of section 2.7.3; it initializes all the state parameters with their prior densities. For this filter, the following number of particles N_p are considered: 1500, 2500, 5000, 7500. This filter is denoted as "Prior".
2. For the second filter, each parameter of the state vector is sampled using the instrumental density providing the best performance in detection. Therefore, for the presence parameter s_k , the marginalized TBD particle filter is chosen. For the position, the "Mix Opt" instrumental is taken with the same parameters as in paragraph 2.7.3.1. For the amplitude parameter, the instrumental density "MAP Init" is chosen with SNR = [3, 13]. The velocity is initialized at the next step after the birth of the particle. Finally the following number of particles N_p are considered: $N_p = 1500$ and $N_{p,c} = 1000$, $N_p = 2500$ and $N_{p,c} = 1500$, $N_p = 5000$ and $N_{p,c} = 3000$, $N_p = 7500$ and $N_{p,c} = 5000$.

Note that we choose an interval of SNR = [3, 13] for the Amplitude parameter whereas the performance is better with SNR = [3, 20] for the instrumental density "MAP Init". We made this choice in order to not penalize the prior density from which the results are not good with an interval of SNR = [3, 20] and thus make the simulation as fair as possible. Results are provided in Figures 2.14 and 2.15, and in Table 2.5. For any number of particles, the Optimal instrumental density outperforms the prior instrumental density both in terms of target detection and estimation ; at the price of a slight increase of the probability of false alarm. Moreover, it is interesting to notice that the Optimal instrumental density is less sensitive to the number of particles than the prior. Indeed, the performance for the Optimal instrumental density for $N_p = 1500$, $N_p = 5000$ and

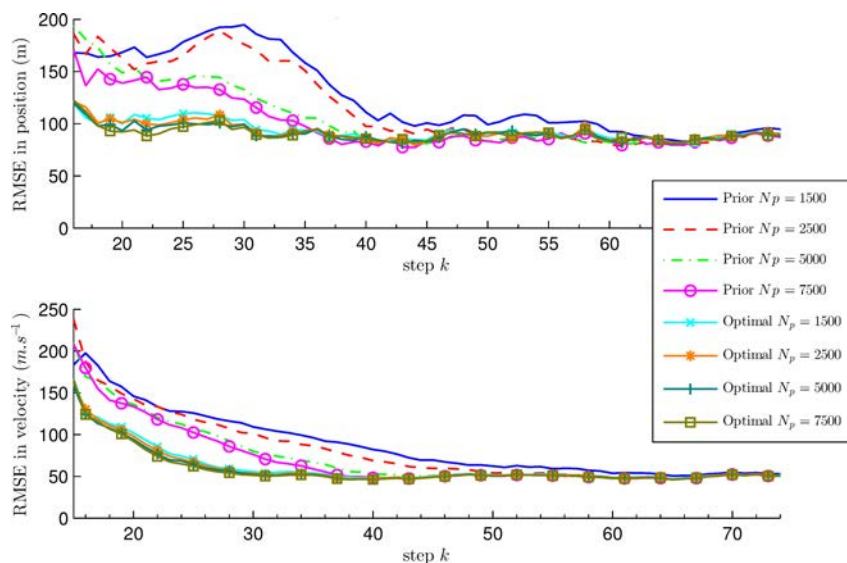


Figure 2.15 – Comparison of the Performance in estimation between the prior instrumental density and the optimal one for several number of particles N_p . SNR = 7dB.

	Prior	Prior	Prior	Prior	Optimal	Optimal	Optimal
N_p	1500	2500	5000	7500	1500	5000	7500
$P_{fa}^{PF} (\times 10^{-3})$	3.95	3.42	3.05	3.42	6.99	6.55	6.85
t_D	76.3%	82.5%	87.5%	89.1%	92.9%	93.2%	93.3%
t_{bD}	0.34%	0.28%	0.21%	0.22%	0.25%	0.27%	0.25%
relative MC run dur.	1	1.45	2.55	4.7	2.98	4.77	6.77

Table 2.5 – Detection performance and relative averaged MC run duration between the prior instrumental density and the optimal one for several number of particles N_p . SNR = 7dB.

$N_p = 7500$ are quite similar while it is sensibly different for the Prior density. Furthermore, the relative averaged MC run durations for the different filter are presented in Table 2.5. It demonstrates that using the Optimal density with a small number of particle is more efficient both in terms of performance and in terms of computational time than using the prior instrumental density with a higher number of particles.

2.7.5 Influence of the target SNR

Lastly, as Track-Before-Detect methods are expected to track low target SNR, it is important to evaluate the performance according to the target SNR. Thus, the Optimal TBD particle filter defined in the previous paragraph with $N_p = 7500$ is applied for different target SNR: 10dB, 7dB, 5dB and 3dB – Note that here we choose an important number of particles (*i.e.* $N_p = 7500$) since one of our objective is to see if TBD particle filter are able to track very low target SNR. Results are provided in Figures 2.16 and 2.17, and in Table 2.6. Clearly, performance highly depends on the target SNR and it seems

difficult to jointly detect and track a target with an SNR below 5dB. Note that, this conclusion does not mean that the TBD particle filter is not able to detect target with very low SNR but rather that it cannot track it accurately. It should also be recalled here that the SNR values indicated do not take into account losses due to the position of the target in the cell: a target with indicated SNR of 5dB may in practice provide here a peak at the output of the range/azimuth matched filter less than 3dB ! In that respect, the probability of presence remains impressively high at low SNR.

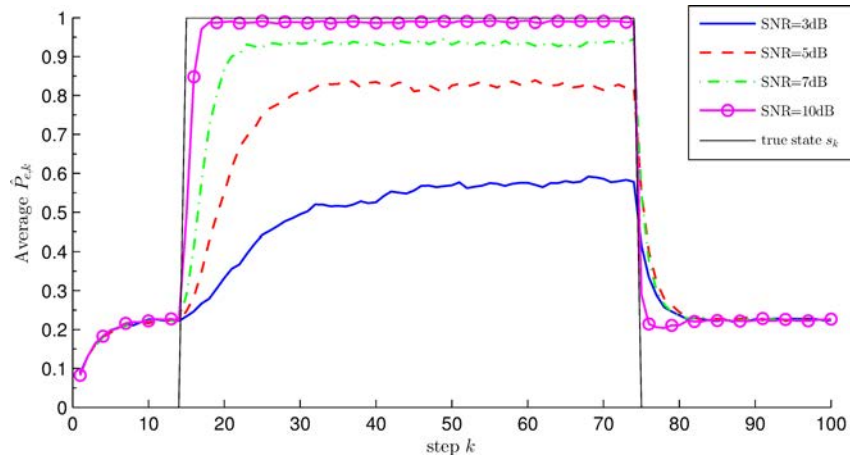


Figure 2.16 – Comparison of the Averaged probability of presence $P_{e,k}$ for the Optimal TBD particle filter with different target SNR. $N_p = 7500$.

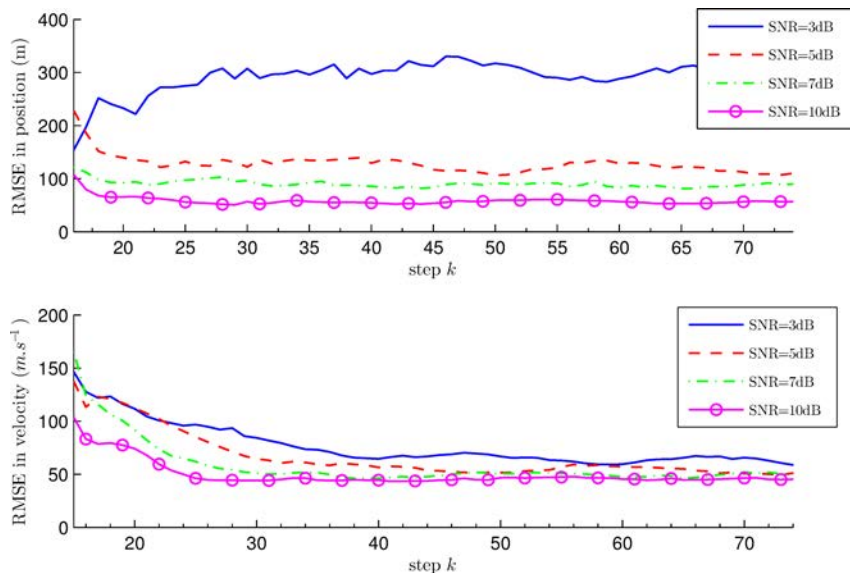


Figure 2.17 – Comparison of the Performance in estimation for the Optimal TBD particle filter with different target SNR. $N_p = 7500$.

target SNR	10	7	5	3
t_D	98%	93.2%	81.9%	36.5%
t_{bD}	0.02%	0.68%	3.22%	13.4%

Table 2.6 – Detection performance for different target SNR with the Optimal TBD particle filter.

2.8 Conclusion

In this chapter, the classic solution for the monotarget Track-Before-Detect problem has been presented. This solution consists in considering an hybrid hidden state (s_k, \mathbf{x}_k) , where \mathbf{x}_k is the classic target state while s_k is a binary variable modelling the absence or the presence of the target, in order to jointly detect and track a single target. For this state-space model, the classic TBD particle filter approximation has been detailed in section 2.4. Two different cases are sampled by this particle filter: one case concerns continuing particles, already alive at previous time step that are propagated according to the target dynamical model; while the other case corresponds to the newborn particles that must be initialized in the target space. We have shown in this chapter that the instrumental density for this latter case must be carefully chosen. Thus, in section 2.5, for each parameter of the state vector, several instrumental densities, which take into account the information of the measurement \mathbf{z}_k in order to initialize the different parameters, have been proposed. In particular, for the position and amplitude, the optimal instrumental densities have been derived and several approximations provided. Concerning the presence parameter s_k , we have shown that it can be sampled according to the posterior probabilities rather than the prior ones. Another solution that uses particles in a more efficient way by only considering particles with the state $s_k = 1$ has been described in section 2.6.

Finally, in section 2.7, Monte Carlo simulations have been used to provide performance in detection and estimation for the different instrumental densities and the different particle filters presented in this chapter. These simulations have allowed to illustrate the importance of using relevant instrumental densities, in particular for the position parameters where it dramatically increases the performance, both in detection and in estimation. Moreover, simulation results also highlighted the importance of carefully dealing with the presence parameter s_k since the Marginalized TBD particle filter and the one using the posterior probabilities to sample s_k outperform the classic TBD particle filter.

Chapter 3

A novel approach for monotarget Track-Before-Detect

3.1 Introduction

In the previous chapter, the classic monotarget TBD particle filters were detailed. These classic methods manage both the detection of the target appearance in the radar window and of its disappearance by the addition of a variable s_k to the target state vector \mathbf{x}_k . This model raised some questions that merit to be discussed:

- Is it appropriate to try to detect the appearance and disappearance of the target in a single algorithm ?
- In particular for very low target SNR, does the initialization of newborn particles (at each iteration) in the whole state-space disturb the estimation of the target state when the particle filter has converged to the actual target state ?
- Lastly, and in the same manner, is it relevant to still initialize newborn particles, that is the most costly part of the TBD particle filter, while the particle filter has already converged to the actual target state ?

Therefore, the aim of this chapter is to try to provide some answers to these questions. To this purpose, we develop an alternative approach to the monotarget TBD problem that allows to manage separately the target appearance and disappearance. More precisely, we propose to model the TBD problem using the quickest change detection framework and then solve it using some particle filter solutions. The aim of quickest detection methods is to detect, as quickly as possible, some (possible) changes in the distribution of a random process while ensuring the smallest probability of error. The TBD problem can be seen as a quickest change detection problem. Indeed, when no target is present in the radar window, the measurement \mathbf{z}_k provided by the radar only contains noise, while after the possible target appearance the measurement consists of the target contribution plus noise. Kligys *et al.* in [KRT98] proposed a solution to the TBD problem in this context. However, their solution is not developed in the particle filter framework that is studied in this thesis. Thus, we propose, in this chapter, a new particle filter solution in this particular framework.

This chapter is organized as follows: in section 3.2 and 3.3, we define a state-model for the target appearance and another one for the target disappearance in the Bayesian change detection framework. Moreover, for each state-model, we propose several particle filters in order to approximate the corresponding Bayesian filter. Then, in section 3.4, we propose to combine the proposed solutions in order to detect both the target appearance and disappearance. Finally, in section 3.5, we evaluate the performance of the particle filters presented in this chapter compared to the classic TBD particle filters detailed in the previous chapter.

3.2 A Bayesian solution for time appearance detection in TBD

3.2.1 State model

The monotarget TBD problem can be seen as a quickest change detection problem [KRT98]. Indeed, let us assume that the target appears at an unknown time step τ_b , then until τ_b the measurement \mathbf{z}_k only consists of noise while after τ_b the measurement \mathbf{z}_k is constituted of the target contribution plus noise. The aim of the TBD application is therefore to detect this change.

In the classic Bayesian quickest change detection framework [TV05], the problem is solved by choosing a prior distribution on the time τ_b . In our TBD application, the prior model must be defined for the random process $(\tau_b, \mathbf{x}_k)_{k \in \mathbb{N}}$ and not only for the variable τ_b . This leads to specify the density $p(\tau_b, \mathbf{x}_{0:k})$ for any k . This density can be rewritten without loss of generality as

$$p(\tau_b, \mathbf{x}_{0:k}) = p(\tau_b) p(\mathbf{x}_{0:k} | \tau_b). \quad (3.1)$$

From this decomposition, this consists in defining a prior distribution for the time of arrival τ_b and for the evolution of the state \mathbf{x}_k knowing the variable τ_b .

3.2.1.1 Time appearance model

The time appearance τ_b is modeled as a geometric random variable, *i. e.*

$$p(\tau_b = i) = \begin{cases} 0, & i = 0, \\ P_b(1 - P_b)^{i-1}, & i \geq 1, \end{cases} \quad (3.2)$$

where $0 < P_b < 1$ denotes the probability of birth. The geometric prior is often encountered in the literature [TV05] because it has interesting properties. In particular, by defining

$$b_k = \begin{cases} 1, & \text{if } \tau_b \leq k, \\ 0, & \text{otherwise,} \end{cases} \quad (3.3)$$

it can be shown that $(b_k)_{k \in \mathbb{N}}$ is a Markov chain with the following transition probability matrix

$$\Pi_{b_k} = \begin{bmatrix} 1 - P_b & P_b \\ 0 & 1 \end{bmatrix}, \quad (3.4)$$

and also that $p(b_k = 1 \mid b_{k-1} = 0) = P_b$, *i.e.* knowing that the target has not yet appeared at step $k - 1$, its probability to show up at step k does not depend on the time instant and is equal to P_b . The proof of these two statements is provided in Appendix A.

3.2.1.2 Target state model

Let us now specify the density $p(\mathbf{x}_{0:k} \mid \tau_b)$. Assuming k is greater than τ_b , we can write

$$p(\mathbf{x}_{0:k} \mid \tau_b) = p(\mathbf{x}_{0:\tau_b-1} \mid \tau_b) p(\mathbf{x}_{\tau_b:k} \mid \tau_b, \mathbf{x}_{0:\tau_b-1}). \quad (3.5)$$

The interest of such a rewriting is to define the evolution of the process $(\mathbf{x}_k)_{k \in \mathbb{N}}$ before and after τ_b . Indeed, as in Chapter 2 where the state \mathbf{x}_k is meaningless (or has no physical meaning) when $s_k = 0$, here the state \mathbf{x}_k has no signification before τ_b and is not related to the measurement equation (3.9). Thus, any prior model can be chosen in this case.

On the contrary, after τ_b the state \mathbf{x}_k represents the state of an actual target and therefore a prior model must be specified in order to model the state evolution. Since it seems reasonable to assume that the evolution of the process after τ_b does not depend on the evolution of the process before τ_b , the density $p(\mathbf{x}_{0:k} \mid \tau_b)$ becomes

$$p(\mathbf{x}_{0:k} \mid \tau_b) = p(\mathbf{x}_{0:\tau_b-1} \mid \tau_b) p(\mathbf{x}_{\tau_b:k} \mid \tau_b). \quad (3.6)$$

Thus, with this independence hypothesis, defining the prior model after τ_b just consists in specifying the density $p(\mathbf{x}_{\tau_b:k} \mid \tau_b)$. In chapter 1, it has been stressed that the Bayesian filter can be derived for the Hidden Markov Model where the hidden process is assumed Markovian. Therefore, in order to adapt the Bayesian filter for our particular model, it is convenient to assume that conditionally to τ_b the evolution of process $(\mathbf{x}_k)_{k \in \mathbb{N}}$, for $k \geq \tau_b$ is Markovian¹, *i.e.*

$$p(\mathbf{x}_{\tau_b:k} \mid \tau_b) = p(\mathbf{x}_{\tau_b} \mid \tau_b) \prod_{i=\tau_b+1}^k p(\mathbf{x}_i \mid \tau_b, \mathbf{x}_{i-1}). \quad (3.7)$$

Then, from Eq. (3.7), the process $(\mathbf{x}_k)_{k \in \mathbb{N}}$ conditionally to τ_b and for $k \geq \tau_b$ is entirely defined by the density at step τ_b , *i.e.* $p(\mathbf{x}_{\tau_b} \mid \tau_b)$, which corresponds to the initialization of the process, and by the transition probabilities $p(\mathbf{x}_i \mid \tau_b, \mathbf{x}_{i-1})$.

By analogy with chapter 2, the density $p(\mathbf{x}_{\tau_b} \mid \tau_b)$ corresponds to the birth density $p_b(\mathbf{x}_k)$ while the transition probabilities $p(\mathbf{x}_i \mid \tau_b, \mathbf{x}_{i-1})$ correspond to the continuing density $p_c(\mathbf{x}_i \mid \mathbf{x}_{i-1})$. We can thus in a similar manner define

$$p(\mathbf{x}_i \mid \tau_b, \mathbf{x}_{i-1}) = \mathcal{N}(\mathbf{x}_i; \mathbf{F}\mathbf{x}_{i-1}, \mathbf{Q}), \quad (3.8)$$

where \mathbf{F} and \mathbf{Q} are the matrices defined in section 2.2.

¹Note that it does not mean that the entire process $(\mathbf{x}_k)_{k \in \mathbb{N}}$ is Markovian, even for $k \geq \tau_b$. In fact, the considered process is only Markovian conditionally to the variable τ_b (for $k \geq \tau_b$) but is generally not Markovian without this conditioning by τ_b .

3.2.2 Measurement model

The measurement model is the same as in section 2.3 with only some slight modification in order to take into account the specificity of the proposed state model. Following 2.3, the measurement equation becomes

$$\mathbf{z}_k = \begin{cases} \rho e^{j\varphi_k} \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k, & \text{if } k \geq \tau_b, \\ \mathbf{n}_k, & \text{otherwise,} \end{cases} \quad (3.9)$$

where $\mathbf{h}(\cdot)$ is the ambiguity function, \mathbf{n}_k is a zero-mean circular complex Gaussian vector with a known covariance matrix $\mathbf{\Gamma}$, φ_k is the random phase uniformly drawn over the interval $[0, 2\pi)$ and ρ is the constant modulus. Although Eq. (3.9) depends on the unknown parameters ρ and φ_k , the same methodology as in paragraphs 2.4.2.2 and 2.4.2.1 can be used in order to remove these parameters, thus allowing to calculate the measurement likelihood $p(\mathbf{z}_k | \tau_b, \mathbf{x}_k)$ (see Eq. (2.23)) which is required in the Bayesian filter. In the same manner, the density $p(\mathbf{z}_k | b_k = 0, \mathbf{x}_k)$ does not depend on the state \mathbf{x}_k and is obtained by Eq. (2.21).

Lastly, note that an additional hypothesis is required in order to derive the Bayesian filter for the proposed state space model. This last hypothesis consists in assuming that

$$p(\tau_b = k | b_{k-1} = 0, \mathbf{z}_{1:k-1}) = p(\tau_b = k | b_{k-1} = 0). \quad (3.10)$$

In other words, it means that the probability that the target appears at step k knowing that it does not appear before is independent of the measurement $\mathbf{z}_{1:k-1}$. In fact, this hypothesis is equivalent to the hypothesis that $\mathbf{z}_{1:k-1}$ conditionally to $b_{k-1} = 0$ is independent to the event $\{\tau_b = k\}$. Indeed, by definition of the conditional probability, the Eq. (3.10) is equal to

$$\begin{aligned} p(\tau_b = k | b_{k-1} = 0, \mathbf{z}_{1:k-1}) &= \frac{p(\tau_b = k, b_{k-1} = 0, \mathbf{z}_{1:k-1})}{p(b_{k-1} = 0, \mathbf{z}_{1:k-1})} \\ &= \frac{p(\tau_b = k, \mathbf{z}_{1:k-1} | b_{k-1} = 0)}{p(\mathbf{z}_{1:k-1} | b_{k-1} = 0)}. \end{aligned} \quad (3.11)$$

Therefore, by assuming that $\mathbf{z}_{1:k-1}$ conditionally to $b_{k-1} = 0$ is independent to the event $\{\tau_b = k\}$, the numerator in Eq. (3.11) factorizes as follows:

$$p(\tau_b = k, \mathbf{z}_{1:k-1} | b_{k-1} = 0) = p(\tau_b = k | b_{k-1} = 0) p(\mathbf{z}_{1:k-1} | b_{k-1} = 0), \quad (3.12)$$

thus leading to Eq. (3.10). This equivalent hypothesis seems reasonable to make since knowing that the target has not appeared until $k - 1$ (*i.e.* $b_{k-1} = 0$) there is no reason that the measurements $\mathbf{z}_{1:k-1}$ should provide information about the target appearance at step k .

3.2.3 Theoretical Bayesian solution

Our objective is now to derive the theoretical Bayesian recursion for the proposed model, *i.e.* to calculate the density $p(\mathbf{x}_k, b_k | \mathbf{z}_{1:k})$ from the density $p(\mathbf{x}_{k-1}, b_{k-1} | \mathbf{z}_{1:k-1})$.

3.2.3.1 Calculation of the posterior state density

Following the same reasoning as in section 2.6, this density can be rewritten as follows:

$$p(\mathbf{x}_k, b_k | \mathbf{z}_{1:k}) = p(b_k | \mathbf{z}_{1:k}) p(\mathbf{x}_k | b_k, \mathbf{z}_{1:k}). \quad (3.13)$$

Since the state \mathbf{x}_k is meaningless when $b_k = 0$, the only probabilities to calculate are the probabilities $p(b_k = 1 | \mathbf{z}_{1:k})$ and $p(b_k = 0 | \mathbf{z}_{1:k})$ which are simply obtained by definition of b_k by

$$p(b_k = 1 | \mathbf{z}_{1:k}) = \sum_{i=1}^k p(\tau_b = i | \mathbf{z}_{1:k}), \quad (3.14)$$

$$p(b_k = 0 | \mathbf{z}_{1:k}) = 1 - p(b_k = 1 | \mathbf{z}_{1:k}), \quad (3.15)$$

and the density $p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$. The latter can be written as

$$p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k, b_k = 1 | \mathbf{z}_{1:k})}{p(b_k = 1 | \mathbf{z}_{1:k})}. \quad (3.16)$$

Using the decomposition of event $\{b_k = 1\}$ in Eq. (A.5) the numerator can be expanded as

$$p(\mathbf{x}_k, b_k = 1 | \mathbf{z}_{1:k}) = \sum_{i=1}^k p(\mathbf{x}_k, \tau_b = i | \mathbf{z}_{1:k}) = \sum_{i=1}^k p(\tau_b = i | \mathbf{z}_{1:k}) p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k}). \quad (3.17)$$

Finally, dividing this expression by the probability $p(b_k = 1 | \mathbf{z}_{1:k})$ and using its decomposition in Eq. (3.14), it comes

$$p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k}) = \sum_{i=1}^k \frac{p(\tau_b = i | \mathbf{z}_{1:k})}{p(b_k = 1 | \mathbf{z}_{1:k})} p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k}) = \sum_{i=1}^k \alpha_{k,i} p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k}), \quad (3.18)$$

where

$$\alpha_{k,i} = \frac{p(\tau_b = i | \mathbf{z}_{1:k})}{p(b_k = 1 | \mathbf{z}_{1:k})} = \frac{p(\tau_b = i | \mathbf{z}_{1:k})}{\sum_{l=1}^k p(\tau_b = l | \mathbf{z}_{1:k})}. \quad (3.19)$$

Clearly $\sum_{i=1}^k \alpha_{k,i} = 1$. Note also that each $\alpha_{k,i}$ corresponds to the probability that the target appears at step i knowing that the target is effectively present. Therefore, the posterior density $p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$ is a mixture density with k components entirely defined by the densities $p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k})$ and the weighting terms $\alpha_{k,i}$.

In a Bayesian perspective, our aim is to calculate recursively each density $p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k})$ and the weighting terms $\alpha_{k,i}$ for all $i \in \{1, \dots, k\}$. However, in the sequel, for the sake of simplicity we will consider the probabilities $p(\tau_b = i | \mathbf{z}_{1:k})$ rather than the quantities $\alpha_{k,i}$ which are simply obtained through a normalization.

Thus, let us assume that at step $k-1$, for all $i \in \{1, \dots, k-1\}$, $p(\tau_b = i | \mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_{k-1} | \tau_b = i, \mathbf{z}_{1:k-1})$ are available. The aim is now to calculate, for all $i \in \{1, \dots, k\}$, $p(\tau_b = i | \mathbf{z}_{1:k})$ and $p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k})$. The next paragraph is dedicated to the calculation of the density components $p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k})$ while the paragraph 3.2.3.3 details the calculation of the probabilities $p(\tau_b = i | \mathbf{z}_{1:k})$.

3.2.3.2 Calculation of the mixture components

Using Bayes rule and the properties of the state-space model, each mixture component $p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k})$ can be rewritten as follows:

$$p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | \tau_b = i, \mathbf{x}_k)}{p(\mathbf{z}_k | \tau_b = i, \mathbf{z}_{1:k-1})}, \quad (3.20)$$

where

$$p(\mathbf{z}_k | \tau_b = i, \mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k | \tau_b = i, \mathbf{x}_k) p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k-1}) d\mathbf{x}_k. \quad (3.21)$$

The density $p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k-1})$ can be obtained for $i \in \{1, \dots, k-1\}$ by the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_{k-1} | \tau_b = i, \mathbf{z}_{1:k-1}) p(\mathbf{x}_k | \tau_b = i, \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}, \quad (3.22)$$

where the transition density $p(\mathbf{x}_k | \tau_b = i, \mathbf{x}_{k-1})$ corresponds, as already mentioned, to the continuing density $p_c(\mathbf{x}_k | \mathbf{x}_{k-1})$ in chapter 2. Thus, each component for $i \in \{1, \dots, k-1\}$ is provided by the classic theoretical Bayesian filter detailed in paragraph 1.2.2 and can be summarized as follows:

$$p(\mathbf{x}_{k-1} | \tau_b = i, \mathbf{z}_{1:k-1}) \xrightarrow[\text{Eq.(3.22)}]{\text{prediction}} p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k-1}) \xrightarrow[\text{Eq.(3.20)}]{\text{update}} p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k}). \quad (3.23)$$

However, it remains to calculate the density $p(\mathbf{x}_k | \tau_b = k, \mathbf{z}_{1:k})$ which corresponds to the target appearance at current step k . Since in this case, the state \mathbf{x}_k does not depend on the previous measurement $\mathbf{z}_{1:k-1}$, the equation (3.20) simplifies to

$$p(\mathbf{x}_k | \tau_b = k, \mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k | \tau_b = k) p(\mathbf{z}_k | \tau_b = k, \mathbf{x}_k)}{p(\mathbf{z}_k | \tau_b = k, \mathbf{z}_{1:k-1})}, \quad (3.24)$$

with

$$p(\mathbf{z}_k | \tau_b = k, \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \tau_b = k) p(\mathbf{z}_k | \tau_b = k, \mathbf{x}_k) d\mathbf{x}_k, \quad (3.25)$$

where $p(\mathbf{x}_k | \tau_b = k)$ is the prior density for the target appearance and corresponds, as already mentioned, to the birth density $p_b(\mathbf{x}_k)$ in chapter 2.

3.2.3.3 Calculation of the probabilities of appearance

Using Bayes rule, each probability $p(\tau_b = i | \mathbf{z}_{1:k})$ for $i \in \{1, \dots, k\}$ can be rewritten from the probability $p(\tau_b = i | \mathbf{z}_{1:k-1})$ for $i \in \{1, \dots, k\}$ as follows:

$$p(\tau_b = i | \mathbf{z}_{1:k}) = \frac{p(\tau_b = i | \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | \tau_b = i, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})}, \quad (3.26)$$

where each quantity $p(\mathbf{z}_k | \tau_b = i, \mathbf{z}_{1:k-1})$ can be obtained from Eq. (3.21) or from Eq. (3.25) when $i = k$. However, only the probabilities $p(\tau_b = i | \mathbf{z}_{1:k-1})$ for $k \in \{1, \dots, k-1\}$ can be obtained from the mixture density posterior density at step $k-1$ (see Eq. (3.17) by

replacing k by $k-1$). Therefore, to be able to calculate all the probabilities $p(\tau_b = i | \mathbf{z}_{1:k})$ at step k , it remains to evaluate the probability $p(\tau_b = k | \mathbf{z}_{1:k-1})$ and the normalization term $p(\mathbf{z}_k | \mathbf{z}_{1:k-1})$.

Let us start with the probability $p(\tau_b = k | \mathbf{z}_{1:k-1})$. By definition of variable b_k , the event $\{\tau_b = k\}$ is included in the more general event $\{b_{k-1} = 0\}$, therefore the probability $p(\tau_b = k | \mathbf{z}_{1:k-1})$ can be rewritten as follows:

$$p(\tau_b = k | \mathbf{z}_{1:k-1}) = p(\tau_b = k, b_k = 0 | \mathbf{z}_{1:k-1}), \quad (3.27)$$

$$= p(b_{k-1} = 0 | \mathbf{z}_{1:k-1}) p(\tau_b = k | b_{k-1} = 0, \mathbf{z}_{1:k-1}), \quad (3.28)$$

where the probability $p(\tau_b = k | b_{k-1} = 0, \mathbf{z}_{1:k-1})$ does not depend on the previous measurements $\mathbf{z}_{1:k-1}$ by hypothesis (see Eq. (3.10)) and is equal, from Eq. (A.8), to P_b . Finally the probability $p(\tau_b = k | \mathbf{z}_{1:k-1})$ can be evaluated from the quantities at previous iteration using the following relationship:

$$p(\tau_b = k | \mathbf{z}_{1:k-1}) = p(b_{k-1} = 0 | \mathbf{z}_{1:k-1}) P_b. \quad (3.29)$$

Recall that the probability $p(b_{k-1} = 0 | \mathbf{z}_{1:k-1})$ can be simply obtained from $p(b_{k-1} = 1 | \mathbf{z}_{1:k-1})$ as follows: $p(b_{k-1} = 0 | \mathbf{z}_{1:k-1}) = 1 - p(b_{k-1} = 1 | \mathbf{z}_{1:k-1})$ (see Eq. 3.15).

Concerning the normalization term $p(\mathbf{z}_k | \mathbf{z}_{k-1})$, the same idea as previously is used, *i.e.* we marginalize over b_k , which leads to

$$p(\mathbf{z}_k | \mathbf{z}_{1:k-1}) = p(\mathbf{z}_k, b_k = 1 | \mathbf{z}_{1:k-1}) + p(\mathbf{z}_k, b_k = 0 | \mathbf{z}_{1:k-1}). \quad (3.30)$$

By using the decomposition of the event $\{b_k = 1\}$ as in Eq. (3.17), the expression can be rewritten as follows:

$$p(\mathbf{z}_k | \mathbf{z}_{1:k-1}) = \sum_{i=1}^k p(\tau_b = i | \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | \tau_b = i, \mathbf{z}_{1:k-1}) + p(b_k = 0 | \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | b_k = 0, \mathbf{z}_{1:k-1}). \quad (3.31)$$

The probability $p(b_k = 0 | \mathbf{z}_{1:k-1})$ is simply obtained as in Eq. (3.15) by

$$p(b_k = 0 | \mathbf{z}_{1:k-1}) = 1 - p(b_k = 1 | \mathbf{z}_{1:k-1}) = 1 - \sum_{i=1}^k p(\tau_b = i | \mathbf{z}_{1:k-1}). \quad (3.32)$$

In the other hand, the quantity $p(\mathbf{z}_k | b_k = 0, \mathbf{z}_{1:k-1})$ does not depend on $\mathbf{z}_{1:k-1}$ by definition of the measurement model and is simply equal to $p(\mathbf{z}_k | b_k = 0)$ (see paragraph 3.2.2).

Finally, each probability $p(\tau_b = i | \mathbf{z}_{1:k})$ can be evaluated with Eq. (3.26) where $p(\mathbf{z}_k | \tau_b = i, \mathbf{z}_{1:k-1})$ and $p(\mathbf{z}_k | \mathbf{z}_{1:k-1})$ are provided respectively by Eq. (3.21) and Eq. (3.31) while the probability $p(\tau_b = k | \mathbf{z}_{1:k-1})$ is obtained with Eq. (3.29).

3.2.4 Particle filter approximation

In general, whereas the parameters of the posterior mixture density $p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$ (*i.e.* the mixture components $p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k})$) can be calculated recursively, in practice the corresponding equations are intractable and we must therefore resort to some

approximations. In the previous section, we demonstrated that the posterior density can be written as mixture. Therefore, we propose to use this particular structure to propose a particle approximation of the density $p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$. In practice, it means that each mixture component will be approximated by a particle filter.

3.2.4.1 Approximation of the mixture components

From Eq. (3.20), each component $p(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k})$ for all $i \in \{1, \dots, k-1\}$ follows the classic Bayes filter recursion. Therefore, each of them can be approximated with a particle filter.

To do so, let, for all $i \in \{1, \dots, k-1\}$,

$$\hat{p}(\mathbf{x}_{k-1} | \tau_b = i, \mathbf{z}_{1:k-1}) = \sum_{n=1}^{N_{p,i}} w_{k-1,i}^n \delta_{\mathbf{x}_{k-1,i}^n}(\mathbf{x}_{k-1}), \quad (3.33)$$

be a particle approximation of the mixture component $p(\mathbf{x}_{k-1} | \tau_b = i, \mathbf{z}_{1:k-1})$ at step $k-1$, where $N_{p,i}$ is the number of particles used for the i^{th} mixture component. Then the unnormalized weights of the particle approximation at step k are obtained, according to Eq. (1.94), by

$$\tilde{w}_{k,i}^n = w_{k-1,i}^n \frac{p_c(\mathbf{x}_{k,i}^n | \mathbf{x}_{k-1,i}^n) p(\mathbf{z}_k | \tau_b = i, \mathbf{x}_{k,i}^n)}{q(\mathbf{x}_{k,i}^n | \tau_b = i, \mathbf{x}_{k-1,i}^n, \mathbf{z}_k)}, \quad (3.34)$$

where $q(\mathbf{x}_{k,i}^n | \tau_b = i, \mathbf{x}_{k-1,i}^n, \mathbf{z}_k)$ is the instrumental distribution used to propagate particle states $\mathbf{x}_{k,i}^n$ (as in chapter 2, the prior is often chosen in that case). Obviously, the normalized weights $w_{k,i}$ are simply obtained through a normalization.

At this point, $k-1$ components have been updated. However recall that the mixture is composed of k components where the last one corresponds to the target appearance at current step (*i.e.* $\tau_b = k$). Using Eq. (3.24), the density component $p(\mathbf{x}_k | \tau_b = k, \mathbf{z}_{1:k})$ can be approximated by

$$\hat{p}(\mathbf{x}_k | \tau_b = k, \mathbf{z}_{1:k}) = \sum_{n=1}^{N_{p,k}} w_{k,k}^n \delta_{\mathbf{x}_{k,k}^n}(\mathbf{x}_k), \quad (3.35)$$

where the unnormalized weights are equal to

$$\tilde{w}_{k,k}^n = \frac{p_b(\mathbf{x}_{k,k}^n) p(\mathbf{z}_k | \tau_b = k, \mathbf{x}_{k,k}^n)}{q(\mathbf{x}_{k,k}^n | \tau_b = k, \mathbf{z}_k)}, \quad (3.36)$$

with $q(\mathbf{x}_k | \tau_b = k, \mathbf{z}_k)$ an instrumental density used to initialize the particle state $\mathbf{x}_{k,k}^n$. As in chapter 2, the choice of the instrumental density for the initialization is crucial for the performance. Fortunately, all the developments made in Chapter 2 concerning the instrumental density for position, velocity and amplitude parameters can be used again here. Finally, the normalized weights $w_{k,k}$ are, again, simply obtained through a normalization.

3.2.4.2 Calculation of the probabilities of appearance

From Eq. (3.26), the computation of the probabilities of appearance $p(\tau_b = i \mid \mathbf{z}_{1:k})$ require to evaluate the normalization terms $p(\mathbf{z}_k \mid \tau_b = i, \mathbf{z}_{1:k-1})$ and $p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})$, and the probabilities $p(\tau_b = i \mid \mathbf{z}_{1:k-1})$ for $i \in \{1, \dots, k\}$.

Concerning the probabilities of appearance $p(\tau_b = i \mid \mathbf{z}_{1:k-1})$ for $i \leq k-1$, if an approximation of the posterior $p(\mathbf{x}_{k-1} \mid b_{k-1} = 1, \mathbf{z}_{1:k-1})$ defined in Eq. (3.18) is assumed available, then these probabilities have already been approximated at previous step; let us denote by $\hat{p}(\tau_b = i \mid \mathbf{z}_{1:k-1})$ their approximation. Then, the probability $p(b_{k-1} = 1 \mid \mathbf{z}_{1:k-1})$ can be simply approximated using Eq. (3.14) and is denoted by $\hat{p}(b_{k-1} = 1 \mid \mathbf{z}_{1:k-1})$. However, it still remains to approximate the probability $p(\tau_b = k \mid \mathbf{z}_{1:k-1})$. From Eq. (3.29), it can be simply done as follows:

$$\hat{p}(\tau_b = k \mid \mathbf{z}_{1:k-1}) = (1 - \hat{p}(b_{k-1} = 1 \mid \mathbf{z}_{1:k-1})) P_b. \quad (3.37)$$

Concerning the normalization terms $p(\mathbf{z}_k \mid \tau_b = i, \mathbf{z}_{1:k-1})$, two cases must be considered, one for the index $i \in \{1, \dots, k-1\}$ and a second one for the case $\tau_b = k$. For each index $i \in \{1, \dots, k-1\}$, from Eq. (3.21),

$$\begin{aligned} p(\mathbf{z}_k \mid \tau_b = i, \mathbf{z}_{1:k-1}) &= \int p(\mathbf{z}_k \mid \tau_b = i, \mathbf{x}_k) p(\mathbf{x}_k \mid \tau_b = i, \mathbf{z}_{1:k-1}) d\mathbf{x}_k, \\ &= \mathbb{E}_{p(\mathbf{x}_k \mid \tau_b = i, \mathbf{z}_{1:k-1})} [p(\mathbf{z}_k \mid \tau_b = i, \mathbf{x}_k)], \end{aligned}$$

i.e. it corresponds to the expectation of $p(\mathbf{z}_k \mid \tau_b = i, \mathbf{x}_k)$ with respect to the density $p(\mathbf{x}_k \mid \tau_b = i, \mathbf{z}_{1:k-1})$. Therefore, as this integral has the form of Eq. (1.74), it can be obtained via a particle approximation of the predicted density $p(\mathbf{x}_k \mid \tau_b = i, \mathbf{z}_{1:k-1})$. This approximation is not directly available but it is however possible to derive an approximation of the latter using the particle approximation at previous step $p(\mathbf{x}_{k-1} \mid \tau_b = i, \mathbf{z}_{1:k-1})$ and the Chapman-Kolmogorov equation (3.22). A possible approximation was proposed (in a completely different context) by Vermaak *et al.* in [VGP05] as

$$p(\mathbf{x}_k \mid \tau_b = i, \mathbf{z}_{1:k-1}) \approx \sum_{n=1}^{N_{p,i}} a_{k,i}^n \delta_{\mathbf{x}_{k,i}^n}(\mathbf{x}_k), \quad (3.38)$$

where the unnormalized predictive weights $\tilde{a}_{k,i}^n$ are equal to

$$\tilde{a}_{k,i}^n = w_{k-1,i}^n \frac{p(\mathbf{x}_{k,i}^n \mid \tau_b = i, \mathbf{x}_{k-1,i}^n)}{q(\mathbf{x}_{k,i}^n \mid \tau_b = i, \mathbf{x}_{k-1,i}^n, \mathbf{z}_k)}, \quad (3.39)$$

and the predictive weights $a_{k,i}^n$ are simply obtained through a normalization by the term

$$C_{k,i} = \sum_{n=1}^{N_{p,i}} \tilde{a}_{k,i}^n. \quad (3.40)$$

Then the approximation $\hat{p}(\mathbf{z}_k \mid \tau_b = i, \mathbf{z}_{1:k-1})$ of the density $p(\mathbf{z}_k \mid \tau_b = i, \mathbf{z}_{1:k-1})$ is computed by substituting in Eq. (3.21) the density $p(\mathbf{x}_k \mid \tau_b = i, \mathbf{z}_{1:k-1})$ with its particle approximation defined in Eq. (3.38), leading to

$$\hat{p}(\mathbf{z}_k \mid \tau_b = i, \mathbf{z}_{1:k-1}) = \sum_{n=1}^{N_{p,i}} a_{k,i}^n p(\mathbf{z}_k \mid \tau_b = i, \mathbf{x}_{k,i}^n). \quad (3.41)$$

Furthermore, by noticing that the unnormalized weights $\tilde{w}_{k,i}^n$ are related to the predictive weights $a_{k,i}^n$ through the following equation:

$$\tilde{w}_{k,i}^n = C_{k,i} a_{k,i}^n p(\mathbf{z}_k | \tau_b = i, \mathbf{x}_{k,i}^n), \quad (3.42)$$

the approximation can be rewritten as

$$\hat{p}(\mathbf{z}_k | \tau_b = i, \mathbf{z}_{1:k-1}) = \frac{1}{C_{k,i}} \sum_{n=1}^{N_{p,i}} \tilde{w}_{k,i}^n. \quad (3.43)$$

Note, however, that when the instrumental density is chosen to be the prior (*i.e.* $p(\mathbf{x}_k | \tau_b = i, \mathbf{x}_{k-1}^n)$), then $C_{k,i} = 1$ and the Eq. (3.43) is simply the sum of the unnormalized weights.

Now, it remains to evaluate the normalization term $p(\mathbf{z}_k | \tau_b = k, \mathbf{z}_{1:k-1})$ for the case $\tau_b = k$. From Eq. (3.25), it is also an expectation with respect to the density $p(\mathbf{x}_k | \tau_b = k)$. However, contrary to the previous case where a particle approximation of the predicted density is required, here the classic importance sampling can be directly applied since the density $p(\mathbf{x}_k | \tau_b = k)$ is known in closed-form (*i.e.* it is the birth density defined in paragraph 3.2.1.2). Thus, the integral (3.25) can be approximated as follows:

$$\begin{aligned} \hat{p}(\mathbf{z}_k | \tau_b = k, \mathbf{z}_{1:k-1}) &= \frac{1}{N_{p,k}} \sum_{n=1}^{N_{p,k}} \frac{p(\mathbf{x}_{k,k}^n | \tau_b = k) p(\mathbf{z}_k | \tau_b = k, \mathbf{x}_{k,k}^n)}{q(\mathbf{x}_{k,k}^n | \tau_b = k, \mathbf{z}_k)} \\ &= \frac{1}{N_{p,k}} \sum_{n=1}^{N_{p,k}} \tilde{w}_{k,k}^n, \end{aligned} \quad (3.44)$$

where $q(\mathbf{x}_k | \tau_b = k, \mathbf{z}_k)$ is an instrumental density and $N_{p,k}$ is the number of particles.

The whole normalization term $\hat{p}(\mathbf{z}_k | \mathbf{z}_{1:k-1})$ can be simply approximated, using its expression provided by Eq. (3.31). Then, each probability $p(\tau_b = i | \mathbf{z}_{1:k})$ for all $i \in \{1, \dots, k\}$ is finally provided by

$$\hat{p}(\tau_b = i | \mathbf{z}_{1:k}) = \frac{\hat{p}(\tau_b = i | \mathbf{z}_{1:k-1}) \hat{p}(\mathbf{z}_k | \tau_b = i, \mathbf{z}_{1:k-1})}{\hat{p}(\mathbf{z}_k | \mathbf{z}_{1:k-1})}. \quad (3.45)$$

3.2.4.3 Dealing with the increasing number of particles and resampling strategies

As previously stated, the proposed particle filter is composed of several particle clouds corresponding to each element of the mixture distribution. By definition of the density $p(\mathbf{x}_k | b_k = 0, \mathbf{z}_{1:k})$ in Eq. (3.18) the number of mixture components at time step k is k and it thus increases with time. Therefore, if at each iteration a new component is initialized with $N_{p,i}$ particles, the total number of particles will be incremented by $N_{p,i}$ and after some iterations the computational cost of the algorithm will become prohibitive. Thus, it is preferable to limit the number of particles. In the sequel, we propose two solutions:

- In the first solution, the number of particles remains the same and constant over time for all the mixture components. Moreover, the mixture components are always resampled separately.

- This first solution may suffer from a degeneracy phenomenon, *i.e.* after some iterations some of the component weights $\alpha_{k,i}$ may be pretty small, so that some components are approximated with a particle filter while they do not really participate in the approximation of the whole density $p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$. Therefore, we propose a second solution that allows to resample over the whole density $p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$ rather than over the mixture components when a severe degeneracy is found over the weight mixture components.

Constant number of particles per mixture component

The easiest way to deal with the increasing number of particles consists in limiting the number of mixture components to an integer $N_{mix,max}$ and discarding the one with the lowest probability $\hat{p}(\tau_b = i | \mathbf{z}_{1:k})$ if the number of components is equal to $N_{mix,max}$. Indeed, if all the mixture components have the same number of particles $N_{p,mix}$, discarding the component with the lowest probability allows to release $N_{p,mix}$ particles that can be used to initialize the new component at next step. The maximum number of particles $N_{p,max}$ is then equal to $N_{p,max} = N_{p,mix} \times N_{mix,max}$. Before going further in the details of the algorithm, let us first explain its general principle. To this purpose, let us assume that at step $k - 1$, $N_{k-1,mix} = N_{mix,max} - 1$ mixture components $p(\mathbf{x}_{k-1} | \tau_b = i, \mathbf{z}_{1:k-1})$ have been kept where their corresponding probability is $p(\tau_b = i, \mathbf{z}_{1:k-1})$. The general principle of the proposed strategy is then the following:

- First, the particles of each remaining component $p(\mathbf{x}_{k-1} | \tau_b = i, \mathbf{z}_{1:k-1})$ are propagated according to the instrumental density $q_c(\mathbf{x}_k | \mathbf{x}_{k-1})$ (often chosen to be the prior).
- Then, a new component is created with $N_{p,mix}$ particles in order to approximate the density $p(\mathbf{x}_k | \tau_b = k, \mathbf{z}_{1:k})$. For this particle approximation, the particles are sampled according to an instrumental density $q_b(\mathbf{x}_k | \mathbf{z}_k)$. Therefore, the number of components is incremented by one, *i.e.* $N_{k,mix} = N_{k-1,mix} + 1$ if the number of mixture components is equal to $N_{mix,max}$.
- The weights for each of the $N_{k,mix}$ mixture components are calculated via Eq. (3.34) for the components present at previous step or via Eq. (3.36) for the new component, thus also allowing to update the probability $\hat{p}(\tau_b = i | \mathbf{z}_{1:k-1})$ at step k and the probability $p(b_k = 1 | \mathbf{z}_{1:k})$.
- If the number of components $N_{k,mix}$ is equal to $N_{mix,max}$, the mixture component with the lowest probability $\hat{p}(\tau_b = i | \mathbf{z}_{1:k})$ is discarded, thus allowing to use its $N_{p,mix}$ particles to initialize a new component at the next iteration. Therefore, the number of components $N_{k,mix}$ is now equal to $N_{k,mix} - 1$. Clearly, the only cases where the number of components will not be equal to $N_{mix,max}$ are the first iterations $k \in \{1, \dots, N_{mix,max} - 1\}$.
- Finally each remaining component is resampled if needed. Note that contrary to the marginalized monotarget particle filter in section 2.6 where a resampling procedure must be performed at each step, here the resampling is optional and can be performed according to the N_{eff} of each component.

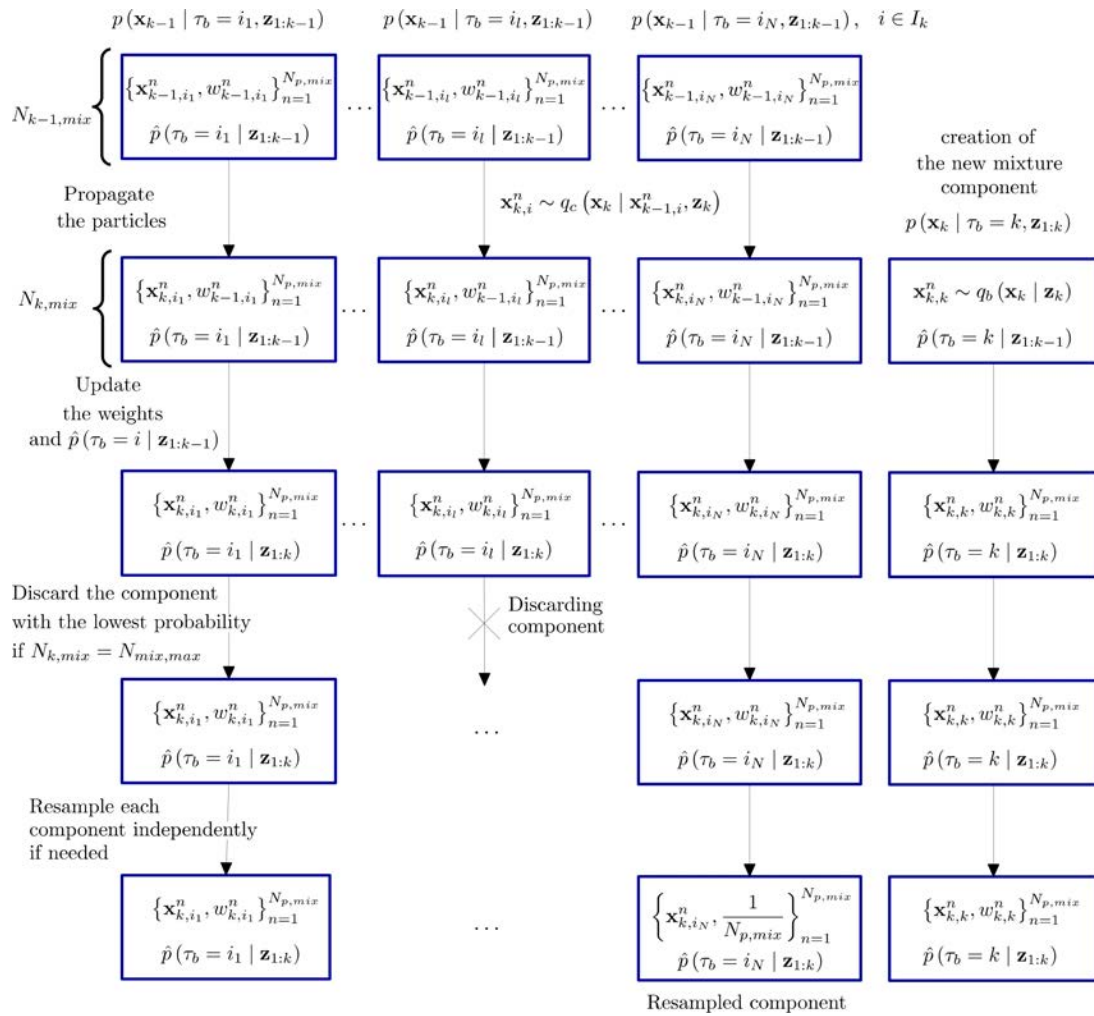


Figure 3.1 – General scheme of the particle filter for time appearance detection with a fixed and constant number of particles per mixture component.

A general scheme of the proposed algorithm is presented in Figure 3.1, where the indexes i_1, \dots, i_N are the set of indexes corresponding to the time index of the $N_{k, mix}$ remaining mixture components.

We will now detail the computations required for the different components of this particle filter. Let $I_k = \{i_1, \dots, i_{N_{k, mix}}\}$ be the set of indexes corresponding to the $N_{k, mix}$ remaining mixture components $\{\tau_b = i_l\}$ ($l \in \{1, \dots, N_{k, mix}\}$). The particle approximation of $p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$ can be rewritten as follows:

$$\hat{p}(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k}) = \sum_{i \in I_k} \hat{\alpha}_{k, i} \hat{p}(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k}), \quad (3.46)$$

where

$$\hat{\alpha}_{k, i} = \frac{\hat{p}(\tau_b = i | \mathbf{z}_{1:k})}{\sum_{l \in I_k} \hat{p}(\tau_b = l | \mathbf{z}_{1:k})}. \quad (3.47)$$

Then, if $N_{k,mix} = N_{mix,max}$, the component with the smallest probability, corresponding to index

$$i_{min} = \arg \min_i \hat{p}(\tau_b = i \mid \mathbf{z}_{1:k}) \quad (3.48)$$

is removed from the set I_k , *i.e.* the set I_k is now equal to

$$I_k = I_k \setminus \{i_{min}\}. \quad (3.49)$$

The posterior density then becomes

$$\hat{p}(\mathbf{x}_k \mid b_k = 1, \mathbf{z}_{1:k}) = \sum_{i \in I_k} \hat{\alpha}'_{k,i} \hat{p}(\mathbf{x}_k \mid \tau_b = i, \mathbf{z}_{1:k}), \quad (3.50)$$

with

$$\hat{\alpha}'_{k,i} = \frac{\hat{p}(\tau_b = i \mid \mathbf{z}_{1:k})}{\sum_{l \in I_{k,min}} \hat{p}(\tau_b = l \mid \mathbf{z}_{1:k})}, \quad (3.51)$$

calculated with the updated set I_k (*i.e.* without the time index i_{min}). The proposed Appearance Time TBD Particle Filter is finally summarized by Algorithm 3.1.

Concerning the estimation, for each mixture component $\hat{p}(\mathbf{x}_k \mid \tau_b = i, \mathbf{z}_{1:k})$ the state \mathbf{x}_k and the posterior covariance matrix can be estimated using the classic estimator defined in Eq. (1.96) and Eq. (1.97). They are respectively denoted by $\hat{\mathbf{x}}_{k|k,i}$ and $\hat{\mathbf{P}}_{k|k,i}$. Finally, the estimators over all the mixture components are simply obtained by:

$$\hat{\mathbf{x}}_{k|k} = \sum_{i \in I_k} \hat{\alpha}_{k,i} \hat{\mathbf{x}}_{k|k,i}, \quad (3.52)$$

$$\hat{\mathbf{P}}_{k|k} = \sum_{i \in I_k} \hat{\alpha}_{k,i} \hat{\mathbf{P}}_{k|k,i}. \quad (3.53)$$

The probability of appearance $p(b_k = 1 \mid \mathbf{z}_{1:k})$ can be approximated by:

$$\hat{p}(b_k = 1 \mid \mathbf{z}_{1:k}) = \sum_{i \in I_k} \hat{p}(\tau_b = i \mid \mathbf{z}_{1:k}). \quad (3.54)$$

Resampling over the whole density $p(\mathbf{x}_k \mid b_k = 1, \mathbf{z}_{1:k})$

The previous strategy allows to limit the number of particles but does not take into account the component weights $\hat{\alpha}_{k,i}$ in the resampling procedure. As a consequence, even if the component with the smallest probability has been removed, some component weights $\hat{\alpha}_{k,i}$ may still be pretty small and participate for a very little part in the estimation of the state \mathbf{x}_k in Eq. (3.52). In fact, this can be seen as a degeneracy phenomenon (see paragraph 1.2.4.4) where after some iterations, one of the mixture weights may be very close to one while the others are almost zero. Therefore, some computational resources are devoted to the calculation of the mixture components that do not actually participate to the state estimation. To avoid this drawback, we propose in the sequel to take into account the weight components in order to resample only the relevant mixture components.

The practical implementation explanation of this solution has been appended to Appendix B; in the sequel, we will only provide a general scheme of this solution and its motivations.

Algorithm 3.1 Appearance Time TBD Particle Filter

Require: mixture components $\{w_{k-1}^i, \mathbf{x}_{k-1,i}^n\}_{n=1}^{N_{p,mix}}$ and probabilities $p(\tau_b = i \mid \mathbf{z}_{1:k-1})$ with $i \in I_{k-1}$ at step $k-1$.

- 1: **for** $i \in I_{k-1}$ **do**
- 2: **for** $n = 1$ to $N_{p,mix}$ **do**
- 3: Draw $\mathbf{x}_{k,i}^n \sim q(\mathbf{x}_k \mid \tau_b = i, \mathbf{x}_{k,i}^n, \mathbf{z}_k)$.
- 4: Compute unnormalized weight $\tilde{w}_{k,i}^n$ according to (3.34).
- 5: **end for**
- 6: Compute $C_{k,i}$ according to (3.40).
- 7: Compute $\hat{p}(\mathbf{z}_k \mid \tau_b = i, \mathbf{z}_{1:k-1})$ according to (3.43)
- 8: Normalisation: $w_{k,i}^n \leftarrow \frac{\tilde{w}_{k,i}^n}{\sum_{l=1}^{N_{p,mix}} \tilde{w}_{k,i}^l}$, $n = 1 \dots N_{p,mix}$.
- 9: **end for**
- 10: **for** new mixture component at step k **do**
- 11: $I_k = I_{k-1} \cup \{k\}$
- 12: $N_{k,mix} = N_{k-1,mix} + 1$
- 13: **for** $n = 1$ to $N_{p,mix}$ **do**
- 14: Draw $\mathbf{x}_{k,k}^n \sim q(\mathbf{x}_k \mid \tau_b = k, \mathbf{z}_k)$.
- 15: Compute unnormalized weight $\tilde{w}_{k,k}^n$ according to (3.36).
- 16: **end for**
- 17: Compute $\hat{p}(\mathbf{z}_k \mid \tau_b = k, \mathbf{z}_{1:k-1})$ according to (3.44)
- 18: Compute $\hat{p}(\tau_b = k \mid \mathbf{z}_{1:k-1})$ according to (3.37).
- 19: Normalization: $w_{k,k}^n \leftarrow \frac{\tilde{w}_{k,k}^n}{\sum_{l=1}^{N_{p,mix}} \tilde{w}_{k,k}^l}$, $n = 1 \dots N_{p,mix}$.
- 20: **end for**
- 21: Compute $\hat{p}(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})$ according to (3.31) where the sum is performed over I_k .
- 22: Compute $\hat{p}(\tau_b = i \mid \mathbf{z}_{1:k})$ according to (3.45), for $i \in I_k$.
- 23: **if** $N_{k,mix} = N_{mix,max}$ **then**
- 24: Find i_{min} according to (3.48).
- 25: Set $I_k = I_k \setminus \{i_{min}\}$.
- 26: Set $N_{k,mix} = N_{k,mix} - 1$.
- 27: **end if**
- 28: **for** $i \in I_k$ **do**
- 29: Compute $N_{\text{eff},i}$ according to Eq. (1.98) for component $\hat{p}(\mathbf{x}_k \mid \tau_b = i, \mathbf{z}_{1:k})$.
- 30: **if** $N_{\text{eff},i} < N_T$ **then**
- 31: Resample $N_{p,mix}$ particles.
- 32: Reset weights: $w_{k,i}^n \leftarrow \frac{1}{N_{p,mix}}$ $n = 1, \dots, N_{p,mix}$.
- 33: **end if**
- 34: **end for**

Ensure: $\{w_{k,i}^n, \mathbf{x}_{k,i}^n\}_{n=1}^{N_{p,mix}}$, $\hat{p}(\tau_b = i \mid \mathbf{z}_{1:k})$, $i \in I_k$.

The central idea of the proposed method is to calculate an effective sample size number over the whole density $\hat{p}(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$ and then resample from this density rather than from the mixture components. As a consequence, in the resampling procedure, the particles belonging to a component with a high component weight $\hat{\alpha}_{k,i}$ will be mostly selected compared to particles belonging to a component with a small $\hat{\alpha}_{k,l}$, so that the number of particles will be different for the different mixture components.

However, resampling over the overall density does not solve the problem of the increasing number of particles. Thus we propose to use the same methodology as in the previous paragraph (*i.e.* discarding the mixture component with the lowest component weight $\hat{\alpha}_{k,i}$). However, since the number of particles per component may be different, the procedure to discard some particles in order to initialize new components has to be changed.

To this purpose, let us assume that $N_{k,mix}$ mixture components are present respectively with $N_{p,i}$ particles per component. Then, we propose the following procedure:

- First, as in the previous paragraph, the total number of particles (denoted as $N_{p,k}^{all}$) is calculated and compared to the maximal number of particles $N_{p,max}$.
- If the number $N_{p,k}^{all}$ is equal to $N_{p,max}$, then as in the previous paragraph, the index i_{min} of the mixture component with the lowest probability is considered. However, here, since the number of particles is different between the components, two cases must be considered:
 - Either $N_{p,i_{min}}$ is equal to a number $N_{p,init}$ and then the component i_{min} is discarded, so that $N_{p,init}$ particles can be used to initialize a new component at next step.
 - Or $N_{p,i_{min}}$ is greater than $N_{p,init}$. Thus, the component i_{min} does not need to be totally discarded. Indeed, the component i_{min} can be kept by reducing the number of particles from $N_{p,i_{min}}$ to $N_{p,i_{min}} - N_{p,init}$. To this purpose, $N_{p,i_{min}} - N_{p,init}$ particles are resampled from the component i_{min} .

In Figure 3.2 a block diagram of the procedure to discard $N_{p,init}$ particles in order to initialize new components is proposed.

Although the procedures to discard $N_{p,init}$ particles and to resample the mixture are thus different from the previous algorithm working with a constant and fix number of particles per component, for all the others steps they follow exactly the same scheme. That is to say, at each new iteration of the algorithm the particles of the $N_{k-1,mix}$ components² are propagated and a new one is created. Then, the weights for each component are calculated and the probabilities $\hat{p}(\tau_b = i | \mathbf{z}_{1:k-1})$ are updated. Finally, $N_{p,init}$ particles are discarded from one of the component (if needed) and the resampling procedure is performed.

Now that the procedure to discard $N_{p,init}$ has been detailed, it remains to explain how the resampling procedure is performed. For the effective sample size number over the

²Note that it means that if resampling is performed over all the mixture components, the resulting particle approximation will be processed exactly in the same manner as the other components, see Appendix B.

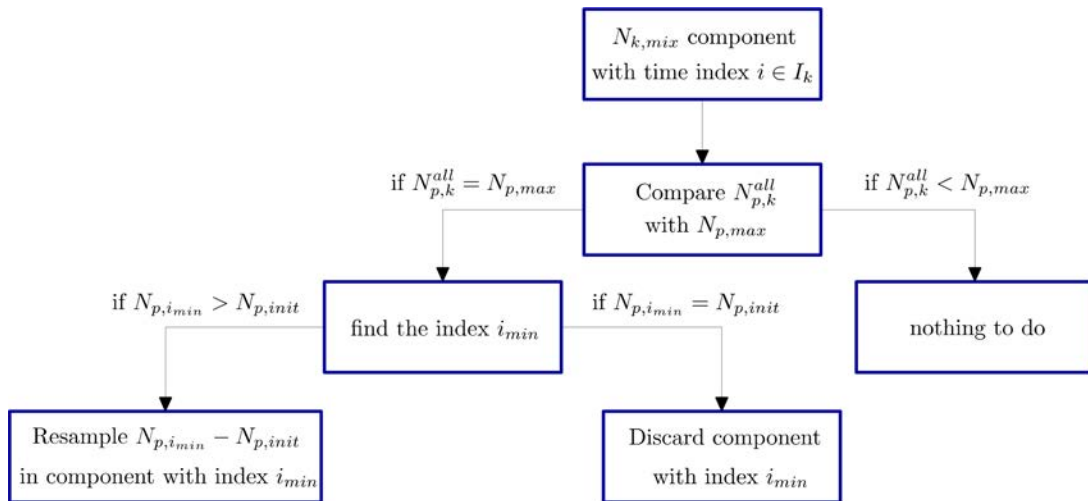


Figure 3.2 – Block diagram to discard $N_{p,init}$ when the number of particles is different between the mixture components.

whole density $\hat{p}(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$, we propose the following definition:

$$N_{\text{eff},k}^{\text{all}} \approx \left(\sum_{i \in I_k} \frac{\hat{\alpha}_{k,i}^2}{N_{\text{eff},i}} \right)^{-1}. \quad (3.55)$$

Then a resampling procedure can be performed if $N_{\text{eff},k}^{\text{all}} < N_{T,k}^{\text{all}}$, with $N_{T,k}^{\text{all}} = \beta_{\text{all}} N_{p,k}^{\text{all}}$ and $0 < \beta_{\text{all}} \leq 1$. However, for some reasons that will be detailed in the sequel, the resampling procedure over all the components does not have to be performed at each iteration and some restrictions have to be introduced. Indeed:

- If the resampling procedure over all the components is performed at each iteration (*i.e.* $\beta_{\text{all}} = 1$), there is no interest in using the mixture structure detailed in this chapter. Indeed, in such a situation, at the end of each iteration only one component will be present. Therefore, at the next iteration only two components will be present, the one from previous step and the new one sampling the event $\{\tau_b = k\}$. However, since a resampling procedure will be performed over this two components, it will still remain one component (gathering the two components before the resampling step) at the end of this step. As a consequence, performing a resampling over all the mixture components only allows to consider two components. Moreover, in this case, the structure of the particle filter is almost equivalent to the marginalized particle filter detailed in section 2.6. Indeed, at the beginning of each iteration one component is present, then a new one is created in order to sample the "birth" case. Parameters and particle filters are updated and finally the two components are resampled jointly in order to create one particle filter component gathering the birth and the continuing particle.
- In the same manner, to avoid that the new mixture components are resampled too quickly from the overall resampling procedure, it is necessary to limit the frequency of this resampling procedure. Let us illustrate this with a generic example. Let us

assume that in the resampling procedure over all the mixture components, $N_{p,mix}^{all} = k_{all}N_{p,init}$ particles are resampled (with k_{all} an integer greater than one). Moreover, let us also assume that $N_{p,max}$ can be factorized as $N_{p,max} = k_{max}N_{p,init}$ ($k_{max} > k_{all}$). Then, if a resampling procedure over all the mixture components is performed, it will remain $N_{p,max} - N_{p,mix}^{all}$ particles allowing to initialize $k_{max} - k_{all}$ new components. However, in the next iterations, if the resampling procedure over all the mixture components is performed too quickly, the particle filter will not have enough time to initialize the $k_{max} - k_{all}$ new components. As a consequence, some particles reserved to initialize new components will be practically never used by the particle filter.

Therefore to avoid the two exposed problems, we propose the following strategies:

- First, $N_{eff,k}^{all}$ is compared to $N_{T,k}^{all}$ without any restriction, and the resampling over all the mixture components is performed if $N_{eff,k}^{all} < N_{T,k}^{all}$. β_{all} has to be chosen not too high to avoid systematic resampling. This step is present in order to avoid a severe degeneracy.
- If $N_{eff,k}^{all} > N_{T,k}^{all}$, an extra condition is considered for resampling over all the mixture components: the resampling procedure will be performed if the total number of particles $N_{p,k}^{all}$ is greater than a number N_p^{min} . If not, no resampling over all the mixture components is done. Thus, if a resampling procedure over all the mixture components was performed at previous step, at the next step the minimal number of particles may not be reached since only one extra components will have been initialized. As a consequence, this new component will continue to explore the state independently from the other components. Of course, the number $N_{eff,k}^{all}$ may still be compared to a number $N_{T,k}^{min} = \beta_{min}N_{p,k}^{all}$ in order to perform the resampling.
- Lastly, if the conditions $N_{p,k}^{all} \geq N_p^{min}$ and $N_{eff,k}^{all} < N_{T,k}^{min}$ are not reached, the mixture components are resampled separately.

A block-diagram of this resampling procedure is proposed in Figure 3.3.

Of course, this strategy is heuristic and no optimality can be ensured; some other strategies may outperform it. On the other hand, note also that the resampling step for the target time appearance detection particle filter offers more possibilities than the particle filters developed in Chapter 2.

3.3 Particle filter for target disappearance time detection

Until now, we have only considered the problem of target appearance detection. In a similar way, the detection of the target disappearance can be done in the Bayesian quickest change detection framework. This case is easier to solve since, as will be seen, no mixture has to be considered. Moreover, it can be shown that, in this case, the model considered in the Bayesian quickest change detection framework (with a geometric prior) is equivalent to the one outlined in the chapter 2 with a particular choice of the transition probability matrix Π . Therefore, in the following, only the key points of the algorithm will be detailed.

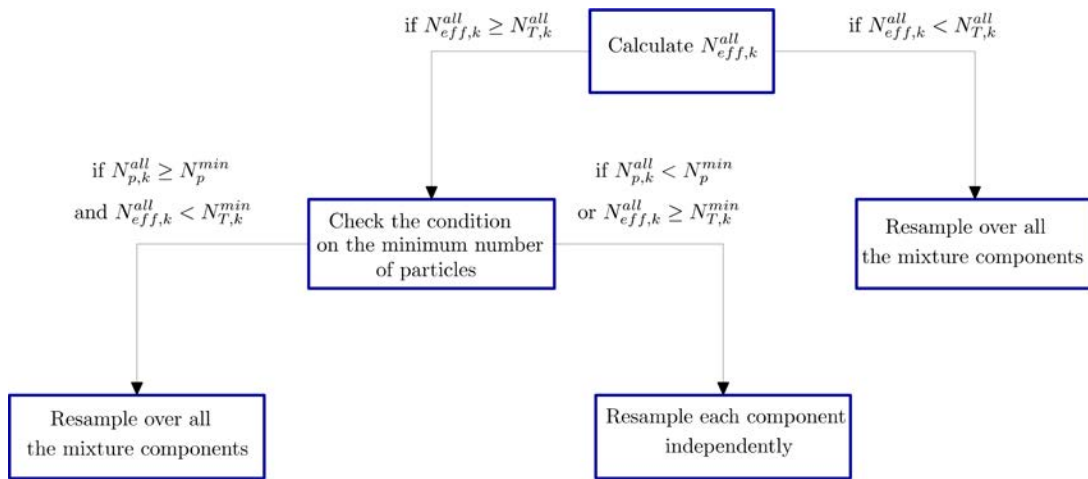


Figure 3.3 – Block-diagram of the resampling procedure that allows to resample over all the mixture components.

3.3.1 State model

Let us define by τ_d the time instant when the target disappears from the radar window. As previously, we propose to consider a geometric prior for the time disappearance τ_d , provided by

$$p(\tau_d = i) = \begin{cases} 0, & i = 0, \\ P_d(1 - P_d)^i, & i \geq 1, \end{cases} \quad (3.56)$$

where $0 < P_d < 1$ denotes the probability of disappearance. Moreover, similarly to the variable b_k introduced in paragraph 3.2.1.1, let us define the variable d_k as

$$d_k = \begin{cases} 1, & \text{if } \tau_d \geq k, \\ 0, & \text{otherwise.} \end{cases} \quad (3.57)$$

Using the same calculation as for the variable b_k in Appendix A, it is easy to show first that

$$p(d_k = 0 \mid d_{k-1} = 1) = P_d \quad (3.58)$$

and secondly that the process $(d_k)_{k \in \mathbb{N}}$ is a two-state Markov chain with the following transition probability matrix

$$\Pi_{d_k} = \begin{bmatrix} 1 & 0 \\ P_d & 1 - P_d \end{bmatrix}, \quad (3.59)$$

where the state $d_k = 0$ is an absorbing state. Lastly, for the initialization step, $p(d_0 = 1) = 1$.

Contrary to the appearance case where the prior model has been specified for the random process $(\tau_b, \mathbf{x}_k)_{k \in \mathbb{N}}$, here it is unnecessary since the target is assumed present at the initial step $k = 0$ and the evolution of the process \mathbf{x}_k can be easily modeled conditionally to the variable d_k . Therefore, as in chapter 2, this amounts to define the evolution of the hybrid process $(\mathbf{x}_k, d_k)_{k \in \mathbb{N}}$ rather than $(\mathbf{x}_k, \tau_d)_{k \in \mathbb{N}}$. Since the process $(d_k)_{k \in \mathbb{N}}$ is Markovian, the entire process can also be assumed Markovian with the same transition probability as in Eq. (2.2), *i.e.*

$$p(\mathbf{x}_k, d_k \mid \mathbf{x}_{k-1}, d_{k-1}) = p(d_k \mid d_{k-1}) p(\mathbf{x}_k \mid d_{k-1}, d_k, \mathbf{x}_{k-1}), \quad (3.60)$$

where the transition probabilities for the variable d_k are provided by Eq. (3.59). Concerning the transition probabilities $p(\mathbf{x}_k | d_{k-1}, d_k, \mathbf{x}_{k-1})$, the only case to consider is the case $d_{k-1} = 1$ and $d_k = 1$, that corresponds to the continuing density defined in Chapter 2, *i.e.*

$$p(\mathbf{x}_k | d_{k-1} = 1, d_k = 1, \mathbf{x}_{k-1}) = p_c(\mathbf{x}_k | \mathbf{x}_{k-1}). \quad (3.61)$$

The other transition densities, either deal with the case $d_k = 0$ where the state \mathbf{x}_k is meaningless or with the case $d_{k-1} = 0$ and $d_k = 1$ that cannot happen due to the particular structure of the transition matrix Π_{d_k} : the target cannot appear anymore once it has disappeared. Lastly, it remains to define the density $p(\mathbf{x}_0 | d_0 = 1)$ that corresponds to the initialization of the process (the case $d_0 = 0$ does not need to be considered since $p(d_0 = 0) = 0$). Contrary to the appearance case where the birth density is often chosen to be non-informative (*e.g.* uniform), here the target is assumed present and therefore it seems reasonable to assume that some information is available about the target state location. For instance, we can choose as initial prior the following density,

$$p(\mathbf{x}_0 | d_0 = 1) = \mathcal{N}(\mathbf{x}_0; \bar{\mathbf{x}}_0, \mathbf{P}_0), \quad (3.62)$$

where $\bar{\mathbf{x}}_0$ is the initial target state mean and \mathbf{P}_0 the initial covariance matrix. In practice, $\bar{\mathbf{x}}_0$ and \mathbf{P}_0 may have been obtained from a previous detection procedure.

3.3.2 Measurement model

The measurement model is defined as in Chapter 2 (see section 2.3), *i.e.*

$$\mathbf{z}_k = d_k \rho e^{j\varphi_k} \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k. \quad (3.63)$$

3.3.3 Bayesian filter and particle filter approximation

The aim is now to compute recursively the density $p(\mathbf{x}_k, d_k | \mathbf{z}_{1:k})$ for any $k \geq 1$, that is to calculate the probability $p(d_k = 1 | \mathbf{z}_{1:k})$ and the density $p(\mathbf{x}_k | d_k = 1, \mathbf{z}_{1:k})$.

Concerning the density $p(\mathbf{x}_k | d_k = 1, \mathbf{z}_{1:k})$, the Bayesian filter can be directly derived via the equation

$$p(\mathbf{x}_k | d_k = 1, \mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_k | d_k = 1, \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | d_k = 1, \mathbf{x}_k)}{p(\mathbf{z}_k | d_k = 1, \mathbf{z}_{1:k-1})}, \quad (3.64)$$

where the density $p(\mathbf{x}_k | d_k = 1, \mathbf{z}_{1:k-1})$ is obtained via the Chapman-Kolmogorov equation where the integration must be performed on \mathbf{x}_{k-1} and d_{k-1} , *i.e.*

$$p(\mathbf{x}_k | d_k = 1, \mathbf{z}_{1:k-1}) = \sum_{d_{k-1}} \int p(\mathbf{x}_k, \mathbf{x}_{k-1}, d_{k-1} | d_k = 1, \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}. \quad (3.65)$$

However, recall that if $d_k = 1$ then $d_{k-1} = 1$. Therefore the sum with respect to d_{k-1} must be done only for $d_{k-1} = 1$ and the Eq. (3.65) simplifies to

$$p(\mathbf{x}_k | d_k = 1, \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_{k-1} | d_k = 1, d_{k-1} = 1, \mathbf{z}_{1:k-1}) p_c(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}. \quad (3.66)$$

Moreover, as it was demonstrated with Eq. (2.78) and Eq. (2.79), the dependence with $d_k = 1$ in Eq. (3.66) can be removed, leading to the classic Chapman-Kolmogorov equation which depends only on the density at the previous step and the transition density, *i.e.*

$$p(\mathbf{x}_k | d_k = 1, \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_{k-1} | d_{k-1} = 1, \mathbf{z}_{1:k-1}) p_c(\mathbf{x}_k | \mathbf{x}_{k-1}) d\mathbf{x}_{k-1}. \quad (3.67)$$

Therefore, if a particle approximation $\{\mathbf{x}_k^n, w_{k,d}^n\}_{n=1}^{N_{p,d}}$ of the posterior $p(\mathbf{x}_{k-1} | d_{k-1} = 1, \mathbf{z}_{1:k-1})$ is available at step $k-1$ (where $N_{p,d}$ is the number of particles) *i.e.*

$$p(\mathbf{x}_{k-1} | d_{k-1} = 1, \mathbf{z}_{1:k-1}) \approx \sum_{n=1}^{N_{p,d}} w_{k-1,d}^n \delta_{\mathbf{x}_{k-1}^n}(\mathbf{x}_{k-1}), \quad (3.68)$$

the unnormalized weights at step k are obtained, according to Eq. (1.94), by

$$\tilde{w}_{k,d}^n = w_{k-1,d}^n \frac{p_c(\mathbf{x}_k^n | \mathbf{x}_{k-1}^n)}{q_c(\mathbf{x}_k^n | \mathbf{x}_{k-1}^n, \mathbf{z}_k)} p(\mathbf{z}_k | d_k = 1, \mathbf{x}_k^n), \quad (3.69)$$

where $q_c(\mathbf{x}_k^n | \mathbf{x}_{k-1}^n, \mathbf{z}_k)$ is any instrumental density (in practice the prior $p_c(\mathbf{x}_k | \mathbf{x}_{k-1})$ is often chosen) and the normalized weights are simply obtained through a normalization. Lastly, the normalization term $p(\mathbf{z}_k | d_k = 1, \mathbf{z}_{1:k-1})$, which is required to calculate the probability $p(d_k = 1 | \mathbf{z}_{1:k})$, is provided by the following equation:

$$p(\mathbf{z}_k | d_k = 1, \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | d_k = 1, \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | d_k = 1, \mathbf{x}_k) d\mathbf{x}_k. \quad (3.70)$$

This normalization term can be approximated, using the same reasoning as the normalization term $p(\mathbf{z}_k | \tau_b = i, \mathbf{z}_{1:k-1})$ in paragraph 3.2.4.2, by the following estimator:

$$\hat{p}(\mathbf{z}_k | d_k = 1, \mathbf{z}_{1:k-1}) = \frac{1}{C_k} \sum_{n=1}^{N_{p,d}} \tilde{w}_{k,d}^n, \quad (3.71)$$

where

$$C_k = \sum_{n=1}^{N_{p,d}} w_{k-1,d}^n \frac{p_c(\mathbf{x}_k^n | \mathbf{x}_{k-1}^n)}{q_c(\mathbf{x}_k^n | \mathbf{x}_{k-1}^n, \mathbf{z}_k)}. \quad (3.72)$$

Lastly, it remains to calculate the probability $p(d_k = 1 | \mathbf{z}_{1:k})$. Using Bayes rule, it can be rewritten as follows:

$$p(d_k = 1 | \mathbf{z}_{1:k}) = \frac{p(d_k = 1 | \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | d_k = 1, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})}. \quad (3.73)$$

Concerning the calculation of quantities $p(d_k = 1 | \mathbf{z}_{1:k-1})$ and $p(\mathbf{z}_k | \mathbf{z}_{1:k-1})$, it is also possible to marginalize over d_k as in Eq. (3.29) and Eq. (3.30). Then, it comes

$$\hat{p}(d_k = 1 | \mathbf{z}_{1:k-1}) = \hat{p}(d_{k-1} = 1 | \mathbf{z}_{1:k-1}) (1 - P_d), \quad (3.74)$$

and

$$\hat{p}(\mathbf{z}_k | \mathbf{z}_{1:k-1}) = \hat{p}(d_k = 1 | \mathbf{z}_{1:k-1}) \hat{p}(\mathbf{z}_k | d_k = 1, \mathbf{z}_{1:k-1}) + \hat{p}(d_k = 0 | \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | d_k = 0, \mathbf{z}_{1:k-1}), \quad (3.75)$$

where $p(\mathbf{z}_k | d_k = 0, \mathbf{z}_{1:k-1}) = p(\mathbf{z}_k | d_k = 0)$ is the likelihood for the case when no target is present and is obtained by Eq. (2.21). Finally, the probability $p(d_k = 1 | \mathbf{z}_{1:k})$ can be estimated by

$$\hat{p}(d_k = 1 | \mathbf{z}_{1:k+1}) = \frac{\hat{p}(d_k = 1 | \mathbf{z}_{1:k-1}) \hat{p}(\mathbf{z}_k | d_k = 1, \mathbf{z}_{1:k-1})}{\hat{p}(\mathbf{z}_k | \mathbf{z}_{1:k-1})}. \quad (3.76)$$

The algorithm scheme for target disappearance is finally explained by Algorithm 3.2.

Algorithm 3.2 Disappearance Time TBD Particle Filter

Require: $\{w_{k-1,d}^n, \mathbf{x}_{k-1}^n\}_{n=1}^{N_{p,d}}, \hat{p}(d_{k-1} = 1 | \mathbf{z}_{1:k-1})$ at step $k - 1$.

- 1: **for** $n = 1$ to $N_{p,d}$ **do**
- 2: Draw $\mathbf{x}_k^n \sim q(\mathbf{x}_k | d_k = 1, \mathbf{x}_{k-1}^n, \mathbf{z}_k)$.
- 3: Compute unnormalized weight $\tilde{w}_{k,d}$ according to Eq. (3.69).
- 4: **end for**
- 5: Compute C_k according to Eq. (3.72)
- 6: Compute $\hat{p}(\mathbf{z}_k | d_k = 1, \mathbf{z}_{1:k-1})$ according to Eq. (3.71).
- 7: Normalization: $w_{k,d}^n \leftarrow \frac{\tilde{w}_{k,d}^n}{\sum_{l=1}^{N_{p,d}} \tilde{w}_{k,d}^l}$, $n = 1 \dots N_{p,d}$.
- 8: Compute $\hat{p}(\mathbf{z}_{k+1} | \mathbf{z}_{1:k})$ according to Eq. (3.75).
- 9: Compute $\hat{p}(d_k = 1 | \mathbf{z}_{1:k})$ according to Eq. (3.76).
- 10: Compute N_{eff} according to Eq. (1.98).
- 11: **if** $N_{\text{eff}} < N_T$ **then**
- 12: Resample $N_{p,d}$ particles
- 13: Reset weights: $w_{k,d}^n \leftarrow \frac{1}{N_{p,d}}$ $n = 1, \dots, N_{p,d}$
- 14: **end if**

Ensure: $\{w_{k,d}^n, \mathbf{x}_k^n\}_{n=1}^{N_{p,d}}, \hat{p}(d_k = 1 | \mathbf{z}_{1:k})$.

3.4 Combination of particle filters for target appearance and disappearance detection

The filters proposed in section 3.2 and 3.3 can only manage either the target appearance or the target disappearance whereas in a TBD perspective it should be desired to manage both the appearance and the disappearance. Therefore, we propose in the sequel to combine the two filters by adding an additional detection stage. As long as no target has been detected, Algorithm 3.1 or B.1 is applied to search for a target appearance. At each step, the target detection is performed as in Chapter 2 by comparing the probability $p(b_k = 1 | \mathbf{z}_{1:k})$ to a given probability P_{init} : if at time step k , $p(b_k = 1 | \mathbf{z}_{1:k}) \geq P_{\text{init}}$, then a target is declared present, and $N_{p,d}$ particles are resampled from the mixture

$$\hat{p}(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k}) = \sum_{i \in I_k} \hat{\alpha}_i \hat{p}(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k}),$$

in order to initialize the disappearance particle filter $\{1/N_{p,d}, \mathbf{x}_k^n\}_{n=1}^{N_{p,d}}$ with $\hat{p}(d_k = 1 | \mathbf{z}_{1:k}) = 1$. This new particle filter is based on Algorithm 3.2 in order to detect the target disappearance time. In the same manner, at each step the probability $p(d_k = 1 | \mathbf{z}_{1:k})$ is

compared to a given probability P_{death} . If $p(d_k = 1 \mid \mathbf{z}_{1:k}) \leq P_{death}$, target disappearance is declared and a new filter for target appearance detection is then initialized.

Of course, target disappearance might be erroneously declared. Therefore, if the corresponding tracking filter was simply deleted and a new one created to detect a target appearance, all the information gathered on the target state would be lost. It might be wiser to initialize one mixture component using the information carried by the particles of the time disappearance filter, thus preserving the information gathered by this filter. More precisely, let us assume that at step k_d , target disappearance was declared. Then, instead of initializing the new time appearance filter at the next step (*i.e.* considering $\{\tau_b = k_d + 1\}$), it might be more convenient to consider that a target has appeared at step k_d with $\hat{p}(\mathbf{x}_{k_d} \mid \tau_b = k_d, \mathbf{z}_{1:k_d}) = \hat{p}(\mathbf{x}_{k_d} \mid d_{k_d} = 1, \mathbf{z}_{1:k_d})$ and $\hat{p}(\tau_b = k_d \mid \mathbf{z}_{1:k_d}) = \hat{p}(d_{k_d} = 1 \mid \mathbf{z}_{1:k_d})$. The required number of particles ($N_{p,mix}$ for Algorithm 3.1 and $N_{p,init}$ for Algorithm B.1) can simply be resampled from $\hat{p}(\mathbf{x}_{k_d} \mid d_{k_d} = 1, \mathbf{z}_{1:k_d})$. For the next iterations, the procedure is exactly the same as the two proposed algorithms for time appearance detection.

The resulting particle filter is called the Appearance Disappearance Detection (ADD) TBD Particle Filter. It is detailed in Algorithm 3.3.

Algorithm 3.3 ADD TBD Particle Filter

```

1: target_is_detected  $\leftarrow$  false
2: for  $k = 1$  to  $N_{it}$  {where  $N_{it}$  is the number of iterations of the algorithm} do
3:   if target_is_detected = false then
4:     Compute  $p(b_k = 1 \mid \mathbf{z}_{1:k})$  with Algorithm 3.1 or Algorithm B.1.
5:     if  $p(b_k = 1 \mid \mathbf{z}_{1:k}) \geq P_{init}$  then
6:       target_is_detected  $\leftarrow$  true
7:       Sample  $N_{p,d}$  particles from  $\hat{p}(\mathbf{x}_k \mid b_k = 1, \mathbf{z}_{1:k})$  to initialize a particle filter for
         Algorithm 3.2.
8:       Set  $p(d_k = 1 \mid \mathbf{z}_{1:k}) = 1$  for this filter.
9:     end if
10:  else
11:    Compute  $p(d_k = 1 \mid \mathbf{z}_{1:k})$  with algorithm 3.2.
12:    if  $p(d_k = 1 \mid \mathbf{z}_{1:k}) \leq P_{death}$  then
13:      target_is_detected  $\leftarrow$  false
14:      Sample  $N_{p,mix}$  or  $N_{p,init}$  particles from  $\hat{p}(\mathbf{x}_k \mid d_k = 1, \mathbf{z}_{1:k})$  to initialize a particle
        filter  $\hat{p}(\mathbf{x}_k \mid \tau_b = k, \mathbf{z}_{1:k})$ ,
15:      Set  $p(\tau_b = k \mid \mathbf{z}_{1:k}) = p(d_k = k \mid \mathbf{z}_{1:k})$ .
16:    end if
17:  end if
18: end for

```

3.5 Simulations and results

In this section, we propose to illustrate the performance of the different TBD algorithms proposed in this chapter via Monte Carlo simulation. One of the main objective of this section is to compare the performance with the classic particle filters detailed in Chap. 2 in

order to measure the possible gain when separating the detection of the target appearance and of the target disappearance as explained in introduction.

3.5.1 Scenario

We consider the same scenario as in chapter 2, that is to say a scenario with a number of iterations $N_{it} = 100$ where a target appears at step $k_b = 15$ and disappears at step $k_d = 75$. For each Monte Carlo run, the initialization of the target state for the position and the velocity at step k_b is done according to the birth density $p_b(\cdot)$ defined in section 2.2 (*i.e.* uniform prior over $\mathcal{D} = [r_{min}, r_{max}] \times [\theta_{min}, \theta_{max}]$ for the position and over $[v_{min}, v_{max}] \times [0, 2\pi]$ for the velocity), with the following parameters:

- $r_{min} = 30$ km, $r_{max} = 36$ km, $\theta_{min} = 35^\circ$ and $\theta_{max} = 55^\circ$,
- $v_{min} = 100$ m.s⁻¹ and $v_{max} = 300$ m.s⁻¹.

For the iterations after k_b the target state \mathbf{x}_k (for the position and the velocity) evolves according to Eq. (2.6) with no noise process (*i.e.* uniform linear motion). The time between two consecutive measurement is set to $T_S = 0.3$ s.

The generation of the raw radar data is done as in the previous chapter with $\mathbf{\Gamma} = \mathbf{I}_{N_c}$ (*i.e.* noise samples are assumed independent). The function $\mathbf{h}(\cdot)$ is defined in paragraph 2.3 with the following parameters:

- For the range axis, $B = 1$ MHz, thus providing a range resolution $\Delta_r = 150$ m, and $T_p = 6.67 \times 10^{-5}$ s.
- For the azimuth axis, $N_a = 70$ and $d = \lambda/2$, corresponding to a resolution $\Delta_\theta = 1.45^\circ$.

Finally, for the parameter ρ several values (following the SNR definition provided in paragraph 2.3.2) will be considered in the simulations.

3.5.2 Methodology for the performance evaluation

As in paragraph 2.7.2, we propose to evaluate the performance in two steps:

- In terms of detection, *i.e.* measuring the capability of the filter to effectively detect the target as quickest as possible while ensuring the smallest probability of false alarm.
- and secondly in terms of estimation, *i.e.* estimating the accuracy of the estimator when the TBD particle filter has detected the target.

To this purpose, we propose to use the same methodology as in the previous chapter. In detection, it means measuring the averaged probability of presence $P_{e,k}$ over N_{MC} Monte Carlo runs, the average percent of time t_D when the target is actually detected and t_{bD} the percent of time when it is badly detected (see 2.7.2.2). In estimation, performance is evaluated with the RMSE in position and velocity from Eq. (2.100) and (2.101).

3.5.3 Comparison between the ADD particle filter and the marginalized particle filter

As we stressed in the beginning of the section, the aim is to see the possible gain by using two different filters for the appearance and the disappearance compared to the classic method of the previous chapter. To this purpose, we propose to compare the performance of the following particle filter:

- The first one, denoted as "ADD Filter, $N_{p,mix}$ constant", consists of the ADD TBD particle filter 3.3 where the particle filter used to detect the target appearance is the "Appearance Time TBD Particle Filter" (detailed by Algorithm 3.1), *i.e.* the number of particles per mixture components is constant over time. Concerning the Appearance Time TBD Particle the specific parameters for this filter are chosen as follows: $N_{p,mix} = 1000$, $N_{mix,max} = 5$, corresponding to a number of particle $N_p = 5000$; each component is resampled at each iteration (*i.e.* $\beta = 1$). Concerning the Disappearance particle filter, the number of particle is set to $N_{p,d} = 1500$ and the resampling procedure is also performed at each iteration.
- The second one, denoted as "ADD Filter, $N_{p,mix}$ variable" consists of the same filter as previously except that the particle filter used for the target appearance is the "Resample All Appearance Time TBD Particle Filter" outlined in paragraph 3.2.4.3 and detailed in Appendix B by Algorithm B.1. For this filter the specific parameters are used: $N_{p,init} = 1000$, $N_{p,max} = 5000$, $N_{p,all} = 3000$, $N_{p,min} = 4000$, $\beta_{all} = 0.1$, $\beta_{min} = 0.5$.
- The last one, denoted as "Marginalized s_k ", consists of the Marginalized Particle Filter detailed in the previous chapter by Algorithm 2.3. The specific parameters of this filter are set with the following values: $N_p = 5000$ and $N_{p,c} = 4000$, *i.e.* at each iteration 1000 particles are initialized.

For all the filters, the probability P_b and P_d are set to 0.1. Concerning the instrumental density $q_b(\cdot | \mathbf{z}_k)$, it is chosen as follows:

- In position, the optimal mixture importance density $q(\cdot | \mathbf{z}_k)$ specified in Eq. (2.39) with the following parameters: $P_{fa} = 0.1$, $\delta_r = 2$, $\delta_\theta = 3$, $N_\rho = 5$ and $P_{\mathcal{D}_{k,\gamma}} = 1$ (*i.e.* particle positions are only initialized in the cells above threshold).
- In amplitude, the prior density is used (*i.e.* uniform prior). The interval for the amplitude parameter ρ is set to $[3, 13]$ (in dB).
- For the velocity, the velocity of birth particles is initialized at the next step, see paragraph 2.5.3 for details.

Concerning the continuing case or alive particle, the prior density is used for the filters. Lastly, the probability P_{init} and P_{death} , they are set respectively to 0.9 and 0.2.

Results are provided in Figures 3.4, 3.5 and Table 3.1.

In detection, the figure 3.4 does not show significant differences between the different particle filters, except that the "ADD Filter, $N_{p,mix}$ variable" seems slightly better for very low SNR (3dB) which is corroborated by the percent time of detection which a little

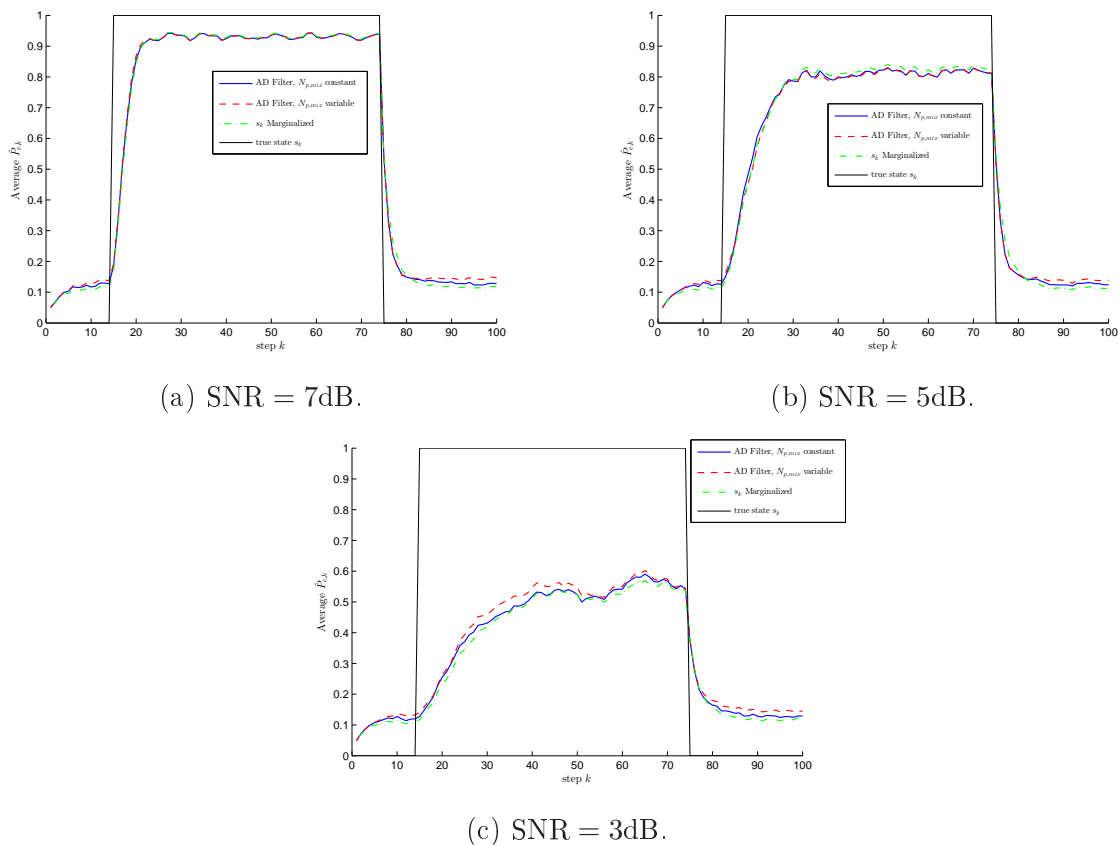
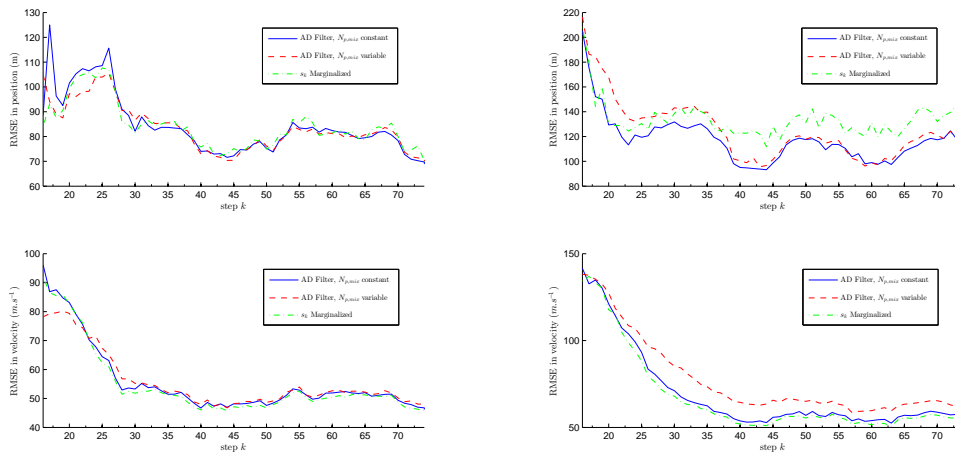


Figure 3.4 – Averaged probability of presence $P_{e,k}$ with different target SNR (7dB, 5dB and 3dB).

better for this filter. In contrast, some important differences can be observed in terms of percent of bad detection t_{bD} that is more important for the " s_k Marginalized" particle filter for all the SNR. This point can be explained by the fact that the " s_k Marginalized" particle filter continues to initialize particle whereas the filter has already detected the target. Therefore, in some situations, especially when the probability of presence $P_{e,k}$ is not close to one, the birth particles may have a non negligible contribution to the target state estimate, even if they are located far away from the actual target position, and thus may lead to a bad detection. However, although the " s_k Marginalized" particle filter has a poorer percent of bad detection t_{bD} , it has a better probability of false alarm.

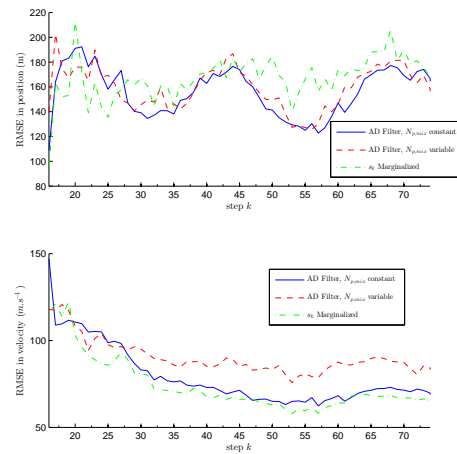
In estimation, above a SNR of 7dB there is no difference between the different particle filters. By cons, from 5 dB and below, the ADD particle filter both for " $N_{p,mix}$ constant" and " $N_{p,mix}$ variable" provide better performance for the estimation of the position. Again, it can be explained by the fact that the " s_k Marginalized" particle filter initializes particles even if it has detected the target. In velocity the "ADD Filter, $N_{p,mix}$ constant" and " s_k Marginalized" filters provide quite similar performance while the "ADD Filter, $N_{p,mix}$ variable" is less efficient.

Lastly, in terms of computational time, the "ADD Filter" both for " $N_{p,mix}$ constant" and " $N_{p,mix}$ variable" is faster than the " s_k Marginalized". This is not surprising since most of the computational resources are devoted to the initialization of particles, so as



(a) SNR = 7dB.

(b) SNR = 5dB.



(c) SNR = 3dB.

Figure 3.5 – RMSE in position and in velocity for the proposed particle filters with different target SNR (7dB, 5dB and 3dB).

"ADD Filter" does not initialize particles when the target is detected the computational time is lower than the " s_k Marginalized" that initializes particles whatever the target is detected or not. Besides, the difference becomes lower with low target SNR since the proportion of time where the filters try to detect the target becomes more important. In fact, the gain in computational time is principally made during the period of time where the target is detected by the filter.

In summary, this simulation has allowed to show the pertinence of separating the detection of the target appearance and disappearance with two different filters. Indeed, it allows to reduce the computational time when the target is detected since, in that case, no particles are initialized without degrading the performance in detection and estimation. Besides, performance is better for the "ADD Filter, $N_{p,mix}$ constant" both in detection (in particular for the percent of bad detection t_bD) and in estimation, but at a cost of a

SNR (dB)	7			5			3		
Filter	1	2	3	1	2	3	1	2	3
$P_{fa}^{PF} (\times 10^{-3})$	4.8	6.53	2.71	4.8	6.53	2.71	4.8	6.53	2.71
t_D	93.2%	93.3%	92.9%	81.1%	80.1%	81.1%	42.7%	44%	41.5%
t_{bD}	0.03%	0.04%	0.13%	0.16%	0.20%	0.98%	0.55%	0.85%	1.99%
rel. MC run duration	1.05	1	2.07	1.04	1	1.85	1.07	1	1.41

Table 3.1 – Detection performance and relative averaged MC run duration for the different particle filters used in the simulation for different target SNR. Filter 1 refers to "AD Filter, $N_{p,mix}$ constant", 2 to "AD Filter, $N_{p,mix}$ constant" and 3 to " s_k Marginalized".

slight raise of the probability of false alarm. Furthermore, it seems that the resampling strategy that consider $N_{p,mix}$ variable over time provides worse performance than the one with $N_{p,mix}$ constant. According to us, this conclusion should be taken with caution since only one set of parameters has been tried, thus it may exist a better set of parameter or even an other resampling strategy which is better.

3.6 Conclusion

In this chapter, we presented an alternative approach to the modeling of the monotarget TBD problem. We shown that it is possible to model the monotarget TBD problem as a quickest detection problem in a Bayesian framework both for the target appearance and disappearance.

In the appearance case, we demonstrated, in section 3.2, that the posterior density of the target state can be expanded as a mixture density. Moreover, in section 3.2.4, we proposed several particle filter approximations, one that considers a constant number of particles per mixture component and an other one that allows a variable number of particles. In the same manner, in the disappearance case which is easier than the appearance case, we outlined the theoretical Bayesian filter and a particle filter approximation. Moreover, in section 3.4, in order to detect both the target appearance and disappearance, we proposed a particle filter that combines the two previous particle filters.

Lastly, in section 3.5, a Monte Carlo simulation was performed to compare the novel approach proposed in this chapter with the monotarget classic particle filters detailed in the previous chapter. This simulation has allowed to show the benefit of using two different particle filters for the target appearance and disappearance. Indeed, the simulation has highlighted that initializing particles when the particle filter has converged to the actual target state may disturb the target estimation and as a consequence the performance in estimation. Moreover, it also highlighted that it sensibly increases the computational time without providing significant gain in estimation or detection (except a slightly lower probability of false alarm). Therefore, according to us, this chapter validates the idea of using specific filters for the target appearance or disappearance. In particular, in chapter 5, this idea will be adapted to the multitarget setting.

Chapter 4

Measurement equation and likelihood calculation for Track-Before-Detect applications

4.1 Introduction

This chapter deals with the calculation of the likelihood of the measurement conditionally to the target state in Track-Before-Detect context. Indeed, in section 1.2.1, we explained that a particle filter requires the calculation (if possible not costly) of the likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$. However, in Track-Before-Detect applications this likelihood cannot be computed directly from the measurement equation (2.8) since this latter depends on the target complex amplitude parameters ρ_k and φ_k that are unknown and may fluctuate over time. Therefore, several strategies have been proposed in the literature in order to release the calculation of the likelihood from these unknown parameters. The first ones [RRG05, DRC08, BDV⁺03] consist in working on the squared-modulus of the complex samples. Using such a strategy allows, in some cases, to calculate the likelihood in a simple manner. On the other hand, it leads to some information loss on the target amplitude parameter. In particular, the spatial coherence of the phase, *i.e.* the fact that the phase of the target amplitude takes the same value in all cells, is then lost, inducing a possible performance degradation. This loss was shown in [DRC12] to severely degrade the performance. Thus, in their article, Davey *et al.* [DRC12] have proposed a new strategy that allows to keep all the information provided by the measurement by working on the complex raw radar data \mathbf{z}_k rather than on the squared-modulus. In particular, this solution allows to keep the spatial coherence of the amplitude parameters. However, in their paper they only investigated the *Swerling 0* fluctuation model and the monotarget case.

Therefore, the objective of this chapter is to extend their work both for amplitude fluctuations of type *Swerling 0*, 1 and 3 and for the multitarget case.

This chapter is organized as follows. In section 4.2 we present the state and measurement models. Then in section 4.3 we present solutions for the likelihood computation from complex and squared modulus measurements. In section 4.4 we derive, when possible, closed forms for the likelihood with *Swerling* fluctuations of type 0, 1 and 3 in the

monotarget and multitarget cases; when not possible, we propose approximations to alleviate the computational time. Finally in section 4.5 we present simulation results that show the gain both in detection and in estimation of the complex measurement method over the squared modulus method.

4.2 Problem Formulation

The measurement model corresponding to the output signal \mathbf{z}_k was presented at the end of the "radar signal processing stage" in chapter 1 section 1.1.8 (see also Figure 1.1). If at a given time index k , N_k targets are present, the output signal (or raw radar data) \mathbf{z}_k is provided by the following equation:

$$\mathbf{z}_k = \sum_{i=1}^{N_k} \rho_{k,i} e^{j\varphi_{k,i}} \mathbf{h}(\mathbf{x}_{k,i}) + \mathbf{n}_k, \quad (4.1)$$

where:

- $\mathbf{h}(\mathbf{x}_{k,i})$ represents the possibly multidimensional ambiguity function of the i^{th} target centered on the target state $\mathbf{x}_{k,i}$. For the sake of simplicity, $\mathbf{h}(\mathbf{x}_{k,i})$ will be denoted $\mathbf{h}_{k,i}$ in the sequel.
- \mathbf{n}_k is a zero mean circular complex Gaussian vector with covariance matrix $\mathbf{\Gamma}$.
- $\varphi_{k,i}$ and $\rho_{k,i}$ are respectively the phase and the modulus of the i^{th} target complex amplitude. All variables $\varphi_{k,1:N_k}$ and $\rho_{k,1:N_k}$ are supposed mutually independent, and independent from \mathbf{n}_k and $\mathbf{x}_{k,1:N_k}$.

Each phase $\varphi_{k,i}$ is supposed to be unknown and uniformly distributed over the interval $[0, 2\pi)$ at each time step k . On the other hand, each modulus $\rho_{k,i}$ is assumed to be drawn from a generic density

$$\rho_{k,i} \sim p_{\vartheta_i}(\rho_k), \quad \text{with } \rho_k \in \mathbb{R}_{\geq 0}, \quad (4.2)$$

where ϑ_i is an unknown static parameter. Note here that these amplitude parameters depend on the time instant k , due to the temporal fluctuation of the target amplitude. The *Swerling* models are convenient in radar to statistically model these amplitude fluctuations over time. The *Swerling* 0 model corresponds to a constant amplitude modulus (*i.e.* no temporal fluctuation); the *Swerling* 1 and 3 models consider slow fluctuations (*i.e.* the modulus fluctuates from burst to burst, where a burst corresponds to a train of pulses, but it is constant from pulse to pulse) respectively modeled by a Rayleigh distribution and a chi-square distribution with four degrees of freedom. Lastly, the *Swerling* 2 and 4 models consider respectively the same fluctuation densities as the *Swerling* 1 and 3 but with fast fluctuations (*i.e.* from pulse to pulse). We do not consider these latter models in this chapter and thus focus only on the *Swerling* fluctuation models of type 0, 1 and 3. The likelihood calculation for these models will be detailed in section 4.4.

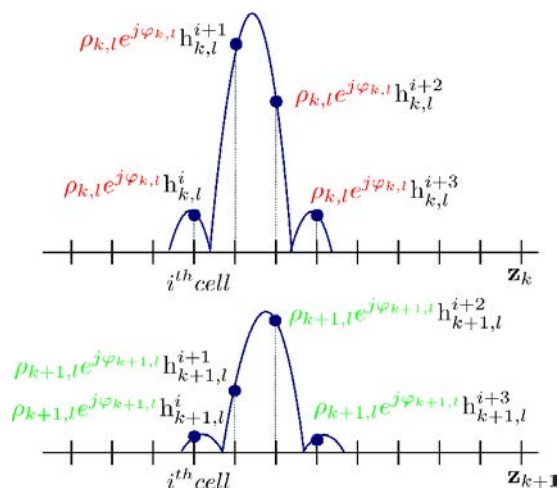


Figure 4.1 – Received signal (noise-free) corresponding to the l^{th} target at two adjacent time steps k and $k + 1$, where dots represent the corresponding measured samples. $\rho_{k,l}$ and $\varphi_{k,l}$ are the same for all cells of \mathbf{z}_k (we denote this feature *spatial coherence*) but their values change independently and randomly over time; there is no *temporal coherence* from step k to step $k + 1$.

4.2.1 Temporal coherence versus spatial coherence

An important point to be stressed here is that variables $\rho_{k,1:N_k}$ and $\varphi_{k,1:N_k}$ are *spatially coherent*: this means that the complex target amplitude $\rho_{k,i} e^{j\varphi_{k,i}}$ is identical over all cells where the signal ambiguity function spreads. Taking into account this information can really increase the performance of the Track-Before-Detect algorithms [DRC12]. On the contrary, these variables $\rho_{k,1:N_k}$ and $\varphi_{k,1:N_k}$ are not assumed coherent over time, *i.e.* from time sample k to $k + 1$, amplitude parameters fluctuate independently. As a consequence, no information can be gathered over time on these parameters. These dependencies are illustrated in Figure 4.1.

4.2.2 State of the art

The objective of this chapter is therefore to compute the measurement likelihood in a general multitarget TBD context with unknown fluctuating amplitude parameters. Several solutions have been provided in the literature, mainly in a monotarget setting. The first solution that deals with the unknown complex amplitude considers a monotarget setting and consists in working on the squared modulus of the complex signal [BDV⁺03, SB01, RAG04, RRG05, DRC08]. For such a radical solution that completely discards the phase dependency, two strategies can be considered to deal with the modulus fluctuation. The first one consists in marginalizing the whole likelihood with respect to the density of the modulus fluctuation [DRC08]. In practice, this leads to intractable integrals that must be approximated numerically. The second strategy consists in marginalizing independently the likelihood in each cell [RRG05]. The advantage of this heuristic second solution is that a closed form can be obtained for fluctuations of type *Swerling* 0, 1 and 3 [MB08]. On the other hand the spatial coherence of the modulus, *i.e.* the fact that the modulus of the target amplitude takes the same value in all cells, is then lost,

		Swerling 0	Swerling 1	Swerling 3
Complex measurement	Monotarget	Eq.(4.15) and [DRC12]		
	Multitarget			
Squared modulus	Monotarget, non coherent	Eq.(4.25) and [MB08]	Eq.(4.49) and [MB08]	Eq.(4.52), Eq.(4.51) and [MB08]
	Multitarget, non coherent		Eq.(4.49) and [BDV ⁺ 03]	
	Monotarget, coherent	Eq. (4.4.2.1) and [DRC12]	Eq. (4.4.2.1) and [DRC12]	(4.4.2.1), and [DRC12]
	Multitarget, coherent			

Table 4.1 – Summary of the state of the art for the likelihood computation with different data types (complex measurements or squared modulus), different Swerling models (type 0, 1 and 3) and different number of targets (mono or multitarget). The squared modulus measurement are splitted between coherent computation and non coherent computation. Each filled cell contains the reference of the equation in this chapter that provides the expression for the likelihood, and the citation of the corresponding paper.

inducing a possible degradation of performance. Note also that the spatial coherence of the phase is lost for both strategies. This loss was shown in [DRC12] to severely degrade the performance.

To avoid this last drawback, Davey *et al.* [DRC12] have proposed a new strategy that allows preserving the spatial coherence of the phase. Their solution consists in directly working on complex measurements and marginalizing the complex likelihood of the whole data over the phase. It provides better performance than solutions based on squared modulus. However, they mainly investigated the case where the modulus does not fluctuate (*i.e.* *Swerling* 0 case); for modulus fluctuations, they only provide a general marginalization formula. One of the contributions of this chapter is an extension of their work with complex measurements to fluctuations of type *Swerling* 1 and 3; we show that closed-forms can be obtained for the monotarget likelihood in both cases.

Furthermore, all the previously discussed strategies with squared modulus or complex measurements were proposed in a monotarget setting. In fact, to our knowledge, the multitarget case has not been investigated in the literature, except for the *Swerling* 1 case with squared modulus [BDV⁺03]. Thus, another contribution of this chapter consists in considering the multitarget case both with squared modulus and complex measurements. In the complex measurement case, we provide a closed-form expression for the likelihood in the *Swerling* 1 case, and we propose in the other fluctuation cases some approximations to alleviate the computational cost. In the squared modulus case, we show that, as

soon as at least two targets are present, all phase dependencies cannot be removed from the likelihood; in fact taking the squared modulus permits to remove only one phase, so that other phases must be marginalized. In that latter case, we also propose some approximations in order to reduce the computational complexity.

The Table 4.1 summarizes the state of the art for the likelihood computation with different data types (complex measurements or squared modulus), different Swerling models (type 0, 1 and 3) and different numbers of target (mono or multitarget). The aim of this chapter is to fill any empty cell in this table.

4.3 Likelihood calculation for Track-Before-Detect applications

In this section, we propose to develop the different methods presented in the previous paragraph "State of Art". We first start by explaining how to calculate the likelihood for Track-Before-Detect applications with the complex measurement and then with the squared-modulus.

4.3.1 Likelihood computation with complex measurements

4.3.1.1 Likelihood from the measurement equation

As previously pointed out, the likelihood $p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k})$, *i. e.* the likelihood of the observation conditionally to the target states cannot be calculated directly from the measurement equation (4.1) since it depends on phase and amplitude parameters $\varphi_{k,1:N_k}$ and $\rho_{k,1:N_k}$ that are unknown and not temporally coherent. Nevertheless, from this equation, it is possible to calculate the likelihood of the measurement \mathbf{z}_k conditionally to the states $\mathbf{x}_{k,1:N_k}$ and the amplitudes parameters $\varphi_{k,1:N_k}$ and $\rho_{k,1:N_k}$, *i. e.* $p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k})$. Indeed, since the noise \mathbf{n}_k is complex Gaussian, the corresponding density is then a complex

Gaussian density with mean $\boldsymbol{\mu}_k = \sum_{i=1}^{N_k} \rho_{k,i} e^{j\varphi_{k,i}} \mathbf{h}_{k,i}$ and covariance matrix $\boldsymbol{\Gamma}$:

$$p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k}) = \frac{1}{\pi^{N_c} \det(\boldsymbol{\Gamma})} \exp \left\{ -(\mathbf{z}_k - \boldsymbol{\mu}_k)^H \boldsymbol{\Gamma}^{-1} (\mathbf{z}_k - \boldsymbol{\mu}_k) \right\}. \quad (4.3)$$

Then by developing Eq. (4.3), it comes

$$p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k}) = \frac{\exp \left\{ -\mathbf{z}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{z}_k \right\}}{\pi^{N_c} \det(\boldsymbol{\Gamma})} \times \exp \left\{ -\sum_{i=1}^{N_k} \rho_{k,i}^2 \mathbf{h}_{k,i}^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_{k,i} + \sum_{i=1}^{N_k} 2\rho_{k,i} |\mathbf{h}_{k,i}^H \boldsymbol{\Gamma}^{-1} \mathbf{z}_k| \cos(\varphi_{k,i} - \xi_{k,i}) - \sum_{i=1}^{N_k} \sum_{l=i+1}^{N_k} 2\rho_{k,i} \rho_{k,l} |\mathbf{h}_{k,i}^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_{k,l}| \cos(\varphi_{k,i} - \varphi_{k,l} - \phi_{k,il}) \right\}, \quad (4.4)$$

where $\xi_{k,i} = \arg(\mathbf{h}_{k,i}^H \boldsymbol{\Gamma}^{-1} \mathbf{z}_k)$ and $\phi_{k,il} = \arg(\mathbf{h}_{k,i}^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_{k,l})$.

4.3.1.2 Marginalizing over the phase and modulus parameters

Since parameters $\varphi_{k,1:N_k}$ and $\rho_{k,1:N_k}$ are assumed to be random variables, it is possible to write the joint likelihood of the measurement \mathbf{z}_k and the amplitude parameters conditionally to the target states $\mathbf{x}_{k,1:N_k}$, that is given by

$$p_{\vartheta_{1:N_k}}(\varphi_{k,1:N_k}, \rho_{k,1:N_k}, \mathbf{z}_k \mid \mathbf{x}_{k,1:N_k}) = p_{\vartheta_{1:N_k}}(\varphi_{k,1:N_k}, \rho_{k,1:N_k} \mid \mathbf{x}_{k,1:N_k}) \times p(\mathbf{z}_k \mid \mathbf{x}_{k,1:N_k}, \varphi_{k,1:N_k}, \rho_{k,1:N_k}). \quad (4.5)$$

From the hypotheses in the measurement model, the density of phases $\varphi_{k,1:N_k}$ and amplitudes $\rho_{k,1:N_k}$ $p_{\vartheta_{1:N_k}}(\varphi_{k,1:N_k}, \rho_{k,1:N_k} \mid \mathbf{x}_{k,1:N_k})$ does not depend on $\mathbf{x}_{k,1:N_k}$ and expands as follows

$$p_{\vartheta_{1:N_k}}(\varphi_{k,1:N_k}, \rho_{k,1:N_k} \mid \mathbf{x}_{k,1:N_k}) = p(\varphi_{k,1:N_k}) p_{\vartheta_{1:N_k}}(\rho_{k,1:N_k}) \quad (4.6)$$

$$= \prod_{i=1}^{N_k} p(\varphi_{k,i}) p_{\vartheta_i}(\rho_{k,i}). \quad (4.7)$$

Finally the likelihood $p_{\vartheta_{1:N_k}}(\mathbf{z}_k \mid \mathbf{x}_{k,1:N_k})$ can be obtained by marginalizing Eq. (4.5) over parameters $\rho_{k,1:N_k}$ and $\varphi_{k,1:N_k}$:

$$p_{\vartheta_{1:N_k}}(\mathbf{z}_k \mid \mathbf{x}_{k,1:N_k}) = \int \cdots \int_{\mathbb{R}_{\geq 0}^{N_k} \times [0, 2\pi)^{N_k}} p(\mathbf{z}_k \mid \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k}) \times p(\varphi_{k,1:N_k}) p_{\vartheta_{1:N_k}}(\rho_{k,1:N_k}) d\varphi_{k,1:N_k} d\rho_{k,1:N_k}. \quad (4.8)$$

First, notice that the spatial coherence is preserved in this formulation thanks to the marginalization. However, this likelihood expression still depends on the static parameters $\vartheta_{1:N_k}$ that have been supposed unknown. It is possible to deal with these static parameters by adding them in the state vector $\mathbf{x}_{k,1:N_k}$ as explained in paragraph 4.3.1.3.

Then, note that most of the Bayesian TBD algorithms require either to calculate the likelihood ratio between the likelihood of the observation conditionally to the state vector and the likelihood of the observation conditionally to the event that no target is present (*i.e.* $N_k = 0$); or the likelihood can be calculated up to a constant (*e.g.* particle filters). As a consequence, the constant term in Eq. (4.4), given by

$$p(\mathbf{z}_k \mid N_k = 0) = \frac{1}{\pi^{N_c} \det(\mathbf{\Gamma})} \exp\{-\mathbf{z}_k \mathbf{\Gamma}^{-1} \mathbf{z}_k\}, \quad (4.9)$$

which is the likelihood conditionally to the event that no target is present, does not need to be calculated, providing directly the likelihood ratio or the likelihood up to this constant. Note that, for the sake of clarity, this constant term will be always discarded in the likelihood expression provided in the rest of the chapter.

At last, an important point is that Eq. (4.8) is often intractable, even for two targets, and must then be computed numerically. However, in section 4.4.1.2, it will be shown that a closed-form can be obtained for the particular *Swerling 1* fluctuation model. For other fluctuation models, the numerical implementation implies the evaluation of multiple

integrals over several parameters and the computational cost may be rapidly prohibitive in the multitarget case. Fortunately, target contributions can in many cases be separated so that the multitarget likelihood becomes equal to the product of monotarget likelihoods that can be computed in closed-form. This separation arises when targets do not interact in the likelihood expression (4.4). This can be translated mathematically by the following condition:

$$|\mathbf{h}_{k,u}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,v}| \approx 0, \text{ for any } (u, v), u \neq v, \quad (4.10)$$

that allows to remove all cross terms in Eq. (4.4). In practice, this hypothesis may arise for instance when $\mathbf{\Gamma} = \mathbf{I}_{N_c}$ and targets are far away from each other. Indeed, for each target the ambiguity vector $\mathbf{h}_{k,i}$ has only significant values in a few number of cells around the target location and can be assumed equal to zero elsewhere, so that the ambiguity vector can be truncated as explained in paragraph 2.4.2.3. Therefore, the scalar product between ambiguity function $\mathbf{h}_{k,u}$ and $\mathbf{h}_{k,v}$ is approximately equal to zero for sufficiently distant targets. Note however that when $\mathbf{\Gamma} \neq \mathbf{I}_{N_c}$, condition (4.10) cannot be verified as straightforwardly and should thus be carefully checked, even for distant targets. Indeed, the inner product induced by matrix $\mathbf{\Gamma}^{-1}$ may mix the components of $\mathbf{h}_{k,u}$ and $\mathbf{h}_{k,v}$ even when they are located far apart from each other.

Finally, the expression of the likelihood $p_{\vartheta_{1:N_k}}(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k})$ becomes under condition (4.10):

$$\begin{aligned} p_{\vartheta_{1:N_k}}(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k}) \\ \propto \exp \left\{ -\sum_{i=1}^{N_k} \rho_{k,i}^2 \mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,i} + \sum_{i=1}^{N_k} 2\rho_{k,i} |\mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{z}_k| \cos(\varphi_{k,i} - \xi_{k,i}) \right\} \\ \propto \prod_{i=1}^{N_k} \exp \left\{ -\rho_{k,i}^2 \mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,i} + 2\rho_{k,i} |\mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{z}_k| \cos(\varphi_{k,i} - \xi_{k,i}) \right\}, \end{aligned} \quad (4.11)$$

where the i^{th} term of the product, denoted by

$$\Xi_{\mathbf{z}_k, \mathbf{x}_{k,i}}(\rho_{k,i}, \varphi_{k,i}) = \exp \left\{ -\rho_{k,i}^2 \mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,i} + 2\rho_{k,i} |\mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{z}_k| \cos(\varphi_{k,i} - \xi_{k,i}) \right\}, \quad (4.12)$$

only depends on parameters $\rho_{k,i}$ and $\varphi_{k,i}$. As variables $\rho_{k,1:N_k}$ and $\varphi_{k,1:N_k}$ are independent, the joint density (4.8) then simply becomes

$$p_{\vartheta_{1:N_k}}(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}) \propto \prod_{i=1}^{N_k} \int_0^{+\infty} \int_0^{2\pi} \Xi_{\mathbf{z}_k, \mathbf{x}_{k,i}}(\rho_{k,i}, \varphi_{k,i}) p(\varphi_{k,i}) p_{\vartheta_i}(\rho_{k,i}) d\varphi_{k,i} d\rho_{k,i}. \quad (4.13)$$

Thus, everything happens as if each target is processed separately. This drastically alleviates the computational complexity of integral (4.8) and allows processing distant targets with parallel filters as we will see in chapter 5 which is dedicated to the Bayesian Multitarget Filter in Track-Before-Detect context. Of course, when condition (4.10) is not verified, this simplification can be done only for separated targets, while targets that cannot be separated must be processed by the same filter.

In the monotarget case, integral (4.8) becomes

$$p_{\vartheta}(\mathbf{z}_k | \mathbf{x}_k) \propto \int_0^{+\infty} \int_0^{2\pi} p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}, \varphi_k, \rho_k) p(\varphi_k) p(\rho_k) d\varphi_k d\rho_k. \quad (4.14)$$

Davey *et al.* [DRC12] have shown in this particular monotarget case that the marginalization can be done over the phase φ_k , providing

$$\begin{aligned} p_{\vartheta}(\mathbf{z}_k | \mathbf{x}_k, \rho_k) &\propto \int_0^{2\pi} p(\mathbf{z}_k | \mathbf{x}_k, \varphi_k, \rho_k) p(\varphi_k) d\varphi_k, \\ &\propto \exp\{-\rho_k^2 \mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k\} \text{I}_0(2\rho_k |\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k|), \end{aligned} \quad (4.15)$$

where I_0 is the modified Bessel function of the first kind, *i.e.*

$$\text{I}_0(x) = \sum_{l=0}^{+\infty} \frac{\left(\frac{x}{2}\right)^{2l}}{(l!)^2}. \quad (4.16)$$

Then, the likelihood is obtained by integrating (4.15) over the generic density $p_{\vartheta}(\rho_k)$ that depends on the fluctuation model considered.

4.3.1.3 Dealing with the unknown static parameters of the modulus fluctuation densities

In a Bayesian perspective, a possible solution to deal with these parameters consists in choosing a prior density for each parameter ϑ_i (for instance a uniform prior over a given interval $[\vartheta_{i,min}, \vartheta_{i,max}]$, where $\vartheta_{i,min}$ and $\vartheta_{i,max}$ are provided) and then in marginalizing also over these parameters. Note that in a filtering perspective the likelihood $p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k})$ is calculated at each iteration step k . It might then be convenient to use the fact that the parameters $\vartheta_{1:N_k}$ are constant in order to estimate them over time. In this perspective, the problem of state-space models with unknown static parameters has been widely studied in the literature [Kit98, Sto02, ADST04].

A popular solution consists in explicitly introducing artificial dynamics on the static parameters [ADST04] and considering them as components of the state vector. Thus, the new state vector for each target becomes $\mathbf{x}'_{k,i} = [\mathbf{x}_{k,i}^T, \vartheta_{k,i}]^T$ where the evolution of parameter $\vartheta_{k,i}$ is Markovian, *i.e.*:

$$\vartheta_{k,i} = \vartheta_{k-1,i} + \varepsilon_{k,i}, \quad (4.17)$$

with $\varepsilon_{k,i}$ a small Gaussian noise, and $\vartheta_{0,i} \sim p_0(\vartheta)$. Then, since parameters $\vartheta_{k,1:N_k}$ belong to the state vector, they do not need to be marginalized in the likelihood expression (4.8) that becomes:

$$p_{\vartheta_{k,1:N_k}}(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}) = p(\mathbf{z}_k | \mathbf{x}'_{k,1:N_k}). \quad (4.18)$$

Finally, in order to alleviate the notations, we will denote by $\mathbf{x}_{k,1:N_k}$ the state vector containing the parameters $\vartheta_{k,1:N_k}$ (*i.e.* $\mathbf{x}'_{k,1:N_k}$). Thus, in the sequel, all the likelihood expressions $p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k})$ for the *Swerling* models studied in this chapter will be provided with the randomized parameter $\vartheta_{k,1:N_k}$.

4.3.2 Likelihood computation with squared modulus

In the previous section, the exact computation of the likelihood from complex measurements has been presented. In this section, a different approach often considered in the

literature, which consists in working only with the squared modulus of the complex data [RRG05, DRC08, BDV⁺03], is exposed. This approach is interesting in applications where only the squared modulus of the data is available but also because it allows to remove the phase dependency in a monotarget setting. This simplifies in some extent the computations, at the cost of loosing the spatial coherence of the phase. Squared modulus were also considered in an application involving two targets with *Swerling 1* amplitude fluctuations [BDV⁺03]. In this specific application, the spatial coherence of the target amplitude was not considered, thus simplifying the computation at the cost of some information loss. We will derive here the general multitarget likelihood in the squared modulus framework. It differs from expressions obtained in the literature since it does not make any approximation and thus properly takes into account the spatial coherence of the complex amplitude. Moreover we show that the squared modulus approach does not allow in the multitarget setting to remove all phase dependencies. Thus, as with complex measurements, these phase variables must be taken into account, for instance by marginalization.

First, let us assume, as in the literature [DRC12, BDV⁺03, SB01], that the covariance matrix has the following expression $\mathbf{\Gamma} = 2\sigma^2\mathbf{I}_{N_c}$, *i.e.* the complex noise samples \mathbf{n}_k are mutually independent. Note however that, since modulus $\rho_{k,1:N_k}$ and phases $\varphi_{k,1:N_k}$ are random variables and spatially coherent at time k , this hypothesis does not allow to establish that signal samples from z_k^l are independent; these samples are independent only conditionally to variables $\rho_{k,1:N_k}$ and $\varphi_{k,1:N_k}$. Then, with a slight abuse of notation, let us denote by $|\mathbf{z}_k|^2$ the vector of squared modulus of the complex signal : $|\mathbf{z}_k|^2 = [|z_k^1|^2, \dots, |z_k^{N_c}|^2]^T$. Since the noise samples z_k^l are independent conditionally to variables $\rho_{k,1:N_k}$ and $\varphi_{k,1:N_k}$, this property also holds for squared modulus of the noise samples $|z_k^l|^2$, thus allowing to expend the likelihood $p(|\mathbf{z}_k|^2 | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k})$ as follows

$$p(|\mathbf{z}_k|^2 | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k}) = \prod_{l=1}^{N_c} p(|z_k^l|^2 | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k}). \quad (4.19)$$

The desired density $p(|\mathbf{z}_k|^2 | \mathbf{x}_{k,1:N_k})$ can be obtained from $p(|\mathbf{z}_k|^2 | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k})$ exactly in the same way as with complex measurements, by marginalizing over all variables $\rho_{k,1:N_k}$ and $\varphi_{k,1:N_k}$. Remark that the hypothesis of independence is absolutely necessary here to establish Eq.(4.19). The condition $\mathbf{\Gamma} = 2\sigma^2\mathbf{I}_{N_c}$ can be generalized to diagonal covariance matrices, but the case where $\mathbf{\Gamma}$ is not a diagonal matrix is much more complicated even for two coupled variables: in that case, squared modulus samples are correlated, thus leading to distributions with no closed-form, for instance multivariate Rayleigh distribution in the *Swerling 1* case [Mal03]. Note also that in practice, this hypothesis is verified with classic matched filtering in presence of white Gaussian noise and an appropriate sampling rate, but it may not be verified anymore when modifying the reception processing, for instance by applying classic weighting windows such as Hamming, Bartlett, Hann, etc. [Har78] that modify the noise correlation after processing.

Before going further into the computation, we would like to highlight here an interesting property that arises when considering squared modulus of complex data, and that has never been discussed to our knowledge in the literature: although N_k targets are present, providing N_k different and independent random phases $\varphi_{k,1:N_k}$, it is possible to show, by changing the set of parameters, that the density $p(|z_k^l|^2 | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi_{k,1:N_k})$ effectively depends only on $N_k - 1$ phase variables. Indeed the variable $|z_k^l|^2$ can be defined up to

an arbitrary phase φ' since $|z_k^l|^2 = |e^{j\varphi'} z_k^l|^2$, and we can write for instance

$$|z_k^l|^2 = \left| \rho_{k,1} h_{k,1}^l + \sum_{i=2}^{N_k} \rho_{k,i} e^{j\varphi'_{k,i}} h_{k,i}^l + n_k^l \right|^2 \quad (4.20)$$

where all $n_k^l = n_k^l e^{-j\varphi_{k,1}}$ are still independent circular symmetric complex Gaussian noise samples, and phases $\varphi'_{k,i} = \varphi_{k,i} - \varphi_{k,1}$ are still uniform variables distributed over the interval $[0, 2\pi)$. Thus, $|z_k^l|^2$ only depends on $N_k - 1$ phase variables. Therefore, taking the squared modulus of the complex signal leads to drop out the dependence of one and only one phase. As a consequence, in a monotarget setting the density of $|z_k^l|^2$ does not depend any longer on the phase φ_k but only on the modulus; this is one of the main reasons to use such a technique for the TBD monotarget algorithms. On the contrary, in the multitarget setting, taking the squared modulus does not remove all dependencies on the phases! This dependency remains present through coherent summations of the target contributions in each cell. Discarding it may lead to losing all the information provided by the spatial coherence of the phase variables.

Conditionally to variables $\mathbf{x}_{k,1:N_k}$, $\rho_{k,1:N_k}$ and $\varphi'_{k,2:N_k}$, each sample $\frac{|z_k^l|^2}{\sigma^2}$ follows a non central chi-square distribution with two degrees of freedom; indeed it corresponds to the sum of the squares of two non-centered Gaussian variables. The density $p(|z_k^l|^2 | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi'_{k,2:N_k})$ is thus provided by:

$$p(|z_k^l|^2 | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi'_{k,2:N_k}) = \frac{1}{2\sigma^2} \exp \left\{ -\frac{|z_k^l|^2}{2\sigma^2} - \frac{\gamma^l(\varphi'_{k,2:N_k}, \rho_{k,1:N_k})}{2} \right\} I_0 \left(\sqrt{\frac{\gamma^l(\varphi'_{k,2:N_k}, \rho_{k,1:N_k}) |z_k^l|^2}{\sigma^2}} \right), \quad (4.21)$$

where $\gamma^l(\varphi'_{k,2:N_k}, \rho_{k,1:N_k})$ is the non centrality parameter equal to

$$\gamma^l(\varphi'_{k,2:N_k}, \rho_{k,1:N_k}) = \frac{\left| \rho_{k,1} h_{k,1}^l + \sum_{i=2}^{N_k} \rho_{k,i} e^{j\varphi'_{k,i}} h_{k,i}^l \right|^2}{\sigma^2}. \quad (4.22)$$

At this step, mono and multitarget cases are different, and we will consider them separately in the following. Finally, note that, as with complex measurements, the likelihood can be computed up to a constant. Therefore terms $\frac{1}{2\sigma^2} \exp \left\{ -\frac{|z_k^l|^2}{2\sigma^2} \right\}$ will be discarded in the rest of the paper.

4.3.2.1 The monotarget case

In a monotarget setting, the non-centrality parameter in each cell becomes

$$\gamma^l(\rho_k) = \frac{\rho_k^2 |h_k^l|^2}{\sigma^2} \quad (4.23)$$

and does not depend on φ_k . The joint likelihood can then be obtained by marginalizing Eq.(4.19) over the parameter ρ_k :

$$p(|\mathbf{z}_k|^2 | \mathbf{x}_k) = \int_0^\infty \prod_{l=1}^{N_c} p(|z_k^l|^2 | \mathbf{x}_k, \rho_k) p_{\vartheta_k}(\rho_k) d\rho_k, \quad (4.24)$$

where $p_{\vartheta_k}(\rho_k)$ is the density for the parameter ρ_k . As for complex measurements, this marginalization allows preserving the spatial coherence of the parameter ρ_k . Since integral (4.24) is, to our knowledge, intractable for *Swerling* fluctuations models of type 1 and 3 (it consists in integrating N_c Bessel functions), it must be in that case approximated numerically. Note that we do not consider the *Swerling* 0 model here, since the integration over the density $p_{\vartheta_k}(\rho_k)$ just consists in replacing the parameter ρ_k by a constant.

To avoid performing a numerical approximation, an heuristic solution was proposed by Rutten *et al.* [RRG05] that consists in first marginalizing independently each sample of the signal $|z_k|^2$ according to $p_{\vartheta}(\rho_k)$, *i.e.*

$$p(|z_k^l|^2 | \mathbf{x}_k) = \int_0^\infty p(|z_k^l|^2 | \rho_k, \mathbf{x}_k) p_{\vartheta_k}(\rho_k) d\rho_k. \quad (4.25)$$

Clearly the spatial coherence of ρ_k is lost since the integration is performed independently for each measurement sample and not over the whole measurement vector. On the other hand, the calculation of integral (4.25) can be done analytically for *Swerling* fluctuation models of type 1 and 3, leading to simple closed-forms expressions. Then, the whole likelihood is calculated by assuming that samples $|z_k^1|^2, \dots, |z_k^{N_c}|^2$ are independent. Under that assumption,

$$p(|\mathbf{z}_k|^2 | \mathbf{x}_k) = \prod_{l=1}^{N_c} p(|z_k^l|^2 | \mathbf{x}_k), \quad (4.26)$$

Recall that this is not true in general because of the spatial coherence of random variable ρ_k that tends to establish a dependency between neighbour measurement samples. Thus, rigorously, measurement samples $|z_k^l|^2$ are independent conditionally to the state \mathbf{x}_k and the parameters ρ_k and φ_k , but they are not generally independent conditionally to the state \mathbf{x}_k only. In other words, if we know the values of the state \mathbf{x}_k and the parameters ρ_k and φ_k , then we know how the state and these parameters influence the different measurement samples, so that the only unknown comes from the independent noise samples. When we only know the state \mathbf{x}_k but not the parameters ρ_k and φ_k , then we do not know exactly the link between the different measurement samples, and they cannot be assumed independent anymore.

It is finally interesting to observe here that, if a similar assumption was used in the complex measurement case (*i.e.* independence of the amplitude parameters from sample to sample, which resorts to removing the spatial coherence of the amplitude parameter), then the likelihood for the complex measurement (without spatial coherence) would be equal to the product of the sample likelihood for each complex sample and become identical to the likelihood with squared modulus (still without spatial coherence). This comes from the fact that when computing the likelihood for one single sample, the phase parameter does not matter, or, in other words, does not provide any information.

4.3.2.2 The multitarget case

As previously discussed, in the multitarget case the parameter γ^l ($\varphi'_{k,2:N_k}, \rho_{k,1:N_k}$) still depends on the N_{k-1} phase variables $\varphi'_{k,2:N_k}$. The likelihood must thus be obtained by marginalization over modulus $\rho_{k,1:N_k}$ and phases $\varphi'_{k,2:N_k}$:

$$p(|\mathbf{z}_k|^2 | \mathbf{x}_{k,1:N_k}) = \int \cdots \int_{\mathbb{R}_{\geq 0}^{N_k} \times [0, 2\pi)^{N_k-1}} \prod_{l=1}^{N_c} p(|z_k^l|^2 | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi'_{k,2:N_k}) p_{\vartheta_{k,1:N_k}}(\rho_{k,1:N_k}) \times p(\varphi'_{k,2:N_k}) d\rho_{k,1:N_k} d\varphi'_{k,2:N_k}. \quad (4.27)$$

As in the monotarget case, this expression is to our knowledge intractable. The same heuristic as in the monotarget case can be used: first marginalizing independently each sample from $\varphi'_{k,2:N_k}$ and $\rho_{k,1:N_k}$ as in (4.25), providing

$$p(|z_k^l|^2 | \mathbf{x}_{k,1:N_k}) = \int \cdots \int_{\mathbb{R}_{\geq 0}^{N_k} \times [0, 2\pi)^{N_k-1}} p(|z_k^l|^2 | \mathbf{x}_{k,1:N_k}, \rho_{k,1:N_k}, \varphi'_{k,2:N_k}) p_{\vartheta_{k,1:N_k}}(\rho_{k,1:N_k}) p(\varphi'_{k,2:N_k}) d\rho_{k,1:N_k} d\varphi'_{k,2:N_k}, \quad (4.28)$$

and then approximating the whole likelihood as in Eq. (4.26). Note, however, that contrary to the monotarget case there is in general no closed-form for the integral (4.28), so that numerical integration must still be performed.

Finally, as with complex measurements, target contributions can often be separated so that the multitarget likelihood then resorts to a product of monotarget likelihoods. This separation is obtained under the condition $h_{k,i}^l h_{k,j}^l \approx 0$, for any $i, j, i \neq j$ that allows to eliminate all cross terms in Eq. (4.22).

4.4 Likelihood computation for Swerling models

In this section, we will derive the measurement likelihood with three different *Swerling* models: *Swerling 0*, *Swerling 1* and *Swerling 3*. For each model, first the case of complex measurements will be considered and second the case of squared modulus measurements. Whenever closed-forms are not obtainable, we will propose approximations that allow to compute the likelihood at a lower computational cost.

4.4.1 Complex measurements

4.4.1.1 Swerling 0 case

The modulus $\rho_{k,i}$ of each target is assumed constant and equal to an unknown constant ρ_i . This corresponds to the following generic fluctuation density for each target:

$$p_{\vartheta_i}(\rho_{k,i}) = \delta_{\vartheta_i}(\rho_{k,i}), \quad (4.29)$$

where $\delta_{\vartheta_i}(\cdot)$ is the delta mass Dirac function at point ϑ_i and where the parameter ϑ_i is thus equal to ρ_i . Whereas parameters $\rho_{1:N_k}$ are unknown, they can be added to the state vector and treated exactly as the other state parameters, as it has been explained

in paragraph 4.3.1.3. Moreover, with this particular fluctuation density, the integration over variables $\rho_{k,1:N_k}$ in Eq. (4.8) just consists in substituting each variable $\rho_{k,i}$ by the constant parameter ρ_i . Since this parameter ρ_i is a priori unknown, it is then replaced by the dynamical parameter $\rho_{k,i}$ as explained in section 4.3.1.3) (note here the slight abuse of notation since $\rho_{k,i}$ refers to the parameter ρ_i evolving over time and not to the value of the amplitude modulus at step k). Finally, the integral (4.8) that corresponds to the complex measurement likelihood must just be computed over parameters $\varphi_{k,1:N_k}$. In the general multitarget case, this integral is, according to our knowledge, intractable and must be approximated except for the particular single target case. A first solution consists in calculating numerically the integral over the domain $[0, 2\pi)^{N_k}$ but this may become rapidly computationally demanding. Thus, we propose to replace the intractable likelihood by its Laplace approximation that has been already successfully used in particle filter application [MBQLG11]. Let $\mathbf{H}_k = [\rho_1 \mathbf{h}_{k,1}, \dots, \rho_{N_k} \mathbf{h}_{k,N_k}]$ and let $\Psi_k = \Psi_k(\varphi_{k,1:N_k}) = [e^{j\varphi_{k,1}}, \dots, e^{j\varphi_{k,N_k}}]^T$. Equation (4.8) can be rewritten as follows:

$$p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}) \propto \int \cdots \int_{[0,2\pi)^{N_k}} \exp \left\{ \Upsilon_{\mathbf{x}_{k,1:N_k}}(\varphi_{k,1:N_k}) \right\} d\varphi_{k,1:N_k}. \quad (4.30)$$

where

$$\Upsilon_{\mathbf{x}_{k,1:N_k}}(\varphi_{k,1:N_k}) = -(\mathbf{z}_k - \mathbf{H}_k \Psi_k(\varphi_{k,1:N_k}))^H \Gamma^{-1} (\mathbf{z}_k - \mathbf{H}_k \Psi_k(\varphi_{k,1:N_k})). \quad (4.31)$$

The integral (4.30) can be approximated using the Laplace method [MBQLG11]. Roughly speaking, the Laplace method consist in using a polynomial approximation of the function $\Upsilon_{\mathbf{x}_{k,1:N_k}}(\cdot)$ of order one at its maximum, thus allowing to evaluate the integral (4.30). The Laplace approximation can be then expressed as follows:

$$p_{SW0}(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}) \approx \exp \left\{ \Upsilon_{\mathbf{x}_{k,1:N_k}}(\hat{\varphi}_{k,1:N_k}) \right\} \frac{(2\pi)^{\frac{N_k}{2}}}{\left| \det \left(-\nabla^2 \Upsilon_{\mathbf{x}_{k,1:N_k}}(\hat{\varphi}_{k,1:N_k}) \right) \right|^{\frac{1}{2}}}, \quad (4.32)$$

where $\hat{\varphi}_{k,1:N_k}$ are the phases maximizing the function $\Upsilon_{\mathbf{x}_{k,1:N_k}}(\cdot)$ and $\nabla^2 \Upsilon(\cdot)$ is the Jacobian matrix calculated with the phases $\hat{\varphi}_{k,1:N_k}$. The phases $\hat{\varphi}_{k,1:N_k}$ cannot be obtained analytically even for two targets and an optimization method such as a gradient descent must be used. However, the function in Eq. (4.31) has the particular structure of a quadratic form in the variable Ψ_k , therefore it is possible to use the classic least square estimator

$$\hat{\Psi}_k = (\mathbf{H}_k^H \Gamma^{-1} \mathbf{H}_k)^{-1} \mathbf{H}_k^H \Gamma^{-1} \mathbf{z}_k \quad (4.33)$$

and to calculate a value close to the actual maximum by taking for each phase $\hat{\varphi}_{k,i}$ the argument of the corresponding component $\hat{\Psi}_{k,i}$, *i.e.*

$$\hat{\varphi}_{k,i} = \arg \left(\hat{\Psi}_{k,i} \right). \quad (4.34)$$

Note that the maximum is not exactly reached with the estimator $\hat{\Psi}_k$ since it may not respect the constraint that all its components have a modulus equal to one (*i.e.* $\hat{\Psi}_k$ is not a vector of phase as $\Psi_k(\varphi_{k,1:N_k})$). In practice, this estimator is in most of the situations

close to the actual maximum. However, in some situations, for instance when components $\mathbf{h}_{k,1:N_k}$ are almost colinear, the difference can be greater. In that latter case, an optimization must be performed or the filter performance will be degraded. A compromise must then be done between the quality of the estimate and the computational time required to reach it.

4.4.1.2 Swerling 1 case

Each modulus $\rho_{k,i}$ follows a Rayleigh distribution:

$$p_{\vartheta_i}(\rho_{k,i}) = p_{SW1}(\rho_{k,i}) = \frac{\rho_{k,i}}{\sigma_{\rho_i}^2} \exp\left(-\frac{\rho_{k,i}^2}{2\sigma_{\rho_i}^2}\right) \quad (4.35)$$

where σ_{ρ_i} is the parameter of the Rayleigh distribution, assumed unknown, such that $\mathbb{E}[\rho_{k,i}^2] = 2\sigma_{\rho_i}^2$ and corresponds to the generic parameter ϑ_i of the density in Eq. (4.2). Obviously, as in the *Swerling 0* case, this parameter can be added to the state vector. Although the integral (4.8) with respect to the *Swerling 1* densities for parameters $\rho_{k,1:N_k}$ and with respect to variables $\varphi_{k,1:N_k}$ seems to be intractable, in practice the density $p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k})$ can be obtained using other probabilistic considerations. Indeed, in the *Swerling 1* model, since $\rho_{k,i}$ follows a Rayleigh distribution with parameter σ_{ρ_i} and $\varphi_{k,i}$ is uniformly distributed over $[0, 2\pi)$, each variable $\rho_{k,i}e^{j\varphi_{k,i}}$ in the measurement equation (4.1) is a zero-mean circular symmetric complex Gaussian variable with variance $2\sigma_{\rho_i}^2$. Therefore \mathbf{z}_k , which is then the sum of independent Gaussian vectors with zero-mean, is a complex Gaussian vector with zero-mean and covariance matrix Σ_{N_k} given by

$$\Sigma_{N_k} = \Gamma + \sum_{i=1}^{N_k} 2\sigma_{\rho_i,k}^2 \mathbf{h}_{k,i} \mathbf{h}_{k,i}^H. \quad (4.36)$$

Clearly, this matrix is definite positive, so that the multi-target likelihood is finally given in closed form by:

$$p_{SW1}(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}) \propto \frac{1}{\det(\Sigma_{N_k})} \exp(-\mathbf{z}_k^H \Sigma_{N_k}^{-1} \mathbf{z}_k). \quad (4.37)$$

In practice, the computation of the likelihood requires the evaluation of $\det(\Sigma_{N_k})$ and $\Sigma_{N_k}^{-1}$ that can be computationally demanding since matrix Σ_{N_k} is a square matrix of size equal to the length of the considered vector $\mathbf{h}_{k,i}$. Fortunately, these quantities can be easily computed by using classic linear algebra formulas. Indeed, the matrix Σ_{N_k} can be written

$$\Sigma_{N_k} = \Gamma + \mathbf{U} \mathbf{V} \mathbf{U}^H, \quad (4.38)$$

with $\mathbf{U} = [\mathbf{h}_{k,1}, \dots, \mathbf{h}_{k,N_k}]$ a matrix with N_k columns and $\mathbf{V} = \text{diag}(2\sigma_{\rho_{1,k}}^2, \dots, 2\sigma_{\rho_{N_k,k}}^2)$. Then using a classic matrix inversion lemma (see [Mur12], p. 117), it comes

$$\Sigma_{N_k}^{-1} = \Gamma^{-1} - \Gamma^{-1} \mathbf{U} (\mathbf{V}^{-1} + \mathbf{U}^H \Gamma^{-1} \mathbf{U})^{-1} \mathbf{U}^H \Gamma^{-1}. \quad (4.39)$$

The inverse of matrix $\mathbf{\Gamma}$ can be pre-computed, while \mathbf{V} is a diagonal matrix and matrix $(\mathbf{V}^{-1} + \mathbf{U}^H \mathbf{\Gamma}^{-1} \mathbf{U})$ is an N_k -by- N_k matrix of much smaller size than $\mathbf{\Sigma}_{N_k}$ as long as the number of targets N_k remains small compared to the number of considered cells. In that case its inversion implies a drastically reduced cost compared to the inversion of $\mathbf{\Sigma}_{N_k}$. Furthermore, the computational cost of the determinant can also be reduced using the matrix determinant lemma (see [Mur12], p. 117)

$$\det(\mathbf{\Sigma}_{N_k}) = \det(\mathbf{V}^{-1} + \mathbf{U}^H \mathbf{\Gamma}^{-1} \mathbf{U}) \det(\mathbf{V}) \det(\mathbf{\Gamma}). \quad (4.40)$$

Note that no hypothesis was made here about the closeness of the targets and therefore this closed-form expression is valid both for distant and close targets. Finally, for the particular monotarget case, the likelihood simply becomes

$$p_{SW1}(\mathbf{z}_k | \mathbf{x}_k) \propto \frac{1}{1 + 2\sigma_{\rho,k}^2 \mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k} \exp\left(\frac{2\sigma_{\rho}^2 |\mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{z}_k|^2}{1 + 2\sigma_{\rho,k}^2 \mathbf{h}_k^H \mathbf{\Gamma}^{-1} \mathbf{h}_k}\right). \quad (4.41)$$

4.4.1.3 Swerling 3 case

Each squared modulus $\rho_{k,i}^2$ follows a chi-square distribution with four degrees of freedom, so that the corresponding density for the modulus $\rho_{k,i}$ is provided by:

$$p_{\vartheta_i}(\rho_{k,i}) = p_{SW3}(\rho_{k,i}) = \frac{8\rho_{k,i}^3}{\nu_{\rho_i}^2} \exp\left(-\frac{2\rho_{k,i}^2}{\nu_{\rho_i}}\right), \quad (4.42)$$

where the parameter ν_{ρ_i} , assumed unknown, is such that $\mathbb{E}[\rho_{k,i}^2] = \nu_{\rho_i}$. Again, this parameter can be added to the state vector as for the *Swerling 0* and *1* case.

According to our knowledge, no closed form can be obtained for Eq (4.8) in the *Swerling 3* case and a numerical approximation must be done, implying the numerical computation of N_k integrals over modulus $\rho_{k,1:N_k}$ and N_k integrals over phases $\varphi_{k,1:N_k}$. However, it is possible to avoid the numerical integration over the parameters $\rho_{k,1:N_k}$ by approximating the chi-square distribution by a Rice distribution; note indeed that the *Swerling 3* model can be viewed as an approximation of a Rice distribution [Ric07]. Using a Rice distribution instead of the *Swerling 3* model, the density of the modulus $\rho_{k,i}$ becomes

$$p_{\text{Rice}}(\rho_{k,i}) = \frac{2\rho_{k,i}(1+a^2)}{\nu_{\rho_i}} \exp\left(-a^2 - \rho_{k,i}^2 \frac{(1+a^2)}{\nu_{\rho_i}}\right) I_0\left(2a\sqrt{\rho_{k,i}^2 \frac{(1+a^2)}{\nu_{\rho_i}}}\right), \quad (4.43)$$

where a is the ratio between the dominant scatterer and the weaker ones. By choosing $a = \sqrt{1 + \sqrt{2}}$, it can be easily checked that densities of the squared modulus $\rho_{k,i}^2$ under *Swerling 3* and Rice models provide the same means and variances [Ric07]. Now consider the complex amplitude $\rho_{k,i} e^{j\varphi_{k,i}}$ where $\rho_{k,i}$ is distributed according to the Rice distribution (4.43). Recall first that this Rice distribution is the distribution of the modulus of a complex Gaussian variable with mean $\mu_{SW3,i} = a\sqrt{\frac{\nu_{\rho_i}}{(1+a^2)}}$ and variance $2\sigma_{SW3,i}^2 = \frac{\nu_{\rho_i}}{(1+a^2)}$.

Then we can replace each variable $\rho_{k,i} e^{j\varphi_{k,i}}$ in (4.1) by a variable $\xi_{k,i} e^{j\psi_{k,i}}$ where the variables $\xi_{k,i}$ and $\psi_{k,i}$ are respectively Gaussian and uniform, and such that $\xi_{k,i} e^{j\psi_{k,i}}$ follows

the same distribution as $\rho_{k,i}e^{j\varphi_{k,i}}$. Conditionally to phases $\psi_{k,1:N_k}$, the observation \mathbf{z}_k is a complex Gaussian vector with mean

$$\boldsymbol{\mu}_{k,SW3} = \sum_{i=1}^{N_k} \mu_{SW3,i} e^{j\psi_{k,i}} \mathbf{h}_{k,i}$$

and covariance matrix

$$\boldsymbol{\Phi}_{N_k} = \boldsymbol{\Gamma} + \sum_{i=1}^{N_k} 2\sigma_{SW3,i}^2 \mathbf{h}_{k,i} \mathbf{h}_{k,i}^H.$$

The density is then given by

$$p_{\text{Rice}}(\mathbf{z}_k \mid \mathbf{x}_{k,1:N_k}, \psi_{k,1:N_k}) \propto \frac{1}{\det(\boldsymbol{\Phi}_{N_k})} \exp\left(-(\mathbf{z}_k - \boldsymbol{\mu}_{k,SW3})^H \boldsymbol{\Phi}_{N_k}^{-1} (\mathbf{z}_k - \boldsymbol{\mu}_{k,SW3})\right). \quad (4.44)$$

Clearly, the computational cost of $\boldsymbol{\Phi}_{N_k}^{-1}$ and $\det(\boldsymbol{\Phi}_{N_k})$ can be reduced as in the *Swerling 1* case. Then, it just remains to marginalize (4.44) over the phases $\psi_{k,1:N_k}$. This marginalization cannot be computed analytically and must then be calculated numerically, except in the monotarget case.

In the particular monotarget case, a closed-form can be obtained both for the chi-square distribution and the Rice distribution. For the chi-square distribution, the expression in Eq. (4.15) must be integrated over density (4.42). The following result (see [GR07], p. 1097 Eq. 6.663)

$$\int_0^{+\infty} x^3 \exp(-\alpha x^2) I_0(\beta x) dx = \frac{2}{\alpha^2} \left(1 + \frac{\beta}{4\alpha}\right) \exp\left(\frac{\beta^2}{4\alpha}\right), \quad (4.45)$$

where $\alpha \in \mathbb{R}_{\geq 0}^*$ and $\beta \in \mathbb{R}$, is used with $\alpha = \frac{2}{\nu_\rho} + \mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_k$ and $\beta = 2 |\mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{z}_k|$. Then, the likelihood for the chi-square *Swerling 3* model in the monotarget case is given by

$$p_{SW3}(\mathbf{z}_k \mid \mathbf{x}_k) \propto \frac{4}{(2 + \nu_{\rho,k} \mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_k)^2} \left(1 + \frac{\nu_{\rho,k} |\mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{z}_k|}{2 + \nu_{\rho,k} \mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_k}\right) \exp\left(\frac{\nu_{\rho,k} |\mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{z}_k|^2}{2 + \nu_{\rho,k} \mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_k}\right). \quad (4.46)$$

For the Rice distribution, it is possible to integrate Eq. (4.44) over the phase ψ , a computation similar to the one providing Eq. (4.15). Then, the likelihood for the Rice *Swerling 3* model in the monotarget setting is equal to

$$p_{\text{Rice}}(\mathbf{z}_k \mid \mathbf{x}_k) \propto \frac{(1 + a^2) \exp(-a^2)}{1 + a^2 + \nu_{\rho,k} \mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_k} \exp\left(\frac{\nu_{\rho,k} |\mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{z}_k|^2 + a^2 (1 + a^2)}{1 + a^2 + \nu_{\rho,k} \mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_k}\right) \times I_0\left(\frac{2a |\mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{z}_k| \sqrt{(1 + a^2) \nu_{\rho,k}}}{1 + a^2 + \nu_{\rho,k} \mathbf{h}_k^H \boldsymbol{\Gamma}^{-1} \mathbf{h}_k}\right). \quad (4.47)$$

4.4.2 Squared modulus measurements

As it has been shown, the likelihood computation with the squared modulus can be done in two ways, either by taking into account the spatial coherence of the phases and modulus with Eq. (4.27) or by marginalizing independently in each cell with Eq. (4.28). As these two cases are different, we treat them separately in the following.

4.4.2.1 The coherent case

In the coherent case, the likelihood is obtained according to Eq. (4.27) by replacing the generic density $p(\rho_{k,i})$ by the density of the fluctuation considered. However, according to our knowledge, it cannot be done analytically for the *Swerling* models and the integral must be approximated numerically. Moreover, note that it can be really intensive in terms of computational resources especially when the number of targets is large since the size of the integration domain increases exponentially with the number of targets. For this reason, we propose an heuristic solution that consists in replacing the parameter $\gamma^l(\rho_{k,1:N_k}, \varphi_{k,2:N_k})$ by its expectation

$$\mathbb{E}[\gamma^l(\rho_{k,1:N_k}, \varphi_{k,2:N_k})] = \sum_{i=1}^{N_k} \frac{\mathbb{E}[\rho_i^2] |h_{k,i}^l|^2}{\sigma^2}, \quad (4.48)$$

where $\mathbb{E}[\rho_i^2]$ only depends on the parameter of the fluctuations density. Thus, integrals (4.27) are simply the product of the densities in Eq. (4.19) for all the cells. This is a strong approximation for the likelihood, but as it will be seen in section 4.5, it gives interesting performance and it is really faster than the numerical integration which is costly in terms of computational resources. In the monotarget case, the likelihood is given by Eq. (4.24) that requires the integration only over parameter ρ_k and therefore the numerical approximation can be done with reasonable cost.

4.4.2.2 The non coherent case

The non coherent case consists in calculating Eq. (4.28) for each cell and then making the product over the N_c . In practice for the *Swerling 0* case, it is not interesting because Eq. (4.28) can be calculated directly; thus it is preferable to still use Eq. (4.27) to calculate the likelihood since it takes into account the spatial coherence of variables $\varphi_{k,2:N_k}$. Nevertheless, for the *Swerling 1* and *3* cases, probabilistic considerations can be used to calculate Eq. (4.28). Indeed, in the *Swerling 1* case Boers *et al.* [BDV⁺03] noticed that each sample $|z_k^l|^2$ follows an exponential distribution with parameter $\lambda_k^l = \frac{1}{2\sigma^2 + \sum_{i=1}^{N_k} 2\sigma_{\rho_i,k}^2 |h_k^l|^2}$, so that

$$p(|z_k^l|^2 | \mathbf{x}_{k,1:N_k}) = \frac{1}{\lambda_k^l} \exp\left(-\frac{|z_k^l|^2}{\lambda_k^l}\right). \quad (4.49)$$

For the *Swerling 3* case, the integration over parameters $\rho_{k,1:N_k}$ can be avoided with the Rice fluctuations. Indeed, by replacing each variable $\rho_{k,i} e^{j\varphi_{k,i}}$ by a variable $\xi_{k,i} e^{j\psi_{k,i}}$, each sample

$$\frac{|z_k^l|^2}{\sigma^2 + \sum_{i=1}^{N_k} \sigma_{SW3,i}^2 |h_k^l|^2}$$

conditionally to variables $\psi_{k,1:N_k}$ follows a non central chi-square distribution with two degrees of freedom and with non-centrality parameter

$$\gamma_{\text{Rice}}^l(\psi'_{k,2:N_k}) = \frac{\left| \mu_{SW3,1} h_{k,1}^l + \sum_{i=2}^{N_k} \mu_{SW3,i} e^{j\psi'_{k,i}} h_{k,i}^l \right|^2}{\sigma^2 + \sum_{i=1}^{N_k} \sigma_{SW3,i}^2 |h_k^l|^2}, \quad (4.50)$$

that does not depend on parameters $\rho_{k,1:N_k}$ anymore. The density of $|z_k^l|^2$ conditionally to $\psi'_{k,2:N_k}$ is given by Eq. (4.21) where σ^2 is substituted by $\sigma^2 + \sum_{i=1}^{N_k} \sigma_{SW3,i}^2 |h_k^l|^2$ and $\gamma^l(\varphi'_{k,2:N_k}, \rho_{k,1:N_k})$ by $\gamma_{\text{Rice}}^l(\psi'_{k,2:N_k})$. Finally the likelihood $|z_k^l|^2$ is obtained by integrating only over variables $\psi'_{k,2:N_k}$.

In the monotarget case, integral (4.25) can be computed analytically both for the Rice distribution and the chi-square distribution. For the Rice distribution, no integration over phase $\psi'_{k,1}$ is required and the likelihood is provided by

$$p_{SW3,\text{Rice}}\left(|z_k^l|^2 \mid \mathbf{x}_k\right) \propto \frac{2\sigma^2(1+a^2)\exp(-a^2)}{2\sigma^2(1+a^2) + \nu_{\rho,k}|h_k^l|^2} \exp\left(\frac{\nu_{\rho,k}\frac{|h_k^l|^2|z_k^l|^2}{2\sigma^2} + 2\sigma^2 a^2(1+a^2)}{2\sigma^2(1+a^2) + \nu_{\rho,k}|h_k^l|^2}\right) I_0\left(\frac{2a|h_k^l||z_k^l|\sqrt{(1+a^2)\nu_{\rho,k}}}{1+a^2 + \nu_{\rho,k}|h_k^l|^2}\right). \quad (4.51)$$

For the chi-square distribution, result (4.45) is used with $\alpha = \frac{\nu_{\rho,k}|h_k^l|^2 + 4\sigma^2}{2\nu_{\rho,k}\sigma^2}$ and $\beta = \frac{|h_k^l||z_k^l|}{\sigma^2}$. Then, integral (4.25) becomes

$$p_{SW3,\chi^2}\left(|z_k^l|^2 \mid \mathbf{x}_k\right) \propto \frac{(4\sigma^2)^2}{(4\sigma^2 + \nu_{\rho,k}|h_k^l|^2)^2} \left(1 + \frac{1}{2} \frac{\nu_{\rho,k}|h_k^l||z_k^l|}{4\sigma^2 + \nu_{\rho,k}|h_k^l|^2}\right) \exp\left(\frac{|z_k^l|^2}{2\sigma^2} \frac{\nu_{\rho,k}|h_k^l|^2}{4\sigma^2 + \nu_{\rho,k}|h_k^l|^2}\right). \quad (4.52)$$

4.4.3 Summary

In this section, we have provided several solutions to compute the likelihood in a Track-Before-Detect context for complex amplitude fluctuations of type *Swerling* 0, 1 and 3. For the computation of the likelihood with the complex measurement, we have shown that a closed-form can be obtained for all the *Swerling* fluctuations considered in the monotarget case. In the multitarget case, a closed-form can be obtained only in the *Swerling* 1 case, while in the other cases a numerical integration must be performed; however we propose several methods in order to alleviate the time calculation. For the likelihood with the squared modulus of the complex measurement, we have derived the right expression in order to keep the spatial coherence information of complex amplitude parameters and we have shown that only the dependency of one phase can be removed, however this leads to an intractable integral for all the *Swerling* models. Then approximations must be performed; we propose a few solutions for such approximations. Table 4.2 presents a sum-up of the different techniques to calculate the likelihood with the existing methods or those proposed in this chapter.

4.5 Simulation and Results

In this section, we first study the performance in detection and estimation of a single target particle filter that considers either complex or squared modulus measurements. We show the improvement of using complex measurements both in detection and in estimation only

		Swerling 0	Swerling 1	Swerling 3
Complex measurement	Monotarget	Eq. (4.15) and [DRC12]	Eq. (4.41)	Eq. (4.46), Eq. (4.47)
	Multitarget	Eq. (4.8), and 4.4.1.1	Eq. (4.37)	Eq. (4.8), Eq. (4.44)
Squared modulus	Monotarget, non coherent	Eq. (4.25) and [MB08]	Eq. (4.49) and [MB08]	Eq. (4.52), Eq. (4.51) and [MB08]
	Multitarget, non coherent	Eq. (4.28)	Eq. (4.49) and [BDV ⁺ 03]	Eq. (4.28) and 4.4.2.2
	Monotarget, coherent	Eq. (4.4.2.1) and [DRC12]	Eq. (4.4.2.1) and [DRC12]	Eq. (4.4.2.1), and [DRC12]
	Multitarget, coherent	Eq. (4.27) and 4.4.2.1	Eq. (4.27) and 4.4.2.1	Eq. (4.27) and 4.4.2.1

Table 4.2 – Summary of the likelihood computation with different data types (complex or squared modulus measurements), different Swerling models (type 0, 1 and 3) and different number of targets (mono or multitarget). The squared modulus measurement case is splitted between coherent computation and non coherent computation. Each cell contains the reference of the equation in this chapter that provides the expression for the likelihood. When this expression previously appeared in the literature, the citation of the corresponding paper is provided as well. Contributions of this chapter are highlighted in bold and italic.

for the *Swerling* 1 and 3 model as Davey *et al.* have already shown the benefits of doing so in the *Swerling* 0 case [DRC12]. Then, we study the behaviour of a simple multitarget particle filter for two close targets. Performance are evaluated in terms of estimation of the two target states and track loss for fluctuations of type *Swerling* 0, 1 and 3.

4.5.1 Single target simulation and results

4.5.1.1 Scenario of the simulation

We consider a scenario with 100 time steps. The target appears at time step $k_b = 10$ and disappears at step $k_d = 75$. At time step k_b , the target state is initialized with the prior distribution $p_b(\cdot)$ defined in section 2.2 and until time step k_d the state is propagated according to Eq. (2.6) (with $q_s = 0$). We also assume that the entire trajectory is contained within area \mathcal{D} (defined in section 2.2.2). The SNR of the target is fixed either to 5, 7 or 10 dB and we consider fluctuations of type *Swerling* 1 and 3. Concerning the measurement model, we use the one defined in section 2.3.

4.5.1.2 Single target particle filter and performance evaluation

TBD particle filter For the simulations, we consider the TBD monotarget particle filter described in section 2.6 (denoted as "Marginalized TBD Particle Filter"). Moreover, for the unknown static parameters $2\sigma_\rho^2$ and ν_ρ that correspond respectively to the parameter of the *Swerling 1* and *Swerling 3* fluctuation densities, we add them to the state vector as explained in paragraph 4.3.1.3. Therefore, for each particle the modulus parameter $2\sigma_{k,p}^2$ is simply propagated according to

$$2\sigma_{k,p}^2 = 2\sigma_{k-1,p}^2 + \epsilon_k, \quad (4.53)$$

where ϵ_k is Gaussian noise, with variance σ_n . Finally, parameters $\sigma_{0,p}^2$ and $\nu_{0,p}$ are drawn uniformly over the interval corresponding to a target SNR between SNR_{\min} and SNR_{\max} for the birth particles.

Concerning the other state parameters (*i.e.* the position and the velocity):

- For the continuing case, state parameters are propagated according to the prior (*i.e.* Eq. (2.6)).
- For the birth case, the position is assumed to be initialized with the instrumental density defined in Eq. (2.41) and that corresponds to initializing the position uniformly over the cells that exceed the threshold $\gamma = -2\sigma^2 \log(P_{fa})$ (where P_{fa} is a given false alarm probability). Note that the approximation of optimal density defined in Eq. (2.39) is not used here. Indeed, such a density cannot be used with the squared-modulus measurements. Therefore, in order to make a fair comparison between the particle filter that uses the squared-modulus measurements with the one that uses the complex measurements, we choose an instrumental density that can be used in both cases. Finally, for the velocity, it is simply assumed to be initialized with the instrumental density defined in paragraph 2.5.3.

Performance evaluation As we explained in section 2.7, the "Marginalized TBD Particle Filter" does not take any decision about the presence or the absence of the target in the radar window. In this case, we have already stressed that it is difficult to properly measure the performance in estimation without making any decision about the target presence or absence, and without taking into account the fact that the filter has effectively converged to the actual target state. Therefore, we propose to use the methodology developed in Chapter 2 that consists in:

- First using the variable $d_{k,i}^T$ detailed in Eq. (2.95) to make the detection.
- Then, using the indicator of good estimate $e_{k,i}$ defined by Eq. (2.96) (for $k \in \{k_b, \dots, k_d - 1\}$) in order to determine if the filter has converged on the actual target state.
- Lastly, estimating the RMSE in position and velocity respectively with Eq. (2.100) and (2.101).

4.5.1.3 Simulations

For the simulation of the target scenario, the following parameters are used: $T=0.3$ s, $v_{min} = 100$ m/s, $v_{max} = 300$ m/s, $SNR_{min} = 3$ dB, $SNR_{max} = 13$ dB, $q_s = 10^{-3}$, $P_{fa} = 0.1$ and $\sigma_n^2 = 0.05$. The transition probabilities for the particle filter are set to $P_b = P_d = 0.05$. The number of continuing particles is set to $N_c = 2000$ and the number of newborn particles to $N_b = 1000$. Concerning the detection strategy, we choose $T_h(d_{k-1} = 0) = 0.9$ and $T_h(d_{k-1} = 1) = 0.2$.

For the simulation of the radar measurements, the parameters used are: $r_{min} = 100$ km, $r_{max} = 120$ km, $\theta_{min} = -10^\circ$, $\theta_{max} = +10^\circ$, $N_r = 40$, $N_\theta = 14$, $\sigma^2 = 0.5$, $B = 1$ MHz, $T_e = 6.67 \times 10^{-5}$ s, $N_a = 70$, $c = 3 \times 10^8$ m.s $^{-1}$. Note that a small radar window is chosen here to avoid using an important number of particles and thus limit the computational cost.

Three filters are used to detect and estimate the hidden target state \mathbf{x}_k , based on different assumptions for the likelihood computation:

1. The first filter, labeled as "Coh Sq-Mod", considers squared modulus to compute the likelihood and takes into account the spatial coherence of the amplitude parameter ρ_k : it corresponds to Eq. (4.24).
2. The second filter, labeled as "Non Coh Sq-Mod", considers squared modulus but does not take into account the spatial coherence of the amplitude parameter ρ_k : it corresponds to Eq. (4.25).
3. The third filter, labeled as "Coh Comp", considers complex measurements and spatial coherence: it corresponds to Eq. (4.14).

$N_{MC} = 1000$ Monte Carlo simulation were performed for performance measurement.

Detection performance In figures 4.2 and 4.3, we present the average of the probability of existence variable $\hat{P}_{k,e}$ which is measured at each step for the *Swerling* 1 and 3 models respectively. In both case, filters that use the complex measurement outperform those that use squared modulus measurements. Furthermore, the difference between the "Coh Sq-Mod" filter and the "Non Coh Sq-Mod" filter is quite small, therefore it seems that taking into account the spatial coherence of the phase is more important than taking into account the modulus information. Moreover, the "Non Coh Sq-Mod" filter requires numerical approximation that leads to increase the computational time for a very small gain in detection.

Estimation performance In figures 4.4 and 4.5, we present the result in terms of RMSE in position and velocity for the *Swerling* 1 and 3 models respectively. As for all the detection results, particle filters that used the complex measurement outperform filters that work on squared modulus measurements. Moreover, note that the RMSE in position seems to be better at the beginning which is not expected since the tracking algorithm should improve the RMSE. However, this can be explained by the fact that the RMSE is calculated only over the iteration where the target has been detected (*i.e.* $d_k^T = 1$) and at the beginning only a few simulations have detected the target (in particular for

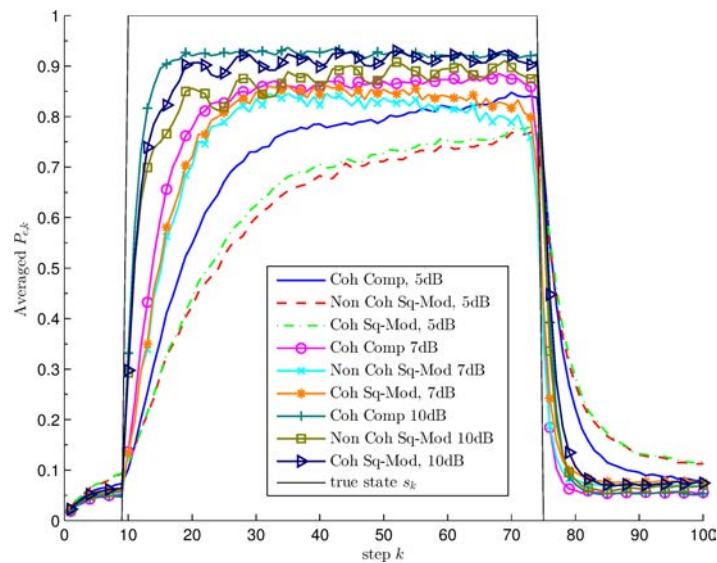


Figure 4.2 – Monte Carlo simulation results for the single target case with the *Swerling 1* model. Average of the probability of existence variable $\hat{P}_{k,e}$. SNR is equal to 5, 7 and 10 dB.

the SNR of 5dB or 7dB) – These detections correspond to favorable cases where the target contribution is not disturb too much by noise. For the next iteration, the filter has detected the target more often than at the beginning, therefore the RMSE is calculated over more Monte-Carlo runs among which less favorable cases. In particular, the cases where the target is located at the edge of the cell that induce a loss in SNR and as a consequence an increase of the RMSE in position.

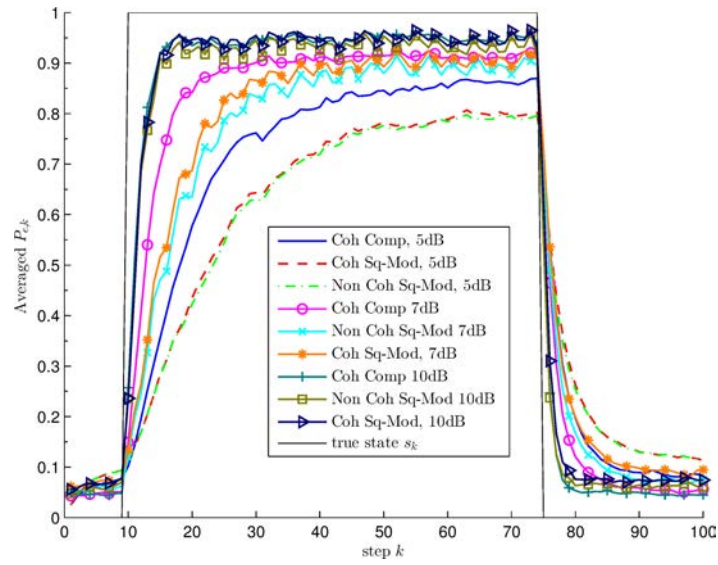


Figure 4.3 – Monte Carlo simulation results for the single target case with the *Swerling* 3 model. Average of the probability of existence variable $\hat{P}_{k,e}$. SNR is equal to 5, 7 and 10 dB.

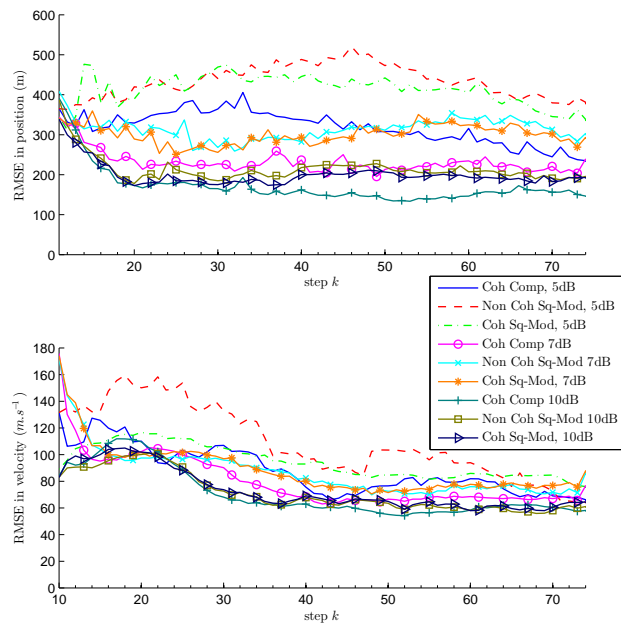


Figure 4.4 – Monte Carlo simulation results for the single target case with the *Swerling* 1 model. Top: RMSE in position. Bottom: RMSE in velocity. SNR is equal to 5, 7 and 10dB.

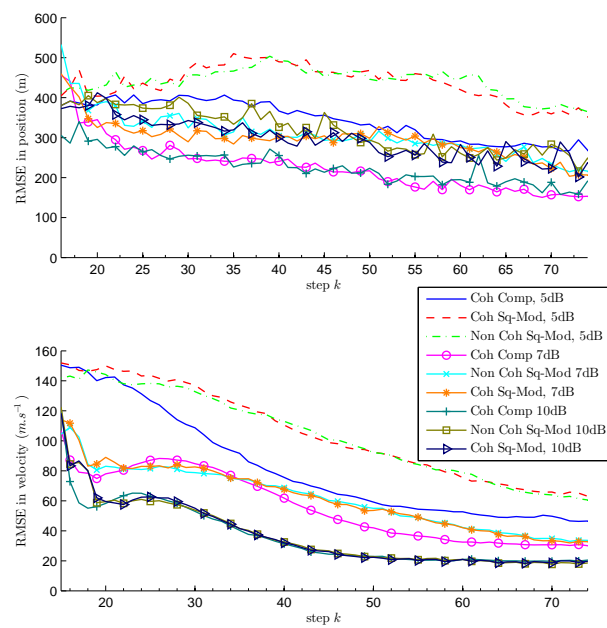


Figure 4.5 – Monte Carlo simulation results for the single target case with the *Swerling* 3 model. Top: RMSE in position. Bottom: RMSE in velocity. SNR is equal to 5, 7 and 10 dB.

4.5.2 Multitarget simulation and results

4.5.2.1 Multitarget scenario

We now consider a scenario with two targets present during all the experiment. Both targets follow a uniform rectilinear trajectory. Target states $\mathbf{x}_{k,1}$ and $\mathbf{x}_{k,2}$ are uniformly initialized over $\mathcal{P} \times \mathcal{C}$ such that:

- the two velocity vectors $(\dot{x}_{k,1}, \dot{y}_{k,1})$, $(\dot{x}_{k,2}, \dot{y}_{k,2})$ form an angle of $\frac{\pi}{4}$,
- the minimum distance between targets is reached at time step $k_c = 35$ and is set to $d_{min} = 150$ m, *i.e.* the minimum distance is equal to the range resolution.

An example of particular trajectories for the two targets is provided in Figure 4.6. Target

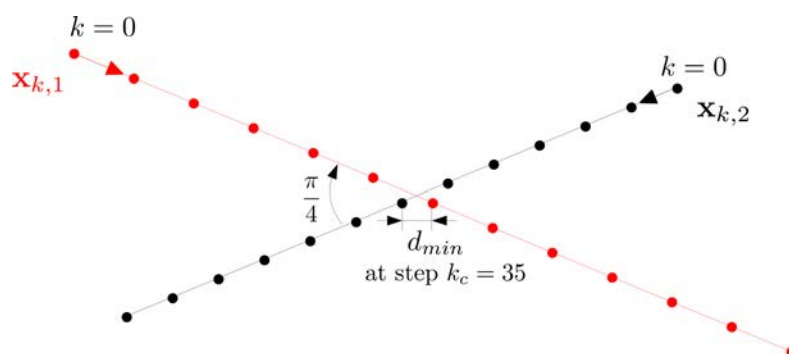


Figure 4.6 – An example of two trajectories where the two velocity vector form an angle of $\frac{\pi}{4}$ and where the minimum distance between the two targets is reached at $k_c = 35$.

SNR are set to 10dB and we consider fluctuations of type *Swerling* 0, 1 and 3. Note that here, we only consider a quite high SNR of 10 dB. Indeed, our main objective in the multi-target case is to demonstrate the importance of taking into account the spatial coherence in the very specific case where targets are close to each other and their contribution in the likelihood mix rather than to determine the performance according to the SNR as in the mono-target case. Considering low SNR target would make difficult to determine if potential particle filter divergences are due to the low SNR or to the target contribution mixing in the likelihood.

4.5.2.2 Multitarget particle filter

For the simulation, we consider here the particle filter proposed by Kreucher *et al.* [KKH05]. We assume that the number of targets is known since the objective here is to measure the effect of the likelihood computation on the particle filter for two close targets. Therefore, $N_k = 2$ and the particle state is defined as $\mathbf{x}_{k,1:N_k}^p = [\mathbf{x}_{k,1}^p, \mathbf{x}_{k,2}^p]^T$, where $\mathbf{x}_{k,1}^p$ and $\mathbf{x}_{k,2}^p$ are the single state vectors of the first and second targets respectively of particle p , $p \in \{1, \dots, N_p\}$. Note here that no presence variable is considered (the presence of the two targets is known a priori by the filter) and thus this filter performs tracking only but cannot perform detection. This choice was motivated in some extent by the computational cost induced by a multi-target filter performing detection and tracking

in a TBD framework and the difficulty to consider simple understandable performance criteria in that case. In the following, we detail the instrumental density used in the particle filter.

At step $k = 0$, each particle target state $\mathbf{x}_{0,i}^p$ is initialized from the actual target state according to the following procedure:

- For the position, a Gaussian noise with variance σ_r^2 is added to the actual target range $r_{0,i} = \sqrt{x_{0,i}^2 + y_{0,i}^2}$ and a Gaussian noise with variance σ_θ^2 is added to the true target bearing $\theta_{0,i} = \arctan\left(\frac{y_{0,i}}{x_{0,i}}\right)$.
- The velocity is initialized around the true velocity in Cartesian coordinates by adding a Gaussian noise with covariance matrix $\sigma_v^2 \mathbf{I}_2$.

For the particle propagation, we consider two cases:

- Either for each particle, state $\mathbf{x}_{k,1:2}^p$ verifies (4.10). Then, the likelihood for each target state $\mathbf{x}_{k,i}^p$ can be computed separately and we propose to use the Independent Partition instrumental density (IP) [KKH05], *i.e.* sample the state of the particles according to the distributions defined by the likelihood of each target.
- Or hypothesis (4.10) is not verified for all the particles and (IP) cannot be used any longer. In that latter case, we just propagate particles according to the prior distribution Eq. (2.6).

4.5.2.3 Calculation of probability of track loss

The probability of track loss is evaluated from N_{MC} Monte Carlo simulation with the following procedure: at each time step k and for each target, we compute the binary loss variable

$$l_{k,i} = \begin{cases} 1 & \text{if } \begin{pmatrix} \hat{r}_{k,i} - r_{k,i} \\ \hat{\theta}_{k,i} - \theta_{k,i} \end{pmatrix}^T \mathbf{P} \begin{pmatrix} \hat{r}_{k,i} - r_{k,i} \\ \hat{\theta}_{k,i} - \theta_{k,i} \end{pmatrix} > \alpha, \\ 0 & \text{otherwise,} \end{cases} \quad (4.54)$$

where $\hat{r}_{k,i} = \sqrt{\hat{x}_{k,i}^2 + \hat{y}_{k,i}^2}$, $\hat{\theta}_{k,i} = \arctan\left(\frac{\hat{y}_{k,i}}{\hat{x}_{k,i}}\right)$, $\mathbf{P} = \begin{pmatrix} \frac{1}{\Delta_r} & 0 \\ 0 & \frac{1}{\Delta_\theta} \end{pmatrix}$ and $\alpha = 5.99$ is the value

of the quantile function of the chi-square distribution with two degrees of freedom evaluated at 0.95. In other words, at each iteration, we check if the position estimator for each target is located within the 0.95% confidence ellipse around the true target position. Finally, a track is declared to be lost if at least one of the variables $l_{k,i}$ equals 1 during at least five consecutive iterations. We define by f_m the loss variable for the m -th Monte Carlo run that takes value 1 if the filter failed to track the two targets during all the experi-

ment and 0, otherwise. Then, the probability of track loss is given by $\hat{P}_{loss} = \frac{1}{N_{MC}} \sum_{m=1}^{N_{MC}} f_m$.

4.5.2.4 Calculation of the Root Mean Square Error (RMSE)

The mean RMSE of the two targets is estimated from N_{MC} Monte Carlo runs with the following procedure: at each iteration, we obtain an estimator of the target state for each target provided by

$$\hat{\mathbf{x}}_{k,i} = \frac{1}{N_p} \sum_{p=1}^{N_p} \mathbf{x}_{k,i}^p, \quad i \in \{1, 2\},$$

and we associate each estimator to a target such that the sum of the Euclidean distances between the estimates and the actual state is minimum. Finally, the RMSE is computed at each iteration k for simulations where both targets have not been declared lost (*i.e.* $l_{k,1} = 0$ and $l_{k,2} = 0$) by taking the mean RMSE of the two targets over these simulations.

4.5.2.5 Simulations

The particle filter is performed with the following parameters: $T = 1$ s, $q_s = 10^{-3}$, $\sigma_r^2 = 3.6 \times 10^{-3}$, $\sigma_\theta^2 = 1.022 \times 10^{-4}$, $\sigma_v^2 = 0.01$ and $\sigma_n^2 = 0.1$. Parameters for the simulation of the radar measurements are the same as for the monotarget simulation, except for the radar window for which we take $r_{min} = 100$ km, $r_{max} = 150$ km, $\theta_{min} = -20^\circ$ and $\theta_{max} = +20^\circ$.

Then, as for the monotarget case, performance is evaluated for the three different ways to calculate the likelihood already defined, *i.e.* "Coh Sq-Mod", "Non Coh Sq-Mod" and "Coh Comp". A fourth one is also used and denoted by "Exp Sq-Mod" (Expectation Squared Modulus) and corresponds to the case where the expectation of the non-centrality parameter is taken to compute the likelihood. Note that for the *Swerling 0* case there is no interest of using the "Non Coh Sq-Mod" method since "Coh Sq-Mod" method requires integration only over $N_k - 1$ phases, therefore we replace this last method by the "Coh Lap" (Coherent Laplace), where the likelihood is calculated via its Laplace approximation (see 4.4.1.1).

When the particle states $\mathbf{x}_{k,1}^p$ and $\mathbf{x}_{k,2}^p$ are well separated, the likelihoods are calculated in closed-form according to the corresponding monotarget likelihood expression. When particle states are too close to each other to be assumed disjoint, the likelihoods are computed according to the multitarget likelihood expressions. When this computation requires a numerical integration, this integration is done over 10 points for each parameter. This small number of integration points is explained by the overall computational cost induced when several parameter dimensions are involved.

Estimation performance The performance in terms of RMSE in position and velocity is presented in figures 4.8, 4.9 and 4.7 for the *Swerling 0*, 1 and 3 models respectively. First we observe that in all cases, "Coh Comp" provides the best performance. Then, the difference between the "Coh Sq-Mod" and "Non Coh Sq-Mod" is quite small so that it does not seem relevant to take into account the spatial coherence of parameters $\rho_{k,1:N_k}$ and $\varphi_{k,1:N_k}$ with squared modulus (at least for relatively high SNR). An other important point is to compare the computational time with respect to performance. Thus, in *Swerling 0* the "Coh Lap" method is approximatively six times faster than "Coh Comp" with almost

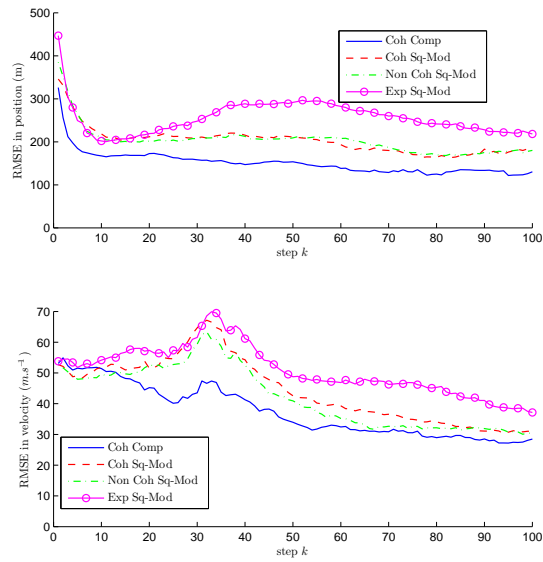


Figure 4.7 – Monte Carlo simulation results in a multi-target setting with the *Swerling 3* model. Top: RMSE in position. Bottom: RMSE in velocity.

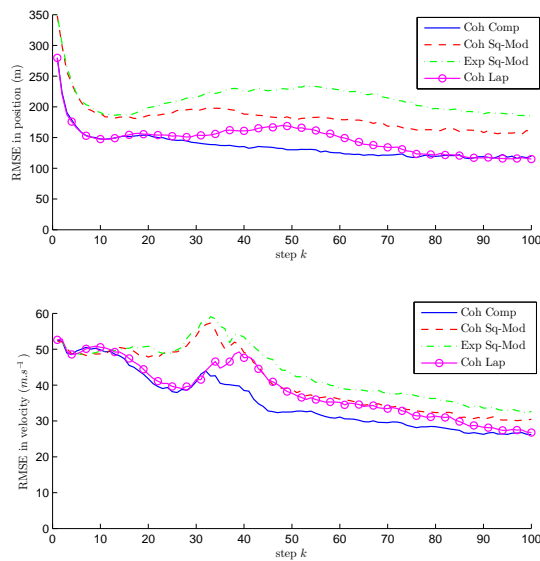


Figure 4.8 – Monte Carlo simulation results in a multi-target setting with the *Swerling 0* model. Top: RMSE in position. Bottom: RMSE in velocity.

the same performance. Likewise, in *Swerling 1* and *Swerling 3*, the "Non Coh Sq-Mod" method is approximately 60 times faster than "Coh Sq-Mod". Finally, note that the RMSE in velocity increases when targets are close. This can be explained by the fact that the likelihood does not depend directly on the velocity.

Track loss performance We present in Table 4.3 the probability of track loss for fluctuations of type *Swerling 0*, 1 and 3. For all the *Swerling* models, the track-loss is

	Probability of track loss		
	<i>Swerling 0</i>	<i>Swerling 1</i>	<i>Swerling 3</i>
"Coh Comp"	1.5×10^{-2}	1.6×10^{-2}	1×10^{-2}
"Coh Sq-Mod"	1.4×10^{-2}	3.1×10^{-2}	1.9×10^{-2}
"Non Coh Sq-Mod"	not defined	4×10^{-2}	1.5×10^{-2}
"Exp Sq-Mod"	2.4×10^{-2}	6.9×10^{-2}	6×10^{-2}
"Coh Lap"	1.5×10^{-2}	not defined	not defined

Table 4.3 – Estimated probability of track loss for the different multitarget particle filters with *Swerling 0*, 1 and 3 fluctuations.

minimum for the "Coh Comp" method, but the "Coh Sq-Mod" and "Non Coh Sq-Mod" methods are relatively close to it. The poorest performance is obtained with the "Exp

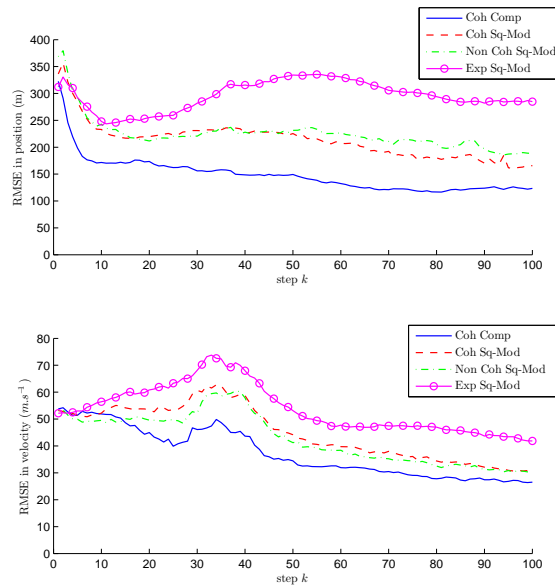


Figure 4.9 – Monte Carlo simulation results in a multi-target setting with the *Swerling 1* model. Top: RMSE in position. Bottom: RMSE in velocity.

Sq-Mod" method where the likelihood is computed with a rough approximation but has the advantage to be much more faster than "Coh Sq-Mod" and "Non Coh Sq-Mod".

4.6 Conclusion

In this chapter, we have investigated different methods for computing the likelihood in a radar Track-Before-Detect context. In practice, the likelihood of the complex measurement depends on the unknown complex amplitude parameters of the targets that must be marginalized. We have shown that closed-form expressions can be obtained in the mono-target case for the *Swerling* models 0, 1 and 3. In the multitarget case, a closed-form expression can be obtained only for the *Swerling 1* case; for the others models, we propose some possible approximations to alleviate the computational time and it may be interesting to investigate other approximations that may lead to faster computational time while preserving acceptable performance. We have also considered the case where the data are the squared modulus of the complex measurements. In that case, no closed-form can be obtained and approximations must be performed. Finally, we have demonstrated via Monte Carlo simulation the benefits of taking into account the spatial coherence of the complex amplitudes both in detection and in estimation compared methods that work on the square modulus of the complex signal. The main conclusions that can be stated based on this work are the following:

- In a TBD context, complex measurements should be used whenever they are available since it appears that the phases information is very important to improve the performance.

- Multitarget likelihood are not simple to compute except for the particular *Swerling 1* case. Thus monotarget likelihood should be computed whenever it is possible to factorize the overall joint density.

Chapter 5

Multitarget Bayesian filter in Track-Before-Detect

5.1 Introduction

In chapter 2, we outlined the classic state model for the TBD problem in a monotarget setting where a variable s_k (taking value 0 or 1) is used to model the presence or the absence of the target. It seems natural to extend the monotarget model to the multitarget setting by considering a process $(N_k, \mathbf{x}_{k,1:N_k})_{k \in \mathbb{N}}$ where N_k is the number of targets and can take values greater than one. In particular, Kreucher *et al.* follow this approach [KKH05] to propose a multitarget particle filter allowing to track several targets in a TBD context. However, their solution suffer from difficulties that may be hard to handle in some situations ; in particular it requires a clustering step in order to sample and estimate the different target states. Moreover, their solution does not fully exploit the particular structure of the likelihood when targets are far apart from each other (see Eq.(4.13)).

Therefore, our main goal, in this chapter, is to propose an alternative strategy allowing to process targets independently (*i.e.* one filter per target) when they do not interact in the likelihood. Thus, we propose, in section 5.3 to consider the following extension of the monotarget model $(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})_{k \in \mathbb{N}}$ – where N_t is the maximum number of targets assumed known – from which we show that each target can be processed independently when they are far apart from each other.

From this model, we then propose in section 5.4 three different particle filters: A first one for detecting the appearance of several targets, a second one to manage the disappearance of several targets and a last one that combines the two previous particle filters in order to manage both the appearance and the disappearance of several targets.

Finally, in section 5.5, we show via Monte Carlo simulations the ability of this strategy to track several targets in a TBD context on simple scenarios.

5.2 Classic Multitarget Bayesian Filter

The measurement model for the multitarget case has already been widely presented in Chap. 4. Therefore, we only present here the multitarget state model, the theoretical

Bayesian filter and its particle approximation.

5.2.1 Multitarget State Model

In chapter 2 and chapter 3, the state model for the monotarget case was extensively detailed, while in chapter 4, N_k targets were considered in order to provide the multitarget measurement equation (4.1) but no prior model was outlined. Thus, in this paragraph, the state model (or prior model) will be detailed for the multitarget case and classic assumptions made in the literature will be provided.

As in the monotarget case where the presence or absence of the target is unknown, in the multitarget case the number of targets is unknown. It is then necessary to model this ignorance. A natural solution is to consider, as in the monotarget case (see section 2.2), an hybrid process $(N_k, \mathbf{x}_{k,1:N_k})_{k \in \mathbb{N}}$, where $N_k \in \mathbb{N}$ is the number of targets and $\mathbf{x}_{k,1:N_k}$ is the multiple target state vector provided by the concatenation of all individual target state vectors $\mathbf{x}_{k,i}$, $i \in \{1, 2, \dots, N_k\}$, *i.e.* $\mathbf{x}_{k,1:N_k} = [\mathbf{x}_{k,1}^T, \mathbf{x}_{k,2}^T, \dots, \mathbf{x}_{k,N_k}^T]^T$. Note here that the size of the state vector is random since it depends on the random variable N_k . Lastly, when $N_k = 0$, the multiple target state vector $\mathbf{x}_{k,1:0}$ is defined as the empty set \emptyset .

In a Bayesian perspective, the process $(N_k, \mathbf{x}_{k,1:N_k})_{k \in \mathbb{N}}$ is assumed Markovian and its joint density can be factorized as follows:

$$p(N_{0:k}, \mathbf{x}_{0:k,1:N_k}) = p(N_0, \mathbf{x}_{0,1:N_0}) \prod_{l=1}^k p(N_l, \mathbf{x}_{l,1:N_l} \mid N_{l-1}, \mathbf{x}_{l-1,1:N_{l-1}}). \quad (5.1)$$

Thus, it is entirely defined by its transition probabilities $p(N_k, \mathbf{x}_{k,1:N_k} \mid N_{k-1}, \mathbf{x}_{k-1,1:N_{k-1}})$ (that will be assumed independent from time index k in the sequel) and the density $p(N_0, \mathbf{x}_{0,1:N_0})$ at step $k = 0$. In practice, it is often convenient to factorize the transition probability as in the monotarget case, first by considering the number of targets N_k and then by expressing the evolution of process $\mathbf{x}_{0,1:N_k}$ conditionally to N_k and N_{k-1} . Mathematically, this leads to consider a transition probability density with the following form:

$$p(N_k, \mathbf{x}_{k,1:N_k} \mid N_{k-1}, \mathbf{x}_{k-1,1:N_{k-1}}) = p(N_k \mid N_{k-1}) p(\mathbf{x}_{k,1:N_k} \mid N_k, N_{k-1}, \mathbf{x}_{k-1,1:N_{k-1}}). \quad (5.2)$$

The process $(N_k)_{k \in \mathbb{N}}$ is a Markov chain, which allows to handle several target appearances or disappearances at each iteration. However, in practice, a simpler model which considers the appearance or disappearance of only one target at each iteration is often used [KKH05]. For this latter model, the process $(N_k)_{k \in \mathbb{N}}$ is an integer-valued random walk: *i.e.*

$$N_k = N_{k-1} + \epsilon_k, \quad (5.3)$$

where $(\epsilon_k)_{k \in \mathbb{N}}$ is an i.i.d sequence taking value -1 , 0 or $+1$. Therefore, the probabilities

$$p(N_k = N_{k-1} + 1 \mid N_{k-1}) = p(\epsilon_k = +1) = P_b, \quad (5.4)$$

$$p(N_k = N_{k-1} - 1 \mid N_{k-1}) = p(\epsilon_k = -1) = P_d, \quad (5.5)$$

do not depend on N_{k-1} and correspond respectively to the classic birth and death event detailed in Chapter 2. In the same manner, the probability

$$p(N_k = N_{k-1} \mid N_{k-1}) = 1 - P_b - P_d \quad (5.6)$$

does not depend on N_{k-1} and corresponds to the case where non target has appeared or disappeared.

Concerning the transition density $p(\mathbf{x}_{k,1:N_k} | N_k, N_{k-1}, \mathbf{x}_{k-1,1:N_{k-1}})$, one hypothesis often encountered in the literature consists in considering that the different target states are independent. Thus, depending on the values of N_k and N_{k-1} , the transition density can be expressed as follows:

$$p(\mathbf{x}_{k,1:N_k} | N_k, N_{k-1}, \mathbf{x}_{k-1,1:N_{k-1}}) = \begin{cases} \prod_{l=1}^{N_k} p_c(\mathbf{x}_{k,l} | \mathbf{x}_{k-1,l}), & \text{if } N_k \leq N_{k-1} \\ p_b(\mathbf{x}_{k,N_k}) \prod_{l=1}^{N_{k-1}} p_c(\mathbf{x}_{k,l} | \mathbf{x}_{k-1,l}), & \text{if } N_k > N_{k-1}, \end{cases} \quad (5.7)$$

where $p_c(\cdot)$ and $p_b(\cdot)$ are respectively the continuing and birth densities detailed in section 2.2.

5.2.2 Theoretical Bayesian Filter

In the multitarget state, the theoretical Bayesian solution is not as simple as in the monotarget case (see section 2.4) where the discrete parameter s_k can only take two values (0 and 1), since here the discrete parameter N_k belongs to \mathbb{N} . However, the multitarget theoretical Bayesian filter still follows the two-step recursion: propagation and update. The aim here is to calculate recursively the posterior density $p(\mathbf{x}_{k,1:N_k}, N_k | \mathbf{z}_{1:k})$. From section 1.2.2, this latter can be rewritten as follows:

$$p(\mathbf{x}_{k,1:N_k}, N_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{x}_{k,1:N_k}, N_k | \mathbf{z}_{1:k-1}) p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k})}{p(\mathbf{z}_k | \mathbf{z}_{1:k-1})}. \quad (5.8)$$

This last equation allows to calculate the probability that exactly l targets are present thanks to the following marginalization:

$$p(N_k = l | \mathbf{z}_{1:k}) = \int p(\mathbf{x}_{k,1:l}, N_k = l | \mathbf{z}_{1:k}) d\mathbf{x}_{k,1:l}, \quad (5.9)$$

and, of course,

$$\sum_{l=0}^{+\infty} p(N_k = l | \mathbf{z}_{1:k}) = 1. \quad (5.10)$$

Concerning the predicted density $p(\mathbf{x}_{k,1:N_k}, N_k | \mathbf{z}_{1:k-1})$, it is obtained by the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_{k,1:N_k}, N_k | \mathbf{z}_{1:k-1}) = \sum_{N_{k-1}=0}^{+\infty} \int p(\mathbf{x}_{k-1,1:N_{k-1}}, N_{k-1} | \mathbf{z}_{1:k-1}) \times p(N_k | N_{k-1}) p(\mathbf{x}_{k,1:N_k} | N_k, N_{k-1}, \mathbf{x}_{k-1,1:N_{k-1}}) d\mathbf{x}_{k-1,1:N_{k-1}}. \quad (5.11)$$

5.2.3 Particle filter approximation

A particle filter approximation of the theoretical Bayesian filter was proposed by Kreucher *et al.* in [KKH05]. Their solution is a generalization of the classic monotarget TBD particle filter detailed in section 2.4: they consider a set of N_p particles $\left\{ \left(N_k^i, \mathbf{x}_{k,1:N_k^i}^i \right), w_k^i \right\}_{i=1}^{N_p}$, where in that case N_k^i belongs to \mathbb{N} , while in the monotarget case the corresponding variable s_k could only take values 0 and 1. Thus, an approximation of the posterior density $p(\mathbf{x}_{k,1:N_k}, N_k \mid \mathbf{z}_{1:k})$ is given by

$$p(\mathbf{x}_{k,1:N_k}, N_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta_{\mathbf{x}_{k,1:N_k^i}^i}(\mathbf{x}_{k,1:N_k}). \quad (5.12)$$

Note here that this particle approximation contains particles with different dimensions since the number of targets per particle may be different.

The first step in the sequential computation of the posterior density $p(\mathbf{x}_{k,1:N_k}, N_k \mid \mathbf{z}_{1:k})$ from the density $p(\mathbf{x}_{k-1,1:N_{k-1}}, N_{k-1} \mid \mathbf{z}_{1:k-1})$ at step $k-1$ consists in drawing the variable N_k for each particle according to an instrumental probability law $q(N_k \mid N_{k-1}, \mathbf{z}_k)$ – in practice, this instrumental probability law is often chosen to be the prior. Recall here that in the proposed model N_k can take only three values¹, i.e. $N_{k-1} - 1$, N_{k-1} or $N_{k-1} + 1$. Then, the particle states $\mathbf{x}_{k,1:N_k^i}^i$ are propagated according to an instrumental density $q(\mathbf{x}_{k,1:N_k} \mid N_k, N_{k-1}, \mathbf{x}_{k-1,1:N_{k-1}}, \mathbf{z}_k)$. Whereas there is no restriction on the choice of the instrumental density, it seems reasonable to choose an instrumental density that has the same structure as $p(\mathbf{x}_{k,1:N_k} \mid N_k, N_{k-1}, \mathbf{x}_{k-1,1:N_{k-1}})$ (defined in Eq. (5.7)). Under that hypothesis the weights are updated according to the following equation:

$$w_k^i \propto w_{k-1}^i \frac{p(N_k^i \mid N_{k-1}^i)}{q(N_k^i \mid N_{k-1}^i, \mathbf{z}_k)} \times \begin{cases} \prod_{l=1}^{N_k} \frac{p_c(\mathbf{x}_{k,l} \mid \mathbf{x}_{k-1,l})}{q_c(\mathbf{x}_{k,l} \mid \mathbf{x}_{k-1,l}, \mathbf{z}_k)}, & \text{if } N_k \leq N_{k-1}, \\ \frac{p_b(\mathbf{x}_{k,N_k})}{q_b(\mathbf{x}_{k,N_k} \mid \mathbf{z}_k)} \prod_{l=1}^{N_{k-1}} \frac{p_c(\mathbf{x}_{k,l} \mid \mathbf{x}_{k-1,l})}{q_c(\mathbf{x}_{k,l} \mid \mathbf{x}_{k-1,l}, \mathbf{z}_k)}, & \text{if } N_k > N_{k-1}. \end{cases} \quad (5.13)$$

Finally, these weights are normalized and a resampling procedure is performed, if required, as in the generic particle filter (see Chapter 1, Algorithm 1.1). A pseudocode of a single cycle of the current particle filter, which is called the Classic Multitarget TBD Particle Filter, is described in Algorithm 5.1.

5.2.4 The invariant permutation problem

An important feature that has not been discussed yet complicates the estimation of the target states: the multitarget posterior density function is invariant under any permutation of the target index [KKH05]. For instance, if the multitarget state contains two

¹As mentioned before, a more general law could be considered for N_k . However, we restrict here to this case to detail a quite simple particle approximation. The extension to a more complicated model for N_k can be derived from the proposed one. Note, nevertheless, that this kind of model may lead to practical issues; in particular, it might be more difficult to initialize properly several new target states at each iteration.

Algorithm 5.1 Classic Multitarget TBD Particle Filter

Require: Particle cloud $\left\{ \left(N_{k-1}^i, \mathbf{x}_{k-1,1:N_{k-1}^i}^i \right), w_{k-1}^i \right\}_{i=1}^{N_p}$ at step $k-1$,

- 1: **for** $i = 1$ to N_p **do**
- 2: Draw N_k^i according to the probability law $p(N_k | N_{k-1}^i)$
- 3: **if** $N_k^i > 0$ **then**
- 4: Draw $\mathbf{x}_{k,1:N_k^i}^i \sim q\left(\mathbf{x}_{k,1:N_k}^i | N_k^i, N_{k-1}^i, \mathbf{x}_{k-1,1:N_{k-1}^i}^i, \mathbf{z}_k\right)$.
- 5: **end if**
- 6: Update particle weight w_k^i according to Eq. (5.13)
- 7: **end for**
- 8: Normalize weights: $w_k^i \leftarrow \frac{w_k^i}{\sum_{l=1}^{N_p} w_k^l}$, $i = 1, \dots, N_p$
- 9: Compute N_{eff} according to Eq. (1.98).
- 10: **if** $N_{\text{eff}} < N_T$ **then**
- 11: Resample N_p particles
- 12: Reset weights: $w_k^i \leftarrow \frac{1}{N_p}$, $i = 1, \dots, N_p$
- 13: **end if**
- 14: **return** $\left\{ \left(N_k^i, \mathbf{x}_{k,1:N_k^i}^i \right), w_k^i \right\}_{i=1}^{N_p}$

individual target state vectors, the posterior density has the same values whatever the order of the target state $\mathbf{x}_{k,1}$ and $\mathbf{x}_{k,2}$, *i. e.*

$$p(\mathbf{x}_{k,1}, \mathbf{x}_{k,2} | \mathbf{z}_{1:k}) = p(\mathbf{x}_{k,2}, \mathbf{x}_{k,1} | \mathbf{z}_{1:k}). \quad (5.14)$$

Therefore, the posterior particle approximation might provide particles with states $\mathbf{x}_{k,1:2}^i = [(\mathbf{x}_{k,1}^i)^T, (\mathbf{x}_{k,2}^i)^T]^T$ or $\mathbf{x}_{k,1:2}^i = [(\mathbf{x}_{k,2}^i)^T, (\mathbf{x}_{k,1}^i)^T]^T$ as illustrated in Figure 5.1. This may not be a problem as long as only the density is considered. However it may become problematic if one wants to estimate the multitarget states, for instance using a classic estimator, as follows:

$$\hat{\mathbf{x}}_{k,1:2} = \left[\frac{1}{N_p} \sum_{i=1}^{N_p} (\mathbf{x}_{k,1}^i)^T, \frac{1}{N_p} \sum_{i=1}^{N_p} (\mathbf{x}_{k,2}^i)^T \right]^T. \quad (5.15)$$

In order to properly estimate the individual target states, it is then necessary to sort the particle state vectors and to partition the state vectors [KKH05] so that the individual target states in a given partition all refer to the same individual target state. In practice, these partitions may be created via a clustering algorithm (such as K-means [HF09]) over the particle state positions.

Moreover, sorting the particle states in ordered partitions may be necessary when using more sophisticated instrumental densities than the prior. Now, the prior is not very efficient in the multitarget case because it blindly samples the general target state without considering the weight of each individual state. Such a strategy tends to create particle states where some individual states sample efficiently the real target state while the others provide worse estimates. This will then spread the particle states over non interesting areas of the multitarget state space.

On the contrary, using partitions enables to consider specific instrumental densities that sample target states individually when target states are sufficiently far apart to be

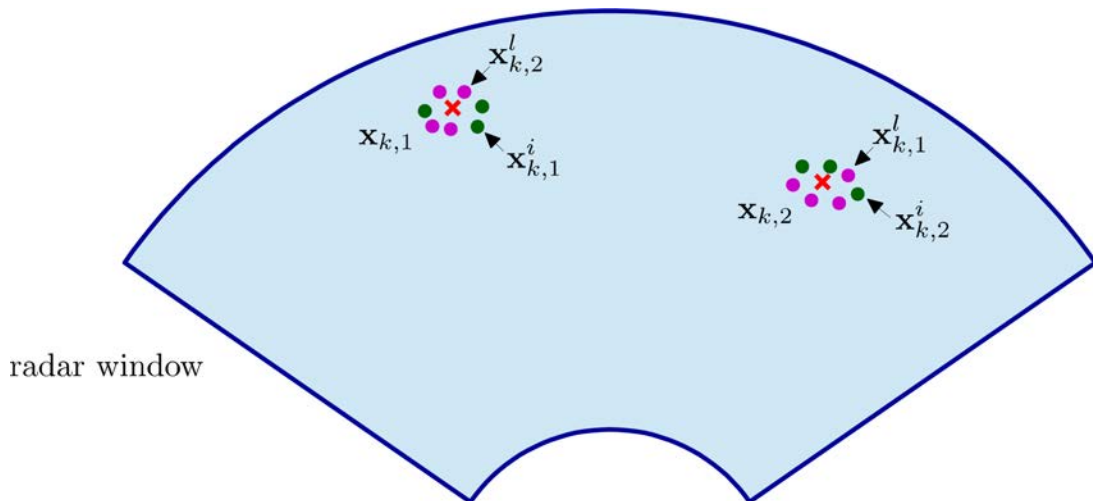


Figure 5.1 – Illustration of the invariant permutation problem. In green, particles with particle target state $\mathbf{x}_{k,1:2}^i = [(\mathbf{x}_{k,1}^i)^T, (\mathbf{x}_{k,2}^i)^T]^T$ while in magenta, particles with particle target state $\mathbf{x}_{k,1:2}^l = [(\mathbf{x}_{k,2}^l)^T, (\mathbf{x}_{k,1}^l)^T]^T$.

considered independent, or by smaller sets of partitions in the case of close targets. This will thus improve the efficiency of the target state sampling. Such a strategy was proposed in [KKH05].

5.2.5 Instrumental densities for the multitarget particle filter

As in the monotarget case, the choice of the instrumental density is crucial to obtain acceptable performance with as few particles as possible. In the multitarget case, as Kreucher *et al.* mentioned in their paper [KKH05], the prior density corresponds to a simple and "naive" solution in order to propagate the particles but, in the other hand, it requires considering a very large number of particles in order to properly sample all the possible combinations between the individual target states.

Therefore, some instrumental densities were proposed in the literature to efficiently propagate the multitarget particle state. The first one was proposed by Orton *et al.* [OF02] and is called the Independent Partition (IP) method. It allows to propagate partitions that do not overlap in an independent manner. The mechanism consists in sampling each particle target state in a partition according to the prior $p_c(\mathbf{x}_{k,t} | \mathbf{x}_{k,t-1}^i)$ where t is the partition number. Then, a discrete density is constructed from these particle states where the weights are provided by the likelihood of the sole partition t , *i.e.*:

$$q(\mathbf{x}_{k,t} | \mathbf{z}_k) = \sum_{i=1}^{N_p} b_{k,t}^i \delta_{\mathbf{x}_{k,t}^i}(\mathbf{x}_{k,t}), \quad (5.16)$$

where,

$$b_{k,t}^i \propto \int_0^{+\infty} \int_0^{2\pi} \Xi_{\mathbf{z}_k, \mathbf{x}_{k,t}}(\rho_{k,t}, \varphi_{k,t}) p(\varphi_{k,t}) p(\rho_{k,t}) d\varphi_{k,t} d\rho_{k,t}, \quad (5.17)$$

and $\Xi_{\mathbf{z}_k, \mathbf{x}_{k,t}}(\cdot, \cdot)$ is detailed in Eq. (4.12). Finally, N_p states $\mathbf{x}_{k,t}^i$ are sampled from the density $q(\mathbf{x}_{k,t} | \mathbf{z}_k)$. Note here that since the prior is no longer used, an additional weighting

term given by $\frac{1}{b_{k,t}^i}$ is induced for each partition in the calculation of the particle weights.

When some partitions overlap, Kreucher *et al.* proposed an other method called the Coupled Partition (CP) method. For each particle i in partition t included in the set of R overlapping partitions, M individual states are sampled from the prior $p_c(\mathbf{x}_{k,t} | \mathbf{x}_{k,t-1}^i)$. Then, a discrete density is built over the M sampled states in the same manner as in the IP method, *i.e.*

$$q(\mathbf{x}_{k,t} | \mathbf{z}_k) = \sum_{m=1}^M b_{k,t}^m \delta_{\mathbf{x}_{k,t}}(\mathbf{x}_{k,t}), \quad (5.18)$$

where,

$$b_{k,t}^m \propto \int_0^{+\infty} \int_0^{2\pi} \Xi_{\mathbf{z}_k, \mathbf{x}_{k,t}}(\rho_{k,t}, \varphi_{k,t}) p(\varphi_{k,t}) p(\rho_{k,t}) d\varphi_{k,t} d\rho_{k,t}. \quad (5.19)$$

Finally, the new state $\mathbf{x}_{k,t}^i$ is sampled from the discrete density in Eq. (5.18). The main difference with the IP method is that here the discrete density in Eq. (5.18) is calculated for each particle while in the IP method only one discrete density is computed over all the particles in the partition.

Lastly, an other important aspect that can be taken into account via the instrumental density is the management of target births and deaths. Indeed, we have seen in Chap. 3 that the solutions developed for the detection of target appearance or target disappearance are quite different. In particular, detecting a target appearance in a large radar window seems more demanding than detecting the disappearance of a single established track, and in particular it requires more particles. Therefore, most of the solutions proposed in the literature consider a two-layer particle filter [GF11]:

- a first filter to detect target disappearances;
- a second filter to detect target appearances.

These two filters are managed by two different instrumental densities. As in the mono-target case, the most important difficulty consists in conveniently sampling the positions of the new targets at each iteration. Garcia-Fernandez in [GF11] proposed to initialize new pre-tracks only in the cells that exceed the threshold $\gamma = -2\sigma^2 \log(P_{fa})$. Then each initialized pre-track is maintained during N_{it}^b iterations; at the end of these N_{it}^b iterations, a statistical test is performed in order to declare if the track is an actual track or a false track. Then all the confirmed pre-tracks are provided to the second layer of the particle filter that propagates the particles using the IP or CP method and manages the track disappearances thanks to a statistical test.

5.2.6 Drawbacks of the existing solutions

The above Bayesian modeling presents the advantage to be very general and can handle almost all the situations encountered in the multitarget case. However, the practical implementation of the particle approximation might require a very large number of particles to ensure acceptable performance. Indeed, if no effort is made to carefully sample the individual target states, the particle approximation may require a lot of particles to properly sample all possible combinations of target states and numbers of targets.

Concerning the IP and CP methods, one major drawback of these two approaches is the need for creating partitions via a clustering algorithm (for instance the K-means algorithm) that may fail to properly sort out the different partitions, in particular in the presence of newborn targets uniformly distributed in the radar window mixed with already clustered existing targets. Moreover, the K-means algorithm requires the prior knowledge of the number of clusters while this number is unknown, it is possible to use well-known criterion such as AIC (Akaike Information Criterion) or MDL (Maximum Description Length) in order to select the number of cluster in the K-means algorithm, but it would increase the already very heavy cost. An other disadvantage of these instrumental densities is the specific resampling procedure performed for each partition t from the discrete density in Eq. (5.16). First, performing this resampling procedure at each iteration might be costly. Then the weights of the discrete instrumental density are only calculated from the current measurement \mathbf{z}_k and thus do not take into account the particle weights at previous step. For high target SNR this will have no consequence. However at low SNR, a noise disturbance may lead to sample most of the particles in a wrong area of the state space.

Lastly, the independence of the targets is taken into account only in the instrumental densities but not in the structure of the Bayesian filter itself. Indeed, most proposed solutions calculate a weight for the multitarget state vector rather than a weight per individual target state, even for sufficiently far away states that may be assumed independent. This may lead to problematic cases where some partitions of a multitarget particle properly sample some of the existing targets while the other partitions do not; the resulting overall weight will tend to underestimate the importance of the well-fitting particles while overestimating the importance of the misfitting particles, and thus bias the estimation. For instance, in the illustration presented in Figure 5.2, the contribution of particle $\mathbf{x}_{k,1}^i$ to the target state estimation of $\mathbf{x}_{k,1}$ will be small (because the overall weight is penalized by the partition $\mathbf{x}_{k,2}^i$) even though it properly samples the target state $\mathbf{x}_{k,1}$.

5.3 A new approach for the multitarget Track-Before-Detect problem

The solution detailed in the previous section considers the overall multitarget state. However, targets far away from each other can be processed independently. Therefore, the aim of this section is to propose a solution that consists in using, whenever it is possible, one particle filter per target rather than an overall filter that samples all target states.

A first solution was proposed by Vo *et al.* in [VVPS10]. In this paper, the authors consider the TBD multitarget problem in the framework of the Random Finite Set (RFS) theory. In particular, Vo *et al.* show that, when considering a particular structure for the likelihood of the measurement conditionally to the random target set, the posterior likelihood can be factorized, thus allowing to process the targets independently. However, the RFS framework used in that paper is not necessary to establish such a property. We propose here an approach based on a probabilistic framework, and in particular a new model that allows to factorize the multitarget posterior density as the product of the individual target posterior densities. Finally, we have seen in Chap. 3 that it could be

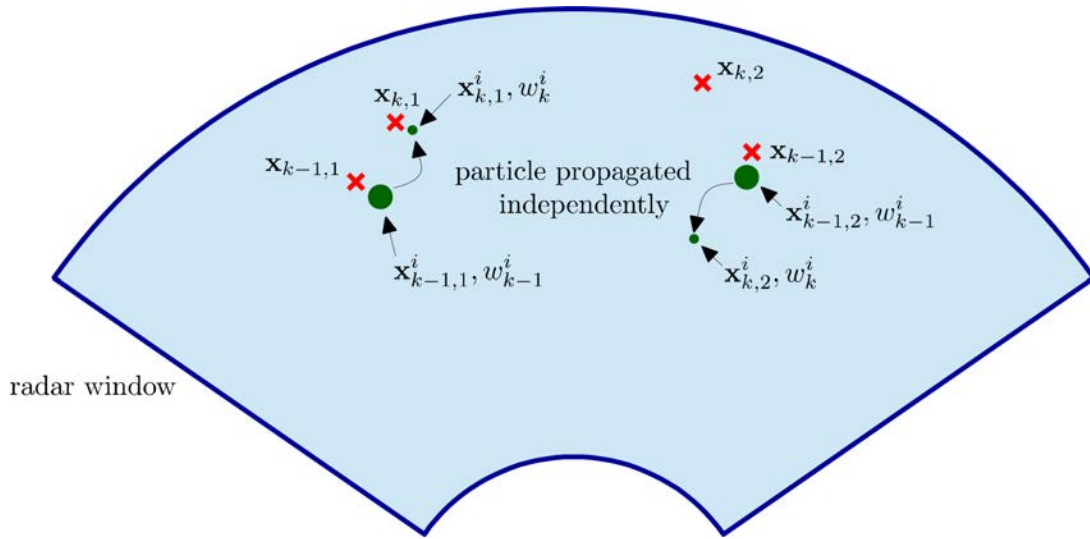


Figure 5.2 – Scheme illustrating the fact that although particle single target states $\mathbf{x}_{k-1,1:2}$ are sampled independently, the resulting multitarget weight w_k^i may be small if one of the single particle target state is badly drawn.

interesting to separate the detection of the target appearance from the detection of the target disappearance. We follow this idea in the multitarget setting.

5.3.1 A new Multitarget State Model

In section 5.2, the number of targets was managed through a variable N_k belonging to \mathbb{N} . This would theoretically allow to manage an infinite number of targets. However, in practice, the number of targets may often be limited to a finite number N_t (first a very large number of targets is very unlikely, and second the capacity of the reception chain to process a large number of targets is usually limited). Furthermore, we saw in the previous multitarget model that the targets are linked via the weight equation (5.13) even if they are assumed to behave independently (see Figure 5.2). Therefore, we propose a new approach that considers a collection of individual TBD target states (*i.e.* (s_k, \mathbf{x}_k)) rather than the overall multitarget state $(N_k, \mathbf{x}_{k,1:N_k})$. This different state model will allow, under some conditions on the likelihood $p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k})$, to factorize the posterior multitarget density as a product of individual target state densities.

To this purpose, let us define by $(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})$ the hybrid multitarget process constituted of a collection of N_t single target states. The idea is now to derive the prior model so that it factorizes as a product of single prior models. The multitarget transition density for this multitarget model can be factorized as in Eq. (2.2), leading to consider as transition density:

$$p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} | s_{k-1,1:N_t}, \mathbf{x}_{k-1,1:N_t}) = \quad (5.20)$$

$$p(s_{k,1:N_t} | s_{k-1,1:N_t}) p(\mathbf{x}_{k,1:N_t} | s_{k-1,1:N_t}, s_{k,1:N_t}, \mathbf{x}_{k-1,1:N_t}).$$

Then, by assuming as in the classic multitarget prior model that the single target processes

$(s_{k,i}, \mathbf{x}_{k,i})$ are independent for any k , the transition probability density factorizes as follows:

$$p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} | s_{k-1,1:N_t}, \mathbf{x}_{k-1,1:N_t}) = \prod_{i=1}^{N_t} p(s_{k,i} | s_{k-1,i}) p(\mathbf{x}_{k,i} | \mathbf{x}_{k-1,i}, s_{k,i}, s_{k-1,i}), \quad (5.21)$$

while the multitarget state density at step $k = 0$ is given by:

$$p(s_{0,1:N_t}, \mathbf{x}_{0,1:N_t}) = \prod_{i=1}^{N_t} p(s_{0,i}) p(\mathbf{x}_{0,i} | s_{0,i}). \quad (5.22)$$

5.3.2 Measurement equation and likelihood for distant target

The measurement equation for the proposed model is similar to the one detailed in section 4.2 with the incorporation of variables $s_{k,1:N_t}$, *i. e.*

$$\mathbf{z}_k = \sum_{i=1}^{N_t} s_{k,i} \rho_{k,i} e^{j\varphi_{k,i}} \mathbf{h}(\mathbf{x}_{k,i}) + \mathbf{n}_k. \quad (5.23)$$

Clearly the factorization of the likelihood in Eq. (4.13) also holds with the addition of variables $s_{k,1:N_t}$: by incorporating the variable $s_{k,1:N_t}$ in Eq. (4.4), the measurement likelihood is given by

$$p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}, s_{k,1:N_t}, \rho_{k,1:N_k}, \varphi_{k,1:N_k}) \propto \exp \left\{ - \sum_{i=1}^{N_k} s_{k,i} \rho_{k,i}^2 \mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,i} + \sum_{i=1}^{N_k} 2s_{k,i} \rho_{k,i} |\mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{z}_k| \cos(\varphi_{k,i} - \xi_{k,i}) - \sum_{i=1}^{N_k} \sum_{l=i+1}^{N_k} 2s_{k,i} s_{k,l} \rho_{k,i} \rho_{k,l} |\mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,l}| \cos(\varphi_{k,i} - \varphi_{k,l} - \phi_{k,il}) \right\}, \quad (5.24)$$

where $\xi_{k,i} = s_{k,i} \arg(\mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{z}_k)$ and $\phi_{k,il} = s_{k,i} s_{k,l} \arg(\mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,l})$.

Let us define, as in Eq. (4.12), the following function:

$$\Xi_{\mathbf{z}_k, (s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})}(\rho_{k,1:N_t}, \varphi_{k,1:N_t}) = p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}, s_{k,1:N_t}, \rho_{k,1:N_k}, \varphi_{k,1:N_k}). \quad (5.25)$$

Then,

$$p(\mathbf{z}_k | \mathbf{x}_{k,1:N_k}, s_{k,1:N_t}) = \int \cdots \int \Xi_{\mathbf{z}_k, (s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})}(\rho_{k,1:N_t}, \varphi_{k,1:N_t}) p(\varphi_{k,i}) p(\rho_{k,i}) d\rho_{k,1:N_t} d\varphi_{k,1:N_t}. \quad (5.26)$$

As in section 4.3.1, under assumption

$$|\mathbf{h}_{k,u}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,v}| \approx 0, \text{ for any } (u, v), u \neq v, \quad (5.27)$$

all the cross terms in Eq. (4.4) can be discarded, and the likelihood function can be expressed as a product of functions that only depend on variables $(s_{k,i}, \mathbf{x}_{k,i})^2$:

$$p(\mathbf{z}_k | s_{k,1:N_t}, \mathbf{x}_{k,1:N_t}) \propto \prod_{i=1}^{N_t} g_{\mathbf{z}_k}(s_{k,i}, \mathbf{x}_{k,i}), \quad (5.28)$$

²Note that, for the sake of simplicity, we do not consider the additional static parameter ϑ_i for the density of amplitude $\rho_{k,i}$. The extension to this model does not present any difficulty.

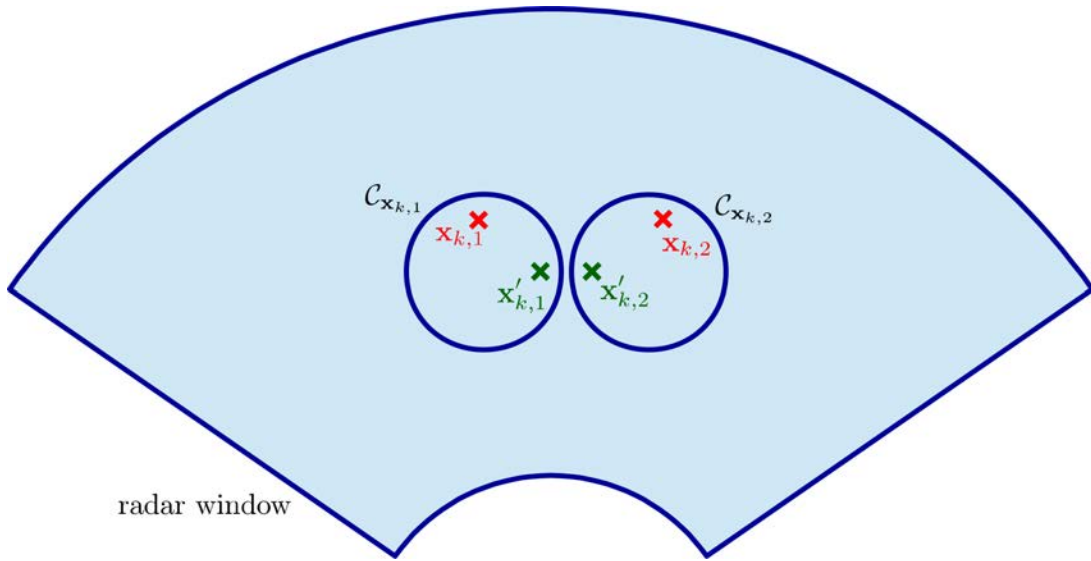


Figure 5.3 – Illustration of the non-interacting hypothesis for different values of $(\mathbf{x}_{k,1}, \mathbf{x}_{k,2})$. For the couple $(\mathbf{x}_{k,1}, \mathbf{x}_{k,2})$ in red the hypothesis is verified since target states are far away from each other, while for the $(\mathbf{x}'_{k,1}, \mathbf{x}'_{k,2})$ in green the hypothesis is not verified since target states are too close.

with

$$g_{\mathbf{z}_k}(s_{k,i}, \mathbf{x}_{k,i}) = \int_0^{+\infty} \int_0^{2\pi} \Xi_{\mathbf{z}_k, (s_{k,i}, \mathbf{x}_{k,i})}(\rho_{k,i}, \varphi_{k,i}) p(\varphi_{k,i}) p(\rho_{k,i}) d\varphi_{k,i} d\rho_{k,i}, \quad (5.29)$$

and

$$\Xi_{\mathbf{z}_k, (s_{k,l}, \mathbf{x}_{k,l})}(\rho_{k,l}, \varphi_{k,l}) = \begin{cases} \exp \left\{ -\rho_{k,i}^2 \mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,i} + 2\rho_{k,i} |\mathbf{h}_{k,i}^H \mathbf{\Gamma}^{-1} \mathbf{z}_k| \cos(\varphi_{k,i} - \xi_{k,i}) \right\}, & \text{if } s_{k,i} = 1, \\ 1, & \text{if } s_{k,i} = 0. \end{cases} \quad (5.30)$$

Lastly, note that this factorization is only true for particular values of target states $(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})$ and might not be verified for other combinations $(s'_{k,1:N_t}, \mathbf{x}'_{k,1:N_t})$. Indeed, for instance, let us assume that the target state $\mathbf{x}_{k,1}$ belongs to a set $\mathcal{C}_{\mathbf{x}_{k,1}}$ while the target state $\mathbf{x}_{k,2}$ belongs to a set $\mathcal{C}_{\mathbf{x}_{k,2}}$. Thus, it may happen that some couples $(\mathbf{x}_{k,1}, \mathbf{x}_{k,2}) \in \mathcal{C}_{\mathbf{x}_{k,1}} \times \mathcal{C}_{\mathbf{x}_{k,2}}$ verifying the non-interacting hypothesis in Eq. (5.27) allowing to factorize the likelihood, while for some other couples $(\mathbf{x}'_{k,1}, \mathbf{x}'_{k,2})$ the non-interacting hypothesis in Eq. (5.27) is not verified. This point is illustrated in Figure 5.3. Therefore the condition of non-interacting target states in $\mathcal{C}_{\mathbf{x}_{k,1}}$ and $\mathcal{C}_{\mathbf{x}_{k,2}}$ should not be confused with $\mathcal{C}_{\mathbf{x}_{k,1}} \cap \mathcal{C}_{\mathbf{x}_{k,2}} = \emptyset$ but can rather be defined as follows:

$$\text{for any } (\mathbf{x}_{k,1}, \mathbf{x}_{k,2}) \in \mathcal{C}_{\mathbf{x}_{k,1}} \times \mathcal{C}_{\mathbf{x}_{k,2}}, \quad |\mathbf{h}_{k,1}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,2}| \approx 0. \quad (5.31)$$

5.3.3 Theoretical Bayesian filter for non-interacting targets

The aim of this section is to demonstrate that, when the likelihood $p_{\vartheta_{1:N_t}}(\mathbf{z}_k | s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})$ can be factorized as in Eq. (5.28) at each iteration step k and for any value of $(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})$,

then the multitarget posterior density $p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} | \mathbf{z}_{1:k})$ factorizes as a product of single target state posterior densities, *i. e.*

$$p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} | \mathbf{z}_{1:k}) = \prod_{i=1}^{N_t} p(s_{k,i}, \mathbf{x}_{k,i} | \mathbf{z}_{1:k}). \quad (5.32)$$

This can be proved by a mathematical induction. To this purpose, let us assume that for any $k \in \mathbb{N}$, the likelihood factorizes as in Eq. (5.28) for all possible values of $(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})$. First, by definition of the state model in section 5.3.1, we have

$$p(s_{0,1:N_t}, \mathbf{x}_{0,1:N_t}) = \prod_{i=1}^{N_t} p(s_{0,i}, \mathbf{x}_{0,i}). \quad (5.33)$$

Thus, the property is verified for $k = 0$. Now let us assume that the property (5.32) is true for a given integer k . By definition of the Bayes filter, the posterior density at step $k + 1$ can be rewritten as follows:

$$p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} | \mathbf{z}_{1:k+1}) = \frac{p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} | \mathbf{z}_{1:k}) p(\mathbf{z}_{k+1} | s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t})}{p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k})}, \quad (5.34)$$

where the predictive density is obtained via the Chapman-Kolmogorov equation:

$$p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} | \mathbf{z}_{1:k}) = \sum_{s_{k,1}, \dots, s_{k,N_t}} \int p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} | \mathbf{z}_{1:k}) \times p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} | s_{k,1:N_t}, \mathbf{x}_{k,1:N_t}) d\mathbf{x}_{k,1:N_t}, \quad (5.35)$$

and the normalization term $p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k})$ is provided by

$$p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k}) = \sum_{s_{k+1,1}, \dots, s_{k+1,N_t}} \int p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} | \mathbf{z}_{1:k}) \times p(\mathbf{z}_{k+1} | s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t}) d\mathbf{x}_{k+1,1:N_t}. \quad (5.36)$$

We will demonstrate that both the predictive density and the normalization can be factorized as a product of single target state functions which will straightforwardly imply that the posterior density at step $k + 1$ also factorizes.

Let us start with the predictive density at step $k + 1$. Using the factorization of the posterior density in Eq. (5.32) at step k for the state $\mathbf{x}_{k,1}$ and the factorization of the transition density in Eq. (5.20), this latter can be rewritten as follows:

$$p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} | \mathbf{z}_{1:k}) = \sum_{s_{k,1}, \dots, s_{k,N_t}} \int p(s_{k,1}, \mathbf{x}_{k,1} | \mathbf{z}_{1:k}) p(s_{k+1,1}, \mathbf{x}_{k+1,1} | s_{k,1}, \mathbf{x}_{k,1}) \times p(s_{k,2:N_t}, \mathbf{x}_{k,2:N_t} | \mathbf{z}_{1:k}) p(s_{k+1,2:N_t}, \mathbf{x}_{k+1,2:N_t} | s_{k,2:N_t}, \mathbf{x}_{k,2:N_t}) d\mathbf{x}_{k,1:N_t}. \quad (5.37)$$

Then, the integration over $\mathbf{x}_{k,1}$ and $s_{k,1}$ can be separated from the variables $(s_{k,2:N_t}, \mathbf{x}_{k,2:N_t})$ leading to

$$p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} \mid \mathbf{z}_{1:k}) = \sum_{s_{k,1}} \int p(s_{k,1}, \mathbf{x}_{k,1} \mid \mathbf{z}_{1:k}) p(s_{k+1,1}, \mathbf{x}_{k+1,1} \mid s_{k,1}, \mathbf{x}_{k,1}) d\mathbf{x}_{k,1} \\ \times \sum_{s_{k,2}, \dots, s_{k,N_t}} \int p(s_{k,2:N_t}, \mathbf{x}_{k,2:N_t} \mid \mathbf{z}_{1:k}) p(s_{k+1,2:N_t}, \mathbf{x}_{k+1,2:N_t} \mid s_{k,2:N_t}, \mathbf{x}_{k,2:N_t}) d\mathbf{x}_{k,2:N_t}. \quad (5.38)$$

Finally, marginalizing over $s_{k+1,2:N_t}$ and $\mathbf{x}_{k+1,2:N_t}$, it comes

$$p(s_{k+1,1}, \mathbf{x}_{k+1,1} \mid \mathbf{z}_{1:k}) = \sum_{s_{k,1}} \int p(s_{k,1}, \mathbf{x}_{k,1} \mid \mathbf{z}_{1:k}) p(s_{k+1,1}, \mathbf{x}_{k+1,1} \mid s_{k,1}, \mathbf{x}_{k,1}) d\mathbf{x}_{k,1}, \quad (5.39)$$

allowing to write the predictive density $p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} \mid \mathbf{z}_{1:k})$, by substituting Eq. (5.39) in Eq. (5.38), as follows:

$$p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} \mid \mathbf{z}_{1:k}) = p(s_{k+1,1}, \mathbf{x}_{k+1,1} \mid \mathbf{z}_{1:k}) p(s_{k+1,2:N_t}, \mathbf{x}_{k+1,2:N_t} \mid \mathbf{z}_{1:k}). \quad (5.40)$$

This last equation indicates that the target state with index 1 is independent from the other states. Of course, the reasoning from Eq. (5.37) to Eq. (5.40) can be iterated for other targets. Thus, the predictive density can be rewritten as the product of the single state predictive density, *i. e.*

$$p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} \mid \mathbf{z}_{1:k}) = \prod_{i=1}^{N_t} p(s_{k+1,i}, \mathbf{x}_{k+1,i} \mid \mathbf{z}_{1:k}). \quad (5.41)$$

In the same manner, using the factorization of the predictive density in Eq. (5.41) and the likelihood in Eq. (5.28), the normalization term $p(\mathbf{z}_{k+1} \mid \mathbf{z}_{1:k})$ also factorizes as follows:

$$p(\mathbf{z}_{k+1} \mid \mathbf{z}_{1:k}) = \prod_{i=1}^{N_t} \sum_{s_{k+1,i}} \int p(s_{k+1,i}, \mathbf{x}_{k+1,i} \mid \mathbf{z}_{1:k}) g_{\mathbf{z}_{k+1}}(s_{k+1,i}, \mathbf{x}_{k+1,i}) d\mathbf{x}_{k+1,i}. \quad (5.42)$$

Therefore, using Eq. (5.42) and Eq. (5.41), the posterior density factorizes as

$$p(s_{k+1,1:N_t}, \mathbf{x}_{k+1,1:N_t} \mid \mathbf{z}_{1:k+1}) = \prod_{i=1}^{N_t} \frac{p(s_{k+1,i}, \mathbf{x}_{k+1,i} \mid \mathbf{z}_{1:k}) g_{\mathbf{z}_{k+1}}(s_{k+1,i}, \mathbf{x}_{k+1,i})}{\sum_{s_{k+1,i}} \int p(s_{k+1,i}, \mathbf{x}_{k+1,i} \mid \mathbf{z}_{1:k}) g_{\mathbf{z}_{k+1}}(s_{k+1,i}, \mathbf{x}_{k+1,i}) d\mathbf{x}_{k+1,i}}, \quad (5.43)$$

where clearly

$$p(s_{k+1,i}, \mathbf{x}_{k+1,i} \mid \mathbf{z}_{1:k+1}) = \frac{p(s_{k+1,i}, \mathbf{x}_{k+1,i} \mid \mathbf{z}_{1:k}) g_{\mathbf{z}_{k+1}}(s_{k+1,i}, \mathbf{x}_{k+1,i})}{\sum_{s_{k+1,i}} \int p(s_{k+1,i}, \mathbf{x}_{k+1,i} \mid \mathbf{z}_{1:k}) g_{\mathbf{z}_{k+1}}(s_{k+1,i}, \mathbf{x}_{k+1,i}) d\mathbf{x}_{k+1,i}}, \quad (5.44)$$

thus demonstrating the factorization of the posterior density at step $k + 1$.

5.3.4 Theoretical Bayesian filter for interacting targets

In the previous section, we have derived the Bayesian filter when targets do not interact. Of course some targets may interact in the likelihood (*e.g.* when they come sufficiently close to each other). In this case, the factorization of the whole posterior density in (5.32) cannot be used anymore. Fortunately, if some targets interact, it does not mean that all the targets should be processed jointly. In fact, it is reasonable to expect that only a small group of targets interacts while the other targets can still be processed independently. We will formalize this more general case in the following. However, since the developments are quite similar to the previous ones, we provide here only the main steps to extend the factorization of the posterior density to groups of interacting targets. The complete development can be found in Appendix C.1.

Let us first define the set of all target indexes $I_{N_t} = \{1, \dots, N_t\}$, and N_g sets of target indexes $I_{int,1}, \dots, I_{int,N_g}$ such that

$$\text{for any } (l, m) \in \{1, \dots, N_g\}, I_{int,l} \cap I_{int,m} = \emptyset, \quad (5.45)$$

and

$$I_{N_t} = \bigcup_{l=1}^{N_g} I_{int,l}. \quad (5.46)$$

Moreover, let us assume that these sets $I_{int,1}, \dots, I_{int,N_g}$ are such that, at each iteration step k , they verify the following hypothesis:

$$\text{for any } (l, m) \in \{1, \dots, N_g\}^2, \text{ for any } (u, v) \in I_{int,l} \times I_{int,m}, |\mathbf{h}_{k,u}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,v}| \approx 0. \quad (5.47)$$

Then, using a similar proof as in the previous paragraph, the posterior multitarget density can be factorized as follows:

$$p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} | \mathbf{z}_{1:k}) = \prod_{i=1}^{N_g} p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} | \mathbf{z}_{1:k}). \quad (5.48)$$

On the other hand, the Bayesian filter for a group of targets $I_{int,i}$ can be obtained as follows:

$$p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} | \mathbf{z}_{1:k}) = \frac{p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} | \mathbf{z}_{1:k-1}) G_{\mathbf{z}_k}(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}})}{\sum_{s_{k,I_{int,i}}} \int p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} | \mathbf{z}_{1:k-1}) G_{\mathbf{z}_k}(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}) d\mathbf{x}_{k,I_{int,i}}}, \quad (5.49)$$

where the function $G_{\mathbf{z}_k}(s_{k,I}, \mathbf{x}_{k,I})$ (I is here any set of indexes) is equal to:

$$G_{\mathbf{z}_k}(s_{k,I}, \mathbf{x}_{k,I}) = \int \Xi_{\mathbf{z}_k, (s_{k,I}, \mathbf{x}_{k,I})}(\rho_{k,I}, \varphi_{k,I}) p(\rho_{k,I}) p(\varphi_{k,I}) d\rho_{k,I} d\varphi_{k,I}. \quad (5.50)$$

Note that here all the cross terms $\mathbf{h}_{k,u}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,v}$, provided that both u and v belong to I , remain contrary to function $g_{\mathbf{z}_k}(s_{k,i}, \mathbf{x}_{k,i})$ in Eq. (5.29) where these cross terms disappeared.

The possibility to factorize the posterior for groups of targets is one of the main difference with the solution proposed by Vo *et al.* [VVPS10] where the factorization is obtained only for single target states.

Let us finally remark that if targets in group I_{int} have interacted until $k-1$ but do not interact after iteration k , then the posterior density $p(s_{k,I_{int}}, \mathbf{x}_{k,I_{int}} \mid \mathbf{z}_{1:k})$ do not factorize as a product of individual target states, *i.e.*

$$p(s_{k,I_{int}}, \mathbf{x}_{k,I_{int}} \mid \mathbf{z}_{1:k}) \neq \prod_{l=1}^{N_{I_{int}}} p(s_{k,l}, \mathbf{x}_{k,l} \mid \mathbf{z}_{1:k}). \quad (5.51)$$

This means that if targets have interacted in the likelihood, they are linked for any future iteration k . Nevertheless, we could expect that asymptotically (*i.e.* when $k \rightarrow +\infty$), the posterior density factorizes.

5.4 Particle filter approximations

Let us now derive a particle filter approximation for the particular Bayesian multitarget filter presented in the previous section. We propose three different particle filters:

- A first filter that manages target disappearances. The idea consists in using, when ever possible, *i.e.* when targets do not interact in the likelihood, the monotarget particle filter outlined in section 3.3; interacting targets will of course be managed jointly.
- A second filter that manages target appearances. The key point here consists in considering that targets appearing in the radar window do not interact in the likelihood. This assumption implies that the instrumental density that samples the particle positions should be carefully designed in order to effectively provide non interacting particle positions.
- Lastly, a third particle filter that manages both target appearances and disappearances.

5.4.1 Disappearance multitarget detection particle filter

The practical implementation of the disappearance multitarget particle filter is quite long and complex and, in particular the way to manage the interacting targets over time. Therefore, in this section we provide only the outline of our solution. The complete description can be found in Appendix C.2.

5.4.1.1 Single and interacting targets particle filters

Let us first detail the particle filter that manages single target disappearance and group of targets disappearance without taking into account the fact that the target state status, *i.e.* whatever the target state interacts with other targets or not, may change over time. To this purpose, let us assume that N_t targets are simultaneously tracked at the current time

instant. If target states $\mathbf{x}_{k,1:N_t}$ do not interact in the likelihood until k (*i.e.* hypothesis in Eq. (5.27) is verified), the whole likelihood factorizes as a product of single target state posterior densities (see Eq. (5.32)). Therefore, rather than approximating the whole multitarget state posterior density with a particle filter, as in section 5.2.3, each single target state posterior density $p(s_{k,i}, \mathbf{x}_{k,i} \mid \mathbf{z}_{1:k})$ can be approximated either by a monotarget particle when it does not interact with the other target filters or by a particle filter that manages a group of interacting targets otherwise.

Concerning the particle filter for single target state, several particle approximations can be considered. We restrict here our attention to the disappearance TBD particle filter detailed in section 3.3, that provides the best performance.

Thus, defining I_{sing} as the set of single targets and using such a particle approximation, each posterior density $p(\mathbf{x}_{k,i} \mid \mathbf{z}_{1:k})$ (for $i \in I_{sing}$) can be approximated as follows:

$$\hat{p}(\mathbf{x}_{k,i} \mid s_{k,i} = 1, \mathbf{z}_{1:k}) = \sum_{p=1}^{N_p} w_{k,i}^p \delta_{\mathbf{x}_{k,i}^p}(\mathbf{x}_{k,i}), \quad (5.52)$$

where the weights $w_{k,i}^p$ can be calculated using Eq. (5.44) leading to

$$w_{k,i}^p \propto w_{k-1,i}^p \frac{p_c(\mathbf{x}_{k,i}^p \mid \mathbf{x}_{k-1,i}^p)}{q_c(\mathbf{x}_{k,i}^p \mid \mathbf{x}_{k-1,i}^p, \mathbf{z}_k)} g_{\mathbf{z}_k}(s_{k,i} = 1, \mathbf{x}_{k,i}^p). \quad (5.53)$$

The densities $p_c(\mathbf{x}_{k,i}^p \mid \mathbf{x}_{k-1,i}^p)$ and $q_c(\mathbf{x}_{k,i}^p \mid \mathbf{x}_{k-1,i}^p, \mathbf{z}_k)$ are respectively the continuing prior density and instrumental density for the continuing case. The probability $\hat{p}(s_k = 1 \mid \mathbf{z}_{1:k})$ can be calculated using Eq. (3.76) (where d_k is replaced by s_k).

In the case of a group of interacting targets, the target states must be processed jointly as explained in section 5.3.4. Thus, for each interacting group of targets $I_{int,i}$, the Bayesian filter in Eq. (5.49) should be used. However, approximating such a Bayesian filter might be difficult due to a complexity increasing with the number of targets. Indeed, if for instance three targets interact, the filter approximation will require the calculation of 2^3 probabilities $p(s_{k,I_{int,i}} \mid \mathbf{z}_{1:k})$ and 2^3 densities $\hat{p}(\mathbf{x}_{k,I_{int,i}} \mid s_{k,I_{int,i}} = 1, \mathbf{z}_{1:k})$. Therefore, for the sake of simplicity, we propose to manage the group of targets by considering that when targets interact: $p(s_{k,I_{int,i}} = 1 \mid \mathbf{z}_{1:k}) = 1$, *i.e.* none of the targets can die.

For each group of targets $I_{int,i}$, we propose the following particle filter approximation:

$$\hat{p}(\mathbf{x}_{k,I_{int,i}} \mid s_{k,I_{int,i}} = 1, \mathbf{z}_{1:k}) = \sum_{p=1}^{N_p} w_{k,I_{int,i}}^p \delta_{\mathbf{x}_{k,I_{int,i}}^p}(\mathbf{x}_{k,I_{int,i}}), \quad (5.54)$$

where

$$w_{k,i}^p \propto w_{k-1,i}^p G_{\mathbf{z}_k}(s_{k,I_{int,i}} = 1, \mathbf{x}_{k,I_{int,i}}). \quad (5.55)$$

Note that in this last equation, we implicitly used the prior distribution as instrumental to propagate the target states $\mathbf{x}_{k,I_{int,i}}$. Finally, weights are normalized and eventually a resampling procedure is performed (if needed).

5.4.1.2 Outline of the proposed particle filter solution

The two particle filters proposed in the previous section assume that the status of each target - whether it belongs to a single track or to a group of interacting targets - does not change. Of course, in real applications, this status may change over time.

Therefore, at each iteration, the status of each target must be updated in order to know if the target should be processed alone or jointly with some other targets. Furthermore, the case of targets that have interacted in the past should also be considered. Indeed, we have seen that such targets will be linked for all the next iterations even if they do not interact anymore. As a consequence, for such targets, it should not be possible to use the single target particle filter, although it allows to dramatically simplify the multitarget tracking problem. In order to solve this problem, we propose the following approximation:

- The groups of interacting targets $I_{int,1:N_g}$ are evaluated at each iteration, using the method provided in section C.2.1.
- If a target (or a group of targets) previously interacted with some other targets but does not at the current step, this target (or this group of targets) is processed independently from other targets with the method the method provided in section C.2.2.

In other words, this last point indicates that the interactions between targets are considered only at the current time step; the past interactions are not taken into account. Note also the sets may differ from iteration k and iteration $k - 1$ but the only available densities correspond to groups defined at iteration $k - 1$. Thus, before performing the particle filter for the sets I_{sing} and $I_{int,1:N_g}$, it is first necessary to reorganize the posterior particle densities from the sets at previous iteration in order to obtain the densities for sets I_{sing} and $I_{int,1:N_g}$, *i.e.* $\hat{p}(\mathbf{x}_{k-1,i} \mid s_{k-1,i} = 1, \mathbf{z}_{1:k-1})$, $i \in I_{sing}$ and $\hat{p}(\mathbf{x}_{k-1,I_{int,l}} \mid s_{k-1,I_{int,l}} = 1, \mathbf{z}_{1:k-1})$, $l \in \{1, \dots, N_g\}$. A method enabling this reorganization is provided in section C.2.2.

Finally, the multitarget disappearance detection particle filter can be summarized as follows:

- First, the sets I_{sing} and $I_{int,1:N_g}$ are evaluated with the method provided in section C.2.1.
- Then, posterior densities at previous step are reorganized in order to calculate the densities $\hat{p}(\mathbf{x}_{k-1,i} \mid s_{k-1,i} = 1, \mathbf{z}_{1:k-1})$, $i \in I_{sing}$ and $\hat{p}(\mathbf{x}_{k-1,I_{int,l}} \mid s_{k-1,I_{int,l}} = 1, \mathbf{z}_{1:k-1})$, $l \in I_{int,1:N_g}$.
- Lastly, the particle filter recursion is performed for each reorganized density.

5.4.2 Appearance Multitarget particle filter

For the Disappearance Multitarget particle filter detailed in the previous section, we proposed to use, when possible, a particle filter per target. The same idea will be developed for the Appearance Multitarget particle filter. However some important differences with the previous algorithm have to be taken into account. Indeed, contrary to the management of target disappearances where particles are already concentrated in the state-space

around the actual target states, in the appearance case, the location of target appearances is unknown, requiring to uniformly sample the whole state-space. Besides, in Chapter 2, we have seen that one of the delicate point for the TBD monotarget particle filter was the initialization of the target state (see section 2.5). Therefore, here a particular attention should be given to the design of the instrumental density for initializing the particle target states – with the additional difficulty that more than one target may appear in the radar window.

In practice, designing such an instrumental density in the general case (for instance, if several targets appear close from each other at the same time) may appear difficult and it is often necessary to consider some simplifying hypotheses. One possibility is to consider that new targets appear sufficiently apart from each other so that they do not interact in the likelihood. Garcia-Fernandez in [GF11] follows this hypothesis to design an instrumental density in order to properly initialize the particle state of birth targets (see section 4.4.2 in [GF11]). As we did in section 2.5.1.2, he initializes particles in the cells that exceed a given threshold γ calculated as $\gamma = -2\sigma^2 \log P_{fa}$ (see Eq. (1.51)). However, in his simulation, he chose a very small probability of false alarm $P_{fa} = 2.10^{-5}$ in order to initialize only a few target states at each iteration. Such a threshold makes difficult to detect low SNR targets and as a consequence to track them. Thus, in order to handle such low SNR targets, we propose some extension to his instrumental density in order to manage a larger P_{fa} .

To this purpose, in the sequel, we will assume that newborn targets appear sufficiently appart from each other and thus do not interact in the likelihood. This hypothesis can be exploited in two different manners in the particle filter framework:

- Either the Bayesian prior can be selected in order to prevent that birth targets appear in the same area and interact in the likelihood. We have not investigated this solution here.
- Or the instrumental density can be chosen in such a manner that the particle target states do not interact in the likelihood. This second strategy will be considered in the following.

5.4.2.1 Outline of the proposed solution

The main idea of the proposed solution consists in using one particle filter per target. Therefore, in order to detect N_t targets, the particle posterior density approximation should factorize as in Eq.(5.32), *i. e.*:

$$\hat{p}(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} | \mathbf{z}_{1:k}) = \prod_{i=1}^{N_t} \hat{p}(s_{k,i}, \mathbf{x}_{k,i} | \mathbf{z}_{1:k}), \quad (5.56)$$

where each particle posterior density $\hat{p}(s_{k,i}, \mathbf{x}_{k,i} | \mathbf{z}_{1:k})$ can be calculated with the algorithm developed in Chap. 2 and 3. However, for the sake of simplicity, we will not consider the solution based on the target appearance time in section 3.3 since, at this point, it seems too complex to manage multiple mixture posterior approximations. Instead, we propose to use the monotarget particle filter of section 2.6 that allows to calculate $\hat{P}_{e,k}^i = \hat{p}(s_{k,i} = 1 | \mathbf{z}_{1:k})$ and $\hat{p}(\mathbf{x}_{k,i} | s_{k,i} = 1, \mathbf{z}_{1:k})$. It should be stressed that in Eq. (5.56)

for any pair (l, m) of targets, none of the particles of target l can interact with particles of targets m . Indeed, if this condition is not verified, then the two targets must be processed jointly and not independently. Thus, if particles belonging to filters l and m (with $l, m \in \{1, \dots, N_t\}$, $l \neq m$) are respectively denoted by $\{\mathbf{x}_{k,l}^p\}_{p=1}^{N_p}$ and $\{\mathbf{x}_{k,m}^p\}_{p=1}^{N_p}$, the previous condition can be expressed as:

$$\forall (p, q) \in \{1, \dots, N_p\}^2, |\mathbf{h}^H(\mathbf{x}_{k,l}^p) \mathbf{\Gamma}^{-1} \mathbf{h}(\mathbf{x}_{k,m}^q)| < \gamma_{\mathbf{h}}. \quad (5.57)$$

5.4.2.2 Managing the interaction between particles

A first solution to prohibit the interaction between particles belonging to different filters consists in keeping one of the interacting particle while "killing" the other interacting particles by setting their weight to zero.

This solution is quite radical, but insures to avoid the interacting issue in all cases and is very simple to implement.

Finding the interacting particles with Eq. (5.57) might be quite long since it requires to calculate the quantity $|\mathbf{h}^H(\mathbf{x}_{k,l}^p) \mathbf{\Gamma}^{-1} \mathbf{h}(\mathbf{x}_{k,m}^q)|$ for all the possible pairs of particles for filters l and m . Therefore, in order to limit the computational resources devoted to the calculation of interactions between particles, we propose to simplify this procedure by working on the cell indexes (u, v) of the particle locations rather than on the scalar products $|\mathbf{h}^H(\mathbf{x}_{k,l}^p) \mathbf{\Gamma}^{-1} \mathbf{h}(\mathbf{x}_{k,m}^q)|$.

Let us define, as in Eq. (2.26), the set of neighborhood cells around the particle target state $\mathbf{x}_{k,i}^p$ as

$$\mathcal{V}_{\mathbf{x}_{k,i}^p} = \{(u, v) \mid |u_{k,i}^p - u| \leq \delta_{h_r}, \text{ and } |v_{k,i}^p - v| \leq \delta_{h_\theta}\}, \quad (5.58)$$

where $(u_{k,i}^p, v_{k,i}^p)$ is the cell location of particle $\mathbf{x}_{k,i}^p$. Then, we define the set of cells that belong to the particle filter approximating the state $\mathbf{x}_{k,i}$ in such a manner:

$$I_{cell,i} = \bigcup_{p=1}^{N_p} \mathcal{V}_{\mathbf{x}_{k,i}^p}. \quad (5.59)$$

Then, two particle filters l and m are declared to interact if the intersection between sets $I_{cell,l}$ and $I_{cell,m}$ is not empty. Let us define by $I_{\cap,(l,m)}$ the intersection between sets $I_{cell,l}$ and $I_{cell,m}$, *i. e.*

$$I_{\cap}^{(l,m)} = I_{cell,l} \cap I_{cell,m}. \quad (5.60)$$

Finally, interacting particles for filter l or m are killed as follows:

$$\forall p \in \{1, \dots, N_p\}, \text{ if } (u_{k,l}^p, v_{k,l}^p) \in I_{\cap}^{(l,m)} \text{ then } w_{k,l}^p = 0. \quad (5.61)$$

As some weights may have been set to zero, the weights $w_{k,l}^p$ for filter l must be renormalized so that $\sum_{p=1}^{N_p} w_{k,l}^p = 1$.

5.4.2.3 Proposed instrumental density

The marginalized TBD particle filter detailed in section 2.6 considers two cases in order to propagate the particles: $N_{p,c}$ "continuing" particles and $N_p - N_{p,c}$ "newborn" particles. Therefore, to extend the monotarget marginalized particles to the multitarget case, two instrumental densities have to be designed.

Instrumental density for the continuing case

The instrumental density for propagating the continuing particles is often chosen to be the prior. However, using the prior density in the multitarget case will not prevent possible interactions between the particle target states since, in that case, the particles for each target state $\mathbf{x}_{k,i}$ will be sampled independently. Therefore, further developments should be made in order to propose an instrumental density that allows to prevent from this issue. As proposed previously, when particles from different particle filters interact, interacting particles for all interacting filters except one can be killed. However, such a strategy does not take into account the information provided by the different particle filters and in particular the probability of appearance $\hat{P}_{e,k}^l$. For instance, if one particle belongs to a particle filter with a high probability $\hat{P}_{e,k}^l$ and interacts with another particle belonging to a particle filter $\hat{P}_{e,k}^m$ with a lower probability, it seems reasonable to keep the particle belonging to the particle filter with the highest probability $\hat{P}_{e,k}^l$.

In order to take into account the information provided by the different particle filters, we propose to sample the continuing particles for the different particle filters in a sequential manner, *i.e.* one filter after another starting with the filter presenting the highest probability of appearance $P_{e,k-1}^l$ at previous step. This solution can be summarized as follows:

1. Sort the probabilities $P_{e,k-1}^l$ in descending order and get the set of ordered indexes $I_{filt,\searrow} = \{i_1, \dots, i_{N_t}\}$.
2. Remove the first element i_1 of the set $I_{filt,\searrow} = \{i_1, \dots, i_{N_t}\}$, *i.e.* $I_{filt,\searrow} = I_{filt,\searrow} \setminus \{i_1\}$.
3. For each filter l in the set $I_{filt,\searrow}$ calculate $I_{\cap}^{(i_1,l)}$ with Eq. (5.60). If the set $I_{\cap}^{(i_1,l)}$ is empty the two filters do not interact and there is nothing to do. On the contrary, some particles of particle filter i_1 and l interact. Then, interacting particles of filter l are killed as follows:
 - Find particles $\mathbf{x}_{k,l}^p$ for which $(u_{k,l}^p, u_{k,l}^p) \in I_{\cap}^{(i_1,l)}$.
 - Set their weights to zero.
 - Normalize the weights $w_{k,l}^p$ such that $\sum_{p=1}^{N_p} w_{k,l}^p = 1$.
4. Go back to step 2 and apply the same procedure.

Instrumental density for the birth case

For the birth case, we propose to extend the work on the instrumental densities for the monotarget case developed in section 2.5. In the sequel, we consider that continuing particles have been already propagated before initializing the birth particles. Once again, the main difficulty is to manage the possible interactions between the particles of the different filters.

As in the continuing case, we propose to initialize particle positions for the different filters in a sequential manner. However, it is here preferable to initialize the different particle filters in the ascending order of the probability of existence $P_{e,k}^l$. Indeed, in the

birth case, it seems reasonable to promote the initialization of particles for particle filters that have the smallest probability of appearance.

An important point to take into account is that continuing particles for the different particle filters are already present in the radar window. Therefore, initializing randomly the position of the birth particles in the set of cells $\mathcal{I}_{k,\gamma}$ – where $\mathcal{I}_{k,\gamma}$ is the set of cells that exceed the threshold γ (see paragraph 2.5.1.2 for details) – may not prevent from interaction between particles of the different filters; some birth particles of a given filter may be initialized in a cell that contains continuing particles of an other filter.

To avoid such a situation, it is first necessary to find which cells in the set $\mathcal{I}_{k,\gamma}$ belong to which filters. Since continuing particles have been already propagated and do not interact by construction, the set of cells $I_{cell,l}^\gamma$, that exceed the threshold γ and that belong to filter l , is simply obtained as follows:

$$I_{cell,l}^\gamma = I_{cell,l} \cap \mathcal{I}_{k,\gamma}. \tag{5.62}$$

Note that this set may be empty. Moreover, the set $\mathcal{I}_{k,\gamma}$ may differ from the union of the sets $I_{cell,l}^\gamma$ since some cells that exceed the threshold may not be considered by any filter. In the sequel, we will denote by $\mathcal{I}_{k,remain}$ the set of cells exceeding the threshold γ and not belonging to any filter and by $N_{\mathcal{I}_{k,remain}}$ the number of remaining cells. Obviously, these cells must be assigned to the different particle filters. We propose the following procedure:

- For each filter, calculate $\alpha_{filt,l} = \frac{1 - P_{e,k}^l}{\sum_{l=1}^{N_t} 1 - P_{e,k}^l}$.
- Randomly assign $\lceil \alpha_{filt,l} N_{\mathcal{I}_{k,remain}} \rceil$ cells to each filter, such as each cell is assigned to only one filter. It should be ensured if possible that at least one cell is assigned to each filter.
- Add the cells randomly assigned to filter l to the set $I_{cell,l}^\gamma$.

Finally, the $N_p - N_{p,c}$ particles of each filter are initialized uniformly over the cells $I_{cell,l}^\gamma$. The weighting term induced by this proposed instrumental distribution is given by:

$$\frac{p_b(r_k, \theta_k)}{q_b(r_k, \theta_k | \mathbf{z}_k)} = \frac{N_{I_{cell,l}^\gamma}}{N_c}, \tag{5.63}$$

where $N_{I_{cell,l}^\gamma}$ is the number of cells in $I_{cell,l}^\gamma$.

5.4.3 Overall TBD multitarget particle filter

In sections 5.4.1 and 5.4.2, two particle filters have been proposed in order to manage respectively the disappearance and the appearance of multiple targets. We now propose to combine the two previous particle solutions. We use here the same strategy as developed in section 3.4 for the monotarget setting. The main difference between the two cases concerns the management of the number of targets. In particular, one would like to avoid that two different filters detect and track the same target. This may arise when particles for an appearance detection filter are initialized near a target already tracked by another filter. To overcome this problem, we assume, in the sequel, that the appearance

multitarget particle filter cannot initialize and propagate particles in cells "belonging" to the disappearance particle filter defined in Eq. (5.59).

Finally one iteration of the overall TBD multitarget particle filter can be summarized as follows:

1. Apply the disappearance multitarget particle filter to update the tracked targets.
2. Determine the cells that are forbidden for the appearance multitarget particle filter.
3. Update the appearance multitarget particle filter.
4. Add any appearance filters with a probability of appearance $P_{e,k}^l$ greater than P_{init} to the set of disappearance multitarget particle filter.

5.5 Simulation and Results

In this section, we evaluate the ability of the overall TBD multitarget particle filter to manage the appearance and disappearance of several targets on quite simple scenarios via Monte Carlo simulations. For the first scenario we simply consider the appearance and disappearance of three targets that do not interact while for the second scenario the crossing of two targets is considered.

As in chapter 2, both detection and estimation performance are evaluated. We propose to evaluate the performance in detection by averaging the estimated number of targets at each iteration over the N_{MC} Monte Carlo runs. The performance in estimation is provided by calculating the RMSE between the estimate target states provided by the particle filters and the actual target states. The computation of the RMSE requires associating the estimated target states with the actual target states. This association is performed so as to minimize the overall summation of all RMSE. If the estimated number of targets is lower than the actual number of targets, all the estimated target states must be used.

5.5.1 Non-interacting targets

We consider a scenario with $N_{it} = 100$ iterations. Three targets are present during the experiment: they appear respectively at $k_{b,1} = 5$, $k_{b,2} = 10$ and $k_{b,3} = 15$, and they disappear respectively at $k_{d,1} = 75$, $k_{d,2} = 80$ and $k_{d,3} = 85$. For each Monte Carlo run, the initialization of the target state for the position and the velocity is done according to the birth density $p_b(\cdot)$ defined in section 2.2 (*i.e.* uniform prior over $\mathcal{D} = [r_{min}, r_{max}] \times [\theta_{min}, \theta_{max}]$ for the position and over $[v_{min}, v_{max}] \times [0, 2\pi]$ for the velocity), with the following parameters:

- $r_{min} = 30$ km, $r_{max} = 42$ km, $\theta_{min} = 30^\circ$ and $\theta_{max} = 60^\circ$,
- $v_{min} = 100$ m.s⁻¹ and $v_{max} = 300$ m.s⁻¹.

Between $k_{b,i}+1$ and $k_{d,i}-1$ the target state $\mathbf{x}_{k,i}$ evolves according to Eq. (2.6) with no noise process (*i.e.* uniform linear motion) and the time between two consecutive measurements is $T_S = 0.3$ s.

Moreover, for each Monte Carlo run, the trajectories of the three targets are carefully drawn so that the targets never interact.

The fluctuations model for each target is assumed to be the *Swerling 0* model and their SNR are set respectively to 5, 7 and 5dB.

We consider here the overall TBD multitarget particle filter detailed in section 5.4.3. This filter has been tested over $N_{MC} = 2000$ Monte Carlo runs with the following parameters :

- Both for the appearance and disappearance multitarget particle filters, we use the following parameters : $\beta = 1$, $q_S = 0.01$, $v_{min} = 100\text{m.s}^{-1}$, $v_{max} = 300\text{m.s}^{-1}$, $\text{SNR}_{min} = 3$ dB, $\text{SNR}_{max} = 13$ dB and $\delta_{h_r} = \delta_{h_\theta} = 2$.
- For the appearance multitarget particle filter, the number of targets N_t is set to 3 (*i.e.* at most three targets can be detected at the same time by the appearance multitarget filter). For each individual particle filter: $P_b = 0.1$, $N_p = 1500$ and $N_{p,b} = 500$ (*i.e.* $N_{p,c} = 1000$). The instrumental density used to propagate the particles is described in paragraph 5.4.2.3, with $P_{fa} = 0.1$. To calculate the set $\mathcal{V}_{\mathbf{x}_{k,i}^p}$ in Eq. (5.59), we take $\delta_{h_r} = \delta_{h_\theta} = 2$ (*i.e.* no interaction between particles in a neighbourhood of two range bearing cells). Lastly, a target is declared to be detected if a filter has a probability of existence $P_{e,k}^l$ greater than 0.9.
- For the disappearance multitarget particle filter, the number of targets N_t is set to 5 (*i.e.* at most five targets can be tracked disappearance multitarget particle filter) For each individual particle filter: $P_d = 0.05$ and $N_p = 1500$. The instrumental density used to propagate the particles is the prior $p_c(\mathbf{x}_k | \mathbf{x}_{k-1})$. Two filters are declared to interact if the distance between the predicted target state estimate is lower than 500 m Lastly, a target is declared to have disappeared if a filter has a probability of existence $P_{e,k}^l$ lower than 0.2.

Figure 5.4 presents the RMSE for each target – Note that the RMSE is displayed with respect to target life iteration, *i.e.* the iterations during which the target is present –, while Figure 5.5 displays the number of targets estimated by the particle filter. Clearly, this solution enables the detection and tracking with some delay of several non interacting targets. However, in Figure 5.4 it seems that the later the target appears the worst is its RMSE, in particular for the velocity. It may be explained by the fact that as long as the first target has not been declared to be detected by its tracking filter, this filter may kill the particles of the other filters (due to the particular structure of the instrumental density).

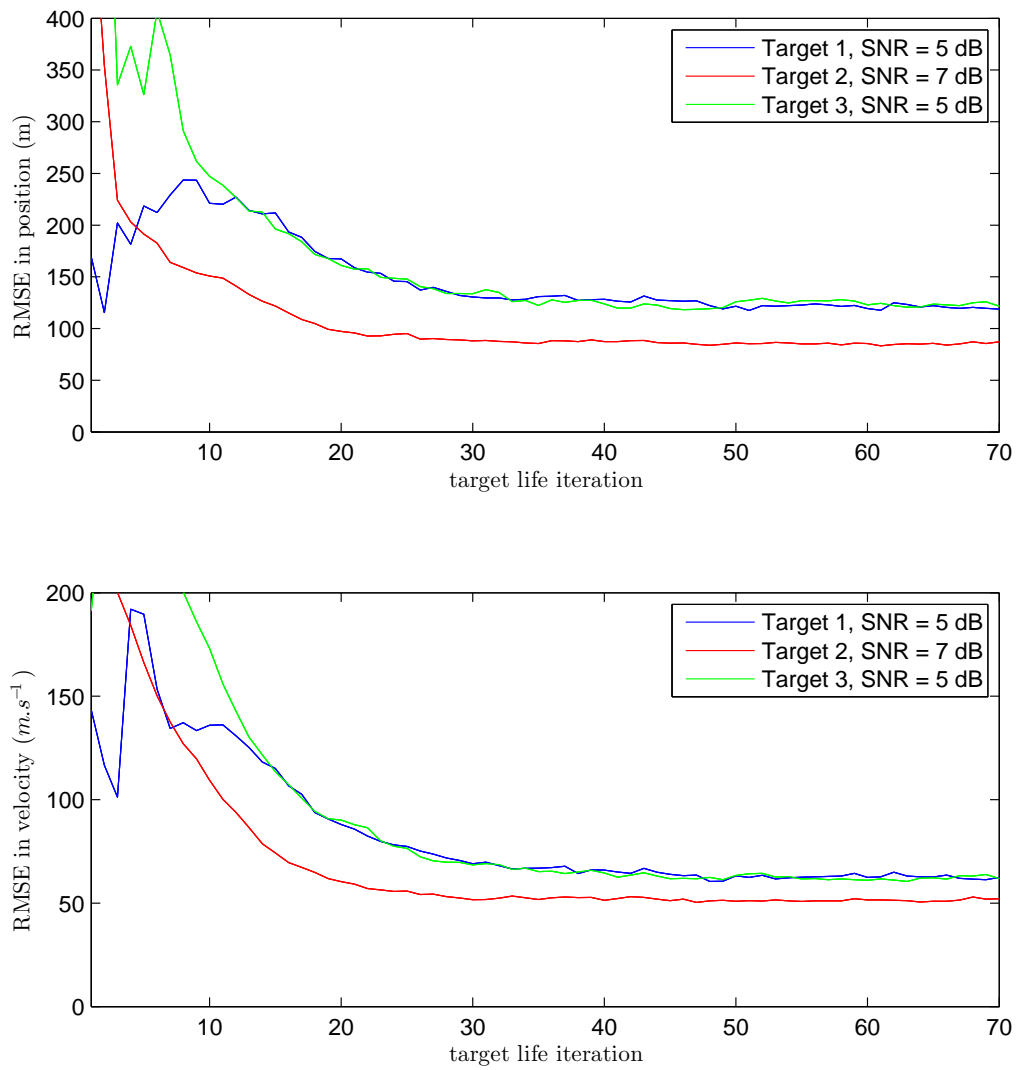


Figure 5.4 – RMSE for the three targets in the non-interacting target scenario.

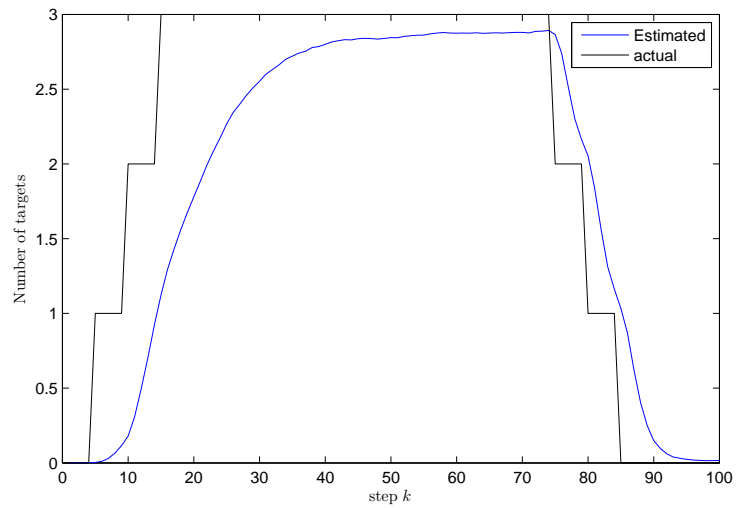


Figure 5.5 – Estimated number of targets. Scenario with three non-interacting targets.

5.5.2 Interacting targets

For this second scenario, the number of iterations is still set to $N_{it} = 100$. Only two targets are present that both appear at step $k_b = 5$ and disappear at step $k_d = 95$. The two target states are drawn as in section 4.5.2.1 : the angle formed by the two velocity vectors is $\pi/4$ and the minimum distance between the two targets is d_{min} . Here targets cross at step $k_c = 40$. The fluctuation model for the two targets is assumed to be the *Swerling 1* model and their SNR are set to 10 dB.

The overall TBD multitarget particle filter is run with the same parameters as in the previous paragraph. $N_{MC} = 2000$ Monte Carlo simulations were run.

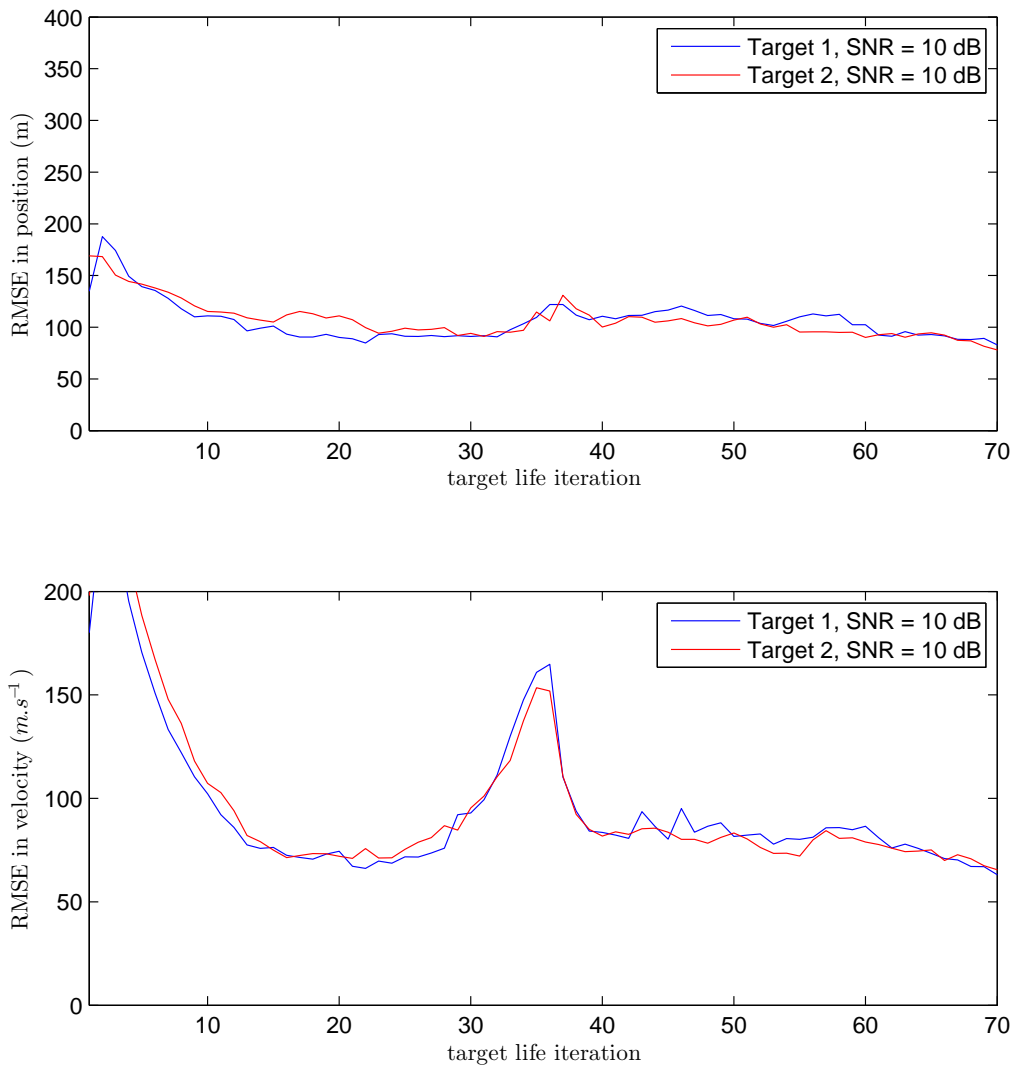


Figure 5.6 – RMSE for the two crossing targets.

Results in terms of RMSE are presented in Figure 5.6 while the estimated number

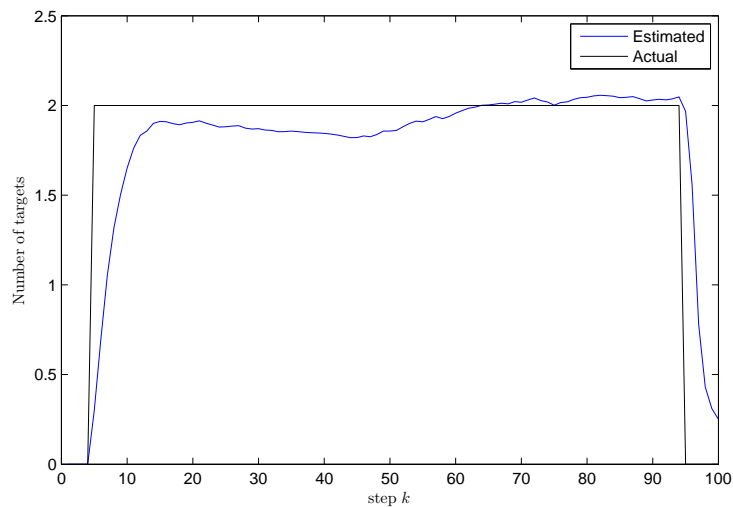


Figure 5.7 – Estimated number of targets. Scenario with two targets crossing.

of targets is presented in Figure 5.7. These two figures show that, in most cases, the proposed solution is able to manage two interacting targets. However some undesirable behaviors have been noticed:

- In Figure 5.7, the estimated number of targets is greater than two after iteration 70. It means that for a small number of Monte Carlo runs, the filter output provided at least three tracks for the two targets. In fact, it appears that when the targets are close to one another, one particle filter diverges from its target and converges to the other one. As a consequence, a new filter is initialized to track the target that was lost, thus leading to three estimated tracks.
- After step k_d , the estimated number of targets should be closer to zero than it is in Figure 5.7 since at 10 dB the target disappearance should be easy to detect. However, we have assumed in the case of interacting targets that when particle filters are grouped (or linked), they cannot be killed; in other words they cannot managed target disappearances. Therefore, if two filters have converged to the same target they cannot be killed anymore since they will be interacting for the remainder of the simulation. As it has just been said, this situation may arise for some Monte Carlo runs and could explain the slow decreasing behavior of the estimated number of targets after the target disappearances.

This two undesirable behaviors are not acceptable and should be managed by the particle filter. This implies that an additional mechanism permitting to prune tracks converging to the same target state is necessary.

5.6 Conclusion

In this chapter, we have first presented the classic multitarget Bayesian filter in a TBD context that process all targets jointly. Then, in section 5.3, we have proposed an other state model that allows to process targets by independent filters when they are sufficiently far apart from each other.

Then, in section 5.4, several particle filter approximations have been proposed. The first particle approximation is dedicated to the appearance of several targets. The main difficulty consists in initializing the particles of the different filters such that they do not interact; we have proposed an instrumental density for that purpose. The second particle filter concerns the disappearance of several targets. The difficult point is to manage the interaction between targets. We have shown that when targets have interacted, they cannot theoretically be considered as independent anymore and have to be processed jointly. However, we have proposed an heuristic procedure in order to consider targets independent even if they have interacted in the past. Finally, the last proposed filter is a combination of the two previous ones that is able to handle the whole TBD multitarget problem.

Lastly, in section 5.5, Monte Carlo simulations have been performed in order to show the ability of this new approach to correctly track, in most situations, several targets. In particular, it has been shown that our solution is able to detect (appearance and disappearance) three targets at low SNR that are far apart from each other and to manage the crossing of two targets at a higher SNR. A few undesirable behaviours have however been observed, implying that further developments and improvements should be brought to the proposed solution.

Conclusion

The aim of this work was to study, develop and propose particle filter methods to detect and track one or several targets in a Track-Before-Detect context.

First, the monotarget TBD problem has been thoroughly investigated. This was motivated by two considerations:

- First, the fact that the monotarget particle filter solutions have not been extensively studied in the literature; in particular the instrumental density for the initialization of the birth particles was not deeply studied in the literature.
- Second, the constant concern that TBD multitarget particle filter solutions based on multitarget particle states are too costly for practical applications and that one should study instead multitarget solutions based on monotarget particle filters as in the classic radar tracking framework where this approach has been successfully used.

In chapter 2, the classic monotarget TBD particle filter generally used in the literature has been studied and our work has focused on proposing some relevant instrumental densities to initialize the particle state. To this purpose, we have considered the optimal instrumental for the initialization of the particle state (which is intractable). It appears that this instrumental density does not depend on the state at previous time step. Thus, all the particles can be initialized from a unique instrumental density making interesting to devote some resources to approximate this particular density. To this purpose, we have proposed several approximations for the optimal instrumental density for the target position using a grid-based approach as well as for the other state parameters such as the amplitude parameter or the presence variable. Then, Monte Carlo simulations have been performed to illustrate the benefits of using such instrumental densities compared to the ones classically used in the literature.

The chapter 3 was motivated by the following questions

- is it relevant to try to detect both the appearance and the disappearance of a target with the same particle filter ?
- is it relevant to still initialize particles whereas the particle filter has converged to the actual target state ?

From these two questions, an alternative modeling to the monotarget TBD problem has been proposed that considers the target appearance and disappearance as two different problems. We have shown that the monotarget TBD problem can be derived as a

Bayesian quickest change point detection problem that allows to consider the target state and its time of appearance or disappearance rather than the presence variable. This has enabled to derive two Bayesian filters, one for the appearance and another one for the disappearance. Different particle filter approximations have been provided for these two theoretical Bayesian filters. Moreover, we have also proposed a particle solution that combines the particle filters previously developed in order to manage both the appearance and the disappearance of a target. Some Monte Carlo simulations have been made in order to evaluate the performance of our approach compared to the classic one showing some benefits in particular in terms of bad detection rate. It also appears that not initializing particles when the particle filter has converged allows to substantially reduce the computational cost without degrading the detection performance. This tends to confirm the intuition that separating the appearance and disappearance detection problems can be more efficient than solving both problems at once.

We focused in chapter 4 on another aspect of the TBD problem: the calculation of the likelihood of the measurement \mathbf{z}_k conditionally to the target states. This computation is of primary importance since it is required for the application of all particle filters. In the TBD framework, this quantity cannot be calculated directly from the measurement equation due to the presence of the unknown target amplitude parameters that may fluctuate randomly and independently over time. A classic heuristic solution to deal with these unknown amplitude parameters consists in considering the squared modulus of the signal $|\mathbf{z}_k|^2$ rather than the complex measurement \mathbf{z}_k . In some cases such a strategy allows to calculate easily the likelihood of the measurement conditionally to the target state – these cases are the *Swerling 0* monotarget case and the *Swerling 1* multitarget state – at the price of a loss of information; in particular the spatial coherence of the amplitude parameters is lost. Moreover, in other situations, this heuristic solution may lead to intractable expressions for the likelihood. In order to overcome these difficulties, Rutten *et al.* have proposed a well-founded approach that consists in marginalizing the likelihood of the measurement over the amplitude parameters while keeping all the information available. However, they have only investigated the *Swerling 0* monotarget case. Thus, we have extended this solution to the multitarget case and to other *Swerling* models. For the monotarget case, we have shown that closed-forms can be obtained for the *Swerling 1* and 3 models. For the multitarget case, we have derived a closed-form expression only for the *Swerling 1* case, while for the other *Swerling* models we have proposed some approximations in order to alleviate the computational time required to calculate the likelihood. Finally, the benefits of calculating the likelihood from the complex measurements \mathbf{z}_k rather than from squared modulus measurements $|\mathbf{z}_k|^2$ have been validated via Monte Carlo simulations.

In the last part of this manuscript (chapter 5), we have tackled the multitarget TBD problem. We have developed a multitarget particle solution that manages targets independently when they are far apart from each other rather than a particle filter that considers the target state jointly. In this perspective, we have shown that it is possible to model the multitarget state as a collection of individual target states $(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})$. By taking advantage of the particular factorization of the measurement likelihood, the whole multitarget posterior density also factorizes as a product of individual target posterior densities, thus allowing to use one filter per target. Moreover we have also shown that

this result can be generalized to groups of targets. Then, as in chapter 3, we have provided some particle filter approximations both for the multitarget appearance case and the disappearance case. For the multitarget appearance case, the main difficulty concerns the initialization of particle states in order to keep the particular structure of one filter per target. To this purpose, we have proposed an instrumental density for the initialization and the propagation of the particle target state that consists, roughly speaking, to kill the interacting particles of the individual particle filters presenting the lowest probability of appearance. For the multitarget disappearance case, the main difficulty consists in managing the interactions between targets. We have proposed an heuristic solution that enables to determine at each iteration if targets interact or not; interacting targets are then processed jointly. Then, as in chapter 3, we combined the two previous particle filters to manage both the target appearances and disappearances. Finally, the proposed particle filter solution was tested via Monte Carlo simulations over two different scenarios. The first one considers the appearance and disappearance of several targets at low SNR that do not interact. Simulation results validated the ability of our solution to handle such scenarios. The second scenario considers the crossing of two targets at a quite high SNR of 10 dB. Here, simulation results have shown that our solution is able to track crossing targets most of the time ; however in some cases the two filter converged to the same target during the crossing, and they were not able to retrieve the two targets afterwards. In that respect, the proposed solution should then be subject to additional improvements. For instance, it may be interesting to develop a particle filter solution that considers that targets may die when they interact or to propose a more sophisticated instrumental density.

Before closing this manuscript, we provide, in the sequel, some perspectives and future works:

- One important work that remains to be done is a comparison with the classic radar tracking algorithms. Indeed, these classic algorithms are very robust and efficient for sufficiently high SNR targets. The main contribution of the TBD method would then be on the detection of low SNR targets. However it should be necessary to quantify the detection gain provided by the TBD approach for this class of targets compared to classic tracking algorithms.
- In all the manuscript, the Doppler parameter was not considered, for simplicity purpose. Of course, this parameter should also be taken into account in a full TBD solution and in particular, as for the other state parameters, it should be interesting to develop a relevant instrumental density taking into account information provided by the measurement to sample this parameter.
- In the measurement equation considered in this work, the noise covariance was assumed to be known. In practice, this hypothesis is unrealistic and therefore, it should be interesting to develop TBD solutions that can handle an unknown covariance matrix, for instance by created an adaptive TBD filter that includes an estimate of the noise and/or clutter covariance in the likelihood computation. Moreover, the Gaussian hypothesis of the noise may be violated, in particular in presence of clutter; of course the advantage of the particle filter solution is its ability

to consider non gaussian densities, but still it would be important to try to adapt the proposed TBD method to non gaussian noise or clutter model, also to evaluate the robustness of the TBD approach to an erroneous statistical noise hypothesis.

- Concerning the processing itself, we have seen that the sample grid provided by the matched filter preprocessing deteriorates somehow the detection performance at a given time instant for targets located at the edge of the resolution cell. This problem arises along all dimensions. Along the range dimension, it seems difficult to overcome because of the analog to digital converter at the reception. However along the angle dimension, it may be interesting to investigate the possibility to discard sampled directions and apply the phase array processing for each particle in its specific direction. This would of course imply applying the TBD algorithms on reception antenna raw data before the FFC processing. Note that a similar procedure could also be applied along the Doppler dimension for pulse trains: for a given particle, the Doppler steering vector considered would then be directly provided by the estimated radial velocity corresponding to the particle state.
- In a similar idea, note that here only point targets were considered. Therefore, the studying the behavior of TBD methods to extended targets, and extending TBD methods to this kind of targets could be of interest.
- Finally, in this work, we have mainly considered simplified cases that permit a better and easier understanding of the algorithmic issue, and also a reduction of the computational cost. However, when considering cases, TBD methods should process large data obtained from range/angle/doppler processing, thus representing many resolution cells to sample. This will represent a very high computational cost in terms of computational resources. Somehow, it will then be important to consider specific computer architectures (*e.g* GPU) that may allow a complex TBD processing on large amount of data.

Finally, it appears that the TBD approach may be a very powerful but very costly method for radar tracking. Clearly it should not be applied to any radar situation: in the presence of sufficiently strong targets, classic radar tracking will certainly perform very well. It may on the contrary be of interest for tracking low SNR targets in surveillance radar applications, provided that subsequent studies demonstrate an interesting performance gain for detecting such targets over classic processing.

Appendix A

Properties of time of appearance τ_b with a geometric prior

When the time of appearance τ_b is modeled by a geometric random variable, *i.e.*

$$p(\tau_b = i) = \begin{cases} 0, & i = 0, \\ P_b(1 - P_b)^{i-1}, & i \geq 1, \end{cases} \quad (\text{A.1})$$

where $0 < P_b < 1$ denotes the probability of birth, it has some interesting properties. Indeed, by defining

$$b_k = \begin{cases} 1, & \text{if } \tau_b \leq k, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{A.2})$$

it can be shown that $(b_k)_{k \in \mathbb{N}}$ is a Markov chain with the following transition probability matrix

$$\Pi_{b_k} = \begin{bmatrix} 1 - P_b & P_b \\ 0 & 1 \end{bmatrix}, \quad (\text{A.3})$$

and also that $p(b_k = 1 \mid b_{k-1} = 0) = P_b$, *i.e.* knowing that the target has not yet appeared at step $k - 1$, its probability to show up at step k does not depend on the time instant and is equal to P_b .

By definition of b_k , the event $\{b_k = 1\}$ can be expressed as follows

$$\{b_k = 1\} = \bigcup_{i=1}^k \{\tau_b = i\}. \quad (\text{A.4})$$

Since the events $\{\tau_b = i\}$ are incompatible,

$$p(b_k = 1) = \sum_{i=1}^k p(\tau_b = i), \quad (\text{A.5})$$

$$p(b_k = 0) = 1 - p(b_k = 1). \quad (\text{A.6})$$

Moreover, for a time appearance variable τ_b modeled by the geometric distribution (A.1), using the definition of conditional probability, *i. e.*

$$p(b_k = 1 \mid b_{k-1} = 0) = \frac{p(b_k = 1, b_{k-1} = 0)}{p(b_{k-1} = 0)}, \quad (\text{A.7})$$

where $p(b_k = 1, b_{k-1} = 0) = p(\tau_b = k)$ by definition of b_k , and noting that $p(b_{k-1} = 1) = \sum_{l=1}^{k-1} p(\tau_b = l)$, it comes

$$p(b_k = 1 \mid b_{k-1} = 0) = \frac{p(\tau_b = k)}{1 - p(b_{k-1} = 1)} = \frac{P_b(1 - P_b)^{k-1}}{1 - \sum_{l=1}^{k-1} P_b(1 - P_b)^{l-1}} = P_b. \quad (\text{A.8})$$

This last equation indicates that knowing that the target has not yet appeared at step $k - 1$, its probability to show up at step k does not depend on the time instant and is equal to P_b .

In other hand, it is easy to show that $(b_k)_{k \in \mathbb{N}}$ is a Markov chain. Indeed, by definition of b_k , the following property holds:

$$b_k = 0 \Rightarrow b_i = 0 \text{ for any } i \leq k - 1, \quad (\text{A.9})$$

and, as a consequence,

$$p(b_k = 0 \mid b_{1:k-2}, b_{k-1} = 0) = p(b_k = 0 \mid b_{k-1} = 0) = 1 - P_b. \quad (\text{A.10})$$

In the same manner, by definition of b_k ,

$$b_{k-1} = 1 \Rightarrow b_k = 1, \quad (\text{A.11})$$

then whatever the sequence $b_{1:k-2}$,

$$p(b_k = 1 \mid b_{1:k-2}, b_{k-1} = 1) = p(b_k = 1 \mid b_{k-1} = 1) = 1. \quad (\text{A.12})$$

Therefore, Eq. (A.10) and (A.12) demonstrate that the process $(b_k)_{k \in \mathbb{N}}$ is Markov with the transition probability matrix in Eq. (A.3). It can be remarked that the state $b_k = 1$ is an absorbing state, *i. e.* once entered in the state $b_k = 1$, the state $b_k = 0$ cannot be reach anymore. Lastly, note that from Eq. (A.1), the probability for the initial state is $p(b_0 = 0) = 1$.

Appendix B

Particle filter for time appearance detection in TBD

The aim of this appendix is to detail the practical implementation of the TBD particle filter that allows to resample over all the mixture components and outlined in paragraph 3.2.4.3. To this purpose, let us first develop the particle approximation $\hat{p}(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$ in Eq. (3.46):

$$\begin{aligned} \hat{p}(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k}) &= \sum_{i \in I_k} \hat{\alpha}_{k,i} \hat{p}(\mathbf{x}_k | \tau_b = i, \mathbf{z}_{1:k}) \\ &= \sum_{i \in I_k} \sum_{n=1}^{N_{p,i}} \hat{\alpha}_{k,i} w_{k,i}^n \delta_{\mathbf{x}_{k,i}^n}(\mathbf{x}_k). \end{aligned} \quad (\text{B.1})$$

Thus, it is possible to calculate the effective sample size $N_{\text{eff},k}^{\text{all}}$ for the overall particle approximation $\hat{p}(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$ from the effective sample size measures $N_{\text{eff},i}$ of the different mixture components using Eq.(1.98) as follows

$$\begin{aligned} N_{\text{eff},k}^{\text{all}} &= \left(\sum_{i \in I_k} \sum_{n=1}^{N_{p,i}} (\hat{\alpha}_{k,i} w_{k,i}^n)^2 \right)^{-1} \\ &= \left(\sum_{i \in I_k} \hat{\alpha}_{k,i}^2 \sum_{n=1}^{N_{p,i}} (w_{k,i}^n)^2 \right)^{-1} \\ &= \left(\sum_{i \in I_k} \frac{\hat{\alpha}_{k,i}^2}{N_{\text{eff},i}} \right)^{-1}, \end{aligned} \quad (\text{B.2})$$

where $N_{\text{eff},i}$ is the effective sample size of the mixture component i . Thus, by defining $N_{p,k}^{\text{all}}$ the total number of particles at step k , *i.e*

$$N_{p,k}^{\text{all}} = \sum_{i \in I_k} N_{p,i}, \quad (\text{B.3})$$

and by $N_{T,k}^{\text{all}} = \beta_{\text{all}} N_{p,k}^{\text{all}}$ with $0 < \beta_{\text{all}} \leq 1$ the threshold for the resampling step, $N_{p,\text{mix}}^{\text{all}}$ particles are resampled from $\hat{p}(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$ if $N_{p,k}^{\text{all}} \leq N_{T,k}^{\text{all}}$. Concerning the number

of particles $N_{p,mix}^{all}$, note that it must be chosen to be smaller than the maximum number of particles $N_{p,max}$ in order to initialize new mixture components for the next iterations. Finally after the resampling procedure the density can be rewritten as follows

$$\begin{aligned}\hat{p}(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k}) &= \frac{1}{N_{p,mix}^{all}} \sum_{n=1}^{N_{p,mix}^{all}} \delta_{\mathbf{x}_k^n, i \in I_k}(\mathbf{x}_k) \\ &= \hat{p}(\mathbf{x}_k | \tau_b \in I_k, \mathbf{z}_{1:k}),\end{aligned}\quad (\text{B.4})$$

which is a "mixture" with one component (*i.e.* $\hat{\alpha}_{k,i \in I_k} = 1$). The probability associated to this mixture is

$$p(\tau_b \in I_k | \mathbf{z}_{1:k}) = \sum_{i \in I_k} p(\tau_b = i | \mathbf{z}_{1:k}). \quad (\text{B.5})$$

For the next iteration, this component is processed as in paragraph 3.2.4.1 and the weight equation (3.34) is almost the same except that $\tau_b = i$ is replaced by $\tau_b \in I_k$ in the equation. As a consequence, a slight difference concerns the propagation of the particles that must be, rigorously speaking, propagate according to $q(\mathbf{x}_{k,i \in I_k}^n | \tau_b \in I_k, \mathbf{x}_{k-1, i \in I_k}, \mathbf{z}_k)$ rather than $q(\mathbf{x}_{k,i} | \tau_b = i, \mathbf{x}_{k-1, i}, \mathbf{z}_k)$. This density can be easily rewritten (following the same reasoning as for the density $p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$ in Eq. (3.16)) as a mixture:

$$q(\mathbf{x}_{k,i \in I_k}^n | \tau_b \in I_k, \mathbf{x}_{k-1, \tau_b \in I_k}, \mathbf{z}_k) = \sum_{i \in I_k} \frac{q(\tau_b = i | \mathbf{z}_k)}{q(i \in I_k | \mathbf{z}_k)} q(\mathbf{x}_k | \tau_b = i, \mathbf{x}_{k-1}, \mathbf{z}_k). \quad (\text{B.6})$$

However, in practice, it is not convenient to sample according to a mixture and, moreover, if the densities $q(\mathbf{x}_k | \tau_b = i, \mathbf{x}_{k-1}, \mathbf{z}_k)$ are the same for all $i \in I_k$, the mixture in Eq. (B.6) simplifies to $q(\mathbf{x}_k | \tau_b = i, \mathbf{x}_{k-1})$ (since the density can be removed from the sum and the probabilities sum to one). Therefore, we propose to approximate the density $q(\mathbf{x}_{k, \tau_b \in I_k}^n | \tau_b \in I_k, \mathbf{x}_{k-1, \tau_b \in I_k}, \mathbf{z}_k)$ by

$$q(\mathbf{x}_{k,i \in I_k}^n | \tau_b \in I_k, \mathbf{x}_{k-1, \tau_b \in I_k}, \mathbf{z}_k) \approx q(\mathbf{x}_{k,i_{all}}^n | \tau_b = i_{all}, \mathbf{x}_{k-1, i_{all}}, \mathbf{z}_k), \quad (\text{B.7})$$

where

$$i_{all} = \arg \max_{i \in I_k} p(\tau_b = i). \quad (\text{B.8})$$

The same approximation can be used for the prior density $p(\mathbf{x}_{k,i \in I_k}^n | \tau_b \in I_k, \mathbf{x}_{k-1, \tau_b \in I_k})$ which is required to evaluate the mixture component weight in Eq. (3.34) – note that it leads to the weight equation (1.99) if the instrumental density is chosen to be the prior. Thus, with this approximation, the mixture for $\tau_b \in I_k$ can be calculated exactly as a mixture component with $\tau_b = i_{all}$ and the algorithm is the same as the one in Algorithm 3.1 except that the number of particles per mixture component may be different: in step "3:" of Algorithm 3.1 $N_{p,mix}$ is replaced by $N_{p,i}$ the number of particles of the i^{th} component (which may vary over time). In the same manner, for the creation of the mixture in Algorithm 3.1, the number of particles $N_{p,mix}$ in step ":14" is replaced by $N_{p,init}$. Note that here $N_{p,init}$ is chosen to be constant at each iteration for a simple implementation but it is not a requirement.

Furthermore, as it was stressed previously, the number $N_{p,mix}^{all}$ must be chosen below the maximum number of particles $N_{p,max}$. Therefore, in the same manner, for a simple

and practicable implementation, we propose to choose $N_{p,mix}^{all} = k_{all}N_{p,init}$ where k_{all} is an integer strictly greater than 1 and $N_{p,max}$ as $N_{p,max} = k_{max}N_{p,init}$, with k_{max} an integer such $k_{max} > k_{all}$. Thus, at the next iterations the particle filter can initialize $k_{max} - k_{all}$ new mixture components with $N_{p,init}$ particles.

However, as for Algorithm 3.1 the number of particles may be equal to $N_{p,max}$ – in particular, if no resampling procedure over all the mixture components has been performed during $k_{max} - k_{all}$ iterations. As a consequence, if nothing is done, no particle is available to initialize a new mixture component for the next iteration. Therefore, in that case it is necessary to remove $N_{p,init}$ particles from the density $p(\mathbf{x}_k | b_k = 1, \mathbf{z}_{1:k})$. To this purpose, we propose to use the same strategy as previously, *i.e.* removing $N_{p,init}$ particles from the component with the lowest probability $\hat{p}(\mathbf{z}_k | \mathbf{z}_{1:k-1})$. Nevertheless, contrary to the previous algorithm, the number of particles in the mixture component i_{min} may be greater (strictly) than $N_{p,init}$ – in practice, it will always be a multiple of $N_{p,init}$. In that case, the mixture does not need to be removed from the set I_k and only $N_{p,init}$ particles can be removed from the mixture component i_{min} . Of course, if the mixture component i_{min} has $N_{p,init}$ particles, this mixture component is removed from the set I_k .

Lastly, an other point has to be discussed, it concerns the minimum number of particles from which the resampling procedure over all the mixture components must be performed. Indeed, let us take the following example, at the first iteration (*i.e.* $k = 1$), a mixture component is initialized with $N_{p,init}$ particles; thus if $N_{eff,1}$ is below $N_{T,1}^{all}$, $k_{all} \times N_{p,init}$ particles will be resampled from this component (which is greater than $N_{p,init}$) whereas this component may have a small probability $p(\tau_b = 1 | \mathbf{z}_1)$ and does not need to be sampled with so many particles. In the same manner, if β_{all} is chosen to be large, the resampling procedure over all the mixture component will be performed almost at each iteration. As a consequence, in the next iterations the algorithm will initialize only one component and resample over all the components whereas if no resampling is performed the algorithm can initialize $k_{max} - k_{all}$ components. To avoid such a situation, we propose to use a two steps resampling procedure depending on the number of particles. First, a severe degeneracy is checked with a β_{all} chosen pretty small (*e.g.* $\beta_{all} = 0.1$) and a resampling procedure is performed if $N_{p,k}^{all} \leq N_{T,k}^{all}$. On the other hand, the number of particles is compared with a number of particles N_p^{min} . If the number of particles is lower, no resampling over all the mixture components is done and each component is resampled separately as in Algorithm 3.1. Whereas, when the number of particles is greater than N_p^{min} two possibilities can arise:

- Either, $N_{p,k}^{all}$ is lower than $N_{T,k}^{min} = \beta_{min}N_{p,k}^{all}$ where β_{min} is chosen greater than β_{all} . Then the resampling procedure is performed over all the mixture components.
- Or, on the other hand, mixture components are resampling separately.

Finally, the proposed Resample All Appearance Time TBD Particle Filter is summarized by Algorithm B.1.

Algorithm B.1 Resample All Appearance Time TBD Particle Filter

Require: mixture components $\{w_{k-1}^i, \mathbf{x}_{k-1,i}^n\}_{n=1}^{N_{p,i}}$ and probabilities $p(\tau_b = i \mid \mathbf{z}_{1:k-1})$ with $i \in I_{k-1}$ at step $k-1$.

- 1: Updating mixture components $\{w_{k-1}^i, \mathbf{x}_{k-1,i}^n\}_{n=1}^{N_{p,i}}$ and probabilities $p(\tau_b = i \mid \mathbf{z}_{1:k-1})$ from line 1 to 23 in Algorithm 3.1 where $N_{p,mix}$ is replaced by the corresponding number of particles in each component.
 - 2: Calculate $N_{\text{eff},k}^{\text{all}}$ according to Eq. (B.2) and $N_{T,k}^{\text{all}}$.
 - 3: **if** $N_{\text{eff},k}^{\text{all}} < N_{T,k}^{\text{all}}$ **then**
 - 4: Resample $N_{p,\text{all}}$ from $\hat{p}(\mathbf{x}_k \mid b_k = 1, \mathbf{z}_{1:k})$.
 - 5: Calculate i_{all} according to Eq. (B.8).
 - 6: **else**
 - 7: **if** $N_{p,k}^{\text{all}} \geq N_p^{\text{min}}$ and $N_{\text{eff},k}^{\text{all}} < N_{T,k}^{\text{min}}$ **then**
 - 8: Resample $N_{p,\text{all}}$ from $\hat{p}(\mathbf{x}_k \mid b_k = 1, \mathbf{z}_{1:k})$.
 - 9: Calculate i_{all} according to Eq. (B.8).
 - 10: **else**
 - 11: **if** $N_{p,k}^{\text{all}} = N_{p,\text{max}}$ **then**
 - 12: Find i_{min} according to (3.48).
 - 13: **if** $N_{p,i_{\text{min}}} = N_{p,\text{init}}$ **then**
 - 14: Set $I_k = I_{k,\text{min}}$.
 - 15: **end if**
 - 16: **for** $i \in I_k$ **do**
 - 17: **if** $i = i_{\text{min}}$ and $N_{p,i_{\text{min}}} > N_{p,\text{init}}$ **then**
 - 18: Resample $N_{p,i} - N_{p,\text{init}}$ particles.
 - 19: Reset weights: $w_{k,i}^n \leftarrow \frac{1}{N_{p,i} - N_{p,\text{init}}}$ $n = 1, \dots, N_{p,i} - N_{p,\text{init}}$.
 - 20: **else**
 - 21: Calculate $N_{T,i} = \beta N_{p,i}$
 - 22: **if** $N_{\text{eff},i} < N_{T,i}$ **then**
 - 23: Resample $N_{p,i}$ particles.
 - 24: Reset weights: $w_{k,i}^n \leftarrow \frac{1}{N_{p,mix}}$ $n = 1, \dots, N_{p,i}$.
 - 25: **end if**
 - 26: **end if**
 - 27: **end for**
 - 28: **end if**
 - 29: **end if**
 - 30: **end if**
- Ensure:** $\{w_{k,i}^n, \mathbf{x}_{k,i}^n\}_{n=1}^{N_{p,mix}}$, $p(\tau_b = i \mid \mathbf{z}_{1:k})$, $i \in I_k$.
-

Appendix C

Multitarget Bayesian filter and particle filters

C.1 Theoretical Bayesian filter for interacting targets

The aim of this appendix is to demonstrate that, if the groups of targets $I_{int,1}, \dots, I_{int,N_g}$ do not interact in the likelihood, the posterior multitarget density can be factorized as follows:

$$p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} \mid \mathbf{z}_{1:k}) = \prod_{i=1}^{N_g} p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} \mid \mathbf{z}_{1:k}). \quad (\text{C.1})$$

First, the Bayesian filter for each group of targets can be derived, using Bayes rule, as follows:

$$p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} \mid \mathbf{z}_{1:k}) = \frac{p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} \mid \mathbf{z}_{1:k-1}) p(\mathbf{z}_k \mid s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})}. \quad (\text{C.2})$$

It should be noted here that the conditioning over variables $\mathbf{z}_{1:k-1}$ in the likelihood expression $p(\mathbf{z}_k \mid s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}, \mathbf{z}_{1:k-1})$ cannot be removed since the conditioning is not performed with respect to all the target states $(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})$. In fact, \mathbf{z}_k is independent from $\mathbf{z}_{1:k-1}$ only when

$$p(\mathbf{z}_k \mid s_{k,1:N_t}, \mathbf{x}_{k,1:N_t}, \mathbf{z}_{1:k-1}) = p(\mathbf{z}_k \mid s_{k,1:N_t}, \mathbf{x}_{k,1:N_t}). \quad (\text{C.3})$$

Fortunately, it does not affect the derivation of the Bayes filter. Indeed, some simplifications arises allowing to easily calculate the likelihood $p(\mathbf{z}_k \mid s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}, \mathbf{z}_{1:k-1})$. To this purpose, notice first that this likelihood can be rewritten as follows:

$$\begin{aligned} & p(\mathbf{z}_k \mid s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}, \mathbf{z}_{1:k-1}) \\ &= \sum_{s_{k,I_{N_t} \setminus I_{int,i}}} \int p(s_{k,I_{N_t} \setminus I_{int,i}}, \mathbf{x}_{k,I_{N_t} \setminus I_{int,i}} \mid s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}, \mathbf{z}_{1:k-1}) \times \\ & \quad p(\mathbf{z}_k \mid s_{k,1:N_t}, \mathbf{x}_{k,1:N_t}) d\mathbf{x}_{k,I_{N_t} \setminus I_{int,i}}. \end{aligned} \quad (\text{C.4})$$

This last equation can be further simplified using Eq. (C.1) since $(s_{k,I_{N_t} \setminus I_{int,i}}, \mathbf{x}_{k,I_{N_t} \setminus I_{int,i}})$ and $(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}})$ are independent. Thus Eq. (C.4) becomes

$$\begin{aligned} & p(\mathbf{z}_k \mid s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}, \mathbf{z}_{1:k-1}) \\ &= \sum_{s_{k,I_{N_t} \setminus I_{int,i}}} \int p(s_{k,I_{N_t} \setminus I_{int,i}}, \mathbf{x}_{k,I_{N_t} \setminus I_{int,i}} \mid \mathbf{z}_{1:k-1}) p(\mathbf{z}_k \mid s_{k,1:N_t}, \mathbf{x}_{k,1:N_t}) d\mathbf{x}_{k,I_{N_t} \setminus I_{int,i}}. \end{aligned} \quad (\text{C.5})$$

Furthermore, using Eq. (5.47), the Eq. (5.25) can be rewritten as follows:

$$\begin{aligned} & \Xi_{\mathbf{z}_k, (s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})}(\rho_{k,1:N_t}, \varphi_{k,1:N_t}) = \\ & \Xi_{\mathbf{z}_k, (s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}})}(\rho_{k,I_{int,i}}, \varphi_{k,I_{int,i}}) \Xi_{\mathbf{z}_k, (s_{k,I_{N_t} \setminus I_{int,i}}, \mathbf{x}_{k,I_{N_t} \setminus I_{int,i}})}(\rho_{k,I_{N_t} \setminus I_{int,i}}, \varphi_{k,I_{N_t} \setminus I_{int,i}}). \end{aligned} \quad (\text{C.6})$$

Thus by defining for any set of indexes I the following function¹

$$G_{\mathbf{z}_k}(s_{k,I}, \mathbf{x}_{k,I}) = \int \Xi_{\mathbf{z}_k, (s_{k,I}, \mathbf{x}_{k,I})}(\rho_{k,I}, \varphi_{k,I}) p(\rho_{k,I}) p(\varphi_{k,I}) d\rho_{k,I} d\varphi_{k,I}, \quad (\text{C.7})$$

the likelihood $p(\mathbf{z}_k \mid s_{k,1:N_t}, \mathbf{x}_{k,1:N_t})$ in Eq. (5.26) factorizes in the following manner:

$$p(\mathbf{z}_k \mid s_{k,1:N_t}, \mathbf{x}_{k,1:N_t}) = G_{\mathbf{z}_k}(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}) \times G_{\mathbf{z}_k}(s_{k,I_{N_t} \setminus I_{int,i}}, \mathbf{x}_{k,I_{N_t} \setminus I_{int,i}}). \quad (\text{C.8})$$

Therefore, by injecting Eq. (C.8) in Eq. (C.4), the likelihood $p(\mathbf{z}_k \mid s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}, \mathbf{z}_{1:k-1})$ also factorizes as follows:

$$\begin{aligned} & p(\mathbf{z}_k \mid s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}, \mathbf{z}_{1:k-1}) = G_{\mathbf{z}_k}(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}) \times \\ & \sum_{s_{k,I_{N_t} \setminus I_{int,i}}} \int p(s_{k,I_{N_t} \setminus I_{int,i}}, \mathbf{x}_{k,I_{N_t} \setminus I_{int,i}} \mid \mathbf{z}_{1:k-1}) G_{\mathbf{z}_k}(s_{k,I_{N_t} \setminus I_{int,i}}, \mathbf{x}_{k,I_{N_t} \setminus I_{int,i}}) d\mathbf{x}_{k,I_{N_t} \setminus I_{int,i}}. \end{aligned} \quad (\text{C.9})$$

In the same manner, using the same reasoning as in Eq. (5.42) the normalization terms $p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})$ can be factorized as follows:

$$\begin{aligned} & p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1}) = \\ & \sum_{s_{k,I_{int,i}}} \int p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} \mid \mathbf{z}_{1:k-1}) G_{\mathbf{z}_k}(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}) d\mathbf{x}_{k,I_{int,i}} \times \\ & \sum_{s_{k,I_{N_t} \setminus I_{int,i}}} \int p(s_{k,I_{N_t} \setminus I_{int,i}}, \mathbf{x}_{k,I_{N_t} \setminus I_{int,i}} \mid \mathbf{z}_{1:k-1}) G_{\mathbf{z}_k}(s_{k,I_{N_t} \setminus I_{int,i}}, \mathbf{x}_{k,I_{N_t} \setminus I_{int,i}}) d\mathbf{x}_{k,I_{N_t} \setminus I_{int,i}}. \end{aligned} \quad (\text{C.10})$$

Finally, injecting Eq. (C.9) and Eq. (C.10), the bayesian filter for the group of target $I_{int,i}$ in Eq. (C.2) simplifies as follows:

$$p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} \mid \mathbf{z}_{1:k}) = \frac{p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} \mid \mathbf{z}_{1:k-1}) G_{\mathbf{z}_k}(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}})}{\sum_{s_{k,I_{int,i}}} \int p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} \mid \mathbf{z}_{1:k-1}) G_{\mathbf{z}_k}(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}}) d\mathbf{x}_{k,I_{int,i}}}. \quad (\text{C.11})$$

¹Note that here, contrary to the definition of function $g_{\mathbf{z}_k}(s_{k,i}, \mathbf{x}_{k,i})$ in Eq. (5.29) where there is no cross terms $\mathbf{h}_{k,u}^H \mathbf{\Gamma}^{-1} \mathbf{h}_{k,v}$, in the definition of the following function all the cross terms remains.

In fact, this last equation is quite similar to the Eq. (5.44) (*i.e.* the bayesian filter for a single non interacting target) and demonstrates that factorization of the posterior density $p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} \mid \mathbf{z}_{1:k})$ can be extended to sets of indexes of non interacting targets, *i.e.*

$$p(s_{k,1:N_t}, \mathbf{x}_{k,1:N_t} \mid \mathbf{z}_{1:k}) = \prod_{i=1}^{N_g} p(s_{k,I_{int,i}}, \mathbf{x}_{k,I_{int,i}} \mid \mathbf{z}_{1:k}). \quad (\text{C.12})$$

C.2 Disappearance multitarget detection particle filter

The aim of this appendix is to detail the practical implementation of the disappearance Multitarget detection particle filter outlined in section 5.4.1.2. The proposed solution requires to determine at each iteration k the interacting sets $I_{int,1:N_g}$ and I_{sing} in order to reorganize the densities calculated at previous step for these sets.

C.2.1 Calculating the sets I_{sing} and $I_{int,1:N_g}$ over time

In order to calculate the single target set I_{sing} and the sets of interacting group $I_{int,1:N_g}$, let us first denote by $I_{int,1:N_g}^-$ the group of interacting targets at iteration $k-1$ (where N_g^- is the number of groups at iteration $k-1$) and I_{sing}^- the index of single targets (*i.e.* those targets that do not interact). At previous iteration $k-1$, the available particle approximations are $\hat{p}(\mathbf{x}_{k-1,I_{int,i}^-} \mid s_{k-1,I_{int,i}^-} = 1, \mathbf{z}_{1:k-1})$ with $i \in \{1, \dots, N_g^-\}$ and $\hat{p}(s_{k-1,l}, \mathbf{x}_{k-1,l} \mid \mathbf{z}_{1:k-1})$ with $l \in I_{sing}^-$. A first possible solution to calculate the interacting groups at current step might be to propagate the particles of each target state $\mathbf{x}_{k-1,i}^p$ ($i \in \{1, \dots, N_t\}$) according to their prior $p_c(\mathbf{x}_k \mid \mathbf{x}_{k-1})$, *i.e.*

$$\mathbf{x}_{k,i}^p \sim p_c(\mathbf{x}_k \mid \mathbf{x}_{k-1,i}^p). \quad (\text{C.13})$$

Then, two targets states $\mathbf{x}_{k,l}$ and $\mathbf{x}_{k,m}$ are declared “interacting” if

$$\text{there exist } (p, q) \in \{1, \dots, N_p\}^2, \text{ such that } |\mathbf{h}^H(\mathbf{x}_{k,l}^p) \mathbf{\Gamma}^{-1} \mathbf{h}(\mathbf{x}_{k,m}^q)| > \gamma_{\mathbf{h}}, \quad (\text{C.14})$$

where $\gamma_{\mathbf{h}}$ is a given positive threshold (eventually equal to zero).

In other words, two targets are declared to be interacting at step k if there exists at least one particle $\mathbf{x}_{k,l}^p$ and one particle $\mathbf{x}_{k,m}^q$ whose positions lead to overlapping ambiguity functions. Note that here we consider interacting states as soon as one pair of particles (p, q) interacts. Of course, this condition can be extended to a minimum number of particles, *i.e.* two targets states can be considered interacting only if a significant minimal number of pairs of particles interact.

However, such a solution might require to evaluate $N_p \times N_p$ (for two target states) scalar products $|\mathbf{h}^H(\mathbf{x}_{k,l}^p) \mathbf{\Gamma}^{-1} \mathbf{h}(\mathbf{x}_{k,m}^q)|$, for the two target case. As a consequence, if the number of targets N_t is large, such a procedure might be costly in terms of computational resources. Thus, in order to alleviate the number of scalar product evaluations, we propose to perform the procedure on the estimated predicted target states $\hat{\mathbf{x}}_{k|k-1,i}$ rather than on all particle target states. To calculate each predicted target states $\hat{\mathbf{x}}_{k|k-1,i}$, we propose a

very simple solution based on the Eq. (1.67) of the Kalman filter equations. Thus, for each target state $\mathbf{x}_{k,i}$ ($i \in \{1, \dots, N_t\}$), the estimated predicted target state $\hat{\mathbf{x}}_{k|k-1,i}$ is calculated as follows:

$$\hat{\mathbf{x}}_{k|k-1,i} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1,i}, \quad (\text{C.15})$$

where \mathbf{F} is the state matrix defined in Eq. (2.6) and $\hat{\mathbf{x}}_{k-1|k-1,i}$ is the estimated target state provided by Eq. (1.96). Finally, targets l and m are declared to be interacting at current step if

$$|\mathbf{h}^H(\hat{\mathbf{x}}_{k|k-1,l})\mathbf{\Gamma}^{-1}\mathbf{h}(\hat{\mathbf{x}}_{k|k-1,m})| > \gamma_{\mathbf{h}}. \quad (\text{C.16})$$

Now, it remains first to calculate the new interacting groups $I_{int,1:N_g}$ and the single target group I_{sing} and then to calculate the particle posterior density approximations for the groups $I_{int,1:N_g}$ and I_{sing} .

Concerning the calculation of the groups $I_{int,1:N_g}$ and I_{sing} , we propose a two-step solution:

- First, find interactions between targets for all possible pairs of targets.
- Then regroup pairs of interacting targets in order to create the groups $I_{int,1:N_g}$.

In order to detail our procedure, let us first define, for a matrix \mathbf{M} of size $N \times M$, by $\mathbf{M}(n, \cdot)$ the n -th row of the matrix and by $\mathbf{M}(\cdot, m)$ its m -th column. To calculate the interactions between pairs of targets, we propose to use a matrix \mathbf{M} (of size $N_t \times N_t$) where each element (l, m) represents a possible interaction between two targets as follows:

$$\mathbf{M}(l, m) = \begin{cases} 1, & \text{if target } l \text{ and } m \text{ interact,} \\ 0, & \text{otherwise.} \end{cases} \quad (\text{C.17})$$

This matrix is symmetric and therefore, it is only necessary to consider the upper or lower part of matrix \mathbf{M} . Moreover, by convention we consider that a target cannot interact with itself, *i.e.* $\mathbf{M}(m, m) = 0$.

Lastly, it remains to calculate the interacting groups $I_{int,1:N_g}$ and the single target group I_{sing} from the matrix \mathbf{M} . This can be done as follows:

- For each row l of the matrix, find the indexes m such that $\mathbf{M}(l, m) = 1$, then regroup these indexes in a set I_{col} .
- If the set I_{col} is empty, it means that the target state l does not interact with any target. Therefore the target l is added to the set I_{sing} .
- If the set I_{col} is not empty, two cases must be considered. In the first case, the target l already belongs to one of the sets $I_{int,1:N_g}$, referred by index $i_{g,l}$. Then the sets $I_{int,i_{g,l}}$ and I_{col} are "merged", *i.e.* $I_{int,i_{g,l}} = I_{int,i_{g,l}} \cup I_{col}$. In the second case, the target l does not belong to any of the sets $I_{int,1:N_g}$. Therefore a new group of interacting targets must be created, *i.e.* $I_{int,N_g+1} = I_{col} \cup \{l\}$. Note that the target must be added to the set I_{int,N_g+1} since by convention it does belong to the set I_{col} . Moreover the number of groups must be updated, *i.e.* $N_g = N_g + 1$.

C.2.2 Reorganization of the particle posterior density at previous step for the sets I_{sing} and $I_{int,1:N_g}$

The last step consists in reorganizing the particle posterior density approximations at previous step: $\hat{p}(\mathbf{x}_{k-1, I_{int,i}^-} | s_{k-1, I_{int,i}^-} = 1, \mathbf{z}_{1:k-1})$ with $i \in \{1, \dots, N_g^-\}$ and $\hat{p}(\mathbf{x}_{k-1,l} | s_{k-1,l} = 1, \mathbf{z}_{1:k-1})$ with $l \in I_{sing}^-$ to obtain the ones for the group $I_{int,1:N_g}$ and I_{sing} . Indeed, the considered multitarget particle filter considers multitarget states, i.e. one particle samples all considered target states within a group. For target states originated from different groups at step $k-1$ and gathered in the same group at step k , it is necessary to resample the corresponding states so as to create new particles that sample the new multitarget state. To this purpose, we propose the following rules:

- For any target index l in I_{sing} , if l also belongs to I_{sing}^- , then the target was single at previous step and is still single at current step. Therefore, there is nothing to do.
- In the same manner, for any groups of targets $I_{int,l}$, if there exists a group of targets $I_{int,m}^-$ such that $I_{int,l} == I_{int,m}^-$ (where $==$ stands for the equality between sets), there is nothing to do.
- In the other cases, the posterior density must be recalculated from the previous sets $I_{int,1:N_g}^-$ and I_{sing}^- . To this purpose, we simply propose to resample N_p particles for each target index l in the new group $I_{int,m}$ from the density provided by the previous sets $I_{int,1:N_g}^-$ and I_{sing}^- .

The pseudo-code for the corresponding algorithm is detailed in Algorithm C.1.

C.2.3 Proposed solution for Disappearance multitarget particle filter

In the previous paragraph, the tools to derive our particle filter solution for Disappearance multitarget particle filter have been detailed. Now, it remains to expose the different steps in order to perform a single recursion of our particle filter. Let us assume that at step $k-1$, the following quantities are available: $\hat{p}(\mathbf{x}_{k-1, I_{int,i}^-} | s_{k-1, I_{int,i}^-} = 1, \mathbf{z}_{1:k-1})$ with $i \in \{1, \dots, N_g^-\}$ and $\hat{p}(\mathbf{x}_{k-1,l} | s_{k-1,l} = 1, \mathbf{z}_{1:k-1})$ with $l \in I_{sing}^-$.

The proposed solution can be derived as follows:

- First, calculate matrix \mathbf{M} with Algorithm.
- Then, calculate the new groups $I_{int,1:N_g}$ and I_{sing} from matrix \mathbf{M} .
- Calculate the densities $\hat{p}(\mathbf{x}_{k-1, I_{int,i}} | s_{k-1, I_{int,i}} = 1, \mathbf{z}_{1:k-1})$ $i \in \{1, \dots, N_g\}$ and $\hat{p}(\mathbf{x}_{k-1,l} | s_{k-1,l} = 1, \mathbf{z}_{1:k-1})$ $l \in I_{sing}$ from the ones with sets $I_{int,1:N_g}^-$ and I_{sing}^- .
- Update weights with Eq. (5.53) and Eq. (5.55).
- Finally, each density is resampled if need.

In Algorithm C.2, we give a pseudo-code algorithm of the proposed particle recursion.

Algorithm C.1 Calculation of the densities $\hat{p}(\mathbf{x}_{k-1, I_{int,i}} | s_{k-1, I_{int,i}} = 1, \mathbf{z}_{1:k-1})$ $i \in \{1, \dots, N_g\}$ and $\hat{p}(\mathbf{x}_{k-1, l} | s_{k-1, l} = 1, \mathbf{z}_{1:k-1})$ $l \in I_{sing}$ from the ones with sets $I_{int, 1:N_g}^-$ and I_{sing}^- .

Require: densities $\hat{p}(\mathbf{x}_{k-1, I_{int,i}^-} | s_{k-1, I_{int,i}^-} = 1, \mathbf{z}_{1:k-1})$ $i \in \{1, \dots, N_g\}$ and $\hat{p}(\mathbf{x}_{k-1, l} | s_{k-1, l} = 1, \mathbf{z}_{1:k-1})$ $l \in I_{sing}^-, I_{int, 1:N_g}^-$ and I_{sing} .

- 1: **for** $l \in I_{sing}$ **do**
- 2: **if** l belongs to I_{sing}^- **then**
- 3: Keep the density $\hat{p}(\mathbf{x}_{k-1, l} | s_{k-1, l} = 1, \mathbf{z}_{1:k-1})$
- 4: **else**
- 5: Get the index $i_{g,l}$ of the group $I_{int, 1:N_g}^-$ such that $l \in I_{int, 1:N_g}^-$.
- 6: Resample $\{\mathbf{x}_{k-1, l}^p\}_{p=1}^{N_p}$ from density $\hat{p}(\mathbf{x}_{k-1, I_{int, i_{g,l}}^-} | s_{k-1, I_{int, i_{g,l}}^-} = 1, \mathbf{z}_{1:k-1})$
- 7: Set $\hat{p}(\mathbf{x}_{k-1, l} | s_{k-1, l} = 1, \mathbf{z}_{1:k-1}) = \frac{1}{N_p} \sum_{p=1}^{N_p} \delta_{\mathbf{x}_{k-1, l}^p}(\mathbf{x}_{k-1, l})$
- 8: **end if**
- 9: **end for**
- 10: **for** $i \in \{1, \dots, N_g\}$ **do**
- 11: Check if there is a group I_{int, i_g}^- such that $I_{int, i} == I_{int, i_g}^-$, if so get the index i_g^-
- 12: **if** i_g^- exists **then**
- 13: Set the density $\hat{p}(\mathbf{x}_{k-1, I_{int, i}} | s_{k-1, I_{int, i}} = 1, \mathbf{z}_{1:k-1}) = \hat{p}(\mathbf{x}_{k-1, I_{int, i_g}^-} | s_{k-1, I_{int, i_g}^-} = 1, \mathbf{z}_{1:k-1})$
- 14: **else**
- 15: Initialize new particle target state $\mathbf{x}_{k-1, I_{int, i}}^p = []$ (empty vector), $p = 1, \dots, N_p$
- 16: **for** $l \in I_{int, i}$ **do**
- 17: **if** l belongs to I_{sing}^- **then**
- 18: Resample $\{\mathbf{x}_{k-1, l}^p\}_{p=1}^{N_p}$ from density $\hat{p}(\mathbf{x}_{k-1, l} | s_{k-1, l} = 1, \mathbf{z}_{1:k-1})$
- 19: Concatenate the state $\mathbf{x}_{k-1, l}^p$ to $\mathbf{x}_{k-1, I_{int, i}}^p$: $\mathbf{x}_{k-1, I_{int, i}}^p = [\mathbf{x}_{k-1, I_{int, i}}^p; \mathbf{x}_{k-1, l}^p]$, $p = 1, \dots, N_p$
- 20: **else**
- 21: Find index $i_{g,l}$ such that $l \in I_{int, i_{g,l}}^-$
- 22: Resample $\{\mathbf{x}_{k-1, l}^p\}_{p=1}^{N_p}$ from density $\hat{p}(\mathbf{x}_{k-1, I_{int, i_{g,l}}^-} | s_{k-1, I_{int, i_{g,l}}^-} = 1, \mathbf{z}_{1:k-1})$
- 23: Concatenate the state $\mathbf{x}_{k-1, l}^p$ to $\mathbf{x}_{k-1, I_{int, i}}^p$: $\mathbf{x}_{k-1, I_{int, i}}^p = [\mathbf{x}_{k-1, I_{int, i}}^p; \mathbf{x}_{k-1, l}^p]$, $p = 1, \dots, N_p$
- 24: **end if**
- 25: **end for**
- 26: Set $\hat{p}(\mathbf{x}_{k-1, I_{int, i}} | s_{k-1, I_{int, i}} = 1, \mathbf{z}_{1:k-1}) = \frac{1}{N_p} \sum_{p=1}^{N_p} \delta_{\mathbf{x}_{k-1, I_{int, i}}^p}(\mathbf{x}_{k-1, I_{int, i}})$
- 27: **end if**
- 28: **end for**
- 29: **return** $\hat{p}(\mathbf{x}_{k-1, I_{int, i}} | s_{k-1, I_{int, i}} = 1, \mathbf{z}_{1:k-1})$, $i \in \{1, \dots, N_g\}$ and $\hat{p}(\mathbf{x}_{k-1, l} | s_{k-1, l} = 1, \mathbf{z}_{1:k-1})$, $l \in I_{sing}$

Algorithm C.2 Proposed disappearance multitarget particle filter.

- Require:** densities $\hat{p}(\mathbf{x}_{k-1, I_{int,i}^-} | s_{k-1, I_{int,i}^-} = 1, \mathbf{z}_{1:k-1})$ $i \in \{1, \dots, N_g^-\}$ and $\hat{p}(\mathbf{x}_{k-1, l} | s_{k-1, l} = 1, \mathbf{z}_{1:k-1})$, $\hat{p}(s_{k, l} = 1 | \mathbf{z}_{1:k})$, $l \in I_{sing}^-$.
- 1: Calculate matrix \mathbf{M} .
 - 2: Calculate the new groups $I_{int, 1:N_g}$ and I_{sing} from matrix \mathbf{M} with Algorithm ??.
 - 3: Calculation of the densities $\hat{p}(\mathbf{x}_{k-1, I_{int,i}} | s_{k-1, I_{int,i}} = 1, \mathbf{z}_{1:k-1})$ $i \in \{1, \dots, N_g\}$ and $\hat{p}(\mathbf{x}_{k-1, l} | s_{k-1, l} = 1, \mathbf{z}_{1:k-1})$ $l \in I_{sing}$ from the ones with sets $I_{int, 1:N_g}^-$ and I_{sing}^- .
 - 4: **for** $l \in I_{sing}$ **do**
 - 5: Propagate particles $\mathbf{x}_{k, l}^p \sim q_c(\mathbf{x}_{k, l} | \mathbf{x}_{k-1, l}^p, \mathbf{z}_k)$
 - 6: Update weights $w_{k, l}^p$ with Eq. (5.53)
 - 7: Calculate $\hat{p}(s_{k, l} = 1 | \mathbf{z}_{1:k})$ with Eq. (3.76)
 - 8: Calculate N_{eff} and resample if needed.
 - 9: **end for**
 - 10: **for** $i \in \{1, \dots, N_g\}$ **do**
 - 11: Propagate particles $\mathbf{x}_{k, I_{int,i}}^p \sim \prod_{l \in I_{int,i}} p_c(\mathbf{x}_{k, l} | \mathbf{x}_{k-1, l}^p, \mathbf{z}_k)$
 - 12: Update weights $w_{k, I_{int,i}}^p$ with Eq. (5.55)
 - 13: **end for**
 - 14: Calculate N_{eff} and resample if needed.
 - 15: **return** $\hat{p}(\mathbf{x}_{k, I_{int,i}} | s_{k, I_{int,i}} = 1, \mathbf{z}_{1:k})$, $i \in \{1, \dots, N_g\}$ and $\hat{p}(\mathbf{x}_{k, l} | s_{k, l} = 1, \mathbf{z}_{1:k})$, $\hat{p}(s_{k, l} = 1 | \mathbf{z}_{1:k})$, $l \in I_{sing}$
-

Bibliography

- [ADST04] C. Andrieu, A. Doucet, S.S. Singh, and V.B. Tadic. Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438, 2004.
- [AM79] B. Anderson and J. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ, 1979.
- [AMGC02] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, Feb 2002.
- [Bar85] Y. Barniv. Dynamic programming solution for detecting dim moving targets. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-21(1):144–156, Jan 1985.
- [BDV⁺03] Y. Boers, J.N. Driessen, F. Verschure, W.P.M.H. Heemels, and A. Juloski. A multi target track before detect application. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on*, volume 9, pages 104–104, June 2003.
- [Bla86] S.S. Blackman. *Multiple-target tracking with radar applications*. Artech House radar library. Norwood, Mass. Artech House, 1986.
- [BLRLG11] M. Bocquel, A. Lepoutre, O. Rabaste, and F. Le Gland. Optimisation d’un filtre particulaire en contexte track-before-detect. In *Actes du 23ème Colloque sur le Traitement du Signal et des Images*, Bordeaux, France, September 2011.
- [BS87] Y. Bar-Shalom. *Tracking and Data Association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.
- [BS09] J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. Wiley Series in Probability and Statistics. Wiley, 2009.
- [BSCS89] Y. Bar-Shalom, K.C. Chang, and H.M. Shertukde. Performance evaluation of a cascaded logic for track formation in clutter. In *Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE International Conference on*, pages 13–17 vol.1, Nov 1989.

- [BSLK01] Y. Bar-Shalom, X. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. John Wiley & Sons, New York, 2001.
- [BST75] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451 – 460, 1975.
- [Cas76] F.R. Castella. Sliding window detection probabilities. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-12(6):815–819, Nov 1976.
- [CD02] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on*, 50(3):736–746, Mar 2002.
- [CEW94] B.D. Carlson, E.D. Evans, and S.L. Wilson. Search radar detection and track with the Hough transform. I. System concept. *Aerospace and Electronic Systems, IEEE Transactions on*, 30(1):102–108, Jan 1994.
- [Dar94] J. Darricau. *Physique et théorie du radar: Concepts de traitement du signal*. Number vol. 3 in *Physique et théorie du radar: Concepts de traitement du signal*. Sodipec, 1994.
- [DdFG01] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Information Science and Statistics. Springer, 2001.
- [DGA00] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, July 2000.
- [DM04] P. Del Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer series in statistics. Springer, New York, 2004.
- [DRC08] S.J. Davey, M.G. Rutten, and B. Cheung. A comparison of detection performance for several track-before-detect algorithms. In *Proceedings of the 11th International Conference on Information Fusion*, pages 1–8, June 2008.
- [DRC12] S.J. Davey, M.G. Rutten, and B. Cheung. Using phase to improve track-before-detect. *Aerospace and Electronic Systems, IEEE Transactions on*, 48(1):832–849, Jan 2012.
- [FBSS83] Thomas E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *Oceanic Engineering, IEEE Journal of*, 8(3):173–184, Jul 1983.
- [GF11] A. F. Garcia-Fernandez. *Detection and Tracking of Multiple Targets using Wireless Sensor Networks*. PhD thesis, Polytechnic University of Madrid, 2011.
- [GR07] I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products*. Elsevier/Academic Press, Amsterdam, seventh edition, 2007.

- [GSS93] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, Apr 1993.
- [Har78] F.J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1):51–83, Jan. 1978.
- [HF09] Robert Hastie, T. Tibshirani and J. Friedman. *The elements of statistical learning: Data mining, inference and prediction*. Springer Series in Statistics. Springer, New-York, 2nd edition, 2009.
- [Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [Kay98] S.M. Kay. *Fundamentals of Statistical Signal Processing: Detection theory*. Prentice Hall Signal Processing Series. Prentice-Hall PTR, 1998.
- [Kit96] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.
- [Kit98] G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215, September 1998.
- [KKH05] C. Kreucher, K. Kastella, and A.O. Hero. Multitarget Tracking using the Joint Multitarget Probability Density. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(4):1396–1414, 2005.
- [KRT98] S. Kligys, B. Rozovsky, and A. Tartakovsky. Detection algorithms and track before detect architecture based on nonlinear filtering for infrared search and track systems. Technical Report CAMS–98.9.1, Center for Applied Mathematical Sciences, University of Southern California, Sep 1998.
- [LC89] F. Le Chevalier. *Principes de traitement des signaux radar et sonar*. Collection technique et scientifique des télécommunications. Masson, Paris, Milan, Barcelone, 1989.
- [LM04] N. Levanon and E. Mozeson. *Radar signals*. John Wiley and Sons, 2004.
- [LR98] J.S. Liu and C. Rong. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.
- [LRG13] A. Lepoutre, O. Rabaste, and F. Le Gland. Exploiting amplitude spatial coherence for multi-target particle filter in track-before-detect. In *Information Fusion (FUSION), 16th International Conference on*, pages 319–326, Istanbul, Turkey, July 2013.

- [LRLG12a] A. Lepoutre, O. Rabaste, and F. Le Gland. Optimized instrumental density for particle filter in track-before-detect. In *Data Fusion Target Tracking Conference : Algorithms Applications, 9th IET*, pages 1–6, London, United Kingdom, May 2012.
- [LRLG12b] A. Lepoutre, O. Rabaste, and F. Le Gland. A particle filter for target arrival detection and tracking in track-before-detect. In *Sensor Data Fusion: Trends, Solutions, Applications (SDF), Workshop on*, pages 13–18, Bonn, Germany, Sept. 2012.
- [LRLG13] A. Lepoutre, O. Rabaste, and F. Le Gland. Filtres particuliers en contexte track-before-detect en présence de fluctuations d’amplitude de type Swerling 1 et 3. In *Actes du 24ème Colloque sur le Traitement du Signal et des Images*, Brest, France, September 2013. GRETSI.
- [LRLG16] A. Lepoutre, O. Rabaste, and F. Le Gland. Multitarget likelihood computation for track-before-detect applications with amplitude fluctuations of type Swerling 0, 1, and 3. *Aerospace and Electronic Systems, IEEE Transactions on*, June 2016.
- [Mal03] R.K. Mallik. On multivariate Rayleigh and exponential distributions. *Information Theory, IEEE Transactions on*, 49(6):1499–1515, June 2003.
- [MB08] M. McDonald and B. Balaji. Track-before-detect using Swerling 0, 1, and 3 target models for small manoeuvring maritime targets. *EURASIP J. Adv. Sig. Proc.*, 2008.
- [MBQLG11] C. Musso, P. Bui Quang, and F. Le Gland. Introducing the Laplace approximation in particle filtering. In *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, July 2011.
- [MOLG01] C. Musso, N. Oudjane, and F. Le Gland. Improving regularized particle filters. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo methods in practice*, Statistics for Engineering and Information Science, pages 247–271. Springer, 2001.
- [Mur12] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive computation and machine learning series. MIT Press, 2012.
- [OF02] M. Orton and W. Fitzgerald. A Bayesian approach to tracking multiple targets using sensor arrays and particle filters. *Signal Processing, IEEE Transactions on*, 50(2):216–223, Feb 2002.
- [Owe13] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.
- [RAG04] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter. Particle filters for tracking applications*. Artech House, 2004.
- [Rei79] Donald B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24:843–854, 1979.

- [RGM04] M.G. Rutten, N.J. Gordon, and S. Maskell. Efficient particle-based track-before-detect in Rayleigh noise. In *Proceedings of the Seventh International Conference on Information Fusion*, volume 4, pages 693–700, June 2004.
- [Ric07] M.A. Richards. Relationship between the gamma, Erlang, chi-square, and Swerling 3/4 probability density functions. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.579.4593&rep=rep1&type=pdf>, Aug. 2007.
- [Rih69] A.W. Rihaczek. *Principles of high-resolution radar*. McGraw-Hill, 1969.
- [RRG05] M.G. Rutten, B. Ristic, and N.J. Gordon. A comparison of particle filters for recursive track-before-detect. In *Proceedings of the Seventh International Conference on Information Fusion*, pages 169–175, 2005.
- [RRL12] O. Rabaste, C. Riché, and A. Lepoutre. Long-time coherent integration for low SNR target via particle filter in track-before-detect. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 127–134, July 2012.
- [SB01] D.J. Salmond and H. Birch. A particle filter for track-before-detect. In *Proceedings of American Control Conference*, pages 3755–3760, 2001.
- [Sko80] M.I. Skolnik. *Introduction to Radar Systems*. McGraw Hill Book Co., New York, 2 edition, 1980.
- [SSH74] R. Singer, R. Sea, and K. Housewright. Derivation and evaluation of improved tracking filter for use in dense multitarget environments. *IEEE Trans. Inf. Theor.*, 20(4):423–432, September 74.
- [Sto02] G. Storvik. Particle filters for state-space models with the presence of unknown static parameters. *Signal Processing, IEEE Transactions on*, 50(2):281–289, Feb 2002.
- [TBS98] S.M. Tonissen and Y. Bar-Shalom. Maximum likelihood track-before-detect with fluctuating target amplitude. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(3):796–809, Jul 1998.
- [Tur60] G.L. Turin. An introduction to matched filters. *Information Theory, IRE Transactions on*, 6(3):311–329, June 1960.
- [TV05] A. Tartakovsky and V.V. Veeravalli. General asymptotic Bayesian theory of quickest change detection. *Theory of Probability & Its Applications*, 49(3):458–497, 2005.
- [VGP05] J. Vermaak, S. J. Godsill, and P. Pérez. Monte Carlo Filtering for Multi-Target Tracking and Data Association. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(1):309–332, 2005.

- [VT02] H.L. Van Trees. *Optimum Array Processing, Part IV of Detection, Estimation, and Modulation Theory*. John Wiley and Sons, 2002.
- [VVPS10] B.N Vo, B.T. Vo, N.T Pham, and D. Suter. Joint detection and estimation of multiple objects from image observations. *Signal Processing, IEEE Transactions on*, 58(10):5129–5141, 2010.
- [Woo53] P.M. Woodward. *Probability and Information Theory: With Applications to Radar*. Number vol. 3 in Electronics and Waves. Elsevier Science & Technology, 1953.
- [WVdM00] E.A. Wan and R. 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*, pages 153–158, 2000.

Résumé : Cette thèse s'intéresse à l'étude et au développement de méthodes de pistage mono et multicible en contexte Track-Before-Detect (TBD) par filtrage particulaire. Contrairement à l'approche classique qui effectue un seuillage préalable sur les données avant le pistage, l'approche TBD considère directement les données brutes afin de réaliser conjointement la détection et le pistage des différentes cibles. Il existe plusieurs solutions à ce problème, néanmoins cette thèse se restreint au cadre bayésien des Modèles de Markov Cachés pour lesquels le problème TBD peut être résolu à l'aide d'approximations particulières. Dans un premier temps, nous nous intéressons à des méthodes particulières monocibles existantes pour lesquels nous proposons différentes lois instrumentales permettant l'amélioration des performances en détection et estimation. Puis nous proposons une approche alternative du problème monocible fondée sur les temps d'apparition et de disparition de la cible; cette approche permet notamment un gain significatif au niveau du temps de calcul. Dans un second temps, nous nous intéressons au calcul de la vraisemblance en TBD – nécessaire au bon fonctionnement des filtres particuliers – rendu difficile par la présence des paramètres d'amplitudes des cibles qui sont inconnus et fluctuants au cours du temps. En particulier, nous étendons les travaux de Rutten *et al.* pour le calcul de la vraisemblance au modèle de fluctuations *Swerling* et au cas multicible. Enfin, nous traitons le problème multicible en contexte TBD. Nous montrons qu'en tenant compte de la structure particulière de la vraisemblance quand les cibles sont éloignées, il est possible de développer une solution multicible permettant d'utiliser, dans cette situation, un seul filtre par cible. Nous développons également un filtre TBD multicible complet permettant l'apparition et la disparition des cibles ainsi que les croisements.

Mots-clés : Pistage, Track-Before-Detect, filtre particulaire, calcul de vraisemblance.

Abstract: This thesis deals with the study and the development of mono and multitarget tracking methods in a Track-Before-Detect (TBD) context with particle filters. Contrary to the classic approach that performs before the tracking stage a pre-detection and extraction step, the TBD approach directly works on raw data in order to jointly perform detection and tracking. Several solutions to this problem exist, however this thesis is restricted to the particular Hidden Markov Models considered in the Bayesian framework for which the TBD problem can be solved using particle filter approximations.

Initially, we consider existing monotarget particle solutions and we propose several instrumental densities that allow to improve the performance both in detection and in estimation. Then, we propose an alternative approach of the monotarget TBD problem based on the target appearance and disappearance times. This new approach, in particular, allows to gain in terms of computational resources. Secondly, we investigate the calculation of the measurement likelihood in a TBD context – necessary for the derivation of the particle filters – that is difficult due to the presence of the target amplitude parameters that are unknown and fluctuate over time. In particular, we extend the work of Rutten *et al.* for the likelihood calculation to several *Swerling* models and to the multitarget case. Lastly, we consider the multitarget TBD problem. By taking advantage of the specific structure of the likelihood when targets are far apart from each other, we show that it is possible to develop a particle solution that considers only a particle filter per target. Moreover, we develop a whole multitarget TBD solution able to manage the target appearances and disappearances and also the crossing between targets.

Keywords: Tracking, Track-Before-Detect, particle filter, likelihood calculation.