

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de 08 Mai 1945 Guelma
Faculté des Sciences et de la Technologie
Département de Génie Electrotechnique et Automatique

THESE



En vue de l'obtention du Doctorat en Génie Electrique

Titre de la thèse :

Poursuite multi-cibles par filtrage particulière

Présenté par : **Abdennour SEBBAGH**

Soutenue publiquement le 03 Juillet 2016

Composition du Jury :

Président :	Pr Boukrouche Abdelhani	Université de Guelma
Rapporteur :	Pr Tebbikh Hicham	Université de Guelma
Examineurs :	Pr Chikouche Djamel	Université de M'Sila
	Pr Bouden Toufik	Université de Jijel
	Pr Lachouri Abderrazek	Université de Skikda

Année 2016

Remerciements

Le travail de cette thèse a été réalisé au sein du laboratoire d'Automatique et Informatique de Guelma (LAIG), dans le cadre du projet de recherche intitulé : « Utilisation des informations visuelles pour la poursuite de cibles manœuvrantes à l'aide de l'approche IMM ».

Je tiens à exprimer toute ma gratitude au Rapporteur de ma thèse Monsieur Hicham Tebbikh, Professeur à l'université de Guelma, qui a assuré la direction de ce travail de thèse pour sa gentillesse et aussi pour ses aides durant les années de travail.

Je tiens également à remercier Monsieur Boukrouche Abdelhani, Professeur à l'université de Guelma pour m'avoir fait l'honneur de présider le jury de ma thèse.

J'adresse mes vifs remerciements à Monsieur Chikouche Djamel, Professeur à l'université de M'Sila, Monsieur Bouden Toufik, Professeur à l'université de Jijel et Monsieur Lachouri Abderrazek, Professeur à l'université de Skikda, pour m'avoir fait l'honneur d'être examinateurs de ce travail de thèse.

Je remercie tous les membres du laboratoire LAIG qui trouvent ici le témoignage de mes reconnaissances.

Refermer ce chapitre des remerciements ne peut se faire sans adresser des remerciements plus décontractés à tous mes collègues de départements : « Electrotechnique & Automatique » et « Electronique & Télécommunication »

Résumé : Cette étude porte sur l'apport des méthodes séquentielles de Monte Carlo ou encore filtres particulières pour la poursuite non linéaire multi-cibles. Dans le cas d'une cible unique, le problème de poursuite se résout classiquement par un filtre de Kalman lorsque les modèles sont linéaires et gaussiens. Des versions étendues de ce filtre peuvent également traiter des modèles linéarisables mais ces algorithmes sont mis en défaut dès que les linéarités sont trop fortes ou que la loi a posteriori à estimer est multimodale. Les méthodes séquentielles de Monte Carlo, intensivement utilisées ces dernières années, proposent une approximation de cette loi par une somme de lois de Dirac centrées en des points dénommés particules et pondérées par la vraisemblance des mesures conditionnellement à ces points. Contrairement aux algorithmes évoqués jusque-là de nature déterministe, les méthodes séquentielles de Monte Carlo sont des algorithmes stochastiques, reposant sur le principe de Monte Carlo. Lors de la présence des cibles multiples, le problème d'association des mesures aux objets cibles doit être résolu pour résoudre celui de l'estimation des objets. Nous proposons dans cette thèse un algorithme de poursuite non linéaire, dénoté NMMPF en anglais, fondé sur la combinaison de l'approche modèles multiples avec une structure particulière (méthodes séquentielles de Monte Carlo). Nous supposons que les associations des mesures aux cibles sont réalisées de manière indépendante d'une mesure à l'autre. Une seconde contribution apportée par nos travaux est la mise en application de cet algorithme en traitement du signal pour suivre la position 2D de cibles supposées observées par des mesures sonar ou radar de gisement. On parle alors de la poursuite mono ou multi-cibles.

Mots-clés : méthodes séquentielles de Monte Carlo, filtres particulières, approche modèles multiples, poursuite multi-cibles, poursuite par mesures d'angles.

Abstract: This thesis is concerned to non linear multiple targets tracking by particle filtering. For only one target, the Kalman filter analytically solves this problem when the model is linear and gaussian. Extended Kalman filters have also been proposed with linearization of the models. However, these filters diverge when the models are highly non linear and when the posteriori law is multimodal. The sequential Monte Carlo methods give an approximation of the posterior law based on a finite sum of Dirac laws centered on points named particles and weighted proportionally to the likelihood of the measures. They are based on the Monte Carlo principle. When dealing with multiple targets, the association problem between the measurements and the targets must be solved to estimate the states. We propose in this work a non linear multiple model particle filter algorithm NMMPF for multiple targets tracking, this algorithm is based on the combination between sequential Monte Carlo methods and multiple model approach and assuming the independence of the association variables. Our second contribution consists of applying this algorithm to bearings-only tracking.

Key-words: sequential Monte Carlo methods, particle filter, multiple model approach, multiple targets tracking, bearings-only tracking

ملخص: نركز في هذا العمل بالخصوص على المتابعة غير الخطية لعدة أهداف (أجسام متحركة) بواسطة الطرق المتتابعة لمنت كارلو (المرشح الجزيئي Filtrage Particulaire). إذا كان الهدف وحيدا فإن مرشح كالمان (Kalman) يمكنه القيام بعملية المتابعة وهذا عندما يكون النموذج خطي و طبيعي . بينما يحل محله مرشح كالمان الممدد عندما يكون النموذج غير خطي وهذا بعد تحويل هذا الأخير إلى نموذج خطي، ولكن مثل هذه المرشحات غالبا ما تخطئ المتابعة خاصة عندما يكون النموذج على درجة كبيرة من عدم الخطية والقانون المستقبلي متعدد النماذج. الطرق المتتابعة لمنت كارلو تعطي قانون مستقبلي متقارب مؤسس على جمع محدود من قوانين ديراك مركزة في نقاط تسمى جزئيات ذات عوامل متناسبة مع دالة حقيقة التشابه للقياسات، هذه الطرق مؤسسة على مبدأ منت كارلو.

عند تواجد عدة أهداف فإن مشكلة التجميع بين الأهداف والقياسات الناتجة عن أجهزة الرصد يجب أن تحل لإيجاد عناصر الحالة. في هذا العمل نقوم بعرض خوارزمية NMMPF لمتابعة عدة أهداف، هذه الخوارزمية مؤسسة على تآلف الطرق المتتابعة لمنت كارلو مع طريقة الأنظمة المتعددة مع فرضية عدم ارتباط متغيرات التجميع.

نقوم بتطبيق هذه الخوارزمية في عملية متابعة الأهداف عن طريق الزوايا فقط الناجمة من أجهزة الرصد.

مفاتيح: الطرق المتتابعة لمنت كارلو، المرشح الجزيئي، طريقة الأنظمة المتعددة، متابعة عدة أهداف

Nomenclature

Tout au long de cette thèse, les variables aléatoires sont représentées par des lettres majuscules et leurs réalisations par des lettres minuscules.

Les acronymes suivants sont précédés de leur équivalent français lorsqu'ils sont issus de la langue anglaise :

KF	Filtre de Kalman, Kalman filter.
EKF	Filtre de Kalman étendu, extended Kalman filter.
UT	Transformé unscented, unscented transform
UKF	Filtre de Kalman « Unscented », unscented Kalman filter.
MM:	Approche modèles multiples, multiple model approach.
PF	Filtre particulaire, particle filter.
MSMC	Méthodes séquentielles de Monte Carlo.
SIS	Echantillonnage pondéré séquentiel, sequential importance sampling.
SIR	Echantillonnage pondéré séquentiel avec rééchantillonnage, sequential importance resampling.
FBO	Filtre bayésien optimal.
PDAF	Filtre à association de données probabilistes, probabilistic data association filter.
JPDAF	Filtre à association de données probabilistes jointes, joint probabilistic data association filter.
MHT	Filtre à hypothèses multiples, multiple hypothesis tracker.
PMHT	Filtre à hypothèses multiples probabilistes, probabilistic multiple hypothesis tracker.
EM	Algorithme d'espérance-maximisation, expectation-maximisation algorithm.
NMMPF	Algorithme modèles multiples non linéaire à base des filtres particuliers, nonlinear multiple model particle filter algorithm.
RMSE	Racine de l'erreur quadratique moyenne, root mean square error.

Notations Utilisées

\mathbb{N}	Ensemble des entiers naturels.
\mathbb{R}	Ensemble des nombres réels.
\triangleq	Egal par définition.
$(\mathcal{D}, \mathcal{F}, \mathbb{P})$	Espace probabilisé sur lequel sont définis les variables aléatoires utilisées.
p	Loi de densité d'une variable aléatoire.
\mathbb{E}_p	Espérance calculée par rapport à la loi de densité p .
\mathbb{P}	Probabilité d'une variable aléatoire.
k	Lettre utilisé pour désigner le pas du temps.
i	Lettre utilisé pour désigner un des M objets.
j	Lettre utilisé pour désigner une des mesures m_k à l'instant k .
n	Lettre utilisé pour désigner une des N particules.
X_k	Processus d'état Markovien à estimer.
n_x	Dimension de l'espace d'état.
T_k	Noyau de transition de X_k .
F_k	Fonction d'évolution du vecteur d'état.
V_k	Bruit d'état.
Σ_V	Matrice de covariance du bruit d'état.
Y_k	Vecteur de mesures.
$Y_{1:k}$	Concaténation de (Y_1, \dots, Y_k) .
m_k	Taille du vecteur de mesures.
H_k	Fonction de mesure reliant une mesure à l'état associé.
W_k	bruit de mesure.
Σ_W	Matrice de covariance du bruit de mesure.
λ	Paramètre de la loi de poisson modélisant le nombre des fausses alarmes.
V	Volume d'observation.
θ_k	Vecteur désignant l'association des mesures aux objets.
Π_k	Vecteur aléatoire des probabilités des associations des mesures aux objets.
$L_{k k}$	Loi conditionnelle de X_k sachant $Y_{1:k}$.
S_k^N	Nuage de N particules à l'instant k .
$L_{k k}^N$	Loi discrète associée au nuage de particules.
s_k^n	Particule n du nuage S_k^N .
ω_k^n	Poids normalisé de la particule s_k^n .

Table de figures

1.1	Graphe de dépendance de la chaîne de Markov cachée constituée par les processus X_k et Y_k	6
1.2	Principe du filtre de Kalman	11
1.3	Principe de unscented transformation	16
2.1	Mesures d'angles simulées du modèle rectiligne	43
2.2	Trajectoire réelle et estimée	44
2.3	RMSE de position (X et Y)	44
2.4	RMSE de vitesse dans le plan horizontal (X, Y)	44
2.5	Mesures d'angles simulées du modèle curviligne	45
2.6	Trajectoire réelle et estimée	46
2.7	RMSE de position (X et Y)	46
2.8	RMSE de vitesse dans le plan horizontal (X, Y)	47
2.9	RMSE de vitesse angulaire Ω	47
3.1	Schéma fonctionnel de l'algorithme d'estimation modèles multiples (cas de non changement de modèles)	53
3.2	Schéma de l'estimateur bayésien de plein ordre (cas de changement de modèles)	56
3.3	Schéma descriptif de l'algorithme GPB-1	58
3.4	Schéma descriptif de l'algorithme GPB-2	61
3.5	Schéma descriptif de l'algorithme IMM	67
5.1	Trajectoires des 3 cibles	99
5.2	Mesures d'angles simulées des 3 cibles en radians	99
5.3	Trajectoires réelles et estimées des 3 cibles	100
5.4	Probabilités des modèles (pour qu'une mesure $j = 1, 2, 3$ soit associée au modèle $i = 1, 2, 3$)	101
5.5	Racine de l'erreur quadratique moyenne (RMSE) de position (X, Y) pour les 3 cibles	102
5.6	Racine de l'erreur quadratique moyenne (RMSE) de vitesse (v_x, v_y) pour les 3 cibles	102
5.7	Racine de l'erreur quadratique moyenne (RMSE) de la vitesse angulaire Ω .	103
5.8	Trajectoires réelles et estimées des 3 cibles pour $N = 1000$ particules	104
5.9	Trajectoires réelles et estimées des 3 cibles pour $N = 2000$ particules	104
5.10	Trajectoires réelles et estimées des 3 cibles avec initialisation biaisée pour $N = 500$ particules	105

Liste des tableaux

- 5.1 Temps de calcul nécessaire pour une itération du NMMPF sur Pentium IV à 3.40 MHz 103

Liste des algorithmes

2.1	Filtre de Monte-Carlo	28
2.2	Filtre particulaire de type bootstrap	29
2.3	Filtre particulaire à échantillonnage pondéré séquentiel (SIS)	32
2.4	Filtre particulaire à échantillonnage pondéré séquentiel avec rééchantillonnage (SIR)	33
2.5	Filtre particulaire avec rééchantillonnage adaptatif	40
4.1	Filtre bayésien optimal pour filtrage linéaire gaussien dans un environnement bruité	77
4.2	Filtre à association de données probabilistes pour filtrage linéaire gaussien dans un environnement bruité	80
4.3	Algorithme PMHT pour filtrage linéaire gaussien dans un environnement bruité.	91
5.1	Algorithme non linéaire modèle multiple à base de filtrage particulaire (NMMPF) avec rééchantillonnage adaptatif	97

Table des matières

Nomenclature	IV
Table de figures	VI
Liste des tableaux	VII
Liste des algorithmes	VIII
Table de matières	IX
Introduction générale	1
1 Filtres de poursuite : Etat de l'art	4
1.1 Filtres de poursuite	4
1.1.1 Introduction	4
1.1.2 Techniques ad hoc	4
1.1.3 Modélisation du problème et filtre bayésien optimal	6
1.1.4 Filtre de Kalman	7
1.1.4.1 Espace d'état	9
1.1.4.2 Formulation récursive	10
1.1.5 Estimation dans des systèmes non linéaires	11
1.1.5.1 Formulation du problème	11
1.1.5.2 Transformation d'incertitude	13
1.1.6 Filtre de Kalman étendu	14
1.1.7 Filtre de Kalman sans odeur (unscented Kalman filter)	16
1.1.7.1 Unscented transformation	16
1.1.7.2 Filtre sans odeur (unscented filter)	18
1.1.7.3 Scaled unscented Kalman filter (SUKF)	19
1.2 Conclusion	22
2 Filtrage particulière et la poursuite mono cible	24
2.1 Méthodes séquentielles de Monte-Carlo : filtres particuliers	24
2.1.1 Filtre de Monte-Carlo et filtre bootstrap	26
2.1.2 Echantillonnage pondéré séquentiel avec rééchantillonnage	30
2.1.3 Techniques de rééchantillonnage	37
2.2 Application à la poursuite mono-cible pour mesure d'angle seul	40
2.2.1 Modélisation mathématique	41
2.2.2 Divers scénarios	42
2.3 Conclusion	47
3 Approche modèles multiples	49
3.1 Estimation à modèles multiples	49
3.1.1 Approche Modèles Multiples (cas de non changement de modèle)	50
3.1.1.1 Calcul des probabilités des modèles	50
3.1.1.2 Calcul des estimations combinées	51
3.1.2 Approche Modèles Multiples (cas de changement de modèles)	53
3.1.2.1 Estimateur pseudo-bayésien généralisé de premier ordre	57

	(GPB-1)	
	3.1.2.2	Estimateur pseudo-bayésien généralisé de deuxième ordre (GPP-2) 60
	3.1.2.3	Estimateur modèles multiples interagissants (IMM) 64
3.2	Conclusion	68
4	Poursuite linéaire multi-cibles	69
4.1	Formulation de la poursuite multi-cibles	69
4.2	Filtrage et association de données avec dépendance des associations	72
4.2.1	Hypothèses d'association	72
4.2.2	Principe de validation des mesures	73
4.2.3	Filtrage d'une cible unique dans un environnement bruité	73
4.2.3.1	Filtre bayésien optimal (FBO)	74
4.2.3.2	Filtre à association de données probabilistes (PDAF)	76
4.2.4	Filtrage multi-cibles dans un environnement bruité	80
4.2.4.1	Filtre joint à association de données probabilistes (JPDAF)	80
4.2.4.2	Filtre à hypothèses multiples (MHT)	82
4.3	Filtrage et association de données avec indépendance des associations	84
4.3.1	Filtre probabiliste à hypothèses multiples (PMHT)	85
4.3.1.1	Algorithme d'espérance-maximisation (EM)	87
4.4	Conclusion	92
5	Poursuite non linéaire multi-cibles : Application aux problèmes par mesures d'angles	94
5.1	Le NMMPF : Algorithme non linéaire modèles multiples à base de filtrage particulière pour la poursuite multi-cibles	94
5.1.1	Présentation de l'algorithme NMMPF	95
5.2	Application du NMMPF à la poursuite multi-cibles par mesure d'angles	96
5.2.1	Scénarios et paramètres	96
5.2.2	Influence de l'augmentation du nombre de particules	103
5.2.3	Influence de l'initialisation	105
5.2.4	Remarque	106
5.3	Conclusion	106
	Conclusion générale	107
	Bibliographies	109
	Annexe	116
	Publications	

Introduction générale

Nous nous intéressons dans ce document de thèse, au problème de filtrage. Il consiste, de façon générale, à estimer au cours du temps un processus d'état à partir de mesures bruitées. Bien que ce système évolue en réalité en temps continu, nous considérons un modèle à temps discret. Nous entendons ici une définition plus précise du filtrage qui se situe dans le triplet de traitements : filtrage, lissage et prédiction. Le filtrage consiste alors à estimer le processus d'état en s'appuyant uniquement sur les mesures passées et présentes. On l'oppose au lissage qui utilise en plus les mesures futures. La prédiction enfin cherche à estimer le processus aux instants futurs à l'aide des mesures passées et présentes.

On peut illustrer ces trois traitements par des opérations effectuées inconsciemment par notre cerveau:

- le nouveau-né procède par filtrage lorsqu'il cherche à suivre l'objet que l'on déplace devant lui alors qu'il ne le voit pas encore nettement;
- le lissage est l'opération intellectuelle que nous faisons en lisant un texte dont un mot nous est inconnu et dont nous devinons le sens à l'aide des mots le précédant et le suivant;
- enfin, tout joueur de tennis doit prédire la trajectoire de la balle s'il souhaite la renvoyer à son adversaire.

Le problème de filtrage, ainsi que ceux de prédiction et de lissage, sont motivés par un grand nombre d'applications réelles parmi lesquelles la poursuite de cibles, le suivi d'entités dans des séquences d'images, la déconvolution de signaux numériques, l'estimation de risque en mathématiques financières et la localisation d'objets en robotique. La modélisation de tels problèmes se fait en général en se donnant un a priori sur la loi d'évolution du modèle à estimer. Dans ce cadre bayésien, la reconstitution optimale du processus d'état consiste à l'estimation de sa loi a posteriori, i.e., la loi de ce processus conditionnellement aux mesures acquises depuis l'instant initial jusqu'à l'instant présent.

Limités par la mémoire finie des ordinateurs, nous sommes contraints de choisir une représentation finie de cette loi. Dans le cas de modèles d'état et de mesures linéaires et de

bruits additifs gaussiens, la loi a posteriori présente l'énorme avantage d'être une gaussienne. Elle peut donc être entièrement caractérisée par sa moyenne et sa covariance. L'étude des modèles linéaires gaussiens a pour cette raison suscité un vif intérêt, donnant naissance au filtre de Kalman [11], [7], qui fournit une expression récursive analytique des deux premiers moments de la loi a posteriori. Des extensions ont également été apportées au filtre de Kalman pour permettre son application à des modèles linéarisables. Les filtres de Kalman étendus obtenus [35], bien qu'encore largement utilisés, sont cependant rapidement mis en défaut lors de non linéarités trop fortes. Outre les non linéarités, le problème de filtrage est souvent perturbé par la présence de mesures non issues des objets, dénommées fausses alarmes et qui conduisent à la multi-modalité de la loi a posteriori. Il est alors vain de vouloir représenter cette loi multimodale par sa moyenne et sa covariance. Les non linéarités des modèles rencontrés, la complexité de leur dynamique ainsi que la gestion de fausses alarmes... sont cependant des obstacles qui nous empêchent de résoudre, dans le cas général le filtre bayésien optimal donnant une expression récursive de la loi a posteriori. Les méthodes séquentielles de Monte Carlo [68], intensivement utilisées pour le filtrage, la prédiction et le lissage mono-cible ces dernières années, proposent une approximation de cette loi par une somme de lois de Dirac centrées en des points dénommés particules et pondérées par la vraisemblance des mesures conditionnellement à ces points. Contrairement, aux algorithmes évoqués jusque-là de nature déterministe, les méthodes séquentielles de Monte Carlo sont des algorithmes stochastiques, reposant sur le principe de Monte Carlo qui consiste à simuler un grand nombre de réalisations d'une variable aléatoire pour en approcher sa loi. Si ce concept est apparu dès le milieu du XX siècle, il a fallu attendre l'augmentation de la puissance de calcul des ordinateurs pour pouvoir les mettre en œuvre dans le cadre du problème de l'estimation séquentielle.

Dans ce contexte d'essor des méthodes particulières dans de nombreux domaines applicatifs, nous nous sommes penchés dans ce travail de thèse sur le difficile problème du filtrage multi-cibles. Les algorithmes déterministes existants se distinguent selon qu'ils supposent la dépendance ou l'indépendance du processus d'association d'une mesure donnée par rapport à l'association des autres mesures, ces algorithmes sont basés sur le filtre de Kalman, à savoir le filtre bayésien optimal (FBO), le filtre à association de données probabilistes (PDAF), le filtre joint à association de données probabilistes

(JPDAF) et le filtre à hypothèses multiples (MHT) [13], [83], estiment de façons plus ou moins optimale la loi de l'ensemble des cibles conditionnellement aux mesures, sous l'hypothèse de dépendance des associations des données, par contre, le filtre probabiliste à hypothèse multiples (PMHT), [83], consiste à relâcher cette hypothèse d'association à fin d'éviter l'énumération exhaustive de ces associations. Nous nous sommes naturellement tournés vers les méthodes séquentielles de Monte Carlo et nous avons étudié leur apport possible dans le cadre du filtrage non linéaire multi-cibles. Notre contribution consiste plus particulièrement à l'élaboration d'un algorithme non linéaire multi-cibles, résultant de la combinaison de l'approche modèles multiples avec des filtres particulières, dénommé nonlinear multiple model particle filter et dénoté NMMPF en anglais. Nous supposons que les associations des mesures aux cibles sont réalisées de manière indépendante d'une mesure à l'autre.

Une seconde contribution apportée par nos travaux est la mise en application de cet algorithme en traitement du signal pour suivre la position 2D de cibles supposées observées par des mesures sonar ou radar de gisement. On parle alors de la poursuite mono ou multi-cibles.

La thèse est organisée en cinq chapitres encadrés par une introduction et une conclusion générales :

Le premier chapitre décrit un état de l'art des filtres de poursuite pour une cible unique. Il débute par le filtre de Kalman et se poursuit par son extension à des modèles linéarisables à savoir le filtre de Kalman étendu et se termine par le filtre de Kalman unscented.

Les méthodes séquentielles de Monte Carlo, ou filtres particulières, sont ensuite présentées en détail au deuxième chapitre, une structure générique commune à de nombreuses variantes de ces méthodes est notamment dégagée. Une mise en œuvre de filtres particulières pour la poursuite mono-cible clôt ce chapitre.

Le troisième chapitre étudie l'approche modèles multiples sans et avec changement de modèle.

Le problème de poursuite à la présence des cibles multiples et de fausses alarmes est étudié dans le quatrième chapitre. Nous exposons les algorithmes déterministes proposés jusqu'à supposant, soit la dépendance, soit l'indépendance des associations d'une mesure à l'autre.

Le cinquième chapitre est réservé au problème de la poursuite non linéaire multi-cibles, pour laquelle nous proposons un algorithme de poursuite que nous avons dénoté NMMPF (en anglais). Nous terminons ce chapitre par des applications en traitement du signal où le NMMPF est mis en œuvre pour traiter des scénarios simulés de poursuite par mesures d'angles uniquement.

Chapitre 1

Filtres de poursuite : Etat de l'art

Nous exposons dans ce premier chapitre les algorithmes traitant du problème de filtrage pour une cible unique. Nous verrons tout d'abord la solution théorique fournie par le filtre optimal bayésien. Dans le cas simple où le problème est linéaire et gaussien, le filtre de Kalman fournit une expression analytique de ce filtre. Dans le cas non linéaire, le filtre optimal bayésien n'est, en général, pas accessible et nous présenterons les algorithmes sous optimaux les plus courants, à savoir le filtre de Kalman étendu et le filtre de Kalman sans odeur, dénommé unscented Kalman filter et dénoté UKF en anglais.

1.1 Filtres de poursuite

1.1.1 Introduction

Le traitement de données incluant des variables aléatoires date du début du XIX^{ème} siècle à partir des apports théoriques de Gauss, qui a notamment mis en œuvre la méthode des moindres carrés déterministes. La théorie de l'estimation progresse un siècle plus tard avec Fisher [1] qui, à travers la notion de densité de probabilité, introduit l'approche du maximum de vraisemblance. Wiener [3] met au point, dans les années quarante, un filtre optimal en utilisant la théorie des processus aléatoires. Kolmogorov [2] dans la même période traite le problème numérique que Wiener avait résolu dans le cas continu. Les travaux de Wiener ont par la suite été étendus pour des systèmes non linéaires, et Kalman [7] a développé un filtre optimal récursif basé sur la représentation d'état et parfaitement adapté à une implémentation sur calculateurs. Ce filtre, qu'un observateur au sens de l'automatique, est parfaitement intégré dans la théorie de l'estimation.

1.1.2 Techniques ad hoc

Le concept qui a débuté la révolution dans les performances du radar de surveillance était d'incorporer les modèles dynamiques dans les systèmes de poursuite. Par l'utilisation des équations de la dynamique newtonienne, tels modèles permettent de prédire la future position de la cible, libérant le radar d'exécuter d'autres tâches entre mises à jour. Les

modèles dynamiques peuvent être basés sur des hypothèses simples de vitesse constante, tel que:

$$\begin{aligned}x(k) &= x(k-1) + v(k-1) \cdot [t_k - t_{k-1}] \\v(k) &= v(k-1)\end{aligned}\quad (1.1)$$

Où x est la position de la cible, v sa vitesse et $[t_k - t_{k-1}]$ la différence de temps entre les instants k et $(k-1)$. Si la vitesse de la cible n'est pas constante, alors des modèles d'accélération constante peuvent être utilisés tel que:

$$\begin{aligned}x(k) &= x(k-1) + v(k-1) \cdot [t_k - t_{k-1}] + \frac{1}{2} a(k-1) \cdot [t_k - t_{k-1}]^2. \\v(k) &= v(k-1) + a(k-1) \cdot [t_k - t_{k-1}] \\a(k) &= a(k-1)\end{aligned}\quad (1.2)$$

Où $a(k)$ est l'accélération de la cible à l'instant k , les méthodes de poursuite basées sur ces modèles estiment la position, la vitesse et, en cas de besoin, l'accélération de la cible connaissant la mesure courante et la mesure prédite en utilisant les équations (1.1) et (1.2). Le filtre de poursuite fondé sur l'hypothèse de vitesse constante est désigné sous le nom du traqueur $\alpha - \beta$ et fonctionne comme suit [73]:

$$\begin{aligned}\hat{x}(k|k) &= \hat{x}(k|k-1) + \alpha [z(k) - \hat{x}(k|k-1)] \\ \hat{v}(k|k) &= \hat{v}(k|k-1) + \frac{\beta}{t_k - t_{k-1}} [z(k) - \hat{x}(k|k-1)]\end{aligned}\quad (1.3)$$

Où $z(k)$ est la mesure à l'instant k de la position de la cible. Le nom traqueur $\alpha - \beta$ est dû aux coefficients α et β , qui sont utilisés comme facteurs pondérés pour faire les mises à jour. Si α et β sont nulles, alors le système se fonde purement sur les prédictions fournies par le modèle dynamique inclus dans le système. Réciproquement, si α et β sont égaux à un, alors le système ignore le modèle dynamique du système et prend en compte uniquement la dernière mesure. Ainsi, en ajustant α et β , le concepteur a une compensation entre le poids que le système place sur les mesures passées, propagées par le modèle dynamique, et le poids que le système place sur la mesure nouvellement présentée.

Le traqueur $\alpha - \beta - \gamma$ opère de façon similaire, en ajoutant un facteur de poids additionnel pour faciliter l'estimation de l'accélération (qui est supposée constante) [21] :

$$\begin{aligned}\hat{x}(k|k) &= \hat{x}(k|k-1) + \alpha [z(k) - \hat{x}(k|k-1)] \\ \hat{v}(k|k) &= \hat{v}(k|k-1) + \frac{\beta}{t_k - t_{k-1}} [z(k) - \hat{x}(k|k-1)]\end{aligned}$$

$$\hat{a}(k|k) = \hat{a}(k|k-1) + \frac{\gamma}{(t_k - t_{k-1})^2} [z(k) - \hat{x}(k|k-1)] \quad (1.4)$$

Ces équations fonctionnent selon le traqueur $\alpha - \beta$ (1.3), avec la prédiction entre les intervalles de mesures exécutées en utilisant le modèle d'accélération constante (1.2), où γ représente le coefficient de pondération employé pour mettre à jour l'estimation d'accélération du modèle.

1.1.3 Modélisation du problème et filtre bayésien optimal

Soit $(D, \mathcal{F}, \mathbb{P})$ un espace probabilisé sur lequel on définit le système dynamique composé des deux processus $(X_k)_{k \in \mathbb{N}}$ et $(Y_k)_{k \in \mathbb{N}^*}$. Le processus $(X_k)_{k \in \mathbb{N}}$ à valeurs dans \mathbb{R}^{n_x} , appelé processus d'état, est inconnu et supposé markovien de distribution initiale $X_0 \sim p_0(dx_0)$ et de noyau de transition T_k :

$$p(X_k | X_{0:k-1} = x_{0:k-1}) = p(X_k | X_{k-1} = x_{k-1}) = T(x_{k-1}) \quad \forall x_{0:k-1} \in (\mathbb{R}^{n_x})^k \quad (1.5)$$

Il est observé via un second processus $(Y_k)_{k \in \mathbb{N}^*}$ à valeurs dans \mathbb{R}^{n_y} , dit processus d'observation ou de mesures. Les variables Y_k sont supposées indépendantes conditionnellement au processus d'état et leur distribution dépend uniquement de l'état courant:

$$p(Y_k | X_k = x_k) = h(y_k | x_k) \quad \forall x_k \in \mathbb{R}^{n_x} \quad (1.6)$$

Les propriétés d'indépendance conditionnelle des mesures et la structure markovienne du processus d'état conduisent à une structure de chaîne de Markov cachée, illustrée en figure 1.1. On désignera par $X_{0:k}$ et $Y_{1:k}$ la suite des variables (X_0, \dots, X_k) respectivement (Y_1, \dots, Y_k) et par $y_{1:k}$ une réalisation de cette dernière suite.

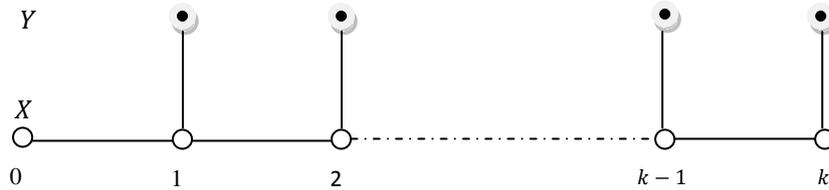


Fig. 1.1 : Graphe de dépendance de la chaîne de Markov cachée constituée par les processus X_k et Y_k

a Cas des mesures à densité

Si l'on suppose que le noyau de transition et la distribution de mesure admettent des densités alors le système dynamique (1.5), (1.6) devient:

$$\begin{cases} X_k | X_{k-1} \sim p(X_k | X_{k-1}); \\ Y_k | X_k \sim h_k(y_k | x_k). \end{cases} \quad (1.7)$$

b Cas de bruits additifs

En pratique, les systèmes étudiés sont souvent modélisés comme des systèmes bruités additivement avec éventuellement la présence d'une suite de commandes connues U_k :

$$\begin{cases} X_k = F_k(X_{k-1}) + V_k + U_k; \\ Y_k = H_k(X_k) + W_k. \end{cases} \quad (1.8)$$

où les fonctions F_k et H_k sont non linéaires. Afin d'assurer le caractère markovien du processus d'état, le bruit $V_k \in \mathbb{R}^{n_v}$ est supposé être un bruit blanc et l'indépendance conditionnelle des observations est assurée par le fait que $W_k \in \mathbb{R}^{n_w}$ soit également un bruit blanc, indépendant de $(V_k)_{k \in \mathbb{N}}$. La distribution initiale est également supposée indépendante des bruits d'état et de mesure. Si les processus de bruit d'état et de mesure admettent des densités respectives notées p_v et p_w , le système précédent est donc caractérisé par les densités du noyau et de la loi de mesure :

$$\begin{cases} T_k(x_{k-1}, x_k) = p_v(x_k - F_k(x_{k-1})); \\ h_k(y_k | x_k) = p_w(y_k - H_k(x_k)). \end{cases} \quad (1.9)$$

En adoptant classiquement une approche bayésienne, l'étude du champ de Markov caché $(X_k)_{k \in \mathbb{N}}$, $(Y_k)_{k \in \mathbb{N}^*}$ suggère plusieurs problèmes d'estimation:

- le problème de filtrage classique, i.e., l'estimation de la distribution conditionnelle $L_{k|k}(x_k) = p(X_k | Y_{1:k} = y_{1:k})$ qui contient l'ensemble de l'information que l'on peut tirer de X_k à partir des réalisations collectées $y_{1:k}$ de $Y_{1:k}$. Ayant résolu le problème de filtrage précédent, on peut alors calculer toute quantité de la forme $\mathbb{E}_{L_{k|k}}(\phi(x_k))$ où ϕ est une fonction intégrable de \mathbb{R}^{n_x} .
- le filtrage des trajectoires, i.e., l'estimation de la distribution conditionnelle $L_{0:k|k}(x_{0:k}) \triangleq p(X_{0:k} | Y_{1:k} = y_{1:k})$.
- le problème de prédiction à c pas, i.e., l'estimation de la distribution conditionnelle $L_{k+c|k}(x_{k+c}) \triangleq p(X_{k+c} | Y_{1:k} = y_{1:k})$ avec $c \in \mathbb{N}^*$.
- le problème de lissage, i.e., l'estimation de la distribution conditionnelle $L_{k|T}(x_k) \triangleq p(X_k | Y_{1:k} = y_{1:k})$ avec $T > k$.

On se concentrera dans ce manuscrit sur le premier problème de filtrage, les problèmes de prédiction et de lissage étant généralement résolus par simple extension de ce dernier. Il est

intéressant, aussi, d'établir l'expression récursive suivante pour $L_{k|k}$ qui constitue le filtre bayésien optimal. Supposons que $L_{k-1|k-1}$ soit connue. Par prédiction de cette distribution, on obtient:

$$L_{k|k-1}(dx_k) = \int_{\mathbb{R}^{n_x}} T_k(x_{k-1}, dx_k) L_{k|k-1}(dx_{k-1}) \quad (1.10)$$

Cette prédiction est alors mise à jour par la donnée de l'observation Y_k en utilisant la règle de Bayes:

$$\begin{aligned} L_{k|k}(dx_k) &= \frac{p(Y_k = y_k | X_k = x_k, Y_{1:k-1} = y_{1:k-1}) p(X_k = x_k | Y_{1:k-1} = y_{1:k-1})}{p(Y_k = y_k | Y_{1:k-1} = y_{1:k-1})} \\ &= \frac{h(y_k | x_k) L_{k|k-1}(dx_k)}{\int_{\mathbb{R}^{n_x}} h(y_k | x_k) L_{k|k-1}(dx_k)}. \end{aligned} \quad (1.11)$$

On notera que le dénominateur de cette dernière expression est une constante de normalisation obtenue par intégration. Remarquons également que les hypothèses de Markov et d'indépendance conditionnelle des mesures permettent l'écriture de (1.10) et (1.11). Cependant, ces formules contiennent des intégrales qui sont rarement accessibles en pratique. Le cas particulier du filtrage linéaire avec bruits additifs gaussiens constitue une exception pour laquelle on dispose alors d'équations analytiques qui sont celles du filtre de Kalman, que nous rappelons dans la prochaine section. Benes présente également une classe particulière de filtres pour laquelle on dispose d'une solution analytique [17]. Dans le cas général, des approximations sont nécessaires et seront effectuées selon les algorithmes, soit pour approximer la loi $L_{k|k}$ par une gaussienne, soit pour approximer les intégrales impliquées.

1.1.4 Filtre de Kalman

Le problème du filtrage consiste à estimer l'état d'un système dynamique aux vues d'observations bruitées des états passés du système. Ce problème est suffisamment général pour trouver de nombreuses applications variées telles que la poursuite trajectoire, la commande optimale...etc. c'est ce contexte applicatif riche qui motive pour l'essentiel la recherche dans ce domaine. Dans ce travail, nous avons choisi de considérer le cas particulier du filtrage en temps discret. Précisément, nous cherchons à estimer à chaque instant d'observation, la valeur d'un processus aléatoire Markovien $(x_k)_{k \geq 1}$. Ce processus d'état est indirectement lié à un processus d'observation $(y_k)_{k \geq 1}$ dont on connaît une réalisation. Le but est alors de calculer (de manière approchée) à chaque instant

d'observation k , la distribution de probabilité conditionnelle p_k de l'état à l'instant courant connaissant les observations passées.

Le filtre de Kalman permet, en effet, le calcul exact et rapide du filtre optimal lorsque les modèles d'état et d'observation ne font intervenir que des fonctions linéaires et des bruits additifs Gaussiens.

1.1.4.1 Espace d'état

Considérant le filtre de Kalman, on suppose d'une part que les observations sont probabilistes et d'autre part que la connaissance de l'évolution entre l'observation k et l'observation $k + 1$ est celle d'un modèle dynamique régissant l'évolution d'un vecteur d'état entre ces deux instants. L'étude la plus complète sur ce type de filtres se trouve dans [11].

Un modèle linéaire gaussien est un modèle où le processus suit une évolution linéaire avec un bruit additif gaussien et où l'observation est une fonction linéaire de l'état, entaché d'un bruit additif gaussien. Ce modèle est mathématiquement décrit par le système suivant:

$$X(k + 1) = F(k)X(k) + G(k)U(k) + D(k)V(k) \quad (1.12)$$

$$Y(k) = H(k)X(k) + W(k) \quad (1.13)$$

Dans les expressions (1.12) et (1.13), $X(k) \in \mathbb{R}^n$ représente l'état du processus, $Y(k) \in \mathbb{R}^p$ est l'observation disponible à l'instant k , $U(k)$ est une entrée de forçage et $V(k) \in \mathbb{R}^s$ et $W(k) \in \mathbb{R}^p$ représentent respectivement un bruit de modélisation et un bruit de mesure. Ces bruits, qui traduisent l'incertitude que l'on a sur le modèle et sur le capteur, sont blancs, gaussiens et de covariances respectives $Q \in \mathbb{R}^{s \times s}$ et $R \in \mathbb{R}^{p \times p}$. Ils sont mutuellement indépendants et indépendants de la condition initiale $X(0)$. Cette condition initiale de l'état du processus est de moyenne $\mathbb{E}[X(0)] = \bar{X}(0)$ et les matrices $F(k) \in \mathbb{R}^{n \times n}$, $G(k) \in \mathbb{R}^{n \times r}$, $D(k) \in \mathbb{R}^{n \times s}$ et $H(k) \in \mathbb{R}^{p \times n}$ sont déterministes. La matrice $F(k)$ est la matrice de transition de l'état de l'instant k à l'instant $k + 1$, $G(k)$ est la matrice de commande, $H(k)$ est la matrice d'observation et $D(k)$ est la matrice qui permet de lier le bruit de modélisation $V(k)$ au vecteur d'état.

1.1.4.2 Formulation récursive

Lorsque les conditions initiales du filtre sont données par $\hat{X}(0) = \bar{X}(0)$ et $P(0)$, les grandeurs $\hat{X}(k)$ et $P(k)$ sont estimées par les étapes de prédiction et de correction suivant les équations :

- Equations de prédiction

$$\hat{X}(k+1|k) = F(k)\hat{X}(k) + G(k)U(k) \quad (1.14)$$

$$P(k+1|k) = F(k)P(k)F^T(k) + D(k)Q(k)D^T(k) \quad (1.15)$$

- Equations de correction (mise à jour)

$$\hat{X}(k+1) = \hat{X}(k+1|k) + K(k+1)Y(k+1|k) \quad (1.16)$$

$$K(k+1) = P(k+1|k)H^T(k+1)S^{-1}(k+1) \quad (1.17)$$

$$P(k+1) = (I - K(k+1)H(k+1))P(k+1|k) \quad (1.18)$$

Avec $S(k+1)$ est la covariance de l'innovation $Y(k+1|k)$.

$$S(k+1) = H(k+1)P(k+1|k)H^T(k+1) + R(k) \quad (1.19)$$

La première étape d'évolution du filtre optimal utilise la connaissance a priori du système. En effet, elle permet la prédiction de l'état du système $\hat{X}(k+1|k)$ et de sa précision $P(k+1|k)$ (matrice de covariance de l'erreur a priori) à partir de l'état $\hat{X}(k)$ et du modèle d'évolution du système. $P(k+1|k)$ est la matrice de covariance associée à l'état $\hat{X}(k+1|k)$.

La deuxième étape utilise la connaissance a posteriori fournie par l'observation. La mesure $Y(k+1)$ permet de calculer une erreur de prédiction de l'observation $\hat{Y}(k+1|k)$. Cette erreur est un terme d'innovation $Y(k+1|k)$:

$$Y(k+1|k) = Y(k+1) - H(k+1)\hat{X}(k+1|k) = Y(k+1) - \hat{Y}(k+1|k) \quad (1.20)$$

Elle sert à pondérer le gain $K(k+1)$ du filtre. Ce gain est généralement appelé gain de Kalman et sert à déduire l'estimation de l'état $\hat{X}(k+1)$ et sa matrice de covariance associée $P(k+1)$. En fait, la matrice $K(k+1)$ minimise la matrice de covariance de l'erreur a posteriori $P(k) = \mathbb{E} \left([X(k) - \hat{X}(k)][X(k) - \hat{X}(k)]^T \right)$.

Si la covariance de l'erreur de mesure R s'approche de zéro, alors :

$$\lim_{R(k) \rightarrow 0} K(k+1) = H^{-1}(k+1) \quad (1.21)$$

Cela veut dire que plus l'erreur de mesure diminue, plus le filtre fait confiance à la véritable mesure $y(k)$ au détriment de $H(k)\hat{X}(k|k-1)$. Inversement, lorsque la covariance de l'erreur a priori tend vers zéro, le gain $K(k+1)$ diminue également :

$$\lim_{P(k|k-1) \rightarrow 0} K(k+1) = 0 \quad (1.22)$$

On remarque que le filtre maintient les deux premiers moments de la distribution sur les états et que la variance de cette distribution est bien:

$$\mathbb{E}(X(k)) = \hat{X}(k) \quad (1.23)$$

$$\mathbb{E}([X(k) - \hat{X}(k)][X(k) - \hat{X}(k)]^T) = P(k) \quad (1.24)$$

$$p(X(k)|Y(k)) \approx N\{X(k); \hat{X}(k), P(k)\} \quad (1.25)$$

Le principe de fonctionnement du filtre est entièrement représenté sur la figure 1.2.

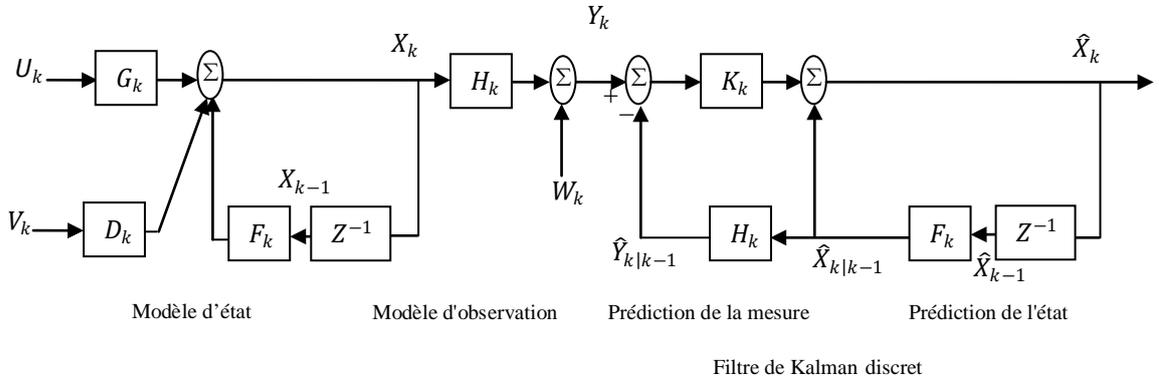


Fig. 1.2: Principe du filtre de Kalman.

1.1.5 Estimation dans des systèmes non linéaires

1.1.5.1 Formulation du problème

Nous souhaitons appliquer le filtre de Kalman sur un modèle non linéaire discret de la forme:

$$X(k+1) = f(X(k), U(k), V(k)) \quad (1.26)$$

$$Y(k) = h(X(k), U(k)) + W(k) \quad (1.27)$$

Où $X(k)$ est le vecteur d'état du système à l'instant k de dimension n , $U(k)$ est le vecteur d'entrée, $V(k)$ est le vecteur de bruit du processus de dimension s , $Y(k)$ est le vecteur d'observation et $W(k)$ est le bruit de mesure. Il est supposé que les vecteurs de bruits $V(k)$ et $W(k)$ ont des moyennes nulles et de covariances :

$$\mathbb{E}(V(i)V^T(j)) = \delta_{ij} Q(i)$$

$$\begin{aligned}\mathbb{E}(W(i)W^T(j)) &= \delta_{ij}R(i) \\ \mathbb{E}(V(i)W^T(j)) &= 0 \quad \forall i, j\end{aligned}\quad (1.28)$$

Où $\delta_{ij} = 1$ pour $i = j$.

Le filtre de Kalman propage les deux premiers moments de la distribution de $X(k)$ périodiquement et possède une structure "prédicteur-estimateur" distinctive. Supposons que $\hat{X}(i|j)$ est l'estimation de $X(j)$, en utilisant l'information d'observation Y^j comprise entre l'instant 1 et j , $Y^j = (y(1), \dots, y(j))$. La covariance de cette estimation est $P(i|j)$. Etant donné l'estimé $\hat{X}(k|k)$, le filtre prédit d'abord l'état futur du système. Idéalement, les quantités prédites sont données par les espérances suivantes:

$$\hat{X}(k+1|k) = \mathbb{E}\left(f(X(k), U(k), V(k)) | Y^k\right) \quad (1.29)$$

$$P(k+1|k) = \mathbb{E}\left([X(k+1) - \hat{X}(k+1|k)][X(k+1) - \hat{X}(k+1|k)]^T | Y^k\right) \quad (1.30)$$

Lorsque $f(\cdot)$ et $h(\cdot)$ sont non linéaires, les valeurs précises de ces statistiques peuvent seulement être calculées si la distribution de $X(k)$, condition sur Y^k est connue. Cependant, nous n'avons aucune forme générale et un nombre potentiellement illimité de paramètres sont exigés. Dans plusieurs applications, la distribution de $X(k)$ est approchée de sorte que seulement un nombre fini et tractable de paramètres doivent être propagé. Par convention, on suppose que la distribution de $\hat{X}(k)$ est gaussienne pour deux raisons. D'abord, la distribution est complètement paramétrée par la moyenne et la covariance. En second lieu, étant donné que seulement les deux premiers moments sont connus, le vecteur estimé $\hat{X}(k+1|k+1)$ est donné par la correction du vecteur prédit $\hat{X}(k+1|k)$ en présence de la mesure courante $Y(k+1)$.

Dans le filtre de Kalman, une règle linéaire de correction ou de mise à jour est indiquée. La règle de mise à jour est:

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)Y(k+1) \quad (1.31)$$

$$P(k+1|k+1) = P(k+1|k) - K(k+1)P_{YY}(k+1)K^T(k+1) \quad (1.32)$$

$$Y(k+1) = Y(k+1) - \hat{Y}(k+1|k) \quad (1.33)$$

$$K(k+1) = P_{XY}(k+1|k)P_{YY}^{-1}(k+1) \quad (1.34)$$

Il est important de noter que ces équations sont seulement en fonction de valeurs prédites des deux premiers moments de $X(k)$ et $Y(k)$. Donc, le problème d'appliquer le

filtre de Kalman à un système non linéaire est la capacité de prédire les deux premiers moments de $X(k)$ et $Y(k)$.

1.1.5.2 Transformation d'incertitude

Le problème de prédiction du futur état ou de l'observation du système peut être exprimé sous la forme suivante :

Supposons que X est une variable aléatoire avec moyenne \bar{X} et covariance P_{XX} . Une deuxième variable aléatoire Z est liée à X par la fonction non linéaire

$$Z = f(X) \quad (1.35)$$

Nous souhaitons calculer la moyenne \bar{Z} et la covariance P_{ZZ} de Z . Pour calculer les statistiques de Z nous devons:

- déterminer la fonction de densité de la distribution transformée,
- évaluer les statistiques de cette distribution.

Dans quelques cas spéciaux (par exemple, lorsque $f(\cdot)$ est linéaire) des solutions exactes existent. Cependant, de telles solutions n'existent pas en général et des méthodes approximatives doivent être employées. Nous préconisons que la méthode devrait rapporter des statistiques consistantes. Idéalement, ceux-ci devraient être efficaces et non biaisées. Les statistiques transformées sont consistantes si l'inégalité :

$$P_{ZZ} = \mathbb{E}([Z - \bar{Z}][Z - \bar{Z}]^T) \geq 0 \quad (1.36)$$

Cette condition est extrêmement importante pour la validité de la méthode de transformation. Si les statistiques ne sont pas consistantes, la valeur de P_{ZZ} est sous-estimée. Si le filtre de Kalman utilise un ensemble inconsistant de statistiques, il y a la possibilité que le filtre diverge. Pour que la transformation soit plus efficace, la valeur P_{ZZ} de l'équation (1.36) devrait être réduite au minimum. En conclusion, il est souhaitable que l'estimation soit non biaisée ou $\bar{Z} \approx \mathbb{E}(Z)$.

Le problème de développer une procédure consistante, efficace et non biaisée de transformation peut être examiné en considérant le développement en séries de Taylor de l'équation (1.35) au voisinage de \bar{X} . Cette série peut s'exprimer par:

$$\begin{aligned} f(X) &= f(\bar{X} + \delta X) \\ &= f(\bar{X}) + \nabla f \cdot \delta X + \frac{1}{2} \nabla^2 f \cdot \delta X^2 + \frac{1}{3!} \nabla^3 f \cdot \delta X^3 + \frac{1}{4} \nabla^4 f \cdot \delta X^4 + \dots \end{aligned} \quad (1.37)$$

Où δX est une variable aléatoire gaussienne de moyenne nulle et de covariance P_{XX} et $\nabla^n f \cdot \delta X^n$ est le terme approprié d'ordre n dans les séries de Taylor multidimensionnelles. En prenant les espérances, on peut montrer que la moyenne et la covariance transformées sont :

$$\bar{Z} = f(\bar{X}) + \frac{1}{2} \nabla^2 f \cdot P_{XX} + \frac{1}{4} \nabla^4 f \cdot \mathbb{E}(\delta X^4) + \dots \quad (1.38)$$

$$P_{ZZ} = \nabla f \cdot P_{XX} \cdot \nabla f^T + \frac{2}{2 \times 4!} \nabla^2 f (\mathbb{E}(\delta x^4) - \mathbb{E}(\delta x^2) \cdot P_{XX}) - \mathbb{E}(P_{XX} \cdot \delta x^2) + P_{XX}^2 (\nabla^2 f)^T + \frac{1}{3!} \nabla^3 f \mathbb{E}(\delta x^4) (\nabla f)^T + \dots \quad (1.39)$$

En d'autres termes, le terme d'ordre n dans la série pour \bar{Z} est en fonction de moments de \bar{X} d'ordre n multiplié par les dérivées d'ordre n de $f(\cdot)$, évaluées au voisinage de $X = \bar{X}$. Si les moments et les dérivés peuvent être évalués correctement jusqu'au $n^{\text{ième}}$ ordre, la moyenne est aussi correcte jusqu'au $n^{\text{ième}}$ ordre. Des commentaires également semblables se retiennent pour l'équation de covariance, bien que la structure de chaque terme soit plus compliquée. Puisque chaque terme dans la série est mesuré par un terme graduel plus petit, les termes d'ordre inférieur dans les séries sont susceptibles d'avoir le plus grand impact. Donc, la marche à suivre de prédiction devrait être concentrée en évaluant les termes d'ordre inférieur. La linéarisation suppose que le deuxième terme et les termes d'ordre supérieur de δX dans l'équation (1.37) peuvent être négligés. Sous cette supposition :

$$\bar{Z} = f(\bar{X}) \quad (1.40)$$

$$P_{ZZ} = \nabla f P_{XX} (\nabla f)^T \quad (1.41)$$

En comparant ces expressions avec les équations (1.38) et (1.39), il est clair que ces approximations sont précises seulement si le deuxième terme et les termes d'ordres supérieurs dans la moyenne et le quatrième terme et les termes d'ordres supérieurs dans la covariance sont négligeables. Cependant, dans plusieurs situations pratiques la linéarisation introduit des biais ou des erreurs significatives.

1.1.6 Filtre de Kalman étendu

La formulation récursive du filtre de Kalman permet à tout instant d'estimer l'état d'un système dynamique linéaire. Dans un contexte plus général, dès l'apparition des non linéarités, la représentation d'état d'un système dynamique peut s'écrire à l'aide des équations suivantes :

$$X(k+1) = f(X(k), k) + G.U(k) + D.V(k) \quad (1.42)$$

$$Y(k) = h(X(k), k) + W(k) \quad (1.43)$$

Les hypothèses faites sur les grandeurs des lois linéaires d'évolution (1.12) et d'observation (1.13) sont toujours considérées comme valables pour les expressions (1.42) et (1.43). Seules les matrices constantes F et H sont, dans le cadre de la non linéarité, remplacées par des fonctions dépendant essentiellement du temps et du vecteur d'état.

Connaissant les moments d'ordre 1 et 2 du vecteur d'état, $\hat{X}(k) \approx N(\bar{X}(k), P(k))$, la formulation récursive du filtre de Kalman étendu est donnée par les équations suivantes :

$$\hat{X}(k+1|k) = f(\hat{X}(k), k) + G.U(k) \quad (1.44)$$

$$P(k+1|k) = F(\hat{X}(k+1|k))P(k|k)F^T(\hat{X}(k+1|k)) + DQ(k)D^T \quad (1.45)$$

$$K(k+1) = P(k+1|k)H^T(\hat{X}(k+1|k))S^{-1}(k+1) \quad (1.46)$$

$$S(k+1) = H(\hat{X}(k+1|k))P(k+1|k)H^T(\hat{X}(k+1|k)) + R(k+1) \quad (1.47)$$

$$\hat{X}(k+1) = \hat{X}(k+1|k) + K(k+1)(Y(k+1) - h(\hat{X}(k+1|k))) \quad (1.48)$$

$$P(k+1) = (I - K(k+1)H(\hat{X}(k+1|k)))P(k+1|k) \quad (1.49)$$

avec les deux définitions suivantes :

$$F(\hat{X}(k+1|k)) = \left. \frac{\partial f(X(k))}{\partial X(k)} \right|_{X(k)=\hat{X}(k)} \quad (1.50)$$

$$H(\hat{X}(k+1|k)) = \left. \frac{\partial h(X(k))}{\partial X(k)} \right|_{X(k)=\hat{X}(k+1|k)} \quad (1.51)$$

Les termes $F(\hat{X}(k+1|k))$ et $H(\hat{X}(k+1|k))$ des expressions (1.50) et (1.51) sont respectivement les matrices jacobiennes de la fonction du système et de la fonction de mesure.

Pour qu'on puisse appliquer le filtre de Kalman sur de tels modèles on doit donc, faire une linéarisation des équations de transition et de mesure autour de $\hat{X}(k)$ et $\hat{X}(k+1|k)$. Cette linéarisation est obtenue par le développement en série de Taylor du premier ordre des deux fonctions $f(X(k))$ et $h(X(k))$ et Le filtre obtenu est appelé filtre de Kalman étendu (FKE). Une étude du comportement du filtre étendu se trouve dans [35]. Cependant, en pratique, le FKE possède deux inconvénients bien connus, d'abord, la linéarisation des

modèles non linéaires peut produire des filtres fortement instables si les suppositions de linéarités locales sont violées. En second lieu, la dérivation de la matrice jacobienne est non triviale dans la plupart des applications et introduit ensuite souvent des difficultés significatives d'exécution.

1.1.7 Filtre de Kalman sans odeur (unscented Kalman filter)

1.1.7.1 Unscented transformation

Unscented transformation (UT) est une nouvelle méthode pour calculer les statistiques d'une variable aléatoire qui subit une transformation non linéaire. Elle est fondée sur l'intuition qu'il est plus facile d'approximer une distribution gaussienne que d'approximer une fonction non linéaire ou une transformation arbitraire [41]. Cette approche est illustrée dans la figure 1.3. Un ensemble de points (ou sigma points) sont choisis de sorte que leur moyenne et leur covariance soient \bar{X} et P_{XX} .

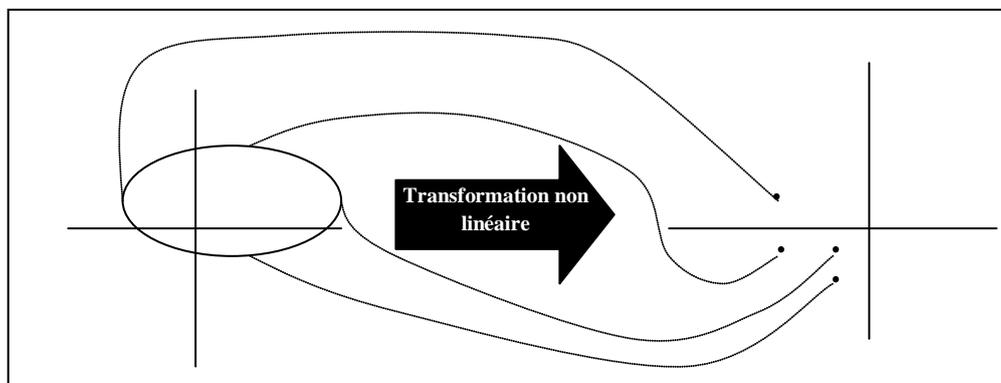


Fig. 1.3: Principe de unscented transformation

La fonction non linéaire est appliquée à chaque point alternativement pour rapporter un nuage de points transformés où \bar{Z} et P_{ZZ} sont leurs statistiques. Bien que cette méthode ait une ressemblance superficielle à des méthodes de type Monte Carlo, il y a une différence extrêmement importante et fondamentale. Les échantillons ne sont pas choisis au hasard mais plutôt selon un algorithme spécifique et déterministe. Des informations d'ordre supérieur sur la distribution peuvent être capturées en utilisant seulement un nombre très petit de points.

Une variable aléatoire X de dimension n , de moyenne \bar{X} et de covariance P_{XX} est approximée par $2n + 1$ points pondérés donnés par:

$$\begin{aligned}
\mathcal{X}_0 &= \bar{X} & \omega_0 &= \frac{\kappa}{n + \kappa} \\
\mathcal{X}_i &= \bar{X} + \left(\sqrt{(n + \kappa)P_{XX}} \right)_i & \omega_i &= \frac{1}{2(n + \kappa)} \\
\mathcal{X}_{i+n} &= \bar{X} - \left(\sqrt{(n + \kappa)P_{XX}} \right)_i & \omega_{i+n} &= \frac{1}{2(n + \kappa)}
\end{aligned}$$

Pour $i = 1, \dots, n$ (1.52)

Où $\kappa \in \mathbb{R}$, $\left(\sqrt{(n + \kappa)P_{XX}} \right)_i$ est la $i^{\text{ème}}$ ligne ou colonne de la racine carrée de la matrice $(n + \kappa)P_{XX}$ et ω_i est le poids associé avec le $i^{\text{ème}}$ point, tel que $\sum_{i=0}^{2n} \omega_i = 1$. La procédure de transformation est comme suit:

- ❖ la fonction non linéaire f est appliquée à chaque point \mathcal{X}_i pour l'ensemble de points transformés $\text{sigma}(Z_i)_{i=0, \dots, 2n}$.

$$Z_i = f(\mathcal{X}_i) \quad (1.53)$$

- ❖ la moyenne est donnée par la somme pondérée des points transformés,

$$\bar{Z} = \sum_{i=0}^{2n} \omega_i Z_i \quad (1.54)$$

- ❖ la covariance est donnée par,

$$P_{ZZ} = \sum_{i=0}^{2n} \omega_i (Z_i - \bar{Z})(Z_i - \bar{Z})^T \quad (1.55)$$

Les propriétés de cet algorithme ont été étudiées en détail dans [47] et [51].

Puisque la moyenne et la covariance de X sont capturées avec précision jusqu'au deuxième ordre, les valeurs calculées de la moyenne et de la covariance de Z sont aussi bien correctes au deuxième ordre. Ceci signifie que la moyenne est calculée à un ordre plus supérieur d'exactitude ou de précision que le filtre de Kalman étendu (FKE), tandis que la covariance est calculée au même ordre d'exactitude. Cependant, il y a d'autres avantages d'exécution ; puisque c'est la distribution de X qui est approchée plutôt que $f(\cdot)$, son développement en séries n'est pas tronqué à un ordre particulier. Il est possible de montrer que l'algorithme de cette transformation peut incorporer partiellement l'information des ordres plus élevés [82], menant à une plus grande précision.

La moyenne et la covariance sont calculées en utilisant des opérations standard de vecteurs et de matrices. Cela signifie que l'algorithme convient à n'importe quel choix du

modèle de processus et l'implémentation est extrêmement rapide parce qu'il n'est pas nécessaire d'évaluer la jacobienne, contrairement au FKE.

1.1.7.2 Filtre sans odeur (unscented Filter)

Les processus de transformation qui se produisent dans un filtre de Kalman comprennent les étapes suivantes:

- la prédiction d'un nouvel état du système $\hat{X}(k+1|k)$ et sa covariance associée $P(k+1|k)$. Cette prédiction doit tenir compte des effets de bruit du processus.
- la prédiction de l'observation prévue $\hat{Y}(k+1|k)$ et la covariance d'innovation $P_{YY}(k+1)$. Cette prédiction devrait inclure des effets de bruit d'observation.
- Finalement, on prédit la matrice de corrélation $P_{XY}(k+1)$.

Ces étapes peuvent être facilement adaptées en restructurant légèrement le vecteur d'état et les modèles du processus et d'observation. D'abord, le vecteur d'état est augmenté avec les termes de processus et de bruit pour donner un vecteur d'ordre $n^a = n + q$.

1. l'application de l'équation (1.52) au système augmenté indiqué par l'équation (1.65) pour créer l'ensemble de points sigma.
2. l'ensemble transformé de points est donné en substituant chaque point à travers le modèle de processus,

$$x_i(k+1|k) = f(x_i^a(k|k), U(k)) \quad (1.56)$$

3. la moyenne prédite est calculée comme suit :

$$\hat{X}(k+1|k) = \sum_{i=0}^{2n^a} \omega_i x_i(k+1|k) \quad (1.57)$$

4. la covariance prédite est calculée comme suit :

$$P(k+1|k) = \sum_{i=0}^{2n^a} \omega_i \left(x_i(k+1|k) - \hat{X}(k+1|k) \right) \left(x_i(k+1|k) - \hat{X}(k+1|k) \right)^T \quad (1.58)$$

5. la fonction non linéaire d'observation $h(\cdot)$ est appliquée à chaque point prédit $x_i(k+1|k)$ pour obtenir les points $Y_i(k+1|k)$:

$$Y_i(k+1|k) = h(x_i(k+1|k), U(k)) \quad (1.59)$$

6. l'observation prédite est calculée par :

$$\hat{Y}(k+1|k) = \sum_{i=0}^{2n^a} \omega_i Y_i(k+1|k) \quad (1.60)$$

7. puisque le bruit d'observation est additif et indépendant, la covariance de l'innovation est :

$$P_{YY}(k+1) = \sum_{i=0}^{2n^a} \omega_i \left(Y_i(k+1|k) - \hat{Y}(k+1|k) \right) \left(Y_i(k+1|k) - \hat{Y}(k+1|k) \right)^T \quad (1.61)$$

8. finalement, la matrice de corrélation est déterminée par :

$$P_{XY}(k+1) = \sum_{i=0}^{2n^a} \omega_i \left(X_i(k+1|k) - \hat{X}(k+1|k) \right) \left(Y_i(k+1|k) - \hat{Y}(k+1|k) \right)^T \quad (1.62)$$

Où le vecteur augmenté est :

$$X^a(k) = \begin{pmatrix} X(k) \\ V(k) \end{pmatrix} \quad (1.63)$$

Le modèle du processus est récrit en fonction de $X^a(k)$ par :

$$X(k) = f(X^a(k), U(k)) \quad (1.64)$$

et la transformation sans odeur utilise $2n^a + 1$ sigma points qui sont tiré de :

$$\hat{X}^a(k|k) = \begin{pmatrix} \hat{X}(k|k) \\ 0_q \end{pmatrix} \quad \text{et} \quad P^a(k|k) = \begin{pmatrix} P(k|k) & P_{XV}(k|k) \\ P_{XV}(k|k) & Q(k) \end{pmatrix} \quad (1.65)$$

1.1.7.3 Scaled unscented Kalman filter (SUKF)

Julier et Uhlmann ont décrit « Unscented Transformation » (UT) qui approxime une distribution de probabilité en utilisant un nombre restreint de points [51], [70]. Ces points sont propagés à travers le véritable système non linéaire et permettent l'estimation précise de la moyenne et la covariance postérieure au deuxième et au troisième ordre, pour n'importe quelle non linéarité [82].

Comme décrit précédemment, UT approxime une variable aléatoire X de dimension n , de moyenne \hat{X} et de covariance P par $2n + 1$ points ou échantillons.

Comme la dimension de l'espace d'état augmente, le rayon de la sphère qui borne tout les points augmente également. Pour commander ceci, nous utilisons la méthode de Julier pour mesurer les points [82]. Les nouveaux points et les poids utilisés pour trouver la moyenne sont:

$$X_i' = X_0 + \alpha(X_i - X_0) \quad (1.66)$$

$$\omega'_i = \begin{cases} \frac{\omega_0}{\alpha^2} + 1 - \frac{1}{\alpha^2} & i = 0 \\ \frac{\omega_i}{\alpha^2} & i \neq 0 \end{cases} \quad (1.67)$$

la covariance est trouvée en utilisant l'ensemble modifié de poids :

$$\omega''_i = \begin{cases} \frac{\omega_0}{\alpha^2} + 2 - \frac{1}{\alpha^2} - \alpha^2 + \beta & i = 0 \\ \frac{\omega_i}{\alpha^2} & i \neq 0 \end{cases} \quad (1.68)$$

Où β est un autre paramètre réglable. La moyenne et la covariance sont données par :

$$Z'_i = f(x'_i) \quad (1.69)$$

$$\hat{Z}' = \sum_{i=0}^{2n} \omega'_i Z'_i \quad (1.70)$$

$$P'_{ZZ} = \sum_{i=0}^{2n} \omega''_i (Z'_i - \hat{Z}') (Z'_i - \hat{Z}')^T \quad (1.71)$$

Il ne reste que de placer les paramètres réglables κ , α et β . Nous suivons les recommandations de Julier de prendre $\beta = 2$, pour capturer une partie du terme du quatrième ordre dans le développement en série de Taylor de la covariance. Nous choisissons :

$$\alpha = \frac{1}{\sqrt{n}}$$

pour rendre le diamètre de points indépendant de la taille du vecteur d'état. UT estime efficacement la moyenne et la covariance transformées [49], ainsi nous nous attendons à ce que l'estimation soit plus précise si la distribution antérieure est approximativement uniformément prélevée. La matrice de covariance estimée P'_{ZZ} est garantie d'être semi définie positive si tous les poids non transformés soient non négatifs, qui établit la condition $\kappa > 0$. Nous voulons également que les poids transformés soient non négatifs pour la robustesse (si un point a un poids essentiellement négatif, alors une non linéarité peut mener à une moyenne estimée biaisée et covariance gonflée), ce qui établit la condition stricte :

$$\kappa > n^2 - n \quad (1.72)$$

Nous choisissons réellement de rendre tous les poids transformés égaux, de sorte que :

$$\kappa = n^2 - \frac{n}{2} \quad (1.73)$$

Le filtre de Kalman sans odeur [46], [56], dénoté UKF en anglais, emploie l'UT pour les deux transformations (modèle de processus et fonction d'observation) exigées par le filtre de Kalman. Il fournit une estimation minimum de l'erreur quadratique moyenne (MMSE) de l'état d'un système discret non linéaire.

$$X(k+1) = f(X(k), U(k), V(k)) \quad (1.74)$$

$$Y(k) = h(X(k), U(k)) + W(k) \quad (1.75)$$

Où $X(k)$ est l'état du système à l'instant k , $f(\cdot)$ et $h(\cdot)$ sont des fonctions non linéaires du modèle d'état et de l'observation, $U(k)$ est le vecteur d'entrée, $V(k)$ est le bruit de processus, $Y(k)$ est l'observation et $W(k)$ est un bruit additif de mesure, avec V et W sont supposés de moyennes nulles et de covariances :

$$\mathbb{E}(V(k)V^T(j)) = \delta_{kj} Q(k)$$

$$\mathbb{E}(W(k)W^T(j)) = \delta_{kj} R(k)$$

$$\mathbb{E}(V(k)W^T(j)) = 0 \quad \forall k, j \quad (1.76)$$

Où $\delta_{kj} = 1$ pour $k = j$.

Une matrice augmentée de covariance est construite avec P , Q , et R sur la diagonale. Les équations (1.40) et (1.52) fournissent alors les points échantillonnés $\mathcal{X}'_i(k+1|k)$ qui indiquent non seulement X mais également V et W .

L'état prédit $\hat{X}(k+1|k)$ et sa covariance $P(k+1|k)$ sont estimés comme suit :

$$\mathcal{X}'_i(k+1|k) = f(\mathcal{X}'_i(k|k), U(k)) \quad (1.77)$$

$$\hat{X}(k+1|k) = \sum_{i=0}^{2n} \omega'_i \mathcal{X}'_i(k+1|k) \quad (1.78)$$

$$P(k+1|k) = \sum_{i=0}^{2n} \omega''_i \left(\mathcal{X}'_i(k+1|k) - \hat{X}(k+1|k) \right) \left(\mathcal{X}'_i(k+1|k) - \hat{X}(k+1|k) \right)^T \quad (1.79)$$

L'observation prédite \hat{Y} , sa covariance P_{YY} et la corrélation P_{XY} sont estimées comme :

$$Y_i(k+1|k) = h(\mathcal{X}'_i(k+1|k), U(k)) \quad (1.80)$$

$$\hat{Y}(k+1|k) = \sum_{i=0}^{2n} \omega'_i Y_i(k+1|k) \quad (1.81)$$

$$P_{YY}(k+1) = \sum_{i=0}^{2n} \omega_i'' \left(Y_i(k+1|k) - \hat{Y}(k+1|k) \right) \left(Y_i(k+1|k) - \hat{Y}(k+1|k) \right)^T \quad (1.82)$$

$$P_{XY}(k+1) = \sum_{i=0}^{2n} \omega_i'' \left(X_i'(k+1|k) - \hat{X}(k+1|k) \right) \left(Y_i(k+1|k) - \hat{Y}(k+1|k) \right)^T \quad (1.83)$$

L'état estimé $\hat{X}(k|j)$ à l'instant k et sa covariance $P(k|j)$, donnant toutes les observations jusque l'instant j , sont mis à jour comme suit :

$$P_{YY}(k+1) = P_{YY}(k+1) + R(k+1) \quad (1.84)$$

$$K(k+1) = P_{XY}(k+1)P_{YY}^{-1}(k+1) \quad (1.85)$$

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1) \left(Y(k+1) - \hat{Y}(k+1|k) \right) \quad (1.86)$$

$$P(k+1|k+1) = P(k+1|k) - K(k+1)P_{YY}(k+1)K^T(k+1) \quad (1.87)$$

où K est le gain du filtre de Kalman.

Pour les fonctions linéaires, UKF est équivalent au filtre de Kalman (FK). La complexité informatique de UKF est identique au FKE, mais elle est plus précise et n'exige pas la dérivation de n'importe quelle Jacobienne.

1.2 Conclusion

Nous nous sommes intéressés dans ce chapitre à l'estimation bayésienne du processus d'état markovien $(x_k)_{k \in \mathbb{N}}$ observé à travers le processus de mesures $(y_k)_{k \in \mathbb{N}^*}$. Plus particulièrement, l'objectif était de fournir une estimation de la loi conditionnelle $L_{k|k} = p(X_k | Y_1, \dots, Y_k)$. Si le filtre optimal fournit une expression récursive de cette loi, il n'est en pratique calculable que dans des cas très restreints tels que le cas linéaire avec bruits additifs gaussiens, i.e., celui du filtre de Kalman dont nous avons rappelé les équations. Pour traiter le cas général, nous avons exposé dans un premier temps les méthodes déterministes développées depuis les années 70. Les filtres de Kalman étendus et le filtre de Kalman unscented reposent sur l'approximation de la loi $L_{k|k}$ par une gaussienne. Ils proposent d'en approximer ses deux premiers moments respectivement par linéarisation locale du modèle d'état et de mesure et par approximation de la loi gaussienne par un nuage déterministe de points. Si les filtres de Kalman étendus restent la méthode la plus utilisée par les ingénieurs à notre époque, il faut cependant souligner qu'aucun résultat de convergence n'est prouvé dans le cas général. Ils sont de toute façon un mauvais choix

lorsque la loi $L_{k|k}$ est multimodale, ce qui nous oblige de chercher un autre filtre capable de résoudre ce problème de non gaussienneté et de multimodalité. La solution est donc sans doute le filtre particulaire ou encore les méthodes séquentielles de Monte Carlo sur lesquelles nous insisterons plus particulièrement dans le deuxième chapitre.

Chapitre 2

Filtrage particulaire et la poursuite mono cible

Dans ce chapitre, nous exposons l'algorithme du filtrage particulaire ou encore méthodes séquentielles de Monte-Carlo. Nous présentons tout d'abord le filtre originel bootstrap. Puis, nous exposons une structure générique de MSMC comprenant les principales variantes des méthodes particulières. Elle comporte deux étapes:

- un échantillonnage pondéré séquentiel selon une loi de proposition générique, dénommé sequential importance sampling et dénoté SIS en anglais. Nous étudierons le choix optimal malheureusement souvent inaccessible et les choix courants réalisés pour déterminer la fonction de proposition.
- un rééchantillonnage des particules pour lequel nous verrons plusieurs techniques.

Pour clore ce chapitre, nous montrerons l'application de cet algorithme réalisé pour la poursuite dans le cadre de cette thèse.

2.1 Méthodes séquentielles de Monte-Carlo: filtres particuliers

Les méthodes particulières sont fondées sur le principe de Monte-Carlo pour estimer et prédire en ligne des systèmes dynamiques.

Reprenons le problème de filtrage non linéaire suivant d'une chaîne de Markov cachée [18] :

$$\begin{aligned} X_0 &\sim L_0(dx_0) \\ X_k | X_{k-1} &\sim p(X_k | X_{k-1}) \\ Y_k &\sim h_k(y_k | x_k) \end{aligned} \quad (2.1)$$

Par exemple, les processus d'état et de mesure pourront être définis par :

$$X_k = F_k(X_{k-1}, V_k) \quad (2.2)$$

$$Y_k = H_k(X_k, W_k) \quad (2.3)$$

où les fonctions F_k et H_k sont non linéaires et V_k et W_k sont des bruits blancs indépendants. On se concentre sur l'estimation de la loi $L_k(X_k) = p(X_k | Y_{1:k})$. Étant donné que cette loi évolue au cours du temps, des méthodes de Monte-Carlo classiques par chaîne de Markov telles que celles exposées dans l'annexe sont peu performantes. En effet, elles nécessitent

la simulation entière d'une chaîne à chaque pas de temps. Une nouvelle classe de méthodes, dénommée "méthodes séquentielles de Monte-Carlo", MSMC, "filtres de Monte-Carlo" ou encore "filtres particulaires" a été abondamment utilisée et étudiée ces dernières années. Ces méthodes permettent de représenter la loi conditionnelle $L_{k|k}$ par une mesure discrète égale à une somme finie de mesures de Dirac concentrées en des points dénommés "particules" et pondérées par des coefficients appelés "poids des particules". Elles consistent à faire évoluer au cours du temps ce système de particules ainsi que leur poids et reposent sur:

1. l'usage récursif du principe d'échantillonnage pondéré;
2. une sélection par rééchantillonnage.

Les particules assorties de leurs poids permettront notamment d'accéder à tous les moments de la loi conditionnelle $L_{k|k}$ et plus généralement à toute fonction de celle-ci. Ces particules constituent ici un maillage aléatoire évolutif de l'espace d'état.

On peut attribuer la première MSMC au filtre de Monte-Carlo pondéré introduit dès les années 70 [65], voire dès les années 50 où l'idée d'un tel filtre est émise dans [6]. Les particules évoluent alors indépendamment les unes des autres au cours du temps, guidées par la loi d'évolution du processus d'état et pondérées par la vraisemblance de la mesure courante. Cette méthode était alors trop novatrice par rapport aux moyens informatiques de l'époque, dans la mesure où le nombre de particules se doit d'être élevé et l'absence de sélection par rééchantillonnage conduisait en pratique à une dégénérescence rapide des poids des particules. Ce principe fut indépendamment repris au début des années 90 en y ajoutant l'étape cruciale de sélection par Del Moral et al dans [3], Gordon et al dans [67], Kitagawa dans [29] et Isard et Blake dans [59]. Les appellations respectives données par leurs auteurs sont alors le filtre particulaire avec interaction ou interacting particle filter, le filtre bootstrap ou bootstrap filter, le Monte-Carlo filter et l'algorithme dit de Condensation pour Conditional Density Propagation. Nous conserverons arbitrairement la dénomination de filtre bootstrap pour désigner cette classe initiale de filtres particulaires, pour laquelle les particules sont propagées selon la loi d'évolution de l'état et rééchantillonnées à chaque étape selon un schéma classique que nous verrons ultérieurement. Reprenons le principe de l'échantillonnage pondéré séquentiel, ce filtre fut généralisé à l'utilisation d'une fonction de proposition quelconque et fut depuis appliqué dans de nombreux domaines: en traitement

du signal radar [37],[60] et [67] pour les calculs bayésiens classiques [40], pour la déconvolution de signaux numériques [44], en analyse d'image [50], pour la modélisation en finance et l'étude des séries temporelles [49], [66], pour le diagnostic médical [52], pour la localisation de robots mobiles [63], pour la sélection de modèles [61] et en traitement de la parole [78]. Notons qu'en dehors du cadre bayésien supposé dans les applications ci-dessus, ces méthodes sont également appliquées sous des formes et des appellations légèrement différentes dans des domaines tels que la science des polymères ou la physique quantique, comme le souligne Iba dans [74] dont le but est de développer les liens entre ces différentes disciplines afin qu'elles s'enrichissent mutuellement.

Les deux principaux écueils des méthodes particulières sont la dégénérescence des poids, lorsque seuls quelques poids sont significatifs, ainsi que l'appauvrissement des états, lorsque toutes les particules sont quasiment égales. De nombreuses améliorations et extensions ont été proposées parmi lesquelles les travaux suivants: [58], [59], [66], [68] et [76]. Ces études portent principalement sur la recherche de fonctions de proposition performantes pour l'échantillonnage pondéré afin de limiter la dégénérescence des poids et des états, ainsi que sur l'utilisation de techniques de rééchantillonnage limitant la dégénérescence inutile des poids [83].

Pour une première approche des méthodes séquentielles de Monte-Carlo, nous conseillons la lecture du tutoriel d'Arulampalam et al. [80] et de celui de Crisan et Doucet [77] en ce qui concerne les questions de convergence. Enfin, l'ouvrage édité par Doucet et al [72] présente ces méthodes ; des aspects théoriques à leurs nombreuses applications, en détaillant un grand nombre de variantes et d'améliorations possibles de l'algorithme.

Notations: une méthode particulaire sera caractérisée par l'évolution d'un ensemble de N particules $(s^n)_{n=1,\dots,N} \in \mathbb{R}^{n_x}$ et de leurs poids $(\omega^n)_{n=1,\dots,N} \in \mathbb{R}$, appelé le nuage de particules et noté $S^N = (s^n, \omega^n)_{n=1,\dots,N}$. La mesure L^N associée à ce nuage est la mesure discrète $L^N \triangleq \sum_{n=1}^N \omega^n \delta_{s^n}$ où δ_s désigne la mesure de Dirac centrée au point s .

2.1.1 Filtre de Monte-Carlo et filtre bootstrap

Le filtre dit de Monte Carlo [83],[8] introduit l'utilisation d'un nuage de particules pour approximer la loi $L_{k|k}$ par une mesure discrète pondérée. L'approximation consiste à discrétiser les équations du filtre optimal. Il faut pour cela être en mesure de:

- générer des échantillons selon la loi initiale $L_0(x)$;

- générer des échantillons selon la loi d'évolution $p(x_k|x_{k-1})$ pour tout x_{k-1} de l'espace d'état.
- évaluer la fonction $l_k(y_k; x_k)$ égale à une constante de normalisation près à la fonction de vraisemblance des mesures $h_k(y_k|x_k)$.

Supposons alors que l'on dispose du nuage $S_{k-1|k-1}^N$ tel que la loi discrète associée $L_{k-1|k-1}^N$ constitue une approximation de la loi $L_{k-1|k-1}$. En remplaçant $L_{k-1|k-1}$ par $L_{k-1|k-1}^N$ dans l'équation de prédiction du filtre optimal (1.10), on obtient une discrétisation de l'intégrale que l'on approxime ensuite par la loi dénotée $L_{k|k-1}^N$ et égale à une somme discrète de lois de Dirac centrées en des particules prédites à partir des particules s_{k-1}^n :

$$\begin{aligned} \int_{\mathcal{R}^{n_x}} p(x_k|x_{k-1}) \sum_{n=1}^N \omega_{k-1}^n \delta_{s_{k-1}^n}(x_{k-1}) &= \sum_{n=1}^N \omega_{k-1}^n p(x_k|s_{k-1}^n) \\ &\approx L_{k|k-1}^N(x_k) \triangleq \sum_{n=1}^N \omega_{k-1}^n \delta_{s_k^n}(x_k) \end{aligned} \quad (2.4)$$

où s_k^n désigne une particule prédite obtenue par propagation selon la loi d'évolution de la chaîne : $s_k^n \sim p(x_k|x_{k-1} = s_{k-1}^n)$.

Le nuage prédit $S_{k|k-1}^N = (s_k^n, \omega_{k-1}^n)_{n=1, \dots, N}$ constitue ainsi une approximation de la loi prédite $L_{k|k-1}$. L'équation de mise à jour du filtre optimal (1.11) énonce ensuite que la loi $L_{k|k}(x_k)$ est égale à une constante de normalisation près au produit $p(y_k|x_k)L_{k|k-1}(x_k)$.

Suivant le principe d'échantillonnage pondéré exposé dans l'annexe au paragraphe A.2.3 [83], on affecte aux éléments s_k^n de l'échantillon les poids non normalisés $\tilde{\omega}_k^n \triangleq \omega_{k-1}^n l_k(y_k; s_k^n)$ que l'on normalise en définissant $\omega_k^n \triangleq \frac{\tilde{\omega}_k^n}{\sum_{n=1}^N \tilde{\omega}_k^n}$. Le nuage $S_k^N = (s_k^n, \omega_k^n)_{n=1, \dots, N}$ constitue alors une approximation de la loi $L_{k|k}$. On peut ensuite poursuivre de même l'estimation de $L_{k+1|k+1}$ récursivement. Les étapes de cet algorithme sont résumées dans l'algorithme 2.1.

Composé uniquement de ces deux étapes de propagation des particules et de mise à jour de leur poids, ce filtre dit filtre de Monte-Carlo n'est pas viable en pratique car il s'avère que la majorité des poids dégénère très rapidement vers zéro. Vous pouvez voir en effet dans l'annexe paragraphe A.2.4 que la taille efficace d'un échantillon issue de la méthode d'échantillonnage pondéré était inversement proportionnelle à la variance des poids selon

Algorithme 2.1 : Filtre de Monte-Carlo

- *Initialisation* :
 - o Pour $n = 1, \dots, N$, générer s_0^n selon $L_0(x)$ et poser $\omega_0^n = 1/N$
- Pour $k = 1, \dots, \text{fin}$:
 - o Pour $n = 1, \dots, N$, générer s_k^n selon $p(x_k | x_{k-1} = s_{k-1}^n)$
 - o Mise à jour des poids :
 - Pour $n = 1, \dots, N$, calculer les poids non normalisés : $\tilde{\omega}_k^n = \omega_{k-1}^n l_k(y_k; s_k^n)$
 - Pour $n = 1, \dots, N$, normaliser les poids : $\omega_k^n = \frac{\tilde{\omega}_k^n}{\sum_{n=1}^N \tilde{\omega}_k^n}$;
 - o Sortie de l'algorithme : le nuage de particules issues de leur poids $S_k^N = (s_k^n, \omega_k^n)_{n=1, \dots, N}$ permet d'approcher la loi a posteriori par :

$$p(x_k | y_{1:k}) \approx \sum_{n=1}^N \omega_k^n \delta_{s_k^n}(x_k)$$

et pour toute fonction ϕ intégrable par rapport à $p(x_k, y_{1:k})$:

$$\mathbb{E}[\widehat{\phi(x_k)}] = \sum_{n=1}^N \omega_k^n \phi(s_k^n)$$

l'équation (A.16). Or, il est montré dans [40] que les poids issus d'un échantillonnage pondéré séquentiel forment une martingale et que par conséquent leur variance augmente inévitablement avec le temps. Ce filtre ne devient en pratique utilisable que lorsque les moyens informatiques sont suffisants et surtout lorsque l'étape de rééchantillonnage permet de limiter cette dégénérescence des poids. Le filtre alors obtenu est connu sous le nom de filtre bootstrap [83]. A la différence du filtre de Monte-Carlo que l'on vient d'exposer, les lois $L_{k|k}$ propagées par le filtre bootstrap sont des lois équipondérées, i.e., $\omega_k^n = 1/N$ pour tout $n = 1, \dots, N$. Supposons ainsi que la loi $L_{k-1|k-1} = \frac{1}{N} \sum_{n=1}^N \delta_{s_{k-1}^n}$. Les particules sont propagées comme précédemment selon la loi $p(x_k | x_{k-1} = s_{k-1}^n)$.

Les particules prédites sont alors notées \tilde{s}_k^n . Les nouveaux poids non normalisés sont égaux à $\tilde{\omega}_k^n = l_k(y_k; \tilde{s}_k^n)$ et les poids normalisés $\omega_k^n = \frac{\tilde{\omega}_k^n}{\sum_{n=1}^N \tilde{\omega}_k^n}$.

L'étape cruciale de rééchantillonnage consiste alors à tirer avec remise N éléments notés $(s_k^n)_{n=1, \dots, N}$ parmi $(\tilde{s}_k^n)_{n=1, \dots, N}$ et cela proportionnellement aux poids $(\omega_k^n)_{n=1, \dots, N}$. On

définit alors le nouveau nuage par $S_k^N \triangleq (s_k^n, 1/N)_{n=1, \dots, N}$. On dispose bien ainsi d'une mesure discrète équipondérée distribuée approximativement selon $L_k | k$.

La dégénérescence des poids est ainsi limitée car les particules de plus faible poids ne sont pas sélectionnées durant le rééchantillonnage. Celles de plus fort poids sont en revanche dupliquées plusieurs fois dans le nuage S_k^N .

L'algorithme du filtre bootstrap est résumé dans l'algorithme 2.2. Cette méthode a connu un vif succès durant les dernières décennies. Cependant, la réussite d'un tel algorithme est intimement liée à la capacité de la loi d'évolution à prédire les particules dans des régions de forte vraisemblance.

Algorithme 2.2 : Filtre particulaire de type bootstrap.

- *Initialisation :*
 - o Pour $n = 1, \dots, N$, générer s_0^n selon $L_0(x)$ et poser $\omega_0^n = 1/N$
- *Pour $k = 1, \dots, \text{fin}$:*
 - o Pour $n = 1, \dots, N$, générer \tilde{s}_k^n selon $p(x_k | x_{k-1} = s_{k-1}^n)$;
 - o *Mise à jour des poids :*
 - Pour $n = 1, \dots, N$, calculer les poids non normalisés : $\tilde{\omega}_k^n = l_k(y_k; \tilde{s}_k^n)$;
 - Pour $n = 1, \dots, N$, normaliser les poids : $\omega_k^n = \frac{\tilde{\omega}_k^n}{\sum_{n=1}^N \tilde{\omega}_k^n}$
 - o *Rééchantillonnage :*
 - Pour $n = 1, \dots, N$, tirer avec remise s_k^n parmi $(\tilde{s}_k^n)_{n=1, \dots, N}$ proportionnellement aux poids $(\tilde{\omega}_k^n)_{n=1, \dots, N}$ puis affecter aux poids ω_k^n la valeur $1/N$.
 - o *Sortie de l'algorithme :* le nuage de particules issues de leur poids $S_k^N = (s_k^n, 1/N)_{n=1, \dots, N}$ permet d'approcher la loi a posteriori par :

$$p(x_k | y_{1:k}) \approx \frac{1}{N} \sum_{n=1}^N \delta_{s_k^n}(x_k)$$

et pour toute fonction ϕ intégrable par rapport à $p(x_k, y_{1:k})$:

$$\mathbb{E}[\widehat{\phi}(x_k)] = \frac{1}{N} \sum_{n=1}^N \phi(s_k^n)$$

Dans la section suivante, nous exposons de façon générique l'échantillonnage pondéré séquentiel. Nous avons volontairement introduit les méthodes particulières par le filtre bootstrap qui déclencha de nombreuses recherches sur ces méthodes [83]. Notons

cependant que ce filtre peut plus généralement être défini non pas sur les états x_k mais sur les trajectoires $x_{1:k}$ en formant chaque particule $s_{0:k}^n$ comme la concaténation $(s_{0:k-1}^n, s_k^n)$. Le nuage obtenu $(s_{0:k}^n, 1/N)_{n=1, \dots, N}$ constitue alors une approximation de la loi $p(x_{0:k} | y_{1:k})$. D'autre part, le principe d'échantillonnage pondéré peut être plus largement appliqué au cas où les particules sont propagées selon une fonction quelconque autre que la loi d'évolution. Une telle propagation permet de mieux guider les particules lors de leur propagation en tenant compte des observations à travers une fonction de proposition. L'algorithme plus général obtenu est l'échantillonnage pondéré séquentiel avec rééchantillonnage (SIR).

2.1.2 Echantillonnage pondéré séquentiel avec rééchantillonnage

Nous nous concentrons tout d'abord sur la représentation de la loi complète $p(x_{0:k}, y_{1:k})$ [83], et non pas uniquement de la loi marginale $p(x_k, y_{1:k})$ tel que cela est réalisé dans [57]. On suppose pour cela que l'on est en mesure de :

- générer des échantillons selon la loi initiale $L_0(x)$.
- générer des échantillons $x_{0:k}$ selon une loi de proposition $g_{0:k}(x_{0:k}; y_{1:k})$.
- d'évaluer la fonction de proposition ainsi que la fonction $l_k(x_k; y_k)$ égale à une constante de normalisation près à la fonction de vraisemblance des mesures.

La notation $g_{0:k}(x_{0:k}; y_{1:k})$ signifie que $g_{0:k}$ est une fonction de $x_{0:k}$ qui dépend des paramètres $y_{1:k}$. Cette notation est similaire à celle utilisée pour désigner la loi normale de moyenne m et de covariance Σ_V par $N(x; m, \Sigma_V)$.

Afin que le ratio $\frac{p(x_{0:k} | y_{1:k})}{g_{0:k}(x_{0:k}; y_{1:k})}$ soit bien défini pour tout x , on suppose que le support de $g_{0:k}(\cdot)$ contient le support de $p(\cdot)$. Reprenons le principe de l'échantillonnage pondéré, exposé en annexe, pour obtenir un échantillon distribué approximativement selon la loi $p(x_{0:k} | y_{1:k})$.

Si l'on tire $(s_{0:k}^n)_{n=1, \dots, N}$ selon la fonction de proposition $g_{0:k}(x_{0:k}; y_{1:k})$ et qu'on leur associe les poids $\omega(s_{0:k}^n) \propto \frac{p(s_{0:k}^n | y_{1:k})}{g_{0:k}(s_{0:k}^n; y_{1:k})}$, alors la loi discrète $\sum_{n=1}^N \omega(s_{0:k}^n) \delta_{s_{0:k}^n}$ est une approximation de $p(x_{0:k} | y_{1:k})$.

Bien que cette technique d'échantillonnage pondéré soit théoriquement viable, elle est contrainte à recalculer à chaque pas de temps k les composantes des instants précédents

$0, \dots, k-1$ de l'état. Afin de disposer d'une formulation récursive de l'expression des particules et de leurs poids ne nécessitant pas de recalculer $s_{0:k-1}^n$ pour obtenir $s_{0:k}^n$ nous imposons maintenant à la loi de proposition $g_{0:k}$ d'être de la forme suivante:

$$g_{0:k}(x_{0:k}; y_{1:k}) = g_{0:k-1}(x_{0:k-1}; y_{1:k-1}) \cdot g_k(x_k; x_{0:k-1}, y_{1:k}). \quad (2.5)$$

Le rééchantillonnage séquentiel consiste alors à tirer s_k^n selon $g_k(s; s_{0:k-1}; y_{1:k})$. Le calcul des poids associés s'exprime alors également récursivement. En effet, par indépendance des mesures conditionnellement aux états, on peut écrire:

$$p(y_{1:k} | x_{0:k}) = p(y_k | x_k) p(y_{1:k-1} | x_{0:k-1}) \quad (2.6)$$

et la relation markovienne sur l'état implique:

$$p(x_{0:k}) = p(x_k | x_{k-1}) p(x_{0:k-1}) \quad (2.7)$$

Les poids non normalisés s'expriment alors:

$$\begin{aligned} \tilde{\omega}_k(x_{0:k}) &= \frac{p(y_{1:k} | x_{0:k}) p(x_{0:k})}{g_{0:k-1}(x_{0:k-1}; y_{1:k-1}) g_k(x_k; x_{0:k-1}, y_{1:k})} \\ &= \frac{p(y_k | x_k) p(y_{1:k-1} | x_{0:k-1}) p(x_{0:k})}{g_{0:k-1}(x_{0:k-1}; y_{1:k-1}) g_k(x_k; x_{0:k-1}, y_{1:k})} \\ &= \frac{\omega_{k-1}(x_{0:k-1})}{p(x_{0:k-1})} \frac{p(y_k | x_k) p(x_{0:k})}{g_k(x_k; x_{0:k-1}, y_{1:k})} \\ &= \omega_{k-1}(x_{0:k-1}) \frac{p(y_k | x_k) p(x_k | x_{k-1})}{g_k(x_k; x_{0:k-1}, y_{1:k})}. \end{aligned} \quad (2.8)$$

Cette formulation récursive permet de propager les particules et de mettre à jour leurs poids séquentiellement. L'algorithme générique séquentiel à échantillonnage pondéré, ou sequential importance sampling en anglais, est décrit dans l'algorithme 2.3. Lorsqu'une étape de rééchantillonnage est introduite, on obtient l'algorithme générique séquentiel à échantillonnage pondéré et rééchantillonnage, dénommé sequential importance resampling et dénoté SIR en anglais. Ce dernier algorithme est résumé en algorithme 2.4. Du point de vue des notations, les filtres particuliers avec rééchantillonnage diffèrent des filtres sans rééchantillonnage par l'utilisation de la notation \tilde{s}_k^n pour décrire les particules prédites et de s_k^n pour décrire les particules après le rééchantillonnage. Dans les algorithmes sans rééchantillonnage, l'unique notation s_k^n est utilisée.

On peut également obtenir de façon plus intuitive une récursion sur la loi $L_{k|k}$: supposons que l'on dispose du nuage $S_{k-1|k-1}^N$, telle que la loi discrète associée $L_{k-1|k-1}^N$, constitue une approximation de la loi $L_{k-1|k-1}$. Si l'on propage les particules du nuage $S_{k-1|k-1}^N$

selon une loi de proposition $g(s; s_{k-1}, y_k)$, l'échantillon obtenu sera distribué selon $L_{k|k}$ si l'on affecte à ses éléments les poids :

$$\tilde{\omega}_k^n \propto \omega_k^n \frac{p(y_k | \tilde{s}_k^n) p(\tilde{s}_k^n | s_{k-1}^n)}{g_k(s_k^n; s_{k-1}^n, y_k)}. \quad (2.9)$$

Algorithme 2.3 : Filtre particulaire à échantillonnage pondéré séquentiel (SIS)

- *Initialisation :*
 - o Pour $n = 1, \dots, N$, générer s_0^n selon $L_0(x)$ et poser $\omega_0^n = 1/N$
- Pour $k = 1, \dots, \text{fin}$:
 - o Pour $n = 1, \dots, N$, générer s_k^n selon $g_k(x_k; s_{0:k-1}^n, y_{1:k})$ et poser

$$s_{0:k}^n = (s_{0:k-1}^n, s_k^n);$$
 - o Mise à jour des poids :

- Pour $n = 1, \dots, N$, calculer les poids non normalisés :

$$\tilde{\omega}_k^n = \omega_{k-1}^n \frac{p(s_k^n | s_{k-1}^n) l_k(y_k; s_k^n)}{g_k(s_k^n; s_{0:k-1}^n, y_{1:k})}$$

- Pour $n = 1, \dots, N$, normaliser les poids : $\omega_k^n = \frac{\tilde{\omega}_k^n}{\sum_{n=1}^N \tilde{\omega}_k^n}$.

- o *Sortie de l'algorithme:* le nuage de particules issues de leur poids $S_{0:k} = (s_{0:k}^n, \omega_k^n)_{n=1, \dots, N}$ permet d'approcher la loi a posteriori par :

$$p(x_{0:k} | y_{1:k}) \approx \sum_{n=1}^N \omega_k^n \delta_{s_k^n}(x_{0:k})$$

et pour toute fonction ϕ intégrable par rapport à $p(x_{0:k} | y_{1:k})$:

$$\mathbb{E}[\phi(\widehat{x_{0:k}})] = \sum_{n=1}^N \omega_k^n \phi(s_{0:k}^n)$$

En particulier, $p(x_k | y_{1:k}) \approx \sum_{n=1}^N \omega_k^n \delta_{s_k^n}(x_k)$.

Le numérateur est la probabilité d'obtenir la particule \tilde{s}_k^n sous l'hypothèse qu'elle est distribuée selon $L_{k|k}$. Le dénominateur est la probabilité d'obtenir une telle particule sous l'hypothèse qu'elle ait été tirée selon la loi de proposition.

L'utilisation d'une loi de proposition g_k différente du noyau de transition T_k du modèle peut être vue comme l'adoption d'un modèle dynamique autre que le modèle dont sont issus les processus d'état et d'observation mais permettant toutefois d'atteindre la loi $p(x_k | y_{1:k})$. Comme exposé dans [71], on peut définir le modèle dynamique dit virtuel:

$$\left\{ \begin{array}{l} X_0 \sim \tilde{L}_0(x_0) \\ X_k \sim \tilde{T}_k(x_k; x_{k-1}, y_{k-1}) \\ Y_k \sim \tilde{h}_k(y_k | x_k) \end{array} \right. \quad (2.10)$$

Algorithme. 2.4 : Filtre particulaire à échantillonnage pondéré séquentiel avec rééchantillonnage (SIR)

- Initialisation :

o Pour $n = 1, \dots, N$, générer s_0^n selon $L_0(x)$ et poser $\omega_0^n = 1/N$

- Pour $k = 1, \dots, \text{fin}$:

o Pour $n = 1, \dots, N$, générer \tilde{s}_k^n selon $g_k(x_k; s_{0:k-1}^n, y_{1:k})$ et poser $\tilde{s}_{0:k}^n = (s_{0:k-1}^n, \tilde{s}_k^n)$;

o Mise à jour des poids :

- Pour $n = 1, \dots, N$, calculer les poids non normalisés : $\tilde{\omega}_k^n = \frac{p(\tilde{s}_k^n | s_{k-1}^n) l_k(y_k; \tilde{s}_k^n)}{g_k(\tilde{s}_k^n; s_{k-1}^n, y_{1:k})}$

- Pour $n = 1, \dots, N$, normaliser les poids : $\omega_k^n = \frac{\tilde{\omega}_k^n}{\sum_{n=1}^N \tilde{\omega}_k^n}$.

o Rééchantillonnage :

Pour $n = 1, \dots, N$, tirer avec remise $s_{0:k}^n$ parmi $(\tilde{s}_{0:k}^n)_{n=1, \dots, N}$ proportionnellement aux poids $(\omega_k^n)_{n=1, \dots, N}$ puis donner aux poids ω_k^n la valeur $1/N$.

o Sortie de l'algorithme: le nuage de particules issues de leur poids $S_{0:k} = (s_{0:k}^n, 1/N)_{n=1, \dots, N}$ permet d'approcher la loi a posteriori par :

$$p(x_{0:k} | y_{1:k}) \approx \frac{1}{N} \sum_{n=1}^N \delta_{s_k^n}(x_{0:k})$$

et pour toute fonction ϕ intégrable par rapport à $p(x_{0:k} | y_{1:k})$:

$$\mathbb{E}[\phi(\widehat{x_{0:k}})] = \frac{1}{N} \sum_{n=1}^N \phi(s_{0:k}^n)$$

En particulier, $p(x_k | y_{1:k}) \approx \frac{1}{N} \sum_{n=1}^N \delta_{s_k^n}(x_k)$.

où le noyau de transition \tilde{T}_k est associé à la fonction de proposition g et où la fonction \tilde{h}_k est telle que pour tout x_{k-1}, y_{k-1} , on ait

$$\tilde{h}_k(y_k | x_k) \tilde{T}_k(x_k; x_{k-1}, y_{k-1}) \propto h_k(y_k | x_k) T_k(x_k; x_{k-1}, y_{k-1}) \quad (2.11)$$

Le filtre optimal appliqué au modèle virtuel défini en (2.10) permet d'estimer récursivement la loi notée $\tilde{L}_{k|k}$ du modèle. Les lois prédites $\tilde{L}_{k|k-1}$ et $L_{k|k-1}$ ne coïncident certes pas, en revanche, du fait de la relation (2.11), les lois $\tilde{L}_{k|k}$ et

$L_{k|k}$ coïncident. Nous verrons ultérieurement que l'utilisation de tels modèles virtuels est particulièrement intéressante car la convergence des méthodes particulières dépend des propriétés du modèle utilisé pour la propagation des particules, i.e., soit le modèle générateur des processus d'état et de mesures, soit le modèle virtuel ci-dessus.

Bien que cet algorithme soit théoriquement viable pour toute fonction g_k telle que les poids soient bien définis, la qualité de l'estimation sera d'autant meilleure que la fonction g_k sera proche de $p(x_{0:k} | y_{1:k})$. Nous détaillons maintenant les principaux choix effectués pour la fonction de proposition g_k .

Utilisation de l'a priori :

Le choix le plus répandu [37], [49] et [50] consiste à propager les particules selon la loi d'évolution a priori du système $g_k(s_k; s_{0:k-1}^n, y_{1:k}) = p(s_k | s_{k-1}^n)$. On retrouve bien alors l'expression des poids du filtre bootstrap : $\tilde{\omega}_k^n = l_k(y_k; s_k^n)$. Ne tenant pas compte de la mesure courante lors de la propagation, le risque est alors que les particules ne couvrent pas de façon optimale les régions de forte vraisemblance. Ce choix rend l'algorithme relativement sensible aux points aberrants [83].

Pour limiter l'augmentation de la variance inconditionnelle des poids, il est intéressant de rechercher une fonction de proposition qui minimise la variance conditionnellement aux mesures et à l'état précédent. La proposition suivante [1] exhibe une telle fonction.

Proposition : *La fonction de proposition $g(s_k; s_{0:k-1}^n, y_{1:k}) = p(s_k | s_{k-1}^n, y_k)$ minimise la variance des poids $\tilde{\omega}_k^n$ conditionnellement aux $s_{0:k-1}^n$ et $y_{1:k}$.*

Preuve :

$$\mathbb{E}_g[\tilde{\omega}_k^n] = \int \tilde{\omega}_{k-1}^n p(y_k | s_k^n) p(s_k^n | s_{k-1}^n) ds_k^n = \tilde{\omega}_{k-1}^n p(y_k | s_{k-1}^n). \quad (2.12)$$

D'autre part,

$$\mathbb{E}_g[(\tilde{\omega}_k^n)^2] = (\tilde{\omega}_{k-1}^n)^2 \int \frac{p(y_k | s_k^n) p(s_k^n | s_{k-1}^n)}{g_k(s_k^n; s_{0:k-1}^n, y_{1:k})} ds_k^n. \quad (2.13)$$

Par conséquent :

$$\text{var}_g \tilde{\omega}_k^n = (\tilde{\omega}_{k-1}^n)^2 \left(\int \frac{p(y_k | s_k^n) p(s_k^n | s_{k-1}^n)}{g_k(s_k^n; s_{0:k-1}^n, y_{1:k})} ds_k^n - (p(y_k | s_{k-1}^n))^2 \right). \quad (2.14)$$

Si l'on choisit $g_k(s; s_{0:k-1}^n, y_{1:k}) = p(s | s_{k-1}^n, y_k)$, on obtient alors par la règle de Bayes :

$$p(s_k^n | y_k, s_{k-1}^n) = \frac{p(y_k | s_k^n, s_{k-1}^n) p(s_k^n | s_{k-1}^n)}{p(y_k | s_{k-1}^n)} = \frac{p(y_k | s_k^n) p(s_k^n | s_{k-1}^n)}{p(y_k | s_{k-1}^n)} \quad (2.15)$$

En insérant (2.15) dans (2.14), on obtient finalement :

$$\text{var}_{p(s|y_k, s_{k-1}^n)} \tilde{\omega}_k^n = 0 \quad (2.16)$$

On peut également obtenir ce résultat plus intuitivement: en utilisant (2.15), on obtient que les poids correspondants à ce choix valent :

$$\tilde{\omega}_k^n = \tilde{\omega}_{k-1}^n p(y_k | s_{k-1}^n). \quad (3.17)$$

Ainsi, pour une valeur donnée de s_{k-1}^n , le poids $\tilde{\omega}_k^n$ prend la même valeur quelque soit la particule tirée selon $g(s_k; y_{1:k}, s_{0:k-1}^n) = p(s_k | y_k, s_{k-1}^n)$. La variance des poids est donc nulle conditionnellement à $s_{0:k-1}^n$.

L'expression (2.17) est particulièrement intéressante dans la mesure où, étant indépendants de s_k^n , les poids peuvent être calculés en parallèle du calcul des nouvelles particules. Cependant, l'utilisation de la loi conditionnelle $p(s | y_k, s_{k-1}^n)$ nécessite, d'une part de savoir générer des échantillons selon cette loi, et d'autre part de savoir évaluer les poids des particules à une constante de normalisation près. Cette évaluation requiert celle de $p(y_k | s_{k-1}^n)$ par l'intermédiaire de l'intégrale $\int_{s_k} p(y_k | s_k) p(s_k | s_{k-1}^n) ds_k$. Dans le cas spécifique où le modèle de mesures est linéaire et où les bruits d'état et de mesure sont additifs et gaussiens, la fonction de proposition g_k est alors gaussienne, ce qui permet le calcul analytique de cette intégrale. Cependant, dans le cas général, on ne sait pas l'évaluer directement. Selon les caractéristiques du système dynamique, plusieurs stratégies ont, en pratique, été adoptées pour être en mesure de tirer des échantillons selon une approximation de cette fonction optimale. En voici les principales:

Les deux variantes suivantes consistent à émettre l'hypothèse approximative que la loi de proposition g est gaussienne et à en estimer les paramètres.

Filtre particulaire de Kalman étendu:

Cet algorithme, dénommé extended Kalman particle filter en anglais, consiste, pour chaque particule, à appliquer un filtre de Kalman étendu par linéarisation autour de la particule [57]. On approxime alors la fonction g_k conditionnellement à cette particule et aux mesures par la loi gaussienne de moyenne et de covariance égales à celles fournies par le filtre de Kalman étendu pour donner une meilleure approximation de g_k .

Filtre particulaire de Kalman unscented:

En remplaçant dans l'algorithme précédent le filtre de Kalman étendu par le filtre de Kalman unscented, on obtient le filtre particulaire de Kalman unscented [69]. Celui-ci est

censé bénéficier de la supériorité du filtre de Kalman unscented sur le filtre de Kalman étendu.

Utilisation de méthodes de Monte-Carlo:

On peut également utiliser un algorithme d'acceptation/rejet ou d'échantillonnage pondéré pour approcher la fonction de proposition. Ces méthodes étant coûteuses en temps de calcul, elles ne sont pas forcément les plus avantageuses.

Utilisation de variables auxiliaires:

Lorsque l'on souhaite tirer des échantillons selon une loi de mélange, l'introduction de variables auxiliaires représentant la composante du mélange est classique [36].

Cette idée a été couplée avec le principe d'échantillonnage pondéré séquentiel dans le filtre particulaire auxiliaire (auxiliary particle filter dénoté ASIR en anglais) proposé dans [66]. La variable auxiliaire n désigne un indice de particules. La première étape consiste alors à tirer R couples (s_k, n) selon la loi conditionnelle:

$$g_k(s_k, n, y_{1:k}) \propto f(y_k | \mu_k^n) p(s_k | s_{k-1}^n) \quad (2.18)$$

où μ_k^n est par exemple la moyenne ou tout autre caractéristique statistique de la loi conditionnelle $p(s_k | s_{k-1}^n)$. Ce tirage s'effectue en tirant tout d'abord un indice n selon $f(y_k | \mu_k^n)$ puis à propager selon la loi de propagation a priori la particule d'indice sélectionné. Les poids de telles particules sont ensuite mis à jour selon la formule:

$$\tilde{\omega}_k^n = \frac{p(y_k | s_k^n)}{f(y_k | \mu_k^{p^n})} \quad (2.19)$$

où l'indice p^n est l'indice de la particule ancêtre de s_k^n . Une fois un tel échantillon obtenu, on en tire N particules qui seront approximativement distribuées selon la loi *a posteriori*.

L'intervention d'indices auxiliaires permet ainsi de propager les particules dont la vraisemblance prédite est élevée. L'ASIR peut donner de meilleurs résultats que le SIR lorsque les pics de la loi de vraisemblance coïncide avec des zones où l'a priori est faible. En revanche, si les deux distributions coïncident, le SIR peut donner de meilleurs résultats dans la mesure où l'introduction des variables auxiliaires augmente inévitablement la covariance de l'échantillon.

Remarquons que l'utilisation de variables auxiliaires est un principe général qui peut être couplé à d'autres méthodes de Monte-Carlo. Par exemple, les travaux de [52] proposent un échantillonneur de Gibbs à la place de l'échantillonnage pondéré.

Ce principe expose cependant au risque que toutes les composantes du mélange ne soient pas représentées, i.e., que certaines particules soient éliminées lors de l'étape de proposition.

Nous avons donc passé en revue différents choix possibles pour la fonction de proposition dans l'échantillonnage séquentiel pondéré qui dépendent du système dynamique que l'on souhaite estimer. Nous exposons maintenant diverses techniques de rééchantillonnage en s'intéressant plus particulièrement à la variance introduite dans l'échantillon lors de cette étape.

2.1.3 Techniques de rééchantillonnage

Comme on l'a vu dans la section précédente, la suite des poids constituant une martingale, leur dégénérescence est inévitable du fait de la croissance de leur variance. De plus en plus de particules deviennent ainsi inefficaces. Introduit initialement dans [26] et [31] pour l'échantillonnage pondéré et dans [37] pour l'échantillonnage pondéré séquentiel, le rééchantillonnage permet de limiter cette dégénérescence. Il consiste à supprimer les particules de trop faible poids et à dupliquer celles de poids élevé. Soit N_k^n pour $n = 1, \dots, N$ le nombre de descendants affectés à chaque particule \tilde{s}_k^n lors de ce rééchantillonnage. Plusieurs techniques de simulation peuvent être mises en pratique. Les suivantes assurent qu'en moyenne le nombre de descendants d'une particule est proportionnel à son poids, i.e., $\mathbb{E}[N_k^n] = N\omega_k^n$. Elles sont tels que les poids sont tous égaux à $1/N$ après rééchantillonnage. En revanche, la variance de N_k^n diffère d'une méthode à l'autre.

Rééchantillonnage multinomial classique:

Dans [44], les entiers N_k^n sont obtenus par tirage selon la loi multinomiale de paramètres N et $(\omega_k^n)_{k=1, \dots, N}$. La variance est alors $\text{var}(N_k^n) = N\omega_k^n(1 - \omega_k^n)$. Cela revient à tirer avec remise des réalisations $(s_k^n)_{k=1, \dots, N}$ selon la loi discrète $(\tilde{s}_k^n)_{k=1, \dots, N}$ pondérée par les poids $(\omega_k^n)_{k=1, \dots, N}$ [37], [83].

Rééchantillonnage résiduel:

De variance moindre, le rééchantillonnage résiduel est partiellement déterministe [58]:

- Dans un premier temps, $N_k^{n,a} = \lfloor N\omega_k^n \rfloor$ descendants sont attribués à la particule \tilde{s}_k^n
- Afin de conserver un nombre de particules constant, les $N - \sum_{k=1}^N N_k^{n,a}$

particules restantes sont dans un second temps tirées proportionnellement à :

$$\frac{N\omega_k^n - N_k^{n,a}}{N - \sum_{n=1}^N N_k^{n,a}}.$$

La variance d'une telle procédure est alors :

$$\text{var}N_k^n = \left(N - \sum_{n=1}^N N_k^{n,a} \right) \frac{N\omega_k^n - N_k^{n,a}}{N - \sum_{n=1}^N N_k^{n,a}} \left(1 - \frac{N\omega_k^n - N_k^{n,a}}{N - \sum_{n=1}^N N_k^{n,a}} \right)$$

inférieure à la variance de la méthode précédente.

Méthodes de variance minimale:

Ces méthodes assurent que les entiers N_k^n ne diffèrent pas de leur valeur moyenne de plus de un. Il en est ainsi du rééchantillonnage stratifié préconisé en annexe dans [49] et dans [65]. Cette technique consiste à tirer N points espacés de $1/N$ de l'intervalle $[0, 1]$. Puis, en assimilant les poids à des intervalles contigus de $[0, 1]$, les quantités N_k^n sont choisies égales au nombre de ces points contenus dans l'intervalle $[\sum_{n=1}^{l-1} \omega_k^l, \sum_{n=1}^l \omega_k^l]$.

Un algorithme de variance équivalente peut être obtenu par un mécanisme de branchement fondé sur l'utilisation d'un arbre binaire, *tree-based branching mechanism* en anglais [62], [83]. Cette dernière référence expose d'autres méthodes de branchement utilisables pour le rééchantillonnage sans forcément assurer un nombre constant de particules au cours du temps.

Si l'étape de rééchantillonnage permet de "rafraîchir" le nuage de particules, elle introduit des erreurs dès que la méthode est stochastique. Elle peut également appauvrir le nuage si elle est pratiquée à outrance. L'idéal est alors de ne rééchantillonner le nuage que lorsque nécessaire, afin de permettre aux particules d'explorer l'espace d'état sur plusieurs pas de temps avant d'être éventuellement supprimées. Pour cela, le critère heuristique de la taille efficace du nuage, introduit dans le premier chapitre et noté N_{eff} , est préconisé dans [57],[44]. Rappelons que l'approximation obtenue pour N_{eff} en annexe, équation (A.16), est la suivante :

$$N_{eff} = \frac{N}{1 + \text{var}_g(\tilde{\omega}(x))}. \quad (2.20)$$

Pour N élevé, l'approximation de $\text{var}_g(\tilde{\omega}(x))$ à l'aide du coefficient de variation des poids $C(\omega_k)$ est proposée dans [44] :

$$C^2(\omega_k) \triangleq \frac{1}{N} \sum_{n=1}^N (N\omega_k^n - 1)^2 = N \sum_{n=1}^N \omega_k^{n^2} - 1 \quad (2.21)$$

Cette approximation repose sur la convergence suivante:

$$\frac{\sum_{n=1}^N \tilde{\omega}_k^n}{N} \xrightarrow[N \rightarrow \infty]{L} 1 \quad (2.22)$$

On obtient alors :

$$N_{eff} = \frac{1}{\sum_{n=1}^N (\omega_k^n)^2}. \quad (2.23)$$

Intuitivement, si un échantillon de taille N comprend M particules de poids $1/M$ et $N - M$ particules de poids nul, alors le nombre de particules efficaces est égal à M . Or, le coefficient de variation des poids vaut dans ce cas :

$$C^2(\omega) = \frac{N}{M} - 1 \quad (2.24)$$

On a donc bien dans ce cas la relation :

$$N_{eff} = M = \frac{N}{1 + C^2(\omega)} \quad (2.25)$$

Les poids ω_k^n étant normalisés, ils appartiennent à l'hyperplan de $[0, 1]^N$ défini par $\sum_{n=1}^N \omega_k^n = 1$. La quantité $\sum_{n=1}^N (\omega_k^n)^2$ représente donc la distance d'un point de l'hyperplan au centre du repère. Elle est donc minimale et vaut alors $1/N$ pour le projeté orthogonal du centre du repère sur l'hyperplan, c'est-à-dire pour le point tel que $\omega_k^n = \omega_k^{n'} \forall n, n' \in 1, \dots, N$. La quantité N_{eff} est donc majorée par N .

L'échantillonnage adaptatif consiste alors à ne rééchantillonner les particules que lorsque N_{eff} est inférieur à un certain seuil fixé N_{seuil} , l'algorithme du filtrage particulaire avec rééchantillonnage adaptative est montré dans la figure 2.5, le nombre de particules efficaces est défini pour l'estimation de $\mathbb{E}_p[\phi(X_k)]$, où ϕ est une fonction de X_k . Le principal avantage de l'approximation dans l'annexe, équation (A.16), et reprise en (2.20) est qu'elle est indépendante de la fonction ϕ . Cependant, comme il est remarqué dans [60], [65] cette approximation peut facilement être mise en défaut.

Algorithme 2.5 : Filtre particulaire avec rééchantillonnage adaptatif

- Initialisation :
 - o Pour $n = 1, \dots, N$, générer s_0^n selon $L_0(x)$ et poser $\omega_0^n = 1/N$
- Pour $k = 1, \dots, \text{fin}$:
- o Pour $n = 1, \dots, N$, générer \tilde{s}_k^n selon $f(x_k | x_{k-1} = s_{k-1}^n, y_k)$;
- o Mise à jour des poids :

- Pour $n = 1, \dots, N$, calculer les poids non normalisés :

$$\tilde{\omega}_k^n = \omega_{k-1}^n \frac{p(\tilde{s}_k^n | s_{k-1}^n) l_k(y_k; \tilde{s}_k^n)}{g_k(\tilde{s}_k^n | s_{k-1}^n, y_k)}$$

- Pour $n = 1, \dots, N$, normaliser les poids : $\omega_k^n = \frac{\tilde{\omega}_k^n}{\sum_{n=1}^N \tilde{\omega}_k^n}$.

- retourner $\mathbb{E}[\widehat{\phi(x_k)}] = \sum_{n=1}^N \omega_k^n \phi(\tilde{s}_k^n)$

- Calculer $N_{eff} = 1/\sum_{n=1}^N (\omega_k^n)^2$

- o Rééchantillonnage :

➤ Si $N_{eff} < N_{seuil}$ alors

- Pour $n = 1, \dots, N$, tirer avec remise s_k^n parmi $(\tilde{s}_k^n)_{n=1, \dots, N}$ proportionnellement aux poids $(\omega_k^n)_{n=1, \dots, N}$ puis donner aux poids ω_k^n la valeur $1/N$.
- Sortie de l'algorithme: le nuage de particules issues de leur poids

$$S_k = (s_k^n, 1/N)_{n=1, \dots, N}$$

permet d'approcher la loi a posteriori par :

$$p(x_k | y_k) \approx \frac{1}{N} \sum_{n=1}^N \delta_{s_k^n}(x_k)$$

et pour toute fonction ϕ intégrable par rapport à $p(x_k | y_k)$:

$$\mathbb{E}[\widehat{\phi(x_k)}] = \frac{1}{N} \sum_{n=1}^N \phi(s_k^n)$$

➤ Si non $s_k^n = \tilde{s}_k^n$ et $\mathbb{E}[\widehat{\phi(x_k)}] = \sum_{n=1}^N \omega_k^n \phi(s_k^n)$

2.2 Application à la poursuite mono-cible par mesure d'angle seul

Pour illustrer l'algorithme de filtrage particulaire, nous allons étudier le problème classique d'estimation de la position d'une cible aérienne par mesure d'angle seul [87].

2.2.1 Modélisation mathématique

Dans le contrôle du trafic aérien, dénommé air traffic control et dénoté ATC en anglais, la cible aérienne a deux types de mouvements ou deux modes de vols :

- o Mouvement uniforme : pour un vol rectiligne à vitesse quasi-constante ;
- o Mouvement curviligne (Manœuvre) : lorsque la cible (avion) tourne, monte ou faire l'atterrissage.

Les modèles de mouvement horizontal et vertical de la cible aérienne peuvent typiquement être découplés [75]. Dans la littérature, il existe beaucoup d'estimateurs qui ont discuté les mouvements de cette cible dans le plan horizontal. Les modes de vol de la cible dans le plan horizontal peuvent être modélisés par :

- o Un modèle de mouvement rectiligne à vitesse quasi-constante pour un mouvement uniforme
- o Un modèle manœuvrant qui peut être implémenté comme un mouvement curviligne.

Le modèle de mouvement rectiligne à vitesse quasi-constante est donné par :

$$X_{k+1} = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix} X_k + \begin{pmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{pmatrix} V_k \quad (2.26)$$

Avec T l'intervalle d'échantillonnage ; X est le vecteur d'état de la cible aérienne, défini comme :

$$X = [\xi \quad \dot{\xi} \quad \eta \quad \dot{\eta}]' \quad (2.27)$$

Avec ξ et η qui désignent les coordonnées cartésiennes du plan horizontal et V_k est un bruit blanc gaussien centré de covariance $Q = \begin{pmatrix} \sigma_X^2 & 0 \\ 0 & \sigma_Y^2 \end{pmatrix}$, utilisé pour modéliser des petites accélérations de turbulence et de changement de vent.

Le manœuvre de la cible aérienne généralement, suit un modèle constant qui est le mouvement curviligne, caractérisé par des vitesses angulaire et rectiligne constantes.

En augmentant le vecteur d'état de l'équation (2.27) par un autre paramètre qui est la vitesse angulaire Ω , on obtient le vecteur d'état augmenté suivant :

$$X = [\xi \quad \dot{\xi} \quad \eta \quad \dot{\eta} \quad \Omega]' \quad (2.28)$$

Le modèle de mouvement curviligne or « the coordinated turn model » est donc, donné par :

$$X_{k+1} = \begin{pmatrix} 1 & \frac{\sin\Omega_k T}{\Omega_k} & 0 & -\frac{1 - \cos\Omega_k T}{\Omega_k} & 0 \\ 0 & \cos\Omega_k T & 0 & -\sin\Omega_k T & 0 \\ 0 & \frac{1 - \cos\Omega_k T}{\Omega_k} & 1 & \frac{\sin\Omega_k T}{\Omega_k} & 0 \\ 0 & \sin\Omega_k T & 0 & \cos\Omega_k T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} X_k + \begin{pmatrix} 0.5T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & 0.5T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{pmatrix} V_k \quad (2.29)$$

En tenant compte que les bruits V_k de l'équation (2.26) et de l'équation (2.29) ont des dimensions différentes.

Les observations sont disponibles à chaque pas de temps k selon l'équation de mesure d'angle suivante :

$$Y_k = \tan^{-1} \left(\frac{Y = \eta}{X = \xi} \right) + W_k \quad (2.30)$$

Avec W_k un bruit gaussien centré de covariance σ_W^2 indépendant de V_k , (X, Y) est la position de la cible aérienne dans le plan horizontal.

Dans cette section, nous faisons quelques simulations pour évaluer l'algorithme du filtrage particulière, nous considérons que la cible est un point dans le plan (X, Y) , pour cela deux modèles cinématiques sont utilisés afin de traquer cette cible.

2.2.2 Divers scénarios

Scénario 1 : mouvement rectiligne à vitesse quasi-constante

Pour le mouvement rectiligne, la position initiale de la cible est :

$$X_0 = \begin{pmatrix} 1500 \\ 10 \\ 1500 \\ 5 \end{pmatrix}$$

La covariance de bruit du modèle dynamique a été fixée à $\sigma_X = \sigma_Y = 0.05 \text{ ms}^{-2}$, les mesures d'angles sont simulées avec un bruit de mesure gaussien de variance $\sigma_W = 0.02$ radians, soit environ 1.5 degrés, toutes les 5 secondes. Les mesures d'angles de la cible sont présentées sur la figure 2.1.

Dans tous les scénarios, nous avons mis en œuvre des filtres séquentiels de type bootstrap, c'est-à-dire que nous avons choisis une fonction de proposition égale à la loi d'évolution,

avec un rééchantillonnage adaptatif, le nombre de particules utilisé est $N = 500$ et le seuil de rééchantillonnage a été fixé à $0.8N$ dans les 100 réalisations.

L'initialisation du filtre a été faite selon une loi gaussienne dont sa moyenne et sa covariance sont :

$$X_0 = \begin{pmatrix} 1600 \\ 11 \\ 1200 \\ 3 \end{pmatrix}$$

$$\text{et } X_{cov} = \begin{pmatrix} 2.10^{+2} & 0 & 0 & 0 \\ 0 & 10^{-3} & 0 & 0 \\ 0 & 0 & 2.10^{+2} & 0 \\ 0 & 0 & 0 & 10^{-3} \end{pmatrix}$$

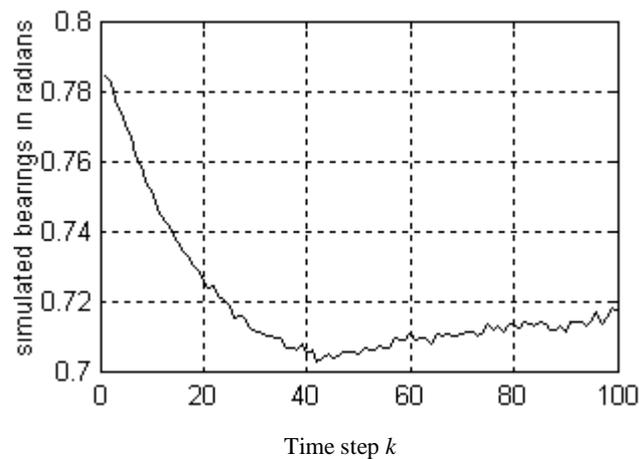


Fig. 2.1: Mesures d'angles simulées du modèle rectiligne

Les résultats obtenus par l'application du filtre particulaire à rééchantillonnage adaptatif sont représentés dans les figures de 2.2 à 2.4. Nous avons pris comme mesure de performance de cet algorithme, la Racine de l'Erreur Quadratique Moyenne, dénommé Root Mean Square Error et dénoté RMSE en anglais, donnée par cette équation :

$$RMSE(X_j) = \sqrt{\frac{\sum_{k=1}^j (X_{réelle}(k) - X_{estimée}(k))^2}{j}} \quad \text{pour : } j = 1, \dots, fin$$

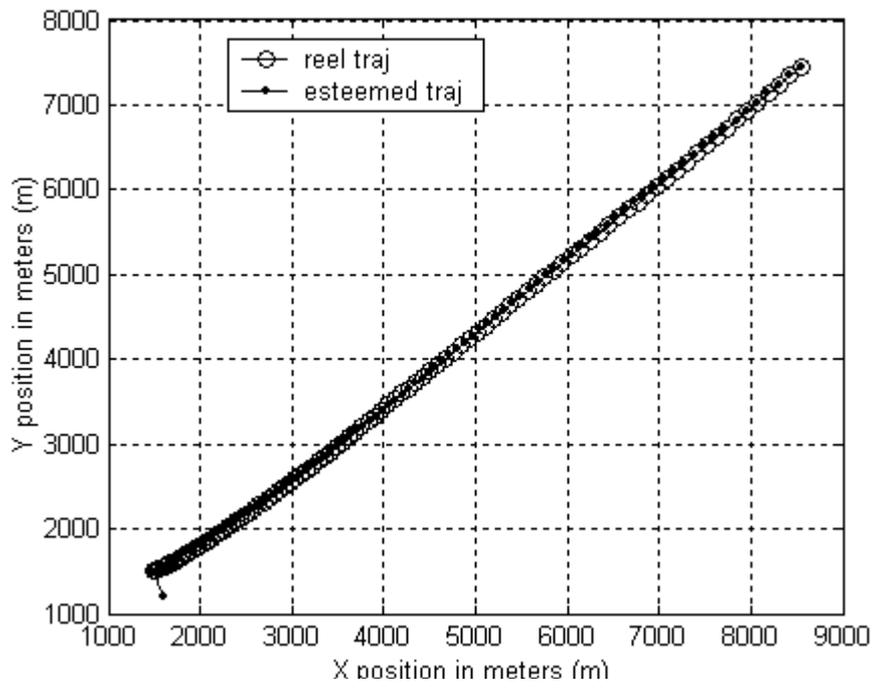


Fig. 2.2 : Trajectoire réelle et estimée

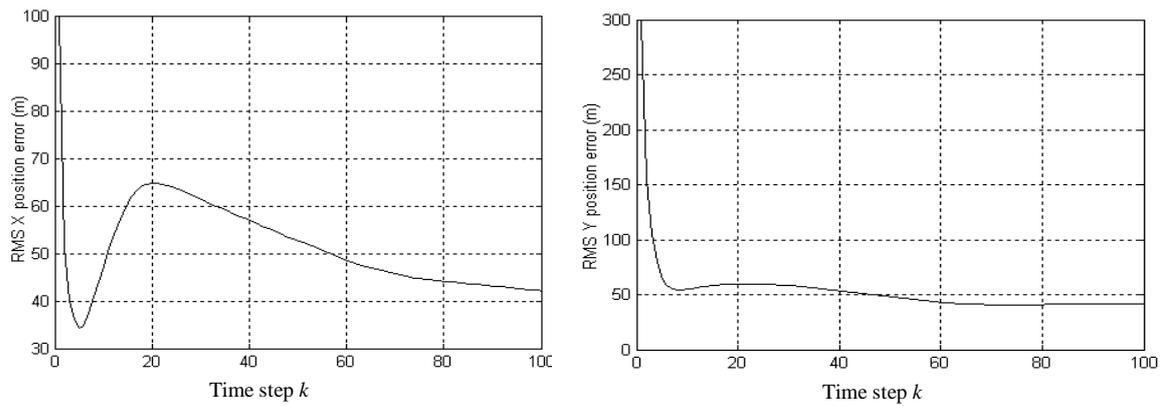


Fig. 2.3 : RMSE de position (X et Y)

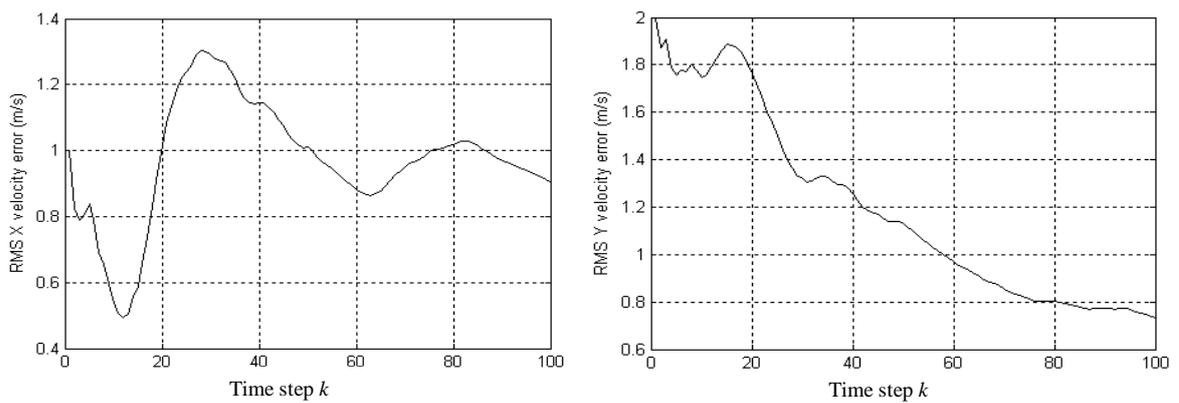


Fig. 2.4 : RMSE de vitesse dans le plan horizontal (X, Y)

Scenario 2: mouvement curviligne où coordinated turn

La position initiale de la cible est :

$$X_0 = \begin{pmatrix} 2000 \\ 10 \\ 1500 \\ 5 \\ 0.00223 \end{pmatrix}$$

Le bruit dynamique de ce mouvement est le même que le premier, les mesures d'angles sont aussi simulées avec des bruits gaussien de déviation $\sigma_W = 0.02$ radians. Les mesures d'angles utilisées de cette cible sont présentées dans la figure 2.5.

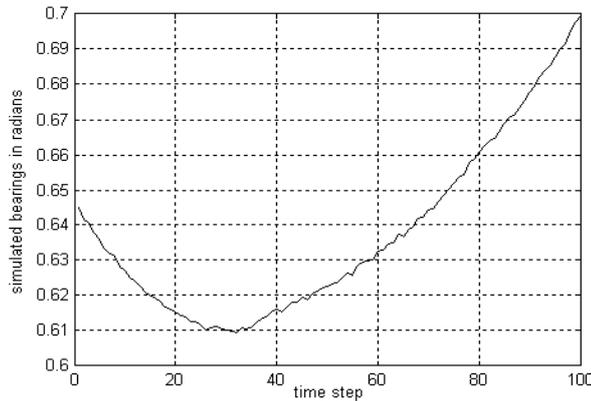


Fig. 2.5 : Mesures d'angles simulées du modèle curviligne

Le vecteur d'initialisation et la matrice de covariance du filtre sont :

$$X_0 = \begin{pmatrix} 2100 \\ 10 \\ 1400 \\ 5 \\ 0.0523 \end{pmatrix}, X_{cov} = \begin{pmatrix} 2.10^{+2} & 0 & 0 & 0 & 0 \\ 0 & 5.10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 2.10^{+2} & 0 & 0 \\ 0 & 0 & 0 & 5.10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 5.10^{-4} \end{pmatrix}$$

Les figures 2.2 et 2.6, démontrent que les trajectoires estimées et réelles sont superposées et presque identiques. Les figures 2.3, 2.4, 2.7, 2.8 et 2.9 présentent l'évolution dans le temps des racines des erreurs quadratiques moyennes (RMSE) de position et de vitesse des mouvements rectiligne et curviligne, il apparaît clairement que les différentes RMSE de position ou de vitesse pour les deux mouvements sont faibles, ce qui montre que l'algorithme a bien traqué la dynamique de la cible et donc, converge vers la vraie trajectoire. D'après ces RMSE faibles, nous pouvons dire que le traqueur à savoir l'algorithme du filtrage particulaire, traque ou dépiste précisément des cibles dont les modèles d'état et /ou de mesure sont non linéaires.

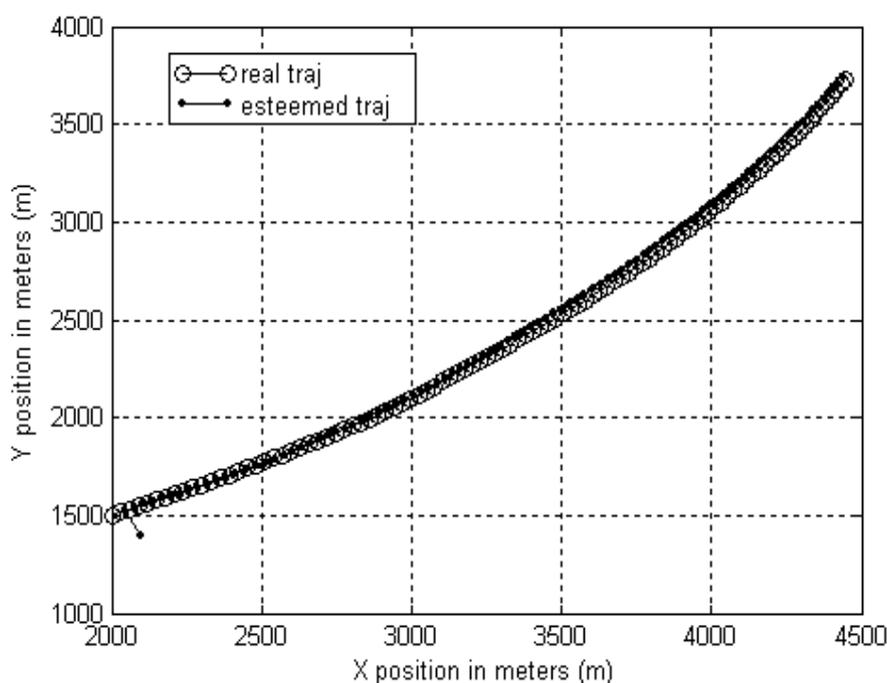


Fig. 2.6 : Trajectoire réelle et estimée

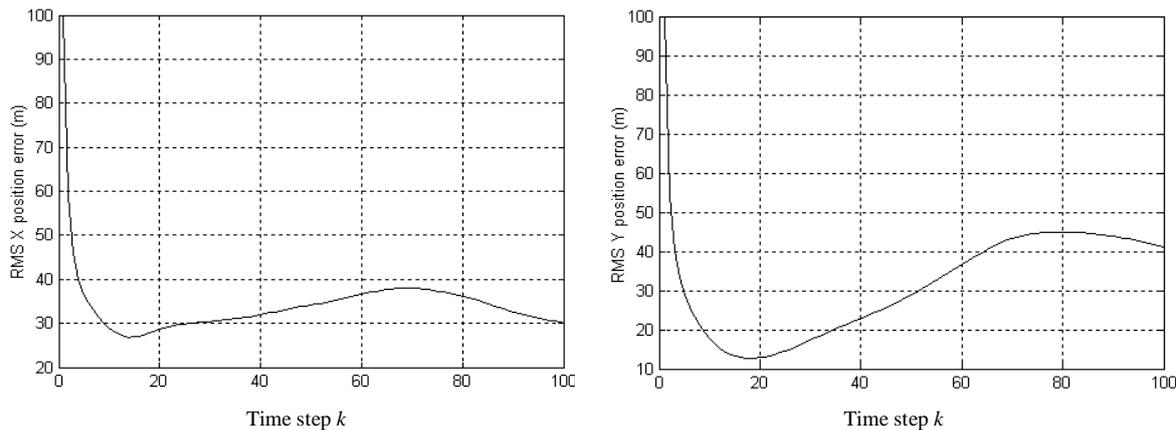


Fig. 2.7 : RMSE de position (X et Y)

En ce qui concerne le temps de calcul, lors de l'utilisation d'un filtre particulaire à rééchantillonnage adaptatif avec $N = 500$ particules pour 200 itérations est de 12 secondes, donc pour une seule itération, le temps est de 60 ms sur une machine de type Pentium IV de fréquence d'horloge égale à 3.40 GHz. On constate ainsi expérimentalement que l'étape de rééchantillonnage est très coûteuse et ce d'autant plus

que le nombre de particules est élevé ce qui donne un autre argument en faveur du rééchantillonnage adaptatif aux dépends du rééchantillonnage systématique.

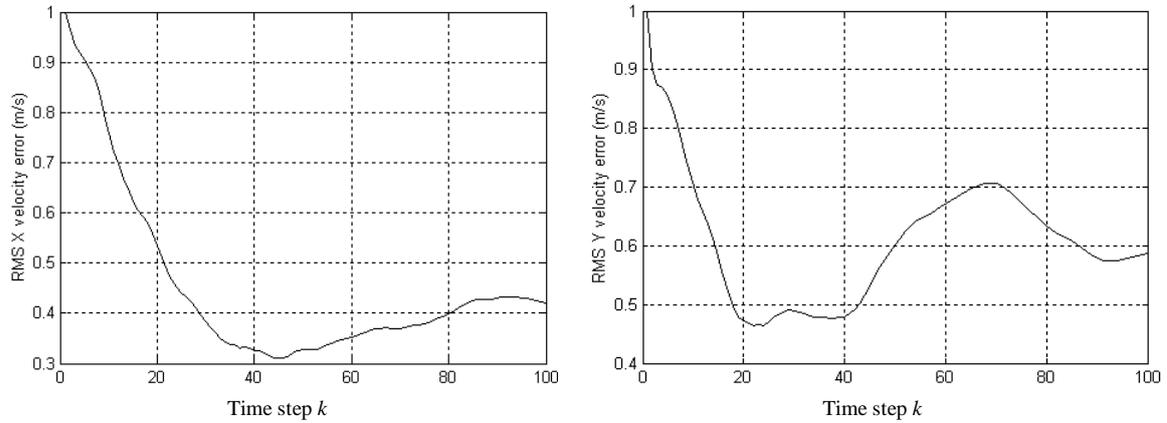


Fig. 2.8 : RMSE de vitesse dans le plan horizontal (X, Y)

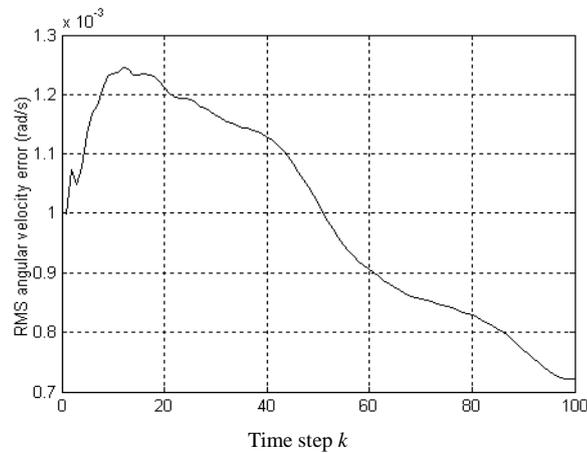


Fig. 2.9 : RMSE de vitesse angulaire Ω

Cette étude montre donc que dans la pratique, le rééchantillonnage systématique peut s'avérer dangereux dans la mesure où il conduit à un appauvrissement des états. Enfin l'économie du temps en ne procédant que si nécessaire au rééchantillonnage est non négligeable.

2.3 Conclusion

Nous nous sommes intéressés dans ce chapitre à l'estimation bayésienne du processus d'état markovien $(X_k)_{k \in \mathbb{N}}$ observé à travers le processus de mesures $(Y_k)_{k \in \mathbb{N}^*}$. Plus particulièrement, l'objectif était de fournir une estimation de la loi conditionnelle $L_{k|k} =$

$p(X_k | Y_1, \dots, Y_k)$. Les méthodes dites de Monte-Carlo ont permis de développer notamment dans le cadre du filtrage une classe de méthodes très performantes, dénommées méthodes séquentielles de Monte-Carlo ou encore filtres particulaires. A chaque instant, la loi $L_{k|k}$ est approximée par une somme pondérée de lois de Dirac centrées en des particules. Ces particules et leur poids évoluent au cours du temps selon un échantillonnage pondéré séquentiel et sont rééchantillonnées afin de n'en conserver que les meilleures au sens de la vraisemblance des observations. Devant la profusion d'études les concernant durant les dernières décennies, nous avons proposé un état de l'art présentant tout d'abord le filtre bootstrap qui est le filtre le plus utilisé puis une classe générique englobant la plupart de ses variantes. Plusieurs choix de fonctions de proposition servant à guider au mieux les particules dans les régions de forte vraisemblance ont été décrites ainsi que plusieurs choix de techniques de rééchantillonnage limitant l'augmentation inutile de la variance des poids des particules.

Nous avons enfin présenté une application du filtre particulaire que nous avons menée pour le suivi d'une cible unique observée à travers des mesures d'angle uniquement dont lesquels nous avons démontré deux scénarios. Nous avons en particulier montré l'avantage pratique du rééchantillonnage adaptatif retardant la dégénérescence des poids par rapport au rééchantillonnage systématique.

Chapitre 3

Approche modèles multiples

La discipline de la poursuite de cibles occupe une grande place dans la littérature, incorporant l'estimation d'état, les techniques d'estimations modèles multiples, les méthodes d'associations des données. Presque tous les systèmes de poursuite modernes utilisent le filtre de Kalman comme un outil central pour l'estimation d'état. Cependant, si la cible est manœuvrante, le succès du filtre de Kalman standard peut être limité. Les structures alternatives utilisant plusieurs filtres de Kalman en parallèle ont montrées un succès dans ce problème. Nous exposons dans ce chapitre deux approches modèles multiples ; la première considère que le modèle en vigueur ne change pas avec le temps, par contre la deuxième suppose que le modèle de la cible peut changer à n'importe quel instant.

3.1 Estimation à modèles multiples

Les techniques décrites dans le chapitre 1 sont extrêmement efficaces pour poursuivre les objets mobiles quand les suppositions de l'algorithme sont satisfaites. Cependant, une préention inhérente à ces méthodes est que le modèle dynamique de la cible est connu pour tout le temps.

Dans l'application de la poursuite de cibles, il y a deux classes fondamentales de modèles: manœuvrant et non manœuvrant [64],[75],[84]. Les modèles non manœuvrants sont employés pour exploiter le fait que la plupart des avions volent le long de chemins droits (trajectoires rectilignes) plus du temps; une telle connaissance introduit la certitude intrinsèque dans le problème d'estimation, qui peut être employée pour réduire la largeur de bande du filtre de poursuite et pour augmenter considérablement la précision des états estimés.

Les modèles manœuvrants sont nécessaires dans un système d'estimation pour décrire le mouvement de la cible quand elle n'est pas « non manœuvrante ». La variété des

manœuvres d'une cible peut (en particulier dans un arrangement militaire) élever le niveau d'incertitude dans le système.

Fondamentalement, les traqueurs de cibles peuvent utiliser deux stratégies pour s'adapter aux changements des modèles dynamiques. La première approche [75] est d'utiliser la mesure pour estimer les paramètres inconnus des manœuvres (des méthodes employant souvent des fenêtres de type glissant) et corriger alors les états estimés en utilisant ces paramètres. L'inconvénient de cette méthode est qu'elle est lente pour s'adapter au changement; elle prend plusieurs périodes d'échantillonnages après le début d'une manœuvre pour pouvoir estimer les paramètres des manœuvres avec n'importe quel niveau d'exactitude.

La deuxième approche pour s'adapter aux changements de la dynamique de cible est d'utiliser une banque parallèle d'estimateurs, chacun accordé à une condition de fonctionnement différente, et de combiner les sorties dans une estimation moyenne pondérée basée sur l'exécution apparente de chaque filtre élémentaire. Cette dernière architecture a un avantage important par rapport à l'ancienne concernant la vitesse d'adaptation. Pratiquement toutes les techniques à modèles multiples, partagent cette même architecture de base et diffèrent seulement de la façon avec laquelle les poids des modèles sont calculés et dans le mélange des estimations des modèles conditionnés entre les cycles de traitement.

3.1.1 Approche Modèles Multiples (cas de non changement de modèle)

La forme de base de l'estimateur adaptatif modèle multiple (EAMM) [79] est obtenue quand on suppose que le modèle en vigueur ne change pas dans le temps.

3.1.1.1 Calcul des probabilités des modèles.

L'événement M_j (mode j) est défini pour représenter la condition que le modèle j est en vigueur. Aucun argument du temps n'est exigé, car on assume que le modèle ne change pas dans le temps. La probabilité a posteriori que le modèle j étant correcte, donnant l'histoire de mesures jusqu'à l'instant k est représentée par :

$$\mu_j(k) \triangleq P(M_j | Z^k) \quad (3.1)$$

En développant Z^k de l'équation (3.1) en fonction de l'histoire de mesures précédentes Z^{k-1} combinée avec la mesure courante $z(k)$ et en utilisant la règle de Bayes à $z(k)$ et M_j :

$$\begin{aligned}
\mu_j(k) &= P(M_j | Z^{k-1}, z(k)) \\
&= \frac{p(M_j, z(k) | Z^{k-1})}{p(z(k) | Z^{k-1})} \\
&= \frac{p(z(k) | M_j, Z^{k-1}) P(M_j | Z^{k-1})}{p(z(k) | Z^{k-1})} \tag{3.2}
\end{aligned}$$

Où la notation $P(\cdot)$ se rapporte à la probabilité d'un événement discret, tandis que $p(\cdot)$ se rapporte à la fonction de densité de probabilité d'une variable continue. Le dénominateur dans (3.2) peut être augmenté en utilisant le développement de probabilité totale au-dessus de tous les modèles:

$$p(z(k) | Z^{k-1}) = \sum_{i=1}^r p(z(k) | M_i, Z^{k-1}) P(M_i | Z^{k-1}) \tag{3.3}$$

Où r est le nombre de modèles présumés (et ainsi le nombre de filtres élémentaires dans la structure). Ceci donne l'équation récursive suivante pour les probabilités des modèles $\mu_j(k)$:

$$\mu_j(k) = \frac{p(z(k) | M_j, Z^{k-1}) \mu_j(k-1)}{\sum_{i=1}^r p(z(k) | M_i, Z^{k-1}) \mu_i(k-1)} \tag{3.4}$$

La représentation de l'espace des paramètres par un certain nombre de modèles discrets est un emploi efficace du théorème de probabilité totale, représentant la pdf de la nouvelle mesure comme une somme pondérée au-dessus de chaque modèle possible.

Le calcul de $p(z(k) | M_j, Z^{k-1})$ est un problème facile pour le cas de modèles fixes (non switching); cette fonction de densité représente la relation entre la mesure entrante (courante) et l'histoire des mesures précédentes. En considérant le modèle de mesure du filtre de Kalman linéaire standard, ceci est évalué comme:

$$p(z(k) | M_j, Z^{k-1}) = N\{z(k); H_j \hat{x}_j(k|k-1), H_j P_j(k|k-1) H_j^T + R_j\} \tag{3.5}$$

Où $\hat{x}_j(k|k-1)$ et $P_j(k|k-1)$ sont l'état et la covariance estimés du filtre du modèle j et H_j et R_j sont respectivement la matrice et la covariance de mesure du modèle j .

3.1.1.2 Calcul des estimations combinées

Les sorties de chaque filtre sont l'état estimé du modèle conditionné \hat{x}_j , la covariance associée P_j et la fonction de vraisemblance du modèle Λ_j . Après que les filtres soient initialisés, ils fonctionnent récursivement sur leurs propres estimés. Leurs fonctions de

vraisemblances sont employées pour faire la mise à jour des probabilités des modèles ; ces probabilités sont utilisées pour combiner les états estimés des modèles conditionnés et leurs covariances.

Sous les suppositions ci-dessus la pdf de l'état du système est une mixture gaussienne [32] avec r termes.

$$p(x(k)|Z^k) = \sum_{j=1}^r \mu_j(k) N\{x(k); \hat{x}_j(k|k), P_j(k|k)\} \quad (3.6)$$

où

$$N\{.\} = (2\pi \cdot \det(P_j(k)))^{-0.5} \exp\left(-0.5[x(k) - \hat{x}_j(k)]^T P_j^{-1}(k)[x(k) - \hat{x}_j(k)]\right) \quad (3.7)$$

L'estimation de la moyenne conditionnelle est formée comme une moyenne pondérée des estimés des filtres élémentaires en utilisant les probabilités des modèles $\mu_j(k)$ comme poids:

$$\hat{x}(k|k) = \sum_{j=1}^r \mu_j(k) \hat{x}_j(k|k) \quad (3.8)$$

La covariance de cette estimation peut également être formée en utilisant une moyenne pondérée, mais en ajoutant le terme de correction qui tient compte de la propagation présentée par les différentes estimations:

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \left(P_j(k|k) + [\hat{x}_j(k|k) - \hat{x}(k|k)][\hat{x}_j(k|k) - \hat{x}(k|k)]^T \right) \quad (3.9)$$

La figure 3.1 montre la structure de l'algorithme « modèles multiples » pour des modèles fixes (non switching). Les estimés des modèles conditionnés par chaque filtre élémentaire à chaque cycle de traitement sont passés directement dans le même filtre au cycle de traitement suivant, car on suppose que le modèle en vigueur ne change pas dans le temps. L'estimation globale combinée est calculée comme moyenne pondérée, comme il est montré dans (3.8) et (3.9).

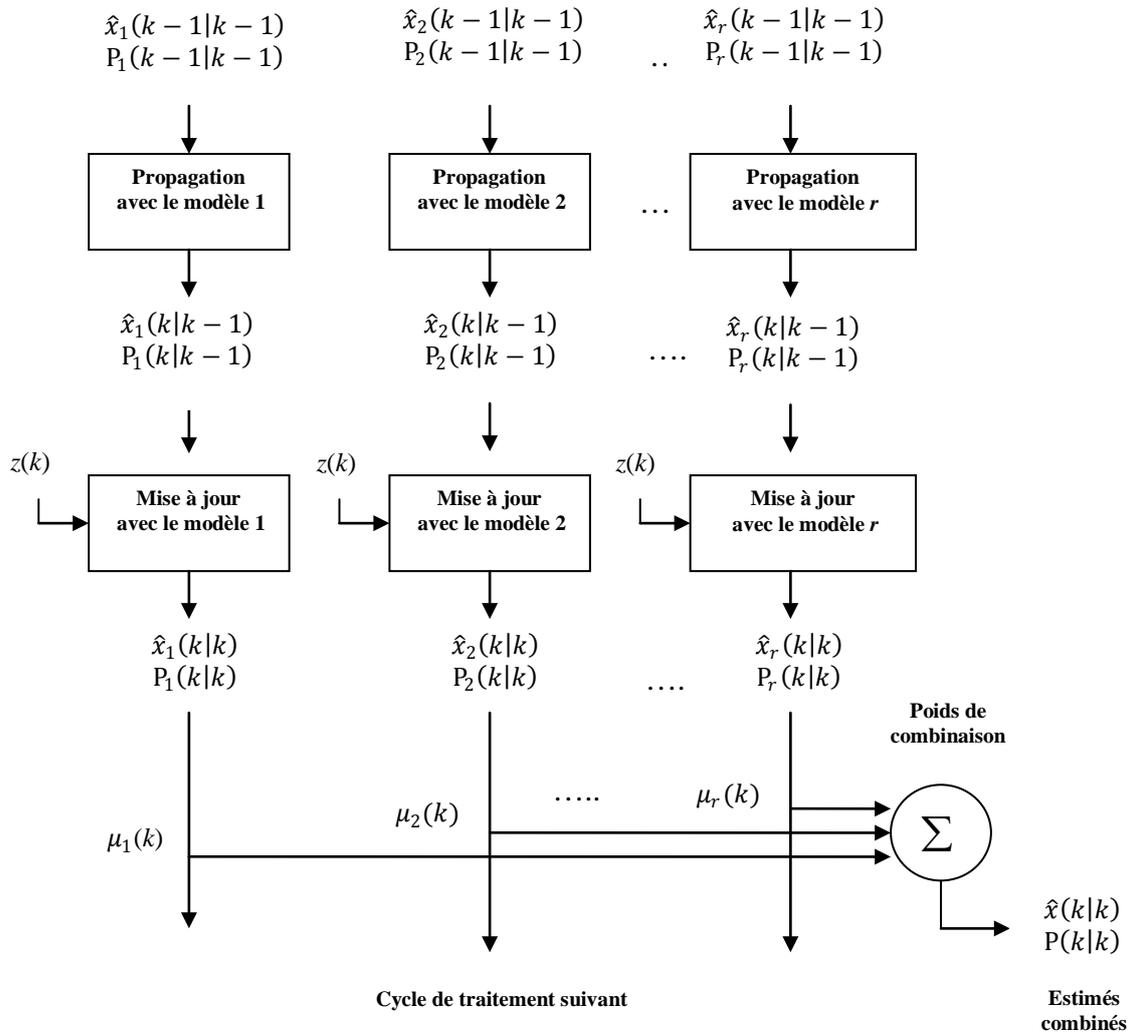


Fig. 3.1 : Schéma fonctionnel de l'algorithme d'estimation à modèles multiples (cas de non changement de modèles)

3.1.2 Approche Modèles Multiples (cas de changement de modèles)

Pour tenir compte des changements du modèle, le modèle en vigueur est autorisé de changer à n'importe quel instant d'échantillonnage et des événements de l'histoire du modèle sont employés pour caractériser le comportement transitoire du système avec le temps [64]. De tels événements prennent la forme :

$$M^{k,l} = \{M_{i_{1,l}}, M_{i_{2,l}}, \dots, M_{i_{k,l}}\} \quad l = 1, \dots, r^k \tag{3.10}$$

$$\text{et } 1 \leq i_{q,l} \leq r \quad q = 1, \dots, k$$

La notation est interprétée comme signification que cette histoire des l modèles possibles à l'instant k se compose de modèles $i_{1,l}$ à l'instant 1, de modèles $i_{2,l}$ à l'instant 2, ...etc où chaque $i_{q,l}$ est l'index d'un nombre de modèle entre 1 et r .

Si des transitions sont permises pour chaque r modèles à n'importe quel instant, alors chaque événement de l'histoire du modèle à l'instant k provoquera r nouveaux événements à l'instant $k + 1$, par conséquent le nombre d'histoires des modèles possibles augmente exponentiellement avec le temps selon r^k .

Par exemple, avec $r = 2$ à l'instant $k = 2$ on a $r^k = 4$ séquences possibles (histoires) :

l	$i_{1,l}$	$i_{2,l}$
1	1	1
2	1	2
3	2	1
4	2	2

Il sera supposé que le processus de saut du modèle est un processus de Markov (chaîne de Markov) avec des probabilités de transitions des modèles connues :

$$\pi_{ij} \triangleq P(M(k) = M_j | M(k-1) = M_i) \quad (3.11)$$

Ces probabilités de transitions des modèles seront supposées invariantes dans le temps et indépendantes de l'état de base ; en d'autres termes, une chaîne de Markov homogène. La pdf de l'état conditionné de la cible sur les mesures doit alors être calculée comme développement de la probabilité totale au-dessus de tous les événements de l'histoire du modèle :

$$p(x(k)|Z^k) = \sum_{l=1}^{r^k} p(x(k)|M^{k,l}, Z^k) P(M^{k,l}|Z^k) \quad (3.12)$$

La probabilité de l'histoire du modèle $P(M^{k,l}|Z^k)$ est développée comme suit :

$$\begin{aligned} P(M^{k,l}|Z^k) &= P(M^{k,l}|z(k), Z^{k-1}) \\ &= \frac{p(M^{k,l}|z(k), Z^{k-1})}{p(z(k)|Z^{k-1})} \\ &= \frac{p(z(k)|M^{k,l}, Z^{k-1})P(M^{k,l}|Z^{k-1})}{p(z(k)|Z^{k-1})} \end{aligned}$$

$$\begin{aligned}
&= \frac{p(z(k)|M^{k,l}, Z^{k-1})P(M_{k,j}, M^{k-1,l'} | Z^{k-1})}{p(z(k)|Z^{k-1})} \\
&= \frac{p(z(k)|M^{k,l}, Z^{k-1})P(M_{k,j} | M^{k-1,l'}, Z^{k-1})P(M^{k-1,l'} | Z^{k-1})}{p(z(k)|Z^{k-1})} \quad (3.13)
\end{aligned}$$

Où l est l'index de l'histoire du modèle courant (entre 1 et r^k), l' est l'index de l'histoire du modèle précédent (entre 1 et r^{k-1}) et j est l'index du modèle courant (entre 1 et r) présumé par l'évènement de l'histoire du modèle $M^{k,l}$. Le dénominateur est développé comme l'expansion de la probabilité totale au-dessus de tous les évènements de l'histoire du modèle comme dans (3.12).

La méthode généralement employée pour évaluer les probabilités de l'évènement de l'histoire du modèle est de supposer que le processus de transition du modèle est un processus de Markov [75], où la probabilité de la transition dépend seulement du nombre de modèles précédents $i_{k-1,l'}$ et pas sur l'histoire du modèle antérieure ou les mesures antérieures:

$$\begin{aligned}
P(M_{k,j} | M^{k-1,l'}, Z^{k-1}) &= P(M_{k,j} | M^{k-1,l'}) \\
&= P(M_{k,j} | M_{i_{1,l'}}, M_{i_{2,l'}}, \dots, M_{i_{k-1,l'}}) \\
&= P(M_{k,j} | M_{i_{k-1,l'}}) \\
&\triangleq \pi_{i_{k-1,l'}j}(k) \quad (3.14)
\end{aligned}$$

où $\pi_{i_{k-1,l'}j}(k)$ est la probabilité de transition du modèle index $i_{k-1,l'}$ à l'instant $k-1$ vers le modèle index j à l'instant k , dont chaque index est un nombre de modèles entre 1 et r . Notons que l'hypothèse de (3.14) fournit un mécanisme pour le calcul des probabilités de la séquence de modèle et le conditionnement dans (3.13) de probabilité de la nouvelle mesure sur la séquence du modèle produit toujours un nombre exponentiellement croissant des hypothèses avec le temps.

La structure de l'estimateur bayésien de plein ordre (cas avec changement de modèles) est montrée sur la figure 3.2. Le diagramme démontre l'augmentation du nombre des filtres qui est exigé: la sortie de chaque filtre au cycle de traitement courant doit être traitée dans le cycle de traitement suivant en utilisant chaque modèle, par conséquent le nombre d'opérations de filtrage au $k^{\text{ième}}$ cycle est r^k .

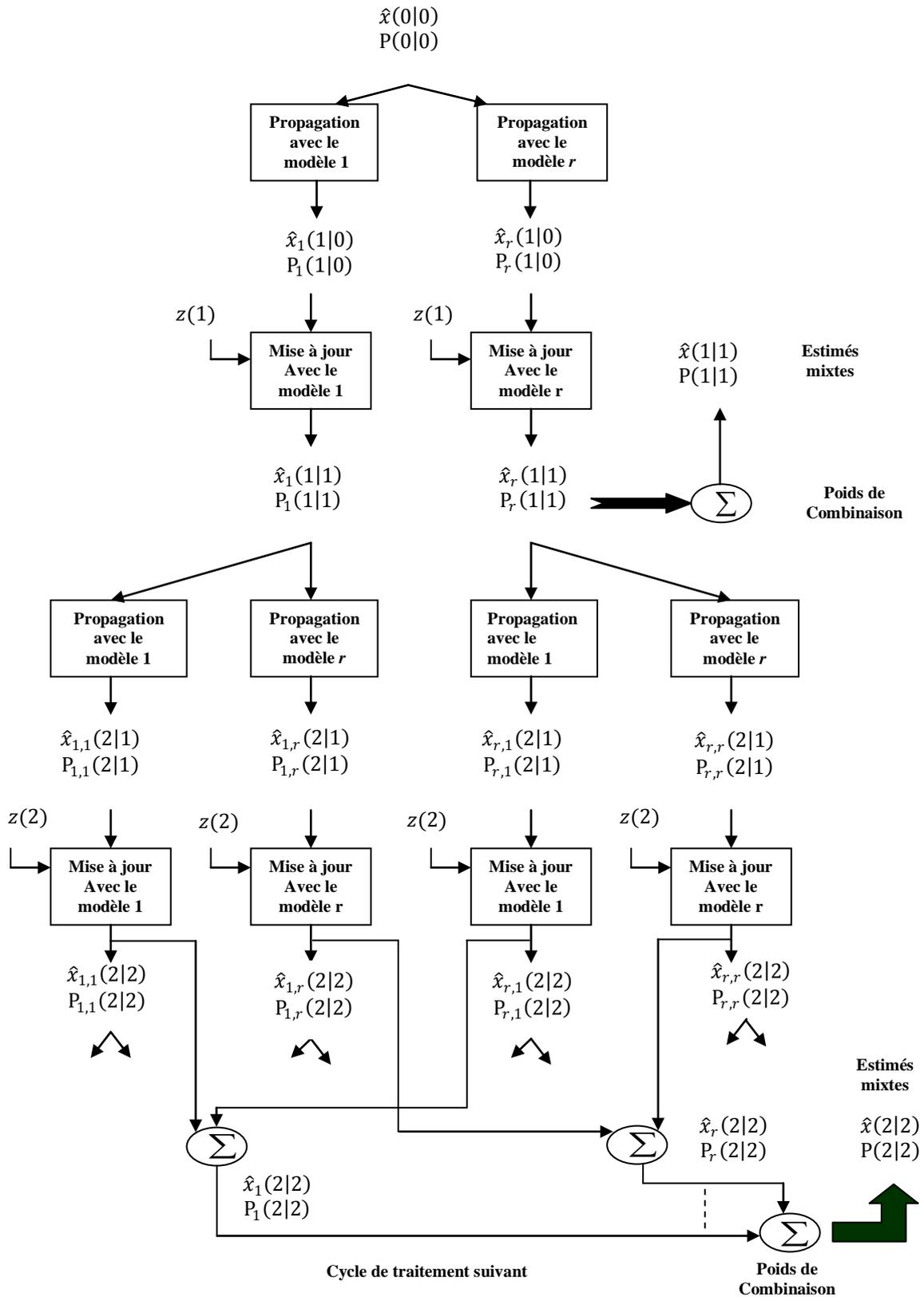


Fig. 3.2 : Schéma de l'estimateur bayésien de plein ordre (cas de changement de modèles)

Les algorithmes les plus utilisés sont décrits dans les sections suivantes.

3.1.2.1 Estimateur pseudo-bayésien généralisé de premier ordre (GPB-1)

L'estimateur pseudo-bayésien généralisé de premier ordre (GPB-1) [75] limite la mémoire des événements de l'histoire ; la séquence des modèles, en combinant les estimations de tous les modèles dans une estimation simple à la fin de chaque cycle de traitement. Au début de chaque cycle de traitement, l'information reportée de l'intervalle de mesure précédente est une estimation combinée simple: Par conséquent la pdf de l'estimation (cas avec changement de modèles) (3.12) prend la version simplifiée :

$$p(x(k)|Z^k) = \sum_{j=1}^r p(x(k)|M_{k,j}, Z^k)P(M_{k,j}|Z^k) \quad (3.15)$$

dont le développement de la probabilité totale au-dessus de l'évènement de l'histoire du modèle entier $M_{k,l}$ est remplacé par le développement au-dessus de l'évènement du modèle simple le plus récent $M_{k,j}$. En développant (3.15), il s'avère que l'histoire de mesures précédente Z^{k-1} est représentée par l'état estimé et sa covariance simples du cycle de traitement précédent, $\{\hat{x}(k-1|k-1), P(k-1|k-1)\}$:

$$\begin{aligned} p(x(k)|Z^k) &= \sum_{j=1}^r p(x(k)|M_{k,j}, Z^k)P(M_{k,j}|Z^k) \\ p(x(k)|Z^k) &= \sum_{j=1}^r p(x(k)|M_{k,j}, z(k), Z^{k-1})P(M_{k,j}|Z^k) \\ p(x(k)|Z^k) &= \sum_{j=1}^r p(x(k)|M_{k,j}, z(k), \hat{x}(k-1|k-1), P(k-1|k-1))P(M_{k,j}|Z^k) \end{aligned} \quad (3.16)$$

En effet, l'approximation de (3.15) et (3.16) signifie de dire que l'histoire des transitions du modèle entier et l'histoire des mesures sont représentables à travers l'estimation simple du cycle de traitement précédent. Une fois la probabilité conditionnelle du modèle dans (3.16) a été évaluée, en utilisant les développements de (3.13) et (3.14), l'estimation combinée est donc calculée comme dans (3.8) et (3.9) et le cycle se répète.

La structure de l'algorithme GPB-1 est montrée sur la figure 3.3, qui exige r filtres en parallèle.

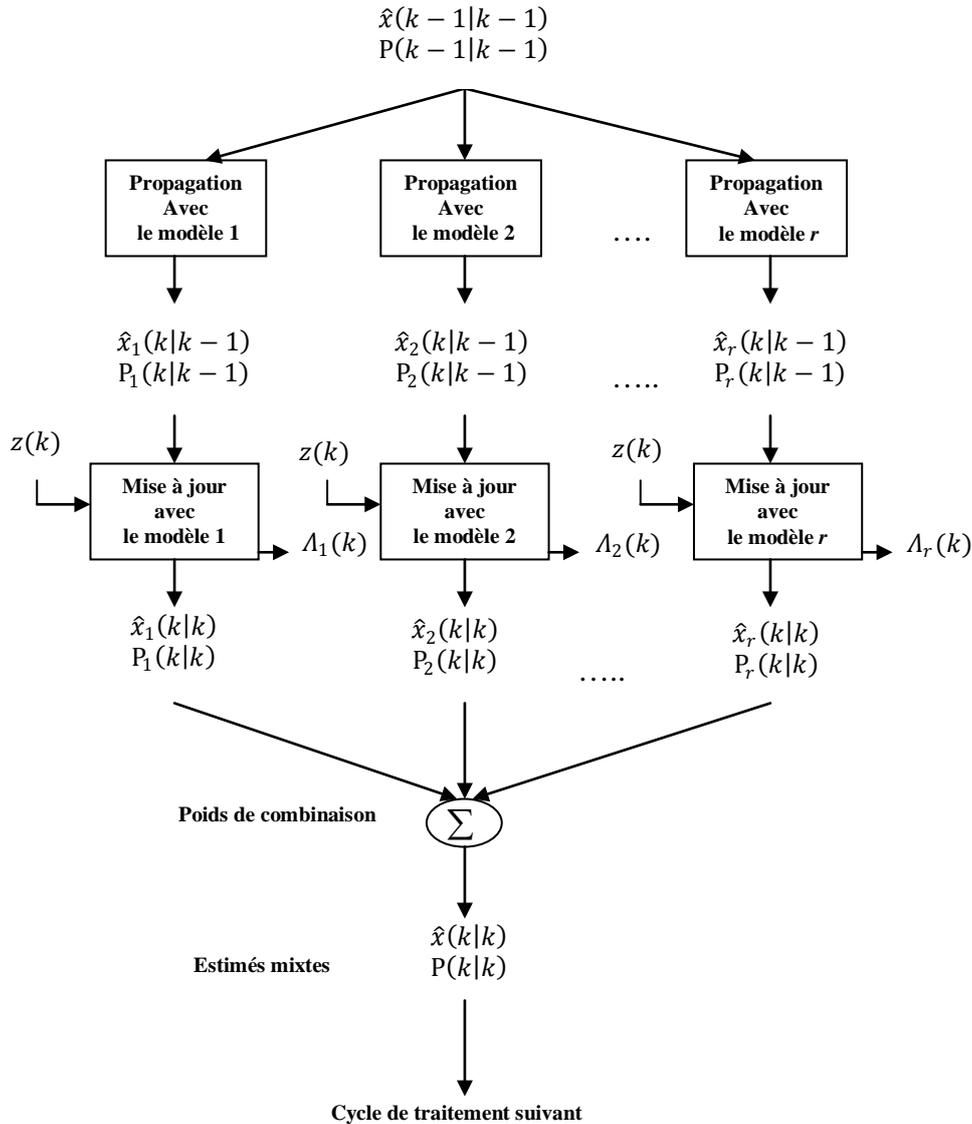


Fig. 3.3 : Schéma descriptif de l'algorithme GPB-1

La sortie de chaque filtre est l'état estimé du modèle conditionné \hat{x}_j , la covariance associée P_j et le mode de la fonction de vraisemblance Λ_j , dénommée *likelihood function* en anglais. Les fonctions de vraisemblances sont utilisées pour faire la mise à jour des probabilités des modèles. Les sorties de tous les filtres sont fusionnées dans une estimation simple \hat{x} et P à chaque cycle de traitement en utilisant les probabilités des modèles, qui est employée comme entrée à chacun des filtres au prochain cycle de traitement, fournissant une approximation très brute du système optimal représenté sur la figure 3.3.

La structure de cet algorithme est :

$$(N_e, N_r) = (1, r) \quad (3.17)$$

où N_e est le nombre d'estimés au début du cycle de l'algorithme et N_r est le nombre de filtres dans l'algorithme.

❖ Algorithme GPB-1:

Un cycle de l'algorithme consiste à effectuer les opérations suivantes :

1. Filtrage

Commençant avec $\hat{x}(k-1|k-1)$, pour chaque filtre on calcule $\hat{x}_j(k|k)$ et sa covariance associée $P_j(k|k)$, les fonctions de vraisemblances $\Lambda_j(k)$ selon les r filtres sont évaluées comme suit :

$$\begin{aligned} \Lambda_j(k) &= p(z(k)|M_j(k), Z^{k-1}) \\ \Lambda_j(k) &= p(z(k)|M_j(k), \hat{x}(k-1|k-1), P_j(k-1|k-1)) \quad j = 1, \dots, r \end{aligned} \quad (3.18)$$

2. Mise à jour de probabilité

$$\begin{aligned} \mu_j(k) &= P(M_j(k)|Z^k) \\ &= P(M_j(k)|z(k), Z^{k-1}) \\ &= \frac{1}{c} p(z(k)|M_j(k), Z^{k-1}) P(M_j(k)|Z^{k-1}) \\ &= \frac{1}{c} \Lambda_j(k) \sum_{i=1}^r P(M_j(k)|M_i(k-1), Z^{k-1}) P(M_i(k-1)|Z^{k-1}) \\ &= \frac{1}{c} \Lambda_j(k) \sum_{i=1}^r \pi_{ij} \mu_i(k-1) \end{aligned} \quad (3.19)$$

où c est la constante de normalisation :

$$c = \sum_{j=1}^r \Lambda_j(k) \sum_{i=1}^r \pi_{ij} \mu_i(k-1) \quad (3.20)$$

3. Combinaison des états estimés et covariances

L'état estimé combiné et sa covariance sont obtenus selon la sommation dans (3.15) comme suit :

$$\hat{x}(k|k) = \sum_{j=1}^r \mu_j(k) \hat{x}_j(k|k) \quad (3.21)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \left(P_j(k|k) + [\hat{x}_j(k|k) - \hat{x}(k|k)][\hat{x}_j(k|k) - \hat{x}(k|k)]^T \right) \quad (3.22)$$

3.1.2.2 Estimateur pseudo-bayésien généralisé de deuxième ordre (GPB-2)

L'estimateur pseudo-bayésien généralisé de deuxième ordre (GPB-2) [75] opère sur des principes semblables à la variante du premier ordre, sauf que la mémoire est permise de prolonger un cycle de traitement additionnel. Encore, la pdf de l'estimation est modifiée (cas avec changement de modèles de plein ordre) de (3.12) à la version simplifiée, que cette fois incorpore le modèle précédent $M_{k-1,i}$ en addition au modèle courant $M_{k,j}$:

$$p(x(k)|Z^k) = \sum_{j=1}^r \sum_{i=1}^r p(x(k)|M_{k,j}, M_{k-1,i}, Z^k) P(M_{k,j}, M_{k-1,i}|Z^k) \quad (3.23)$$

Manipulant (3.23) et supposant que l'histoire $\{M_{k-1,i}, Z^{k-1}\}$ est représentée par les estimés combinés du lième modèle dans le cycle de traitement précédent $\{\hat{x}_i(k-1|k-1), P_i(k-1|k-1)\}$, et que (selon le modèle de Markov) la transition du modèle dépend seulement du modèle précédent mais pas de l'histoire de mesure:

$$\begin{aligned} p(x(k)|Z^k) &= \sum_{j=1}^r \sum_{i=1}^r p(x(k)|M_{k,j}, M_{k-1,i}, Z^k) P(M_{k-1,i}|M_{k,j}, Z^k) P(M_{k,j}|Z^k) \\ &= \sum_{j=1}^r \sum_{i=1}^r p(x(k)|M_{k,j}, z(k), \{M_{k-1,i}, Z^{k-1}\}) \mu_{i|j}(k-1|k) \mu_j(k) \end{aligned} \quad (3.24)$$

Donc, à l'instant $k-1$ il y a r états estimés et covariances, chacun est prédit et mis à jour à l'instant k sous r hypothèses, à savoir :

$$\hat{x}_{ij}(k|k) \triangleq \hat{x}\{k|k; M_j(k), \hat{x}_i(k-1|k-1), P_i(k-1|k-1)\} \quad i, j = 1, \dots, r \quad (3.25)$$

$$P_{ij}(k|k) \triangleq P\{k|k; M_j(k), P_i(k-1|k-1)\} \quad i, j = 1, \dots, r \quad (3.26)$$

Après la mise à jour, les estimés correspondant au même modèle sont combinés avec les pondérations $\mu_{i|j}(k-1|k)$ résultant r estimés $\hat{x}_j(k|k)$. En d'autres termes, les r^2 hypothèses sont fusionnées dans r à la fin de chaque cycle d'estimation. Ceci est illustré à la figure 3.4, qui montre la structure de l'algorithme. Comparé avec la structure de l'algorithme GPB-1, représentée sur la figure 3.3, l'algorithme GPB-2 utilise r^2 filtres, ainsi il peut maintenir r estimations et propage chaque estimation avec chacun de r filtres à chaque intervalle de traitement.

La structure de l'algorithme GPB-2 est :

$$(N_e, N_f) = (r, r^2) \tag{3.27}$$

Où N_e est le nombre d'estimés au début du cycle de l'algorithme et N_f est le nombre de filtres dans l'algorithme.

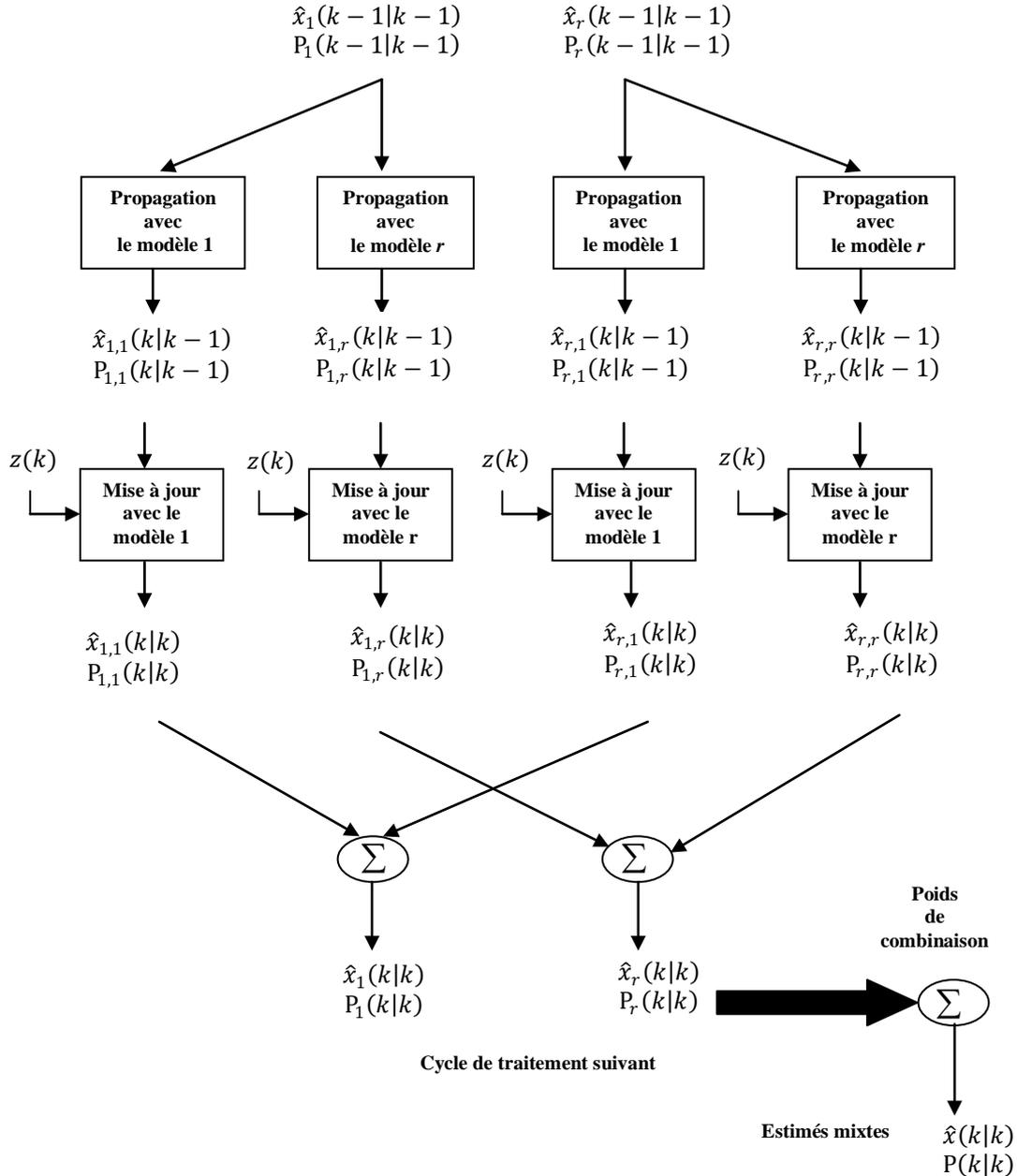


Fig. 3.4 : Schéma descriptif de l'algorithme GPB-2

❖ Algorithme GPB-2:

Un cycle de l'algorithme consiste à effectuer les opérations suivantes :

1. Filtrage

Commençant avec $\hat{x}_i(k-1|k-1)$ et $P_i(k-1|k-1)$ on calcule $\hat{x}_{ij}(k|k)$ et sa covariance associée $P_{ij}(k|k)$ à travers le filtre assorti de $M_j(k)$. Les fonctions de vraisemblances correspondantes à ces r^2 filtres sont :

$$\Lambda_{ij}(k) = p(z(k)|M_j(k), M_i(k-1), Z^{k-1}) \quad (3.28)$$

Elles sont évaluées comme suit:

$$\Lambda_{ij}(k) = p\left(z(k)|M_j(k), \hat{x}_i(k-1|k-1), P_i(k-1|k-1)\right) \quad i, j = 1, \dots, r \quad (3.29)$$

2. Calcul de probabilités de fusionnement

La probabilité que le modèle i avait, en effet, à l'instant $k-1$ si le modèle j est, en effet, à l'instant k conditionnée sur Z^k est :

$$\begin{aligned} \mu_{i|j}(k-1|k) &\triangleq P(M_i(k-1)|M_j(k), Z^k) \\ &= P(M_i(k-1)|z(k), M_j(k), Z^{k-1}) \\ &= \frac{1}{c_j} p(z(k), M_j(k)|M_i(k-1), Z^{k-1}) P(M_i(k-1)|Z^{k-1}) \\ &= \frac{1}{c_j} p(z(k)|M_j(k), M_i(k-1), Z^{k-1}) P(M_j(k)|M_i(k-1), Z^{k-1}) \\ &\quad \cdot P(M_i(k-1)|Z^{k-1}) \end{aligned} \quad (3.30)$$

donc les probabilités de fusionnement sont:

$$\mu_{i|j}(k-1|k) = \frac{1}{c_j} \Lambda_{ij}(k) \pi_{ij} \mu_i(k-1) \quad i, j = 1, \dots, r \quad (3.31)$$

Avec

$$c_j = \sum_{i=1}^r \Lambda_{ij}(k) \pi_{ij} \mu_i(k-1) \quad j = 1, \dots, r \quad (3.32)$$

Les probabilités de transition des modèles π_{ij} sont supposées connues, leur choix fait partie du processus de conception de l'algorithme.

3. Fusionnement

L'état estimé correspondant à $M_j(k)$ est obtenu en combinant les estimés de (3.25) selon la sommation intérieure dans (3.24) comme suit:

$$\hat{x}_j(k|k) = \sum_{i=1}^r \hat{x}_{ij}(k|k) \mu_{ij}(k-1|k) \quad j = 1, \dots, r \quad (3.33)$$

La covariance correspondante est :

$$P_j(k|k) = \sum_{i=1}^r \mu_{ij}(k-1|k) \left(P_{ij}(k|k) + [\hat{x}_{ij}(k|k) - \hat{x}_j(k|k)][\hat{x}_{ij}(k|k) - \hat{x}_j(k|k)]^T \right) \quad j = 1, \dots, r \quad (3.34)$$

4. Mise à jour de probabilités des modèles

Ceci se fait comme suit :

$$\begin{aligned} \mu_j(k) &\triangleq P(M_j(k)|z(k), Z^{k-1}) \\ &= \frac{1}{c} p(z(k), M_j(k)|Z^{k-1}) \\ &= \frac{1}{c} \sum_{i=1}^r p(z(k), M_j(k)|M_i(k-1), Z^{k-1}) P(M_i(k-1)|Z^{k-1}) \\ &= \frac{1}{c} \sum_{i=1}^r p(z(k)|M_j(k), M_i(k-1), Z^{k-1}) P(M_j(k)|M_i(k-1), Z^{k-1}) \\ &\quad \cdot P(M_i(k-1)|Z^{k-1}) \quad j = 1, \dots, r \end{aligned} \quad (3.35)$$

donc les probabilités de mises à jour des modèles sont:

$$\mu_j(k) = \frac{1}{c} \sum_{i=1}^r \Lambda_{ij}(k) \pi_{ij} \mu_i(k-1) = \frac{c_j}{c} \quad j = 1, \dots, r \quad (3.36)$$

où c_j est décrit dans (3.32) et c est la constante de normalisation :

$$c = \sum_{j=1}^r c_j \quad (3.37)$$

5. Combinaison des états estimés et covariances

L'état estimé combiné et sa covariance sont :

$$\hat{x}(k|k) = \sum_{j=1}^r \mu_j(k) \hat{x}_j(k|k) \quad (3.38)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \left(P_j(k|k) + [\hat{x}_j(k|k) - \hat{x}(k|k)][\hat{x}_j(k|k) - \hat{x}(k|k)]^T \right) \quad (3.39)$$

3.1.2.3 Estimateur modèles multiples interagissants (IMM)

L'estimateur modèle multiple interagissant [25],[75],[85] dénommé Interacting Multiple Model en anglais et dénoté IMM, est une méthodologie qui réalise l'exécution comparable à l'estimateur GPB-2 à l'aide seulement de r filtres élémentaires mais pas de r^2 . L'algorithme peut être obtenu en considérant les limitations inhérentes au problème: si seulement r filtres élémentaires sont permis, alors l'entrée au $j^{\text{ième}}$ filtre devrait être la meilleure estimation de l'état à l'instant $k - 1$, conditionnée sur l'événement que le modèle j est en vigueur à l'instant k (le nouvel instant d'échantillonnage), $p(x(k-1)|M_{k,j}, Z^{k-1})$. En utilisant cette expression comme point de départ, nous suivons une itération simple de l'algorithme à travers le calcul de la même fonction à la période d'échantillonnage suivante.

Après un cycle de mise à jour de la propagation du filtre de Kalman standard au $k^{\text{ième}}$ instant d'échantillonnage, la sortie du $j^{\text{ième}}$ filtre élémentaire sera $p(x(k)|M_{k,j}, Z^k)$. La condition pour l'algorithme IMM est donc de combiner les estimés de r filtres élémentaires pour calculer les entrées $p(x(k)|M_{k+1,i}, Z^k)$ de chaque filtre élémentaire pour le prochain cycle de traitement.

La pdf globale formée en utilisant l'information de tous les r filtres représente l'information totale contenue par le système à l'instant k :

$$p(x(k)|Z^k) = \sum_{j=1}^r p(x(k)|M_{k,j}, Z^k) P(M_{k,j}|Z^k) \quad (3.40)$$

Le but de l'entremêlement est ainsi de masser (3.40) dans l'expansion nécessaire à l'entrée du prochain cycle de traitement:

$$p(x(k)|Z^k) = \sum_{i=1}^r p(x(k)|M_{k+1,i}, Z^k) P(M_{k+1,i}|Z^k) \quad (3.41)$$

Le dernier facteur dans (3.41) est facilement évalué en utilisant l'hypothèse de Markov selon (3.14):

$$\begin{aligned} P(M_{k+1,i}|Z^k) &= \sum_{j=1}^r P(M_{k+1,i}|M_{k,j}, Z^k) P(M_{k,j}|Z^k) \\ &= \sum_{j=1}^r P(M_{k+1,i}|M_{k,j}) P(M_{k,j}|Z^k) \end{aligned} \quad (3.42)$$

Notons que si $\pi(k+1|k)$ est la matrice de transitions de Markov telle que:

$$(\pi)_{ij} = P(M_{k+1,i} | M_{k,j})$$

Alors (3.42) est simplement une multiplication de la matrice $\pi(k+1|k)$ par le vecteur dont ces éléments sont les probabilités $P(M_{k,j} | Z^k)$, donnant le vecteur des composants qui représentent les probabilités $P(M_{k+1,i} | Z^k)$.

Le facteur principal dans la somme de (3.41) est alors augmenté en utilisant le théorème de probabilité totale au-dessus du modèle précédent de l'index j :

$$p(x(k) | M_{k+1,i}, Z^k) = \sum_{j=1}^r p(x(k) | M_{k+1,i}, M_{k,j}, Z^k) P(M_{k,j} | M_{k+1,i}, Z^k) \quad (3.43)$$

où les probabilités en arrière de transition sont calculées par :

$$\begin{aligned} P(M_{k,j} | M_{k+1,i}, Z^k) &= \frac{P(M_{k,j}, M_{k+1,i} | Z^k)}{P(M_{k+1,i} | Z^k)} \\ &= \frac{P(M_{k+1,i} | M_{k,j}, Z^k) P(M_{k,j} | Z^k)}{P(M_{k+1,i} | Z^k)} \\ &= \frac{P(M_{k+1,i} | M_{k,j}, Z^k) P(M_{k,j} | Z^k)}{\sum_{n=1}^r P(M_{k+1,i} | M_{k,n}, Z^k) P(M_{k,n} | Z^k)} \end{aligned} \quad (3.44)$$

Selon l'hypothèse de Markov, la probabilité de transition $P(M_{k+1,i} | M_{k,j}, Z^k)$ ne dépend pas de l'histoire de mesure Z^k .

En supposant que l'histoire Z^k de l'estimateur est convenablement modélisée par les r estimations du cycle de traitement précédent (chaque estimation conditionnée sur un modèle différent $M_{k,j}$), (4.43) est alors approximée par une densité gaussienne simple:

$$\begin{aligned} p(x(k) | M_{k+1,i}, Z^k) &\approx \sum_{j=1}^r p(x(k) | M_{k+1,i}, \hat{x}_j(k|k), P_j(k|k)) P(M_{k,j} | M_{k+1,i}, Z^k) \\ &\approx N\{x(k); \hat{x}^i(k|k), P^i(k|k)\} \end{aligned} \quad (3.45)$$

dont la moyenne et la covariance de la gaussienne sont données par:

$$\hat{x}^i(k|k) = \sum_{j=1}^r P(M_{k,j} | M_{k+1,i}, Z^k) \hat{x}_j(k|k) \quad i = 1, \dots, r \quad (3.46)$$

$$P^i(k|k) = \sum_{j=1}^r P(M_{k,j} | M_{k+1,i}, Z^k) \left(P_j(k|k) + [\hat{x}_j(k|k) - x^i(k|k)] [\hat{x}_j(k|k) - x^i(k|k)]^T \right) \dots \dots \dots (3.47)$$

avec :

$$\mu_{i|j}(k|k) = P(M_{k,j} | M_{k+1,i}, Z^k) \quad (3.48)$$

Les probabilités a posteriori exigées des modèles $P(M_{k,j} | Z^k)$ pour (3.44), sont récursivement calculées en utilisant les expressions :

$$\begin{aligned} P(M_{k,j} | Z^k) &= P(M_{k,j} | z(k), Z^{k-1}) \\ &= \frac{p(M_{k,j}, z(k) | Z^{k-1})}{p(z(k) | Z^{k-1})} \\ &= \frac{p(z(k) | M_{k,j}, Z^{k-1}) P(M_{k,j} | Z^{k-1})}{p(z(k) | Z^{k-1})} \end{aligned} \quad (3.49)$$

Comme discuté dans (3.42), $P(M_{k,j} | Z^{k-1})$ peut être augmentée ou développée en utilisant le théorème de la probabilité totale comme:

$$\begin{aligned} P(M_{k,j} | Z^{k-1}) &= \sum_{i=1}^r P(M_{k,j} | M_{k-1,i}, Z^{k-1}) P(M_{k-1,i} | Z^{k-1}) \\ &= \sum_{i=1}^r P(M_{k,j} | M_{k-1,i}) P(M_{k-1,i} | Z^{k-1}) \end{aligned} \quad (3.50)$$

où l'hypothèse que la probabilité de transition du modèle ne dépend pas de l'histoire de mesure est de nouveau rappelée.

Ainsi, la substitution dans (3.50) et en augmentant le dénominateur en utilisant le théorème de la probabilité totale, (3.49) devient:

$$P(M_{k,j} | Z^k) = \frac{p(z(k) | M_{k,j}, Z^{k-1}) \sum_{i=1}^r P(M_{k,j} | M_{k-1,i}) P(M_{k-1,i} | Z^{k-1})}{\sum_{n=1}^r p(z(k) | M_{k,n}, Z^{k-1}) P(M_{k,n} | Z^{k-1})} \quad (3.51)$$

Notons que le dénominateur est le facteur de normalisation nécessaire pour assurer que la somme de probabilités des modèles conditionnels est égale à 1.

Selon les techniques modèles multiples précédentes, l'estimation combinée est calculée à chaque cycle de traitement pour donner la sortie de l'estimateur. Un schéma fonctionnel de l'algorithme IMM est montré dans la figure 3.5. La structure est très semblable à la structure des modèles fixes (cas de non changement de modèles) représentée sur la figure 3.1: il y a r filtres, dont chacun est fourni avec une entrée différente. Cependant, plutôt que de passer la sortie de chaque filtre directement dans le même filtre au prochain cycle de traitement, l'algorithme mélange les estimations selon le modèle de transition de Markov afin de permettre au système de réagir aux changements du modèle en vigueur. D'une

autre façon, l'entrée au filtre assorti du modèle i est obtenue par une interaction de r filtres, qui consiste à mélanger les estimés $\hat{x}_j(k|k)$ avec les pondérations $\mu_{ij}(k|k)$, appelées probabilités mixtes. Les détails de cet algorithme (IMM) peuvent être trouvés dans [25], [75] et [86].

La structure de l'algorithme IMM est :

$$(N_e, N_f) = (r, r) \tag{3.52}$$

où N_e est le nombre d'estimations au début du cycle de traitement et N_f est le nombre de filtres dans l'algorithme.

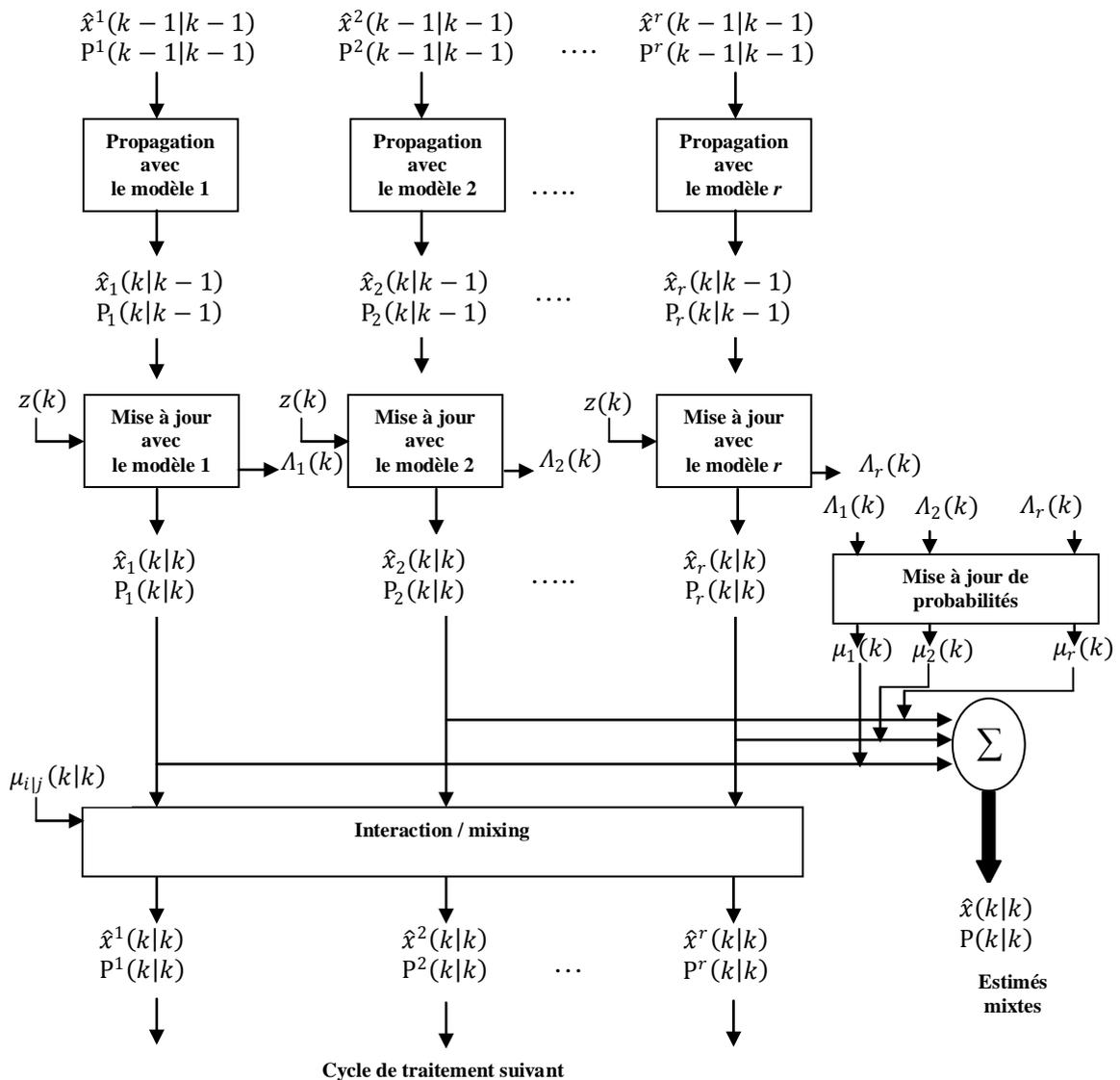


Fig. 3.5 : Schéma descriptif de l'algorithme IMM

3.2 Conclusion

Les sections précédentes ont présenté les approches d'estimation modèles multiples généralement utilisées. L'estimateur modèle multiple traditionnel est fondé sur l'hypothèse que le modèle en vigueur ne change pas avec le temps; mais, si on tient compte des changements du modèle, c'est à dire le modèle en vigueur est autorisé de se changer à n'importe quel instant. Donc, ces changements prolongent l'algorithme pour fournir des performances adéquates dans un environnement avec changement de modèles (switching model). Tous les estimateurs (cas avec changement de modèles) utilisent un modèle de Markov pour des probabilités de transition ; l'algorithme le plus utilisé est l'IMM, qui fournit une performance semblable au GPB-2 à une fraction du coût de calcul.

Chapitre 4

Poursuite linéaire multi-cibles

Le problème de filtrage d'une cible unique est souvent perturbé par la présence de mesures qui ne résultent pas de la détection de la cible considérée, soit parce qu'elles sont issues d'une autre cible présente dans le volume d'observation, soit parce qu'elles ont été générées par un processus d'élaboration de mesure, sans être issues d'une cible. Ces dernières mesures sont généralement appelées des fausses alarmes. A la réception du vecteur de mesures, rien ne nous indique cependant leur origine, ce qui nécessite d'émettre des hypothèses sur les associations entre les mesures et les cibles. Dans ce chapitre, nous exposerons tout d'abord la modélisation du problème de filtrage linéaire multi-cibles que nous venons de décrire. Différents algorithmes déterministes permettent de résoudre ce problème. Nous les exposerons en distinguant pour cela deux classes de filtres, selon que l'association d'une mesure à une cible est considérée de façon dépendante ou pas de l'association des autres mesures.

4.1 Formulation de la poursuite multi-cibles

Soit M le nombre de cibles suivies. Cet entier sera supposé connu et constant dans certains algorithmes et inconnu et variable dans d'autres. Nous utiliserons toujours l'exposant i pour faire référence à l'indice d'une cible parmi les M présentes. Le vecteur X_k désigne maintenant la concaténation des états (X_k^1, \dots, X_k^M) où $(X_k^i)_{k \in \mathbb{N}}$ est à valeurs dans $\mathbb{R}^{n_x^i}$ pour $i = 1, \dots, M$. On supposera au minimum que le processus X_k est markovien:

$$X_k = F_k(X_{k-1}, V_k) \quad (4.1)$$

En pratique, les différentes cibles seront généralement modélisées selon des dynamiques indépendantes. On a alors [83]:

$$X_k^i = F_k^i(X_{k-1}^i, V_k^i) \quad \forall i = 1, \dots, M. \quad (4.2)$$

A chaque instant $k \in \mathbb{N}$, on dispose d'un vecteur de mesure de longueur m_k :

$$Y_k = (y_k^1, \dots, y_k^{m_k}). \quad (4.3)$$

La lettre j sera utilisée en exposant pour désigner un indice de mesures parmi les m_k mesures présentes à l'instant k . Certaines mesures sont issues de cibles détectées et d'autres sont des fausses alarmes. On suppose généralement que le nombre de fausses alarmes suit une loi de Poisson de paramètre λV où λ désigne un nombre réel et V désigne le volume d'observation et qu'elles sont uniformément réparties dans ce volume. Il est également possible qu'une cible ne soit pas détectée à un instant et qu'aucune mesure n'en soit issue. Du fait de la présence de fausses alarmes et de la possibilité de non détection des cibles, le nombre de mesures m_k n'est en général pas égal au nombre de cibles M .

L'incertitude sur l'origine des mesures est double [83]:

- la présence de fausses alarmes implique qu'une mesure ne provient pas obligatoirement d'une cible.
- la présence de plusieurs cibles implique qu'une mesure qui n'est pas une fausse alarme peut a priori provenir de chacune des cibles.

Pour modéliser cette incertitude, on introduit la variable Θ_k qui indique l'association entre les mesures et les cibles. Ainsi $\Theta_k^j = i$ signifie que la mesure y_k^j provient de la cible d'indice i si $i = 1, \dots, M$ et qu'elle est une fausse alarme si i vaut 0. Pour les mesures qui ne sont pas de fausses alarmes, on suppose que le modèle suivant est vérifié :

$$Y_k^j = H_k^i(X_k^i, W_k^j) \quad (4.4)$$

Aucun modèle cinématique n'est associé au modèle des fausses alarmes, indicé par 0. L'indexation des composantes du vecteur de mesure Y_k étant arbitraire, on suppose que toutes les mesures ont la même probabilité a priori d'être issues d'une cible donnée. Ainsi, les variables aléatoires $(\Theta_k^j)_{j=1, \dots, m_k}$ ont supposées identiquement distribuées. Leur loi commune est définie à l'aide des probabilités π_k^i suivantes:

$$\pi_k^i \triangleq \mathbb{P}(\Theta_k^j = i) \quad \forall j = 1, \dots, m_k. \quad (4.5)$$

La probabilité π_k^i du modèle i est ainsi la probabilité a priori qu'une mesure quelconque soit associée au modèle i . La dénomination de « modèle » recouvre celle de la cible si $i = 1, \dots, M$ et le modèle des fausses alarmes si $i = 0$. Intuitivement, la probabilité π_k^i représente l'observabilité de la cible i pour $i = 1, \dots, M$. Le vecteur $\pi_k = (\pi_k^0, \pi_k^1, \dots, \pi_k^M)$ est considéré comme une réalisation du vecteur aléatoire $\Pi_k = (\Pi_k^0, \Pi_k^1, \dots, \Pi_k^M)$ où l'a priori suivant est mis sur la loi de Π_k :

$$p(\Pi_k) = p(\Pi_0)p(\Pi_k^1, \dots, \Pi_k^M | \Pi_k^0) \quad (4.6)$$

Où $p(\Pi_k) = p(\Pi_0)p(\Pi_k^1, \dots, \Pi_k^M | \Pi_k^0)$ est uniforme sur l'hyperplan $[0, 1 - \Pi_k^0]^M$ vérifiant $\sum_{i=1}^M \Pi_k^i = 1 - \Pi_k^0$. On suppose ainsi que chaque mesure, si elle n'est pas une fausse alarme, a la même probabilité a priori d'être associée à chacune des cibles. De plus, la probabilité qu'une mesure soit associée au modèle des fausses alarmes est calculable et dépend du nombre de mesures m_k et de la densité des fausses alarmes λV , [83]:

$$\begin{aligned} \pi_k^0 \triangleq \mathbb{P}(\Theta_k^j = 0) &= \sum_{l=0}^{m_k} \mathbb{P}(\Theta_k^j = 0 | N_k^0 = l) \mathbb{P}(N_k^0 = l) \\ &= \sum_{l=0}^{m_k} \frac{l}{m_k} \exp(\lambda V) \frac{(\lambda V)^l}{l!} \end{aligned} \quad (4.7)$$

Où N_k^0 est le nombre de mesures qui sont des fausses alarmes à l'instant k . Chaque terme de la somme est calculé selon le raisonnement suivant : supposant que l mesures sur les m_k soient des fausses alarmes, alors la probabilité qu'une mesure quelconque soit une fausse alarme vaut $\frac{l}{m_k}$, i.e., $\mathbb{P}(\Theta_k^j = 0 | N_k^0 = l) = \frac{l}{m_k}$.

Enfin, on suppose qu'à chaque instant les mesures sont indépendantes conditionnellement à l'état, i.e. :

$$p(Y_k | X_k) = \prod_{j=1}^{m_k} p(Y_k^j | X_k) \quad (4.8)$$

On s'intéresse au problème de filtrage de la loi $L_{k|k}$ qui revient à l'estimation des M lois $L_{k|k}^i(dx_k) \triangleq \mathbb{P}(X_k^i | Y_{1:k} = y_{1:k})$ pour $i = 1, \dots, M$. Du fait de la non connaissance des associations entre les mesures et les objets, nous ne sommes pas en mesure de découpler cette estimation en M estimations indépendantes. Le filtrage avec incertitude sur l'origine des mesures nécessite ainsi la résolution de l'association de données. Nous verrons que plusieurs hypothèses d'indépendance peuvent être émises conduisant à l'indépendance ou non des composantes Θ_k^j de la variable d'association Θ_k . Nous nous contentons pour l'instant de les énoncer et nous les étudierons dans les sections suivantes.

Nous nous intéressons tout d'abord à la résolution du problème de filtrage dans un environnement bruité en supposant les hypothèses (H1) et (H2) qui induit la dépendance des associations des différentes mesures.

- (H1) Une mesure peut provenir d'une cible ou être une fausse alarme.
- (H2) A un instant donné, une cible peut être à l'origine d'aucune ou d'au plus une mesure.
- (H3) A un instant donné, une cible peut être à l'origine d'aucune ou de plusieurs mesures.

4.2 Filtrage et association de données avec dépendance des associations

Les algorithmes suivants sont construits sous l'hypothèse de modèles d'état et de mesures linéaires avec des bruits additifs gaussiens. Les processus d'état sont de plus supposés indépendants [83]:

$$\begin{cases} X_k^i = F_k^i X_{k-1}^i + V_k^i & \forall i = 1, \dots, M \\ Y_k^j = H_k^j X_k^i + W_k^j & \text{si } \Theta_k^j = i \end{cases} \quad (4.9)$$

Où $(V_k^i)_{k \in \mathbb{N}}$ pour $i = 1, \dots, M$ et $(W_k^j)_{k \in \mathbb{N}}$ pour $j = 1, \dots, m_k$ sont des bruits blancs gaussiens indépendantes de matrices de covariance respectives Σ_V^i et Σ_W^j .

Les méthodes suivantes sont fondées sur la représentation de la loi $L_{k|k}^i = \mathbb{P}(X_k^i | Y_{1:k})$ par une gaussienne et l'utilisation du filtre de Kalman. Etant donnée l'incertitude sur l'origine des mesures, la probabilité de chaque association est évaluée et utilisée pour pondérer les estimations. Le vecteur d'association est soumis aux deux hypothèses d'association (H1) et (H2) détaillées dans le paragraphe suivant.

4.2.1 Hypothèses d'association

(H1) une mesure peut provenir d'une cible ou être une fausse alarme.

Cette hypothèse traduit l'exclusivité et l'exhaustivité du processus d'association. En pratique, cela nous permettra d'utiliser le théorème des probabilités totales pour décomposer un calcul de probabilité selon l'ensemble des réalisations possibles du vecteur d'association. L'égalité suivante sera notamment vérifiée à chaque pas de temps:

$$\sum_{i=0}^M \mathbb{P}(\Theta_k^j = i) = 1 \quad \forall j = 1, \dots, m_k \quad (4.10)$$

L'hypothèse suivante énonce la dépendance des variables $(\Theta_k^j)_{j=1, \dots, m_k}$ entre-elles.

(H2) A un instant donné, une cible peut être à l'origine d'aucune ou d'au plus une mesure.

Cette dépendance s'écrit :

$$\forall k = 1, \dots, \text{fin} \quad \forall i = 1, \dots, M \quad \forall j, j' \in \{1, \dots, m_k\} \text{ tel que } j \neq j', \quad \Theta_k^j = i \Rightarrow \Theta_k^{j'} \neq i \quad (4.11)$$

4.2.2 Principe de validation des mesures

Pour limiter la complexité des calculs, une approximation consiste à ne considérer que certaines mesures selon le test de validation ou fenêtrage statistique suivant:

$$\begin{cases} H_0 : \text{ la mesure } y_k \text{ ne provient pas de la cible } x_k \\ H_1 : \text{ la mesure } y_k \text{ provient de la cible } x_k \end{cases} \quad (4.12)$$

Seules les mesures pour lesquelles H_1 est acceptée sont sélectionnées. Ce test est fondé sur les hypothèses suivantes. On suppose que la loi $L_{k|k-1}$ est une loi gaussienne $N(X_k; \bar{X}_{k|k-1}, P_{k|k-1})$. La mesure prédite, l'innovation et la covariance de l'innovation sont alors définies par

$$y_{k|k-1} \triangleq H_k \bar{X}_{k|k-1}, \quad (4.13)$$

$$\tilde{y}_{k|k-1} \triangleq y_k - y_{k|k-1}, \quad (4.14)$$

et

$$S_k = H_k P_{k|k-1} H_k^T + \Sigma_W \quad (4.15)$$

On suppose alors que la mesure associée à X_k est distribuée selon la loi Gaussienne centrée en la mesure prédite $y_{k|k-1}$ et de covariance S_k . On définit la distance de Mahalanobis associée:

$$\epsilon_k(y_k) \triangleq (y_k - y_{k|k-1})^T S_k^{-1} (y_k - y_{k|k-1}) = \tilde{y}_{k|k-1}^T S_k^{-1} \tilde{y}_{k|k-1}. \quad (4.16)$$

La mesure y_k est alors validée si la distance $\epsilon_k(y_k)$ est inférieure à un certain seuil γ , soit:

$$\begin{cases} H_1 & \text{acceptée si } (y_k - y_{k|k-1})^T S_k^{-1} (y_k - y_{k|k-1}) \leq \gamma. \\ H_0 & \text{acceptée sinon} \end{cases} \quad (4.17)$$

Le seuil γ est choisi de tel sorte que la probabilité que la mesure correcte se trouve dans la région $\{y_k / \epsilon_k = (y_k - y_{k|k-1})^T S_k^{-1} (y_k - y_{k|k-1}) \leq \gamma\}$ soit égale à P_g . En pratique, la probabilité P_g est fréquemment fixée à $P_g = 0.99$.

4.2.3 Filtrage d'une cible unique dans un environnement bruité

On se restreint tout d'abord au cas où le nombre d'objets M est égal à un mais où l'origine des mesures est incertaine du fait de la présence de fausses alarmes parmi les mesures. Selon les hypothèses (H1) et (H2), il existe deux types d'associations [83]:

- Soit toutes les mesures validées sont supposées être des fausses alarmes et les

m_k composantes du vecteur θ_k sont nulles. Une seule réalisation de Θ_k est de ce type.

- Soit, $m_k - 1$ composantes sont nulles et une composante vaut un. Selon le choix de cette dernière composante, il y a m_k réalisations de Θ_k de ce type.

Il y a donc $\rho_k = 1 + m_k$ réalisations possibles pour Θ_k . On suppose que ces réalisations sont ordonnées et l'on désigne par $\theta_k^{\#l}$ la lième réalisation de Θ_k . Si celle-ci est du second type, on désignera par j^l l'indice de la mesure associée à la cible. Nous convenons que le symbole # alourdit l'écriture mais sa présence permet de différencier une réalisation $\theta_k^{\#l}$ du vecteur Θ_k d'une réalisation θ_k^j de la composante Θ_k^j de Θ_k .

Nous exposons en premier lieu le filtre bayésien optimal qui prend en compte l'ensemble des associations de l'instant initial à l'instant courant pour estimer l'état courant.

4.2.3.1 Filtre bayésien optimal (FBO)

Initialement proposé dans [69], le filtre bayésien optimal consiste à décomposer la loi $L_{k|k}$ selon le théorème des probabilités totales sur l'ensemble des réalisations possibles pour $\Theta_{1:k}$.

$$p(X_k | Y_{1:k}) = p(X_k | Y_{1:k}, m_{1:k}) = \sum_{\theta_{1:k}} p(X_k | \theta_{1:k}, Y_{1:k}, m_{1:k}) \mathbb{P}(\theta_{1:k} | Y_{1:k}, m_{1:k}) \quad (4.18)$$

La première égalité de l'équation précédente rappelle que le conditionnement par rapport à m_k est implicite lors du conditionnement par rapport à Y_k . Une réalisation $\theta_{1:k}^{\#l}$ du vecteur $\Theta_{1:k}$ se résume à la donnée d'un vecteur de mesures $y_{1:k}^{\#l} = (y_1^{j^l}, \dots, y_k^{j^l})$ avec éventuellement aucune mesure à un instant donné, si elles sont toutes associées au modèle des fausses alarmes. Elle s'exprime récursivement comme la concaténation d'une réalisation $\theta_k^{\#u}$ de Θ_k . Le nombre de réalisations possibles pour $\Theta_{1:k}$ vaut ainsi $\rho_{1:k} = \prod_{k=1}^{fin=T} (1 + m_k^v)$ où m_k^v est le nombre de mesures validées à l'instant k . La somme précédente s'écrit donc :

$$p(X_k | Y_{1:k}) = \sum_{l=1}^{\rho_{1:k}} p(X_k | Y_{1:k} = y_{1:k}^{\#l}, m_{1:k}) \mathbb{P}(\theta_{1:k} = \theta_{1:k}^{\#l} | Y_{1:k}, m_{1:k}) \quad (4.19)$$

Les lois $p(X_k | Y_{1:k} = y_{1:k}^{\#l})$ ne présentent pas d'incertitude sur l'origine des mesures. Pour un modèle d'état et de mesure linéaire gaussien, ce sont donc des lois gaussiennes dont la moyenne et la covariance peuvent s'estimer par un filtre de Kalman. La loi $L_{k|k}$ est ainsi

une somme de gaussiennes sur les réalisations possibles du vecteur $\Theta_{1:k}$ et pondérées par leur probabilités. On utilisera dans les équations à venir la décomposition de $y_{1:k}^{\#l}$ selon le passé et le présent $y_{1:k}^{\#l} = (y_{1:k-1}^{\#s}, y_k^{j^l})$.

$$\begin{aligned} p(X_k | Y_{1:k}) &= \sum_{l=1}^{\rho_{1:k}} p(X_k | Y_{1:k-1} = y_{1:k-1}^{\#s}, Y_k = y_k^{j^l}) \mathbb{P}(\Theta_{1:k} = \theta_{1:k}^{\#l} | Y_{1:k}, m_{1:k}) \\ &= \sum_{l=1}^{\rho_{1:k}} \frac{p(y_k^{j^l} | X_k) p(X_k | Y_{1:k-1} = y_{1:k-1}^{\#s})}{p(y_k^{j^l} | y_{1:k-1}^{\#s})} \mathbb{P}(\Theta_{1:k} = \theta_{1:k}^{\#l} | Y_{1:k}, m_{1:k}) \end{aligned} \quad (4.20)$$

Le filtre de Kalman fournit une estimation de la moyenne $\bar{X}_{k|k}^{\#l}$ et la covariance $P_{k|k}^{\#l}$ de la loi $p(X_k | Y_{1:k-1} = y_{1:k-1}^{\#s}, Y_k = y_k^{j^l})$:

$$\bar{X}_{k|k}^{\#l} = \bar{X}_{k|k-1}^{\#s} + K_k^{\#s} (y_k^{j^l} - y_{k|k-1}^{\#s}) \quad (4.21)$$

où $y_{k|k-1}^{\#s}$ est la mesure prédite à partir de la suite de mesures $y_{1:k-1}^{\#s}$ et $K_k^{\#s}$ le gain associé :

$$K_k^{\#s} = P_{k|k-1}^{\#s} H_k (S_k^{\#s})^{-1}. \quad (4.22)$$

En notant $\beta_{1:k}^l \triangleq \mathbb{P}(\Theta_{1:k} = \theta_{1:k}^{\#l} | Y_{1:k}, m_{1:k})$, la moyenne de $L_{k|k}$ s'écrit donc :

$$\bar{X}_{k|k} \triangleq \mathbb{E}(X_k | Y_{1:k}) = \sum_{l=1}^{\rho_{1:k}} \bar{X}_{k|k}^{\#l} \beta_{1:k}^l. \quad (4.23)$$

Des calculs menés dans [83] donnent l'expression suivante de la covariance :

$$P_{k|k} = \sum_{l=1}^{\rho_{1:k}} \beta_{1:k}^l P_{k|k}^{\#l} + \sum_{l=1}^{\rho_{1:k}} \beta_{1:k}^l \bar{X}_{k|k}^{\#l} \bar{X}_{k|k}^{\#l T} - \bar{X}_{k|k} \bar{X}_{k|k}^T. \quad (4.24)$$

Les probabilités a posteriori $\beta_{1:k}^l$ s'obtiennent récursivement par application de la règle de Bayes :

$$\beta_{1:k}^l = \frac{1}{c} p(Y_k | \Theta_{1:k} = \theta_{1:k}^{\#l}, Y_{1:k-1}, m_{1:k}) p(\Theta_k = \theta_k^{\#l} | Y_{1:k-1}, \Theta_{1:k-1} = \theta_{1:k-1}^{\#s}, m_{1:k}) \beta_{1:k-1}^s \quad (4.25)$$

où c est une constante de normalisation telle que la somme $\sum_{l=1}^{\rho_{1:k}} \beta_{1:k}^l$ vaille un. Le calcul explicite des facteurs du produit précédent donnent :

$$p(Y_k | \Theta_{1:k} = \theta_{1:k}^{\#l}, Y_{1:k-1}, m_{1:k}) = \begin{cases} \frac{1}{\sqrt{m_k-1}} \frac{1}{p_g} N(y_k^{j^l}; y_{k|k-1}^{\#s}, S_k^{\#s}) & \text{si } j^l \neq 0 \\ \frac{1}{\sqrt{m_k}} & \text{si } j^l = 0 \end{cases} \quad (4.26)$$

et

$$p(\Theta_k = \theta_k^{\#l} | Y_{1:k-1}, \Theta_{1:k-1} = \theta_{1:k-1}^{\#s}, m_{1:k}) = p(\Theta_k = \theta_k^{\#l} | m_k) = \begin{cases} \frac{\frac{1}{m_k} P_g P_d}{D} & \text{si } j^l \neq 0 \\ \frac{(1-P_g P_d) \frac{\mu_f(m_k)}{\mu_f(m_k-1)}}{D} & \text{si } j^l = 0 \end{cases} \quad (4.27)$$

avec au dénominateur

$$D = P_g P_d + (1 - P_g P_d) \frac{\mu_f(m_k)}{\mu_f(m_k - 1)}. \quad (4.28)$$

où μ_f est la loi de densité des fausses alarmes, généralement une loi de Poisson de paramètre λV . L'algorithme complet est résumé dans l'algorithme 4.1.

Théoriquement ce filtre est optimal, cependant la croissance exponentielle du nombre $\rho_{1:k}$ de réalisations possibles du vecteur d'association $\Theta_{1:k}$ va de pair avec la croissance exponentielle de la mémoire nécessaire ce qui empêche son utilisation pratique. Il était cependant intéressant de le présenter pour deux raisons.

Tout d'abord, le filtre à hypothèses multiples, dénommé multiple hypothesis tracker et dénoté MHT en anglais, est dans une certaine mesure une extension de ce filtre, comme nous allons le voir dans la section suivante. D'autre part, la connaissance du filtre optimal permet de mieux apprécier les approximations commises dans le filtre à association probabiliste, probabilistic data association filter dénoté PDAF en anglais, qui en est l'approximation la plus courante et que nous exposons maintenant.

4.2.3.2 Filtre à association de données probabilistes (PDAF)

Ce filtre est une approximation du filtre bayésien optimal ne tenant pas compte des associations passées mais uniquement du vecteur d'association courant [10], [13]. Il constitue une extension du filtre de Kalman dans lequel l'innovation classique est remplacée par une combinaison d'innovations associées à chaque hypothèse d'association et pondérées par leur probabilité d'occurrence. La loi $L_{k|k}$ est décomposée selon le théorème des probabilités totales sur les réalisations possibles de Θ_k et non de $\Theta_{1:k}$. Le nombre de cibles étant restreint à un, nous avons deux situations : soit toutes les mesures sont des fausses alarmes, i.e., $\theta_k^j = 0 \forall j = 1, \dots, m_k$, soit la mesure d'indice j provient de la cible, i.e., $\theta_k^j = 1$:

Algorithme 4.1 : Filtre bayésien optimal pour filtrage linéaire gaussien dans un environnement bruité.

$$\text{- Initialisation : } \begin{cases} \bar{X}_0 = \mathbb{E}(X_0); \\ P_0 = \mathbb{E}((X_0 - \bar{X}_0)(X_0 - \bar{X}_0)^T). \end{cases}$$

- Pour $k = 1, \dots, \text{fin}$:

- Prédiction :

$$\text{Pour } s = 1, \dots, \rho_{1:k-1} \begin{cases} p(X_k | \theta_{1:k-1}^{\#s}, Y_{1:k-1}) = N(X_k; \bar{X}_{k|k-1}^{\#s}, P_{k|k-1}^{\#s}); \\ \bar{X}_{k|k-1}^{\#s} = F_k \bar{X}_{k-1|k-1}^{\#s}; \\ P_{k|k-1}^{\#s} = F_k P_{k-1|k-1}^{\#s} F_k^T + \Sigma_V; \\ y_{k|k-1}^{\#s} = H_k \bar{X}_{k|k-1}^{\#s}. \end{cases}$$

- Association:

- Sélection des m_k^v mesures validées de volume V_k^v .

- Calcul des probabilités d'association :

Pour $l = 1, \dots, \rho_k$ et $y_{1:k}^{\#l} = (y_{1:k-1}^{\#s}, y_k^{j^l})$:

$$\beta_{1:k}^l = \mathbb{P}(\Theta_{1:k} = \theta_{1:k}^{\#l} | Y_{1:k}, m_{1:k}) = \begin{cases} \frac{1}{m_k^v} P_d N(y_k^{j^l}; y_{k|k-1}^{\#s}, S_k^{\#s}) \beta_{1:k-1}^s & \text{si } j^l = 1, \dots, m_k^v; \\ \frac{1}{V_k^v} (1 - P_g P_d) \frac{\mu_f(V_k^v)}{\mu_f(V_k^v - 1)} \beta_{1:k-1}^s & \text{sinon.} \end{cases}$$

- Mise à jour :

Pour $l = 1, \dots, \rho_k$ et $y_{1:k}^{\#l} = (y_{1:k-1}^{\#s}, y_k^{j^l})$:

$$\begin{cases} p(X_k | \Theta_{1:k}^{\#l}, Y_{1:k}) = N(X_k; \bar{X}_{k|k}^{\#l}, P_{k|k}^{\#l}); \\ K_k^l = P_{k|k-1}^{\#s} H_k^T (H_k P_{k|k-1}^{\#s} H_k^T + \Sigma_W)^{-1}; \\ \bar{X}_{k|k}^{\#l} = \bar{X}_{k|k-1}^{\#s} + K_k^l (y_k^{j^l} - y_{k|k-1}^{\#s}); \\ P_{k|k}^{\#l} = (I - K_k^l H_k) P_{k|k-1}^{\#s}. \end{cases}$$

$$\bar{X}_{k|k} = \sum_{l=1}^{\rho_{1:k}} \beta_{1:k}^l \bar{X}_{k|k}^{\#l}$$

$$P_{k|k} = \sum_{l=1}^{\rho_{1:k}} \beta_{1:k}^l P_{k|k}^{\#l} + \sum_{l=1}^{\rho_{1:k}} \beta_{1:k}^l \bar{X}_{k|k}^{\#l} \bar{X}_{k|k}^{\#l T} - \bar{X}_{k|k} \bar{X}_{k|k}^T.$$

$$\begin{aligned} p(X_k | Y_{1:k}) &= p(X_k | Y_{1:k}, m_{1:k}) = \sum_{\Theta_k} p(X_k | Y_{1:k}, \Theta_k, m_{1:k}) \mathbb{P}(\Theta_k | Y_{1:k}, m_{1:k}) \\ &= \sum_{j=1}^{m_k} p(X_k | Y_{1:k-1}, y_k^j) \underbrace{\mathbb{P}(\Theta_k^j = 1 | Y_{1:k}, m_{1:k})}_{\triangleq \beta_k^j} \\ &\quad + p(X_k | Y_{1:k-1}) \underbrace{\mathbb{P}(\Theta_k^j = 0 \forall j = 1, \dots, m_k | Y_{1:k}, m_{1:k})}_{\triangleq \beta_k^0} \end{aligned}$$

$$= \sum_{j=1}^{m_k} \frac{p(y_k^j | X_k, Y_{1:k-1}) p(X_k | Y_{1:k-1})}{p(y_k^j | Y_{1:k-1})} \beta_k^j + p(X_k | Y_{1:k-1}) \beta_k^0. \quad (4.29)$$

On commet alors l'approximation fondamentale suivante:

$$p(X_k | Y_{1:k-1}) = N(X_k; \bar{X}_{k|k-1}, P_{k|k-1}). \quad (4.30)$$

On considère ainsi que la loi prédite $p(X_k | Y_{1:k-1})$ est une gaussienne alors qu'elle est en réalité égale à une somme de gaussiennes. Chaque gaussienne correspond à une réalisation des associations passées comme cela a été vu au paragraphe précédent via l'équation (4.19). La moyenne et la covariance de $L_{k|k}$ se calculent récursivement sous cette hypothèse selon les formules suivantes:

$$\begin{aligned} \bar{X}_{k|k} &= \sum_{j=1}^{m_k} \left(\bar{X}_{k|k-1} + K_k (y_k^j - H_k \bar{X}_{k|k-1}) \right) \beta_k^j + \bar{X}_{k|k-1} \beta_k^0 \\ &= \bar{X}_{k|k-1} \underbrace{\sum_{j=0}^{m_k} \beta_k^j}_{=1} + K_k \underbrace{\sum_{j=1}^{m_k} \beta_k^j \tilde{y}_k^j}_{=\tilde{y}_k} \end{aligned} \quad (4.31)$$

où on rappelle que $\tilde{y}_k^j = y_k^j - \bar{y}_{k|k-1}$ est l'innovation. La mise à jour de la covariance se fait selon l'équation (non directe) suivante :

$$P_{k|k} = \beta_k^0 P_{k|k-1} + (1 - \beta_k^0) P_k^c + \tilde{P}_k \quad (4.32)$$

avec

$$P_k^c = (I_d - K_k H_k) P_{k|k-1}; \quad (4.33)$$

$$\tilde{P}_k = K_k \left(\sum_{j=1}^{m_k} \beta_k^j (\tilde{y}_k^j)^T \tilde{y}_k^j - \tilde{y}_k^T \tilde{y}_k \right) K_k^T. \quad (4.34)$$

Les probabilités d'association β_k^j s'obtiennent à nouveau en utilisant la règle de Bayes :

$$\begin{cases} \beta_k^j = \frac{1}{c} p(Y_k | \theta_k^j = 1, m_k, Y_{1:k-1}) \mathbb{P}(\theta_k^j = 1 | m_k, Y_{1:k-1}) & \text{pour } j = 1, \dots, m_k; \\ \beta_k^0 = \frac{1}{c} p(Y_k | \theta_k^j = 0 \forall j, m_k, Y_{1:k-1}) \mathbb{P}(\theta_k^j = 0 \forall j | m_k, Y_{1:k-1}). \end{cases} \quad (4.35)$$

où c est une constante de normalisation telle que la somme $\sum_{j=0}^{m_k} \beta_k^j$ vaille un. Sous l'hypothèse (4.30), on suppose que les mesures associées à X_k sont distribuées selon une gaussienne autour de la mesure prédite $y_{k|k-1}$ et de covariance S_k . On obtient ainsi:

$$\begin{cases} p(Y_k | \Theta_k^j = 1, Y_{1:k-1}, m_k) = \frac{1}{\sqrt{m_k-1}} \frac{1}{P_g} N(\tilde{y}_k^j; 0, S_k) \text{ pour } j = 1, \dots, m_k; \\ p(Y_k | \Theta_k^j = 0 \forall j, Y_{1:k-1}, m_k) = \frac{1}{\sqrt{m_k}}. \end{cases} \quad (4.36)$$

et

$$\begin{cases} \mathbb{P}(\Theta_k^j = 1 | Y_{1:k-1}, m_k) = \mathbb{P}(\Theta_k^j = 1 | m_k) = \frac{1}{m_k} \frac{P_g P_d}{D} \text{ pour } j = 1, \dots, m_k; \\ \mathbb{P}(\Theta_k^j = 0 \forall j | Y_{1:k-1}, m_k) = \frac{(1 - P_g P_d) \frac{\mu_f(m_k)}{\mu_f(m_k-1)}}{D}. \end{cases} \quad (4.37)$$

avec au dénominateur :

$$D = P_g P_d + (1 - P_g P_d) \frac{\mu_f(m_k)}{\mu_f(m_k - 1)}. \quad (4.38)$$

L'algorithme complet est résumé dans l'algorithme 4.2. L'approximation (4.30) permet d'assimiler la loi prédite $p(X_k | Y_{1:k-1})$ à une loi gaussienne. L'information du passé est donc résumée par sa moyenne et covariance prédites et mises à jour par un unique filtre de Kalman dont l'innovation est une somme pondérée des innovations selon l'association courante. La complexité du PDAF est donc similaire à celle du filtre de Kalman. L'algorithme bayésien optimal met en œuvre quant à lui autant de filtres de Kalman que de réalisations possibles du vecteur d'association $\Theta_{1:k}$ et les combine entre-eux proportionnellement à la probabilité de chaque association. Dans le cas où la somme de gaussiennes mises en œuvre dans le FBO contient des composantes de moyennes très distinctes, la structure de cette somme est détruite par l'hypothèse (4.30) du PDAF. Pour y remédier, des filtres intermédiaires entre le PDAF et le FBO ont été proposés, que nous nous contenterons de citer:

- Les filtres à mémoire de taille k_m qui sont construits sur le même principe que le FBO en ne tenant compte cependant que des réalisations de $\Theta_{k-k_m:k}$ au lieu de celles de $\Theta_{1:k}$ [12].
- Des techniques de réduction de mélange, qui permettent de réduire le nombre de composantes issues d'un FBO ou d'un filtre à mémoire de taille m_k . Dans [28], la réduction est effectuée soit en fusionnant les couples de composantes les plus similaires, soit en assimilant aux composantes de plus forte probabilité leurs composantes voisines.

Algorithme 4.2 : Filtre à association de données probabilistes pour filtrage linéaire gaussien dans un environnement bruité.

- Initialisation :
$$\begin{cases} \bar{X}_0 = \mathbb{E}(X_0); \\ P_0 = \mathbb{E}((X_0 - \bar{X}_0)(X_0 - \bar{X}_0)^T). \end{cases}$$
- Pour $k = 1, \dots, \text{fin}$:
 - Prédiction :
$$\begin{cases} p(X_k | Y_{1:k-1}) \approx N(X_k; \bar{X}_{k|k-1}, P_{k|k-1}); \\ \bar{X}_{k|k-1} = F_k \bar{X}_{k-1|k-1}; \\ P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + \Sigma_V; \\ \bar{y}_{k|k-1} = H_k \bar{X}_{k|k-1}. \end{cases}$$
 - Association:
 - Sélection des m_k^v mesures validées de volume V_k^v .
 - Calcul des probabilités d'association :

$$\begin{cases} \beta_k^j = \mathbb{P}(\Theta_k^j = 1 | Y_{1:k}, m_{1:k}) = \frac{1}{m_k^v} P_g N(y_k^j; \bar{y}_{k|k-1}, S_k) \quad \text{pour } j = 1, \dots, m_k^v; \\ \beta_k^0 = \frac{1}{V_k^v} (1 - P_g P_d) \frac{\mu_f(V_k^v)}{\mu_f(V_k^v - 1)}. \end{cases}$$
 - Mise à jour :

$$\tilde{y}_k^j = y_k^j - \bar{y}_{k|k-1} \quad \text{pour } j = 1, \dots, m_k^v.$$

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + \Sigma_W)^{-1};$$

$$\bar{X}_{k|k} = \bar{X}_{k|k-1} + K_k \sum_{j=1}^{m_k^v} \beta_k^j \tilde{y}_k^j;$$

$$P_k^c = (I_d - K_k H_k) P_{k|k-1};$$

$$\tilde{P}_k = K_k \left(\sum_{j=1}^{m_k^v} \beta_k^j (\tilde{y}_k^j)^T \tilde{y}_k^j - \tilde{y}_k^T \tilde{y}_k \right) K_k^T;$$

$$P_{k|k} = \beta_k^0 P_{k|k-1} + (1 - \beta_k^0) P_k^c + \tilde{P}_k.$$

Nous allons maintenant exposer les algorithmes traitant du filtrage de plusieurs objets dans un environnement bruité, i.e., nous devons faire face à une double incertitude sur l'origine des mesures.

4.2.4 Filtrage multi-cibles dans un environnement bruité

4.2.4.1 Filtre joint à association de données probabilistes (JPDAF)

Nous supposons dans ce paragraphe que le nombre de cibles M est constant et connu. Les composantes du vecteur d'association sont toujours supposées indépendantes selon

(H2). Une première solution au filtrage multi-cibles consiste à mener un PDAF pour chaque cible. En pratique, les filtres se reportent alors en fait tous sur la même trajectoire dès que l'ambiguïté sur l'origine des mesures est trop importante. Le JPDAF est une extension du PDAF pour le filtrage multi-cibles intégrant un principe d'exclusion, qui rend les estimations des différentes cibles dépendantes via le vecteur d'association Θ_k . On décompose la loi $L_k^i|k$ selon le théorème des probabilités totales sur les réalisations possibles de Θ_k [83]:

$$p(X_k^i | Y_{1:k}) = \sum_{\Theta_k} p(X_k^i | \Theta_k, Y_{1:k}) \mathbb{P}(\Theta_k | Y_{1:k}) \quad (4.39)$$

L'énumération des réalisations est maintenant plus complexe avec plusieurs cibles qu'avec une seule. On regroupe d'une part celles qui associent une mesure à la cible i que l'on notera $\Theta_k \supset i$ et d'autre part celles qui n'associent aucune mesure à la cible i auquel cas on propage simplement l'état prédit :

$$p(X_k^i | Y_{1:k}) = \sum_{\Theta_k \supset i} p(X_k^i | \Theta_k, Y_{1:k}) \mathbb{P}(\Theta_k^j = i | Y_{1:k}) \\ + p(X_k^i | \Theta_k^j \neq i \forall j, Y_{1:k-1}) \mathbb{P}(\Theta_k^j \neq i \forall j | Y_{1:k-1}) \quad (4.40)$$

Étant donné l'hypothèse (H2), une mesure au plus peut être associée à la cible i . Par conséquent la première somme de l'égalité précédente peut se réécrire comme une somme sur l'indice de mesure.

$$p(X_k^i | Y_{1:k}) = \sum_{j=1}^{m_k} p(X_k^i | \Theta_k^j = i, Y_{1:k}) \mathbb{P}(\Theta_k / \Theta_k^j = i | Y_{1:k}) \\ + p(X_k^i | \Theta_k^j \neq i \forall j, Y_{1:k-1}) \mathbb{P}(\Theta_k^j \neq i \forall j | Y_{1:k-1}) \quad (4.41)$$

L'hypothèse approximative (4.30) est à nouveau faite, supposant que la loi prédite de chaque cible est distribuée selon une gaussienne. L'estimation de chaque état se fait selon un filtre de Kalman comme dans le PDAF. La seule différence réside dans l'évaluation des probabilités des associations. Comme il existe plusieurs associations vérifiant la contrainte $\Theta_k^j = i$, le calcul de la probabilité $\mathbb{P}(\Theta_k^j = i | Y_{1:k})$ nécessite l'énumération exhaustive de toutes ces associations.

$$\mathbb{P}(\Theta_k^j = i | Y_{1:k}) = \sum_{\Theta_k \supset i} \mathbb{P}(\Theta_k | Y_{1:k})$$

$$= \sum_{\theta_k^1} \dots \sum_{\theta_k^{j-1}} \sum_{\theta_k^{j+1}} \dots \sum_{\theta_k^{m_k}} \mathbb{P}(\theta_k^1, \dots, \theta_k^j = i, \dots, \theta_k^{m_k} | Y_{1:k}). \quad (4.42)$$

La probabilité d'une association se calcule comme précédemment à l'aide de la règle de Bayes:

$$\mathbb{P}(\theta_k | Y_{1:k}) = \frac{1}{c} p(Y_k | \theta_k, Y_{1:k-1}) p(\theta_k | Y_{1:k-1}) \quad (4.43)$$

Du fait de l'indépendance conditionnelle des mesures, on peut écrire:

$$p(Y_k | \theta_k, Y_{1:k-1}) = \prod_{j=1}^{m_k} p(Y_k^j | \theta_k^j, Y_{1:k-1}) \quad (4.44)$$

avec

$$p(Y_k^j | \theta_k^j = i, Y_{1:k-1}) = \begin{cases} N(Y_k^j; y_{k|k-1}^i, S_k^i) & \text{si } \theta_k^j \neq 0; \\ \frac{1}{V} & \text{sinon.} \end{cases} \quad (4.45)$$

où $y_{k|k-1}^i \triangleq H_k^i X_{k|k-1}^i$ est la mesure prédite associée à la cible i selon l'équation de mesure du système (4.9). On paramètre la probabilité des fausses alarmes par V_k le volume total et non pas par les volumes $V_k^{i,v}$ qui sont variables d'une cible à l'autre.

D'autre part, la probabilité a priori d'une association dépend du nombre de mesures m_k , de la probabilité de détection P_d , du nombre de fausses alarmes m_f et de la densité μ_f des fausses alarmes :

$$p(\theta_k | Y_{1:k-1}) = p(\theta_k) = \frac{m_f!}{m_k!} \mu_f(m_f) \prod_{i=1}^M P_d^{D^{c_k}(i)} (1 - P_d)^{1 - D^{c_k}(i)} \quad (4.46)$$

Où $D^{c_k}(i) = \{1, 0\}$ indique si la cible i est détectée ou pas.

L'estimation de la loi $L_{k|k}^i$ par le JPDAF est donc identique à celle proposée par PDAF hormis le calcul des probabilités d'associations qui nécessite l'énumération des associations.

Un inconvénient du JPDAF est que le nombre de cibles suivies est supposé connu et constant, or il peut être intéressant de pouvoir initialiser de nouvelles pistes au cours du scénario. C'est ce que propose le filtre à hypothèses multiples, dénommé multiple hypothesis tracker et dénoté MHT.

4.2.4.2 Filtre à hypothèses multiples (MHT)

Le nombre de cibles est supposé supérieur à un, mais non connu a priori et pouvant croître

au cours du temps. L'hypothèse d'association (H1) signifie donc qu'une mesure peut provenir d'une cible existante ou d'une nouvelle cible, ou être une fausse alarme. Comme le filtre bayésien optimal exposé en début de chapitre, le MHT décompose l'estimation de la loi $L_{k|k}$ sur l'ensemble des réalisations possibles de $\Theta_{1:k}$. La possibilité d'apparition d'une nouvelle cible à chaque instant est certes très intéressante d'un point de vue applicatif mais, cumulé à la prise en compte de tout l'historique des associations, le nombre de ces réalisations croît exponentiellement et rend l'algorithme inutilisable tel qu'il est. La probabilité de chaque association se calcule toujours en utilisant la règle de Bayes [83]:

$$\beta_{1:k}^l = \frac{1}{c} p(Y_k | \Theta_{1:k} = \theta_{1:k}^{\#l}, Y_{1:k-1}, m_{1:k}) p(\Theta_k = \theta_k^{\#l} | Y_{1:k-1}, \Theta_{1:k-1} = \theta_{1:k-1}^{\#s}, m_{1:k}) \beta_{1:k-1}^s \quad (4.47)$$

On utilise à nouveau l'indépendance conditionnelle des associations pour calculer $p(Y_k | \Theta_{1:k} = \theta_{1:k}^{\#l}, Y_{1:k-1}, m_{1:k})$ comme dans le FBO en ajoutant toutefois la loi de densité des mesures associées à de nouvelles sources. Le second facteur $p(\Theta_k = \theta_k^{\#l} | Y_{1:k-1}, \Theta_{1:k-1} = \theta_{1:k-1}^{\#s}, m_{1:k})$ s'écrit également comme pour le FBO en intégrant une loi de densité sur le nombre de mesures issues de nouvelles cibles. On trouvera les détails de ces calculs dans [15] et [27].

Le nombre d'associations à énumérer étant trop élevé, il est nécessaire d'insérer des techniques heuristiques d'élagage des branches de l'arbre dont les probabilités sont trop faibles. Le nombre de branches supprimées doit être assez élevé pour que la complexité de l'algorithme soit réduite. Toutefois, il peut être fatal d'éliminer trop rapidement une hypothèse de probabilité faible à un instant donné qui se serait révélée intéressante après plusieurs pas de temps. On peut alors mettre en œuvre le principe de mémoire limitée. Sans rentrer dans les variations et optimisations nombreuses de cet algorithme, nous renvoyons à [15] pour plus de détails.

Nous venons de présenter les principaux algorithmes déterministes prenant en compte l'incertitude sur l'origine des mesures sous une hypothèse de dépendance des associations. Le filtre bayésien optimal et le PDAF traitent uniquement l'incertitude due aux fausses alarmes. Le JPDAF et le MHT traitent quant à eux également l'incertitude due à la présence de plusieurs objets. Tous ces algorithmes s'appliquent dans le cas particulier de modèles d'état et de mesure linéaires avec des bruits additifs gaussiens. L'hypothèse (H2)

de dépendance des associations implique l'énumération exhaustive des associations. Cette contrainte est plus ou moins forte selon la prise en compte du passé qui est moindre dans le (J)PDAF que dans le FBO et MHT.

- Tout d'abord, la technique de validation des mesures a permis de réduire le nombre de mesures prises en compte et par conséquent le nombre d'associations possibles entre les mesures et les cibles. L'estimation de la loi conditionnelle à toutes les mesures $L_{k|k}^i = p(X_k^i | Y_{1:k})$ est en fait approximée par celle de la loi conditionnelle aux mesures validées $p(X_k^i | Y_{1:k}^v)$.
- Dans le PDAF et son extension le JPDAF, la loi prédite $L_{k|k-1}$ est approximée par une gaussienne alors que dans la version bayésienne optimale il s'agit en réalité d'une somme pondérée de gaussiennes sur toutes les réalisations possibles des associations. Cette approximation permet de s'affranchir de l'énumération exhaustive des réalisations du vecteur d'association $\Theta_{1:k-1}$.
- Dans le MHT, un élagage de certaines branches de l'arbre des hypothèses est nécessaire pour diminuer le nombre de branches qui croît exponentiellement avec le temps.

Ces approximations cherchent toutes à lutter contre le nombre trop important d'associations à énumérer. Or, comme on va le voir maintenant, cette énumération n'a pas lieu d'être si l'on relâche l'hypothèse d'association (H2) au profit d'une hypothèse d'indépendance des associations que nous dénommerons (H3).

4.3 Filtrage et association de données avec indépendance des associations

On reprend un système dynamique général pour lequel les cibles évoluent indépendamment, défini par les équations (4.2) et (4.4) :

$$\begin{cases} X_k^i = F_k^i(X_{k-1}^i, V_k^i) & \forall i = 1, \dots, M. \\ Y_k^j = H_k^i(X_k^i, W_k^i) & \text{si } \Theta_k^j = i \end{cases} \quad (4.48)$$

où $(V_k^i)_{k \in \mathbb{N}}$ pour $i = 1, \dots, M$ et $(W_k^j)_{k \in \mathbb{N}}$ pour $j = 1, \dots, m_k$ sont des bruits blancs gaussiens indépendants.

L'originalité de l'algorithme du PMHT que nous présentons maintenant est de relâcher l'hypothèse de dépendance des composantes du vecteur d'association en remplaçant l'hypothèse (H2) par l'hypothèse (H3) :

(H3) A un instant donné, une cible peut être à l'origine d'aucune ou de plusieurs mesures.

L'association peut donc se faire indépendamment d'une mesure à l'autre, ce qui se traduit par l'indépendance des composantes du vecteur Θ_k . La loi a priori du vecteur d'association s'écrit donc:

$$p(\Theta_k) = \prod_{j=1}^{m_k} p(\Theta_k^j). \quad (4.49)$$

L'hypothèse (H3) est souvent remise en cause dans les applications en traitement du signal. Le processus physique de génération des mesures ne permet en effet pas qu'une cible soit à l'origine de plusieurs mesures. Cependant, si l'on se tourne vers des applications en analyse d'images, il n'y a rien d'artificiel à considérer qu'un objet puisse produire plusieurs mesures, si les mesures sont des points caractéristiques de l'image vérifiant certaines propriétés de luminance ou de mouvement. Nous allons voir de plus dans le paragraphe suivant que le filtre probabiliste à hypothèse multiples, dénommé probabilistic multiple hypothesis tracker et dénoté PMHT en anglais, est optimal une fois cette hypothèse admise.

4.3.1 Filtre probabiliste à hypothèses multiples (PMHT)

Contrairement aux algorithmes précédents tels que le (J)PDAF et le MHT, le PMHT considère les vecteurs d'association comme des vecteurs aléatoires à estimer, en plus des vecteurs d'états. Ainsi, les mesures ne sont plus associées "en dur" à des cibles mais sont associées à tous les cibles selon les probabilités de chaque association. Le PMHT a été initialement proposé dans [43]. La présentation qui suit s'inspire des travaux [53] et [54]. Une autre différence essentielle avec les algorithmes de filtrage précédents est que le PMHT n'est pas un algorithme de filtrage mais de lissage. Il ne fournit en effet pas d'estimation de la loi $L_{k|k}$ à chaque instant mais procède, une fois que toutes les mesures $y_{1:T}$ sont reçues, à l'estimation de $X_{0:T}$. Dans le domaine de la poursuite de cibles, on parle alors d'algorithme de type "batch".

Afin d'alléger les notations, les lettres capitales $\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}$ et $\boldsymbol{\Pi}$ désigneront dans ce paragraphe les ensembles respectifs de vecteurs (Y_1, \dots, Y_T) , (X_0, \dots, X_T) , $(\Theta_1, \dots, \Theta_T)$ et (Π_1, \dots, Π_T) . On rappelle que la composante Π_k^i de Π_k est la variable aléatoire qui représente la loi de probabilité du modèle i , i.e., la probabilité qu'une mesure quelconque

soit associée à la cible i si $i = 1, \dots, M$ ou au modèle des fausses alarmes si $i = 0$. Le point de vue adopté dans [53] et [54] consiste alors à voir la loi de vraisemblance des mesures conditionnellement aux états et aux probabilités de chaque modèle comme une loi de mélange en la décomposant de la façon suivante :

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \mathbf{\Pi}) &= \prod_{k=1}^T p(Y_k|X_k, \Pi_k) = \prod_{k=1}^T \sum_{\Theta_k} p(Y_k|X_k, \Pi_k, \Theta_k) p(\Theta_k|X_k, \Pi_k) \\ &= \prod_{k=1}^T \sum_{\Theta_k} p(Y_k|X_k, \Pi_k, \Theta_k) p(\Theta_k) \end{aligned} \quad (4.50)$$

où la notion \sum_{Θ_k} , désigne la somme $\sum_{\Theta_k^1=0}^M \dots \sum_{\Theta_k^{m_k}=0}^M$. Le passage de la première à la seconde ligne est obtenue en supposant l'indépendance de Θ_k et X_k , hypothèse que nous n'avons pas émise jusque-là.

Par indépendance conditionnelle des mesures d'une part et surtout par indépendance des composantes du vecteur Θ_k d'après (4.49), on peut ensuite décomposer ce produit sous la forme :

$$p(\mathbf{y}|\mathbf{x}, \mathbf{\Pi}) = \prod_{k=1}^T \sum_{\Theta_k} \prod_{j=1}^{m_k} p(Y_k^j|X_k, \Pi_k, \Theta_k) p(\Theta_k^j). \quad (4.51)$$

On évite ainsi l'énumération exhaustive des associations. Chaque composante Θ_k^j décrivant le même ensemble $0, \dots, M$, on peut factoriser la somme sur les associations et obtenir alors l'écriture :

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \mathbf{\Pi}) &= \prod_{k=1}^T \prod_{j=1}^{m_k} \sum_{\Theta_k^j=0}^M p(Y_k^j|X_k, \Pi_k, \Theta_k) p(\Theta_k^j) \\ &= \prod_{k=1}^T \prod_{j=1}^{m_k} \left(\pi_k^0 \mu_f(Y_k^j) + \sum_{i=1}^M p(Y_k^j|X_k^i, \Theta_k^j = i) \pi_k^i \right) \end{aligned} \quad (4.52)$$

où μ_f désigne la loi de densité des fausses alarmes. La dernière expression pour la vraisemblance dans (4.52) est bien celle d'une loi de mélange de densités. Remarquons que cette forme de loi de mélange ne peut être atteinte si l'on ne suppose pas l'indépendance de Θ_k et X_k . On recherche les valeurs optimales de X qui la maximisent. Cependant, ne connaissant pas les paramètres de pondération $\left((\pi_k^i)_{i=0, \dots, M} \right)_{k=1, \dots, T}$, cette optimisation ne

peut se faire directement. Le principe du PMHT consiste alors à utiliser l'algorithme d'espérance-maximisation, dénommé expectation-maximisation en anglais et dénoté EM, pour estimer le paramètre $O = \left(\pi_k^0, (X_k^i, \pi_k^i)_{i=1, \dots, M} \right)_{k=0, \dots, T}$ qui maximise cette vraisemblance. L'algorithme EM a été initialement présenté dans [14] pour les problèmes aux données incomplètes. Avant de détailler l'application de l'algorithme EM dans le cadre du PMHT, nous en rappelons le principe :

4.3.1.1 Algorithme d'espérance-maximisation (EM)

Soit Y les mesures supposées être des données incomplètes et dont la loi dépend du paramètre inconnu O . Les données manquantes sont désignées par θ . On note alors O^* l'estimateur du maximum de vraisemblance qui maximise la log-vraisemblance des données incomplètes :

$$O^* \triangleq \arg \max_O LV(O) = \arg \max_O \log p(Y|O) \quad (4.53)$$

Du fait que les données θ soient cachées, cet estimateur est rarement accessible directement et l'on manipulera plutôt la log-vraisemblance des données complètes:

$$LV_c(O) = \log p(Y, \theta|O). \quad (4.54)$$

Le principe de l'algorithme EM consiste à estimer le maximum de vraisemblance associé à LV_c en maximisant itérativement l'espérance conditionnelle:

$$Q(O|O') \triangleq \mathbb{E}(\log p(Y, \theta|O) | Y, O') \quad (4.55)$$

Une valeur initiale O_0 est tout d'abord donnée au paramètre O . Supposons qu'après l'itération τ , nous disposons de l'estimée O_τ . On calcule alors l'estimée $O_{\tau+1}$ selon le schéma suivant en deux étapes :

- A l'étape E, on calcule l'espérance :

$$Q(O|O_\tau) = \mathbb{E}(\log p(Y, \theta|O) | Y, O_\tau) \quad (4.56)$$

- A l'étape M, on calcule la nouvelle estimée $O_{\tau+1}$ qui maximise l'espérance calculée à l'étape E précédente:

$$O_{\tau+1} \triangleq \arg \max_O Q(O|O_\tau) \quad (4.57)$$

En utilisant l'inégalité de Jensen, on peut montrer que:

- l'estimateur du maximum de la vraisemblance des données incomplètes $LV(O)$ est un point fixe de la fonction $O' \rightarrow \arg \max_O Q(O|O')$.
- la log-vraisemblance $LV(O_\tau)$ croît avec τ , ce qui assure la convergence de la

suite $(O_\tau)_{\tau \in \mathbb{N}}$ vers des points stationnaires.

Les travaux de [18] montrent que l'estimateur fourni par l'algorithme EM converge vers un maximum local de la vraisemblance des données incomplètes, sous des conditions suffisantes de régularité. Cependant, il existe plusieurs limitations à cet algorithme:

- chacune des deux étapes peut ne pas être calculable analytiquement;
- l'estimée finale peut dépendre fortement de l'initialisation;
- la convergence peut se révéler très lente;
- il peut y avoir stagnation des estimées autour de points de selle ou de plateau de vraisemblance.

Afin de contourner ces inconvénients, des extensions stochastiques de l'algorithme EM ont été proposées comme les algorithmes Monte-Carlo EM, dénotés MCEM, l'algorithme stochastique EM, dénoté SEM. Les MCEM [29], [30] utilisent des méthodes de Monte-Carlo pour approximer l'espérance calculée à l'étape E. Quant au SEM [19], il propose de simuler les données manquantes selon un échantillonneur de Gibbs utilisant leur loi conditionnelle aux observations. On pourra trouver une présentation de ces algorithmes stochastiques et une comparaison dans le cas d'une loi de mélange dans [48].

Un large domaine d'études et d'application de l'algorithme EM est celui de l'estimation des paramètres d'une loi de mélange. Les travaux de [20] en sont à l'origine.

D'autre part, l'algorithme EM peut également être utilisé pour maximiser non pas le critère du maximum de vraisemblance mais celui du maximum a posteriori. Il est montré dans [14] qu'il suffit pour cela d'ajouter la log-vraisemblance de la loi a priori, i.e., de remplacer le terme $Q(O|O')$ par $Q(O|O') + \log p(O)$. Le PMHT proposé dans [43] correspond à l'application de l'algorithme EM aux lois de mélange en ajoutant la dimension temporelle et en utilisant le critère du maximum a posteriori. On en détaille maintenant les équations.

Soit $\mathcal{O} = (\mathbf{X}, \mathbf{\Pi}) = \left(\pi_k^0, (X_k^i, \pi_k^i)_{i=1, \dots, M} \right)_{k=0, \dots, T}$ le vecteur à estimer est composé du processus caché et des lois de probabilités Π_k^i de chaque modèle. Le vecteur de mesures \mathbf{y} est le vecteur de données incomplètes à partir duquel on obtient le

vecteur de données complètes $(\mathbf{y}, \boldsymbol{\theta})$.

Etape E :

Soit \mathcal{O}_τ l'estimé de \mathcal{O} obtenu à l'itération τ . Des calculs mènent à l'expression suivante de l'espérance :

$$\begin{aligned} M(\mathcal{O}|\mathcal{O}_\tau) &= \mathbb{E}[\log p(\mathbf{y}, \boldsymbol{\theta}|\mathcal{O})|\mathbf{y}, \mathcal{O}_\tau] + \log p(\mathcal{O}) \\ &= \sum_{i=0}^M \sum_{k=1}^T \left(\sum_{j=1}^{m_k} \omega_{k,\tau+1}^{j,i} \right) \log \pi_k^i + \sum_{i=1}^M \sum_{k=1}^T \sum_{j=1}^{m_k} \log \left(p(y_k^j | x_k^i) \right) \omega_{k,\tau+1}^{j,i} \\ &\quad + \sum_{i=1}^M \log \left(p(x_0^i) \right) + \sum_{i=1}^M \sum_{k=1}^T \log \left(p(x_k^i | x_{k-1}^i) \right) \end{aligned} \quad (4.58)$$

où $\omega_{k,\tau+1}^{j,i}$ est un poids mesurant la probabilité a posteriori d'affecter la mesure j au modèle i :

$$\omega_{k,\tau+1}^{j,i} = p(\theta_k^j = i | Y_k^j, \mathcal{O}_{k,\tau}) \frac{\pi_{k,\tau}^i p(y_k^j | x_{k,\tau}^i)}{p(y_k^j | \mathcal{O}_{k,\tau})}. \quad (4.59)$$

Etape M :

Les paramètres $\boldsymbol{\Pi}$ et $\boldsymbol{\mathcal{X}}$ apparaissant dans des termes distincts de la somme obtenue à la dernière égalité dans (4.58), on peut simplifier considérablement la maximisation en la décomposant en deux maximisations indépendantes. La première s'effectue par rapport à $\boldsymbol{\Pi}$ et la seconde par rapport à $\boldsymbol{\mathcal{X}}$.

- Les termes $\log \pi_k^i$ apparaissent linéairement et la maximisation sous la contrainte $\sum_{i=0}^M \pi_k^i = 1$ pour tout $k = 1, \dots, T$ s'obtient pour :

$$\pi_{k,\tau+1}^i = \frac{1}{m_k} \sum_{j=1}^{m_k} \omega_{k,\tau+1}^{j,i}. \quad (4.60)$$

- La seconde maximisation revient à trouver, pour $i = 1, \dots, M$, les états (x_0^i, \dots, x_T^i) qui maximisent :

$$\sum_{k=1}^T \sum_{j=1}^{m_k} \log \left(p(y_k^j | x_k^i) \right) \omega_{k,\tau+1}^{j,i} + \log \left(p(x_0^i) \right) + \sum_{k=1}^T \log \left(p(x_k^i | x_{k-1}^i) \right) \quad (4.61)$$

La solution dépend alors du modèle dynamique de l'application. Si aucune solution analytique n'est disponible, on peut avoir recours à un algorithme numérique d'optimisation de type Gauss-Newton par exemple.

Dans le cas linéaire gaussien, il est plus aisé de rechercher le maximum de l'exponentielle de cette dernière expression :

$$p(x_0^i) \prod_{k=1}^T \left(p(x_k^i | x_{k-1}^i) \prod_{j=1}^{m_k} (p(y_k^j | x_k^i))^{\omega_{k,\tau+1}^{j,i}} \right). \quad (4.62)$$

En utilisant le caractère gaussien du bruit de mesures, on peut introduire une mesure fictive \tilde{y}_k^j et sa matrice de covariance $\tilde{\Sigma}_{W^i}$ définies ci-dessous pour faire apparaître la densité d'une loi gaussienne:

$$\begin{aligned} \prod_{j=1}^{m_k} p(y_k^j | x_k^i)^{\omega_{k,\tau+1}^{j,i}} &\propto \prod_{j=1}^{m_k} N(y_k^j; H_k^i x_k^i, (\omega_{k,\tau+1}^{j,i})^{-1} \Sigma_{W^i}) \\ &\propto \prod_{j=1}^{m_k} N(\tilde{y}_k^j; H_k^i x_k^i, (m_k \pi_{k,\tau+1}^i)^{-1} \Sigma_{W^i}) \end{aligned} \quad (4.63)$$

où

$$\tilde{y}_k^i = \frac{1}{m_k \pi_{k,\tau+1}^i} \sum_{j=1}^{m_k} \omega_{k,\tau+1}^{j,i} y_k^j \quad (4.64)$$

Il ne reste que d'appliquer le filtre de Kalman en utilisant le centroïde des mesures à l'instant courant \tilde{y}_k^i et de covariance $\tilde{\Sigma}_{W^i}$ définie par :

$$\tilde{\Sigma}_{W^i} = \frac{\Sigma_{W^i}}{m_k \pi_{k,\tau+1}^i}. \quad (4.63)$$

Les filtres peuvent donc être menés indépendamment pour chaque objet.

L'algorithme du PMHT est résumé dans l'algorithme 4.3 dans le cas de modèles d'état et de mesure linéaires avec des bruits gaussiens. Lorsque le modèle dynamique ne vérifie pas ces propriétés, l'étape de maximisation ne peut en général être résolue analytiquement. Cependant, l'algorithme du PMHT est toujours applicable en résolvant cette étape par des algorithmes numériques adéquats. Il peut de plus être facilement étendu à la présence de récepteurs de mesures multiples. En pratique, la mise en œuvre du PMHT peut cependant poser plusieurs problèmes, parmi lesquels :

- L'initialisation de l'algorithme nécessite d'affecter une valeur initiale aux vecteurs de paramètres \mathcal{X} et Π , i.e., pour l'ensemble des instants de 0 à T , ce qui est beaucoup plus complexe que d'affecter une valeur à l'instant initial comme dans les algorithmes précédents de filtrage. Pour initialiser \mathcal{X} , on peut choisir une valeur à

l'instant initial et la propager selon la loi d'évolution mais sans tenir compte des mesures. Cependant, si la vraie trajectoire est très différente de la trajectoire ainsi prédite, le PMHT risque de converger seulement vers un maximum local. Il s'avère malheureusement que le PMHT est très sensible à la qualité de l'initialisation.

- Le critère d'arrêt de l'algorithme EM doit également être fixé. On peut considérer un nombre fixe d'itérations, de l'ordre d'une dizaine en général, ou utiliser un critère d'arrêt mesurant la différence entre deux estimées successives.

Malgré son élégance théorique, obtenue grâce à l'hypothèse d'indépendance des associations, les performances du PMHT sont parfois décevantes comparées à celle du JPDAF. Plusieurs comportements du PMHT peuvent en effet mettre en péril l'estimation :

- Tout d'abord, l'algorithme n'adapte pas sa zone de recherche en fonction des données et de la qualité de l'estimation.
- Si l'algorithme diverge de la vraie trajectoire, il peut se reporter sur une fausse alarme pour améliorer son estimation sans réaliser la divergence.
- La contrepartie de l'hypothèse (H3) est que l'algorithme accepte autant de mesures qu'il en existe et les intègre dans l'estimation même si un nombre d'entre-elles sont des fausses alarmes.

Algorithme 4.3 : Algorithme PMHT pour filtrage linéaire gaussien dans un environnement bruité.

$$- \text{Initialisation : } \begin{cases} \mathcal{O}_0 = (\mathbf{X}_0, \mathbf{\Pi}_0) \\ \text{avec } \begin{cases} \mathbf{X}_0 = (X_{0,0}, \dots, X_{T,0}) \text{ et } X_{k,0} = (X_{k,0}^1, \dots, X_{k,0}^M); \\ \mathbf{\Pi}_0 = (\Pi_{0,0}, \dots, \Pi_{T,0}) \text{ et } \Pi_{k,0} = (\Pi_{k,0}^1, \dots, \Pi_{k,0}^M). \end{cases} \end{cases}$$

- Pour $\tau = 1, \dots, \tau_{fin}$:

- Mise à jour de $\mathbf{\Pi}$: calculer pour tout $k = 1, \dots, T$ et tout $i = 1, \dots, M$:

$$\pi_{k,\tau+1}^i = \frac{1}{m_k} \sum_{j=1}^{m_k} \omega_{k,\tau+1}^{j,i} \text{ ou } \omega_{k,\tau+1}^{j,i} = \frac{\pi_{k,\tau}^i p(y_k^j | x_{k,\tau}^i)}{p(y_k^j | \mathcal{O}_{k,\tau})}.$$

- Mise à jour de \mathbf{X} :

- Calcul du centroïde des mesures et de la matrice de covariance associée pour tout $k = 1, \dots, T$ et tout $i = 1, \dots, M$:

$$\begin{cases} \tilde{y}_k^i = \frac{1}{m_k \pi_{k,\tau+1}^i} \sum_{j=1}^{m_k} \omega_{k,\tau+1}^{j,i} y_k^j; \\ \tilde{\Sigma}_{W^i} = \frac{\Sigma_{W^i}}{m_k \pi_{k,\tau+1}^i}. \end{cases}$$

- Calcul de $\mathbf{X}_{\tau+1}$ par application de filtre de Kalman avec lissage, pour $i = 1, \dots, M$, utilisant les mesures \tilde{y}_k^i et les matrices de covariance $\tilde{\Sigma}_{W^i}$

Le lecteur intéressé trouvera plus de détails sur ces thèmes, à savoir, les problèmes de mise en œuvre du PMHT, ses déficiences ainsi que quelques propositions pour y remédier dans l'article de Willett et al [81] qui le compare avec le PDAF.

Les deux sections précédentes ont permis d'exposer la structure des principaux algorithmes déterministes pour le filtrage multi-cibles en environnement bruité. Dans la première d'entre-elles, nous avons supposé la dépendance des associations conduisant aux algorithmes tels que le FBO, le (J)PDAF et le MHT. Cette hypothèse a été relâchée dans la seconde, au profit de l'indépendance des associations d'où l'algorithme du PMHT. Comme on l'a vu au chapitre 2, dans le cas de poursuite d'une cible unique, la structure d'estimation par filtre de Kalman peut être avantageusement remplacée dans le cas de modèles d'états ou de mesures non linéaires par des méthodes particulières. Nous abordons maintenant le centre de ce travail de thèse à savoir l'apport des méthodes particulières pour le filtrage multi-cibles.

4.4 Conclusion

Il est rare que les algorithmes traitant du filtrage d'une cible unique puissent être appliqués sous leur forme initiale présentée aux chapitres 1 et 2. En effet, les mesures acquises proviennent parfois, soit d'une autre cible que la cible d'intérêt, soit sont des fausses alarmes. Des extensions des algorithmes précédents doivent donc être effectuées pour tenir compte de l'incertitude sur l'origine des mesures. Cette incertitude est modélisée par l'introduction d'une variable d'association déterminant les associations entre les mesures et les cibles. Selon les algorithmes, l'association d'une mesure est considérée indépendamment ou pas des associations des autres mesures. Fondés sur l'utilisation du filtre de Kalman, un premier ensemble de filtres, le FBO, le PDAF, le JPDAF et le MHT estiment de façon plus ou moins optimale la loi de l'ensemble des cibles conditionnellement aux mesures, sous l'hypothèse de dépendance des associations. Les sous-optimalités introduites permettent de réduire le nombre de réalisations différentes du vecteur d'association à considérer, qui croît exponentiellement avec le temps sous cette hypothèse de dépendance. L'originalité du PMHT consiste à relâcher cette hypothèse afin d'éviter l'énumération exhaustive de ces associations. Les états et les probabilités de chaque modèle sont estimés conjointement par un algorithme d'espérance-maximisation. Suite à la mise en évidence de l'apport des méthodes particulières pour le problème de filtrage d'une

cible unique, il était naturel de se demander quel pouvait être l'apport de telles méthodes pour le problème de filtrage non linéaire multi-cibles.

Chapitre 5

Poursuite non linéaire multi-cibles : Application aux problèmes par mesures d'angles

Suite à l'émergence des méthodes particulières décrites au chapitre 2, nous étudions leur application à la poursuite non linéaire multi-cibles. Notre contribution concernant ces méthodes consiste en l'élaboration d'un algorithme non linéaire résultant de la combinaison de l'approche modèle multiple décrite au chapitre 3 avec les filtres particuliers. Pour explorer la capacité de cet algorithme à la poursuite multi-cibles, nous l'appliquons sur divers scénarios. Les scénarios considérés sont des scénarios simulés au préalable et les objets cibles sont des cibles ponctuelles du plan 2D.

5.1 Le NMMPF : Algorithme non linéaire modèles multiples à base de filtrage particulière pour la poursuite multi-cibles

Soit M le nombre de cibles à suivre (supposé ici constant et connu). On considère le modèle dynamique général défini par un processus d'état $X_k = (X_k^1, \dots, X_k^M)$ markovien :

$$X_k = F_k(X_{k-1}, V_k), \quad (5.1)$$

où les processus d'états sont éventuellement indépendants :

$$X_k^i = F_k^i(X_{k-1}^i, V_k^i) \quad \text{pour } i = 1, \dots, M \quad (5.2)$$

Le vecteur d'observation à l'instant k est désigné par $y_k = (y_k^1, \dots, y_k^{m_k})$, l'observation y_k^j est alors une réalisation du processus :

$$Y_k^j = H_k^i(X_k^i, W_k^j) \quad (5.3)$$

Les bruits $(V_k^i)_{k \in \mathbb{N}}$ pour $i = 1, \dots, M$ et $(W_k^j)_{k \in \mathbb{N}}$ pour $j = 1, \dots, m_k$ sont des bruits blancs indépendants.

Nous supposons de plus que les fonctions H_k^i sont telles qu'on peut leur associer des fonctions l_k^i définies par $l_k^i(y_k^j; x_k^i) \propto p(y_k^j | x_k^i)$.

Nous supposons qu'une mesure peut provenir de l'un des M objets cibles et qu'une cible donnée peut produire une ou plusieurs mesures à chaque instant.

5.1.1 Présentation de l'algorithme NMMPF

L'algorithme non linéaire modèles multiples à base de filtrage particulière, dénommé nonlinear multiple model particle filter algorithm et dénoté NMMPF en anglais, consiste alors à combiner l'approche modèles multiples décrite au chapitre 3 avec les filtres particulières. Pour se faire, nous devons construire un banc de filtre composé de M modèles différents couvrant les probables mouvements observés, c'est-à-dire, nous mettons un filtre particulière pour chaque cible. Le rôle de l'approche modèle multiple [88] intervient donc, après l'opération du filtrage particulière, par interaction entre les filtres pour affecter chaque mesure à la cible correspondante.

Nous rappelons que le nombre de cibles est constant et connu et égal à M , la solution proposée par notre algorithme NMMPF consiste alors, à mettre un filtre particulière pour chaque cible. Nous désignerons par $s_k^{n,i}$ la composante de la cible i de la $n^{\text{ième}}$ particule. Les estimations des différentes cibles se font donc d'une manière indépendante et l'espace d'état des particules n'est pas celui du vecteur concaténé $X_k = (X_k^1, \dots, X_k^M)$ mais un filtre particulière est mené pour chaque cible.

Pendant l'opération du filtrage, l'initialisation du nuage $S_0 = (s_0^n, 1/N)_{n=1, \dots, N}$ s'effectue en échantillonnant chaque composante $s_0^{n,i}$ pour $i = 1, \dots, M$ selon la loi $p(X_0^i)$ et indépendamment des autres composantes. Supposons construit le nuage $S_{k-1} = (s_{k-1}^n, \omega_{k-1}^n)_{n=1, \dots, N}$ avec $\sum_{n=1}^N \omega_{k-1}^n = 1$, la prédiction se fait alors pour chaque sous particule selon la loi d'évolution de la cible à laquelle elle est associée :

$$\text{Pour } n = 1, \dots, N \quad s_{k|k-1}^n = \begin{pmatrix} F_k^1(s_{k-1}^{n,1}, v_k^{n,1}) \\ \vdots \\ F_k^M(s_{k-1}^{n,M}, v_k^{n,M}) \end{pmatrix}. \quad (5.4)$$

L'estimation par filtrage particulière de la loi conditionnelle $L_{k|k}^i$ de chaque cible pour $i = 1, \dots, M$ pour chaque mesure est donnée comme suit :

$$L_{k|k}^i = p(X_k^i | Y_k^j) \quad j = 1, \dots, m_k \quad (5.5)$$

A partir de ces lois et à l'aide de l'approche modèles multiples, nous pouvons estimer la loi conditionnelle finale $L_{k|k}$ de tous les filtres pour chaque mesure par:

$$p(X_k | Y_k^j, Y_{1:k-1}) = \sum_{i=1}^M p(X_k^i | Y_k^j) \mu_i^j(k) \quad \text{pour } j = 1, \dots, m_k \quad (5.6)$$

où μ_i^j est la probabilité pour que la mesure j soit associée à la cible i , avec :

$$\sum_{i=1}^M \mu_i^j(k) = 1 \quad \text{pour } j = 1, \dots, m_k \quad (5.7)$$

L'estimé final pour chaque mesure dans l'algorithme NMMPF [88] est donc la somme des estimés des différents filtres pondérés par leurs probabilités, i.e., probabilité pour qu'une mesure j soit associée à la cible i . Sachant que ces probabilités à l'instant initial $k = 0$, sont égales pour chaque mesure:

$$\text{pour } j = 1, \dots, m_k, \quad \mu_i^j(0) = \frac{1}{M} \quad \text{pour } i = 1, \dots, M \quad (5.8)$$

Une description générale de l'algorithme NMMPF est présentée dans l'algorithme 5.1.

5.2 Application du NMMPF à la poursuite multi-cibles par mesure d'angles

5.2.1 Scénarios et paramètres

Nous considérons à nouveau le scénario simulé de poursuite en utilisant uniquement des mesures d'angles comme à la fin du chapitre 2. Le nombre de cibles est fixé à trois. Chacune est assimilée à un point de l'espace 2D et elle est paramétrée par ses coordonnées cartésiennes et son vecteur de vitesse instantanée dans le repère cartésien est : $X_k^i = (X_k^i \ v_k^i \ Y_k^i \ v_k^i)$ pour $i = 1, 2$ et $X_k^i = (X_k^i \ vx_k^i \ Y_k^i \ vy_k^i \ \Omega_k^i)$ pour $i = 3$, où Ω est la vitesse angulaire. Notons que les cibles 1 et 2 ont des mouvements rectilignes et elles ont des lois évolutions identiques qui diffèrent seulement de l'ajout du paramètre d'accélération à la cible 2, i.e., elles sont respectivement à vitesse et à accélération constantes. Par contre la loi d'évolution de la cible 3 est totalement différente, elle a un mouvement circulaire à vitesse de rotation constante.

Pour cela, nous avons considéré pour les cibles 1 et 2, une loi d'évolution identique à celle de (2.26) :

$$X_{k+1} = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix} X_k + \begin{pmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{pmatrix} V_k \quad (5.9)$$

Algorithme 5.1 : Algorithme non linéaire modèle multiple à base de filtrage particulière (NMMPF) avec rééchantillonnage adaptatif

- Initialisation de l'état et sa covariance pour chaque filtre : X_0^i, P_0^i pour $i = 1, \dots, M$.

- Initialisation des probabilités des modèles : $\mu_i^j(0) = \frac{1}{M}$ pour $i = 1, \dots, M$.

- For $k = 1, \dots, \text{fin}$

- Opération de filtrage (utilisant un banc de filtres particulières)

For $i = 1, \dots, M$.

o Initialisation : $\begin{cases} s_0^{n,i} = p(X_0^i) \\ \omega_0^n = \frac{1}{N} \end{cases} \quad n = 1, \dots, N.$

o For $j = 1, \dots, m_k$

- Générer $(s_k^{n,i})$ selon $f(X_k^i | X_{k-1}^i = s_{k-1}^{n,i}, Y_k^j = y_k^j), n = 1, \dots, N$.

- Mise à jour des poids :

$$\begin{cases} \text{calcul des poids non normalisés:} \\ \tilde{\omega}_k^{n,j} = \omega_{k-1}^{n,j} \frac{p(s_k^{n,i} | s_{k-1}^{n,i}) l_k(y_k^j, s_k^{n,i})}{f(s_k^{n,i} | s_{k-1}^{n,i}, y_k^j)} \quad \text{pour } n = 1, \dots, N \\ \text{normalisation des poids:} \\ \omega_k^{n,j} = \frac{\tilde{\omega}_k^{n,j}}{\sum_{n=1}^N \tilde{\omega}_k^{n,j}} \quad \text{pour } n = 1, \dots, N \end{cases}$$

- Retourner $\mathbb{E}(\phi(X_k^i)_j) = \sum_{n=1}^N \omega_k^{n,j} \phi(X_k^i)_j$

- Calculer $N_{eff}^j = \frac{1}{\sum_{n=1}^N (\omega_k^{n,j})^2}$

- Rééchantillonnage si $N_{eff}^j < N_{seuil}$.

- Mise à jour des probabilités des modèles

o For $j = 1, \dots, m_k$.

- $\Lambda_i^j(k) \triangleq p(y^j(k) | Y_{k-1}^j, M_i) = p(Y_i^j(k)) = N(Y_i^j(k); 0, S_i^j(k)) \quad i = 1, \dots, M$

- $\mu_i^j(k) = \frac{\Lambda_i^j(k) \mu_i^j(k-1)}{\sum_{i=1}^M \Lambda_i^j(k) \mu_i^j(k-1)} \quad i = 1, \dots, M.$

- Combinaison des estimations

o For $j = 1, \dots, M$

- $\hat{X}^j(k|k) = \sum_{i=1}^M \mu_i^j(k) \hat{X}_i^j(k|k)$

- $P_j(k|k) = \sum_{i=1}^M \mu_i^j(k) \left(P_i^j(k|k) + [\hat{X}_i^j(k|k) - \hat{X}^j(k|k)][\hat{X}_i^j(k|k) - \hat{X}^j(k|k)]^T \right)$

Et pour la cible 3, une loi d'évolution identique à celle de (2.29) :

$$X_{k+1} = \begin{pmatrix} 1 & \frac{\sin\Omega_k T}{\Omega_k} & 0 & -\frac{1 - \cos\Omega_k T}{\Omega_k} & 0 \\ 0 & \cos\Omega_k T & 0 & -\sin\Omega_k T & 0 \\ 0 & \frac{1 - \cos\Omega_k T}{\Omega_k} & 1 & \frac{\sin\Omega_k T}{\Omega_k} & 0 \\ 0 & \sin\Omega_k T & 0 & \cos\Omega_k T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} X_k + \begin{pmatrix} 0.5T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & 0.5T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{pmatrix} V_k \quad (5.10)$$

Où V_k est un vecteur gaussien de moyenne nulle et de matrice de covariance :

$$\Sigma_{V^i} = \begin{pmatrix} \sigma_{X^i}^2 & 0 \\ 0 & \sigma_{Y^i}^2 \end{pmatrix} \quad (5.11)$$

Le pas de temps T est fixé à 4 secondes dans les expérimentations suivantes et l'écart-type du bruit de dynamiques est identique pour toutes les cibles et vaut $\sigma_X = \sigma_Y = 0.05ms^{-2}$.

Les trois cibles sont observées via la loi de mesure d'angle suivante :

$$Y_k = \tan^{-1} \left(\frac{Y = \eta}{X = \xi} \right) + W_k \quad (5.12)$$

avec W_k un bruit gaussien centré de covariance σ_W^2 indépendant de V_k et (X, Y) est la position de la cible aérienne dans le plan horizontal.

Dans tous les scénarios, le nombre de mesures est supposé constant et égal à 3 à chaque instant k .

Les positions initiales des trois cibles ainsi que leurs vitesses sont :

$$X_0^1 = \begin{pmatrix} 3000 \\ 10 \\ 500 \\ 5 \end{pmatrix}; \quad X_0^2 = \begin{pmatrix} 500 \\ 10 \\ 3000 \\ 5 \end{pmatrix}; \quad X_0^3 = \begin{pmatrix} 2000 \\ 10 \\ 1500 \\ 5 \\ 0.00223 \end{pmatrix}.$$

Les trajectoires des trois cibles sont présentées dans la figure 5.1. Nous supposons que chaque cible produit une mesure à chaque pas de temps $T = 4$ secondes selon l'équation (5.12), les mesures d'angles des trois cibles sont simulées avec un bruit de mesure gaussien de variance $\sigma_W = 0.02$ radians, soit environ 1.5 degrés, qui sont représentées dans la figure 5.2.

Dans ce cas, nous avons besoin d'un banc de filtres composé de trois filtres particuliers. L'initialisation de ces filtres a été faite suivant une loi gaussienne, dont les vecteurs moyennes et leurs matrices de covariances sont données par :

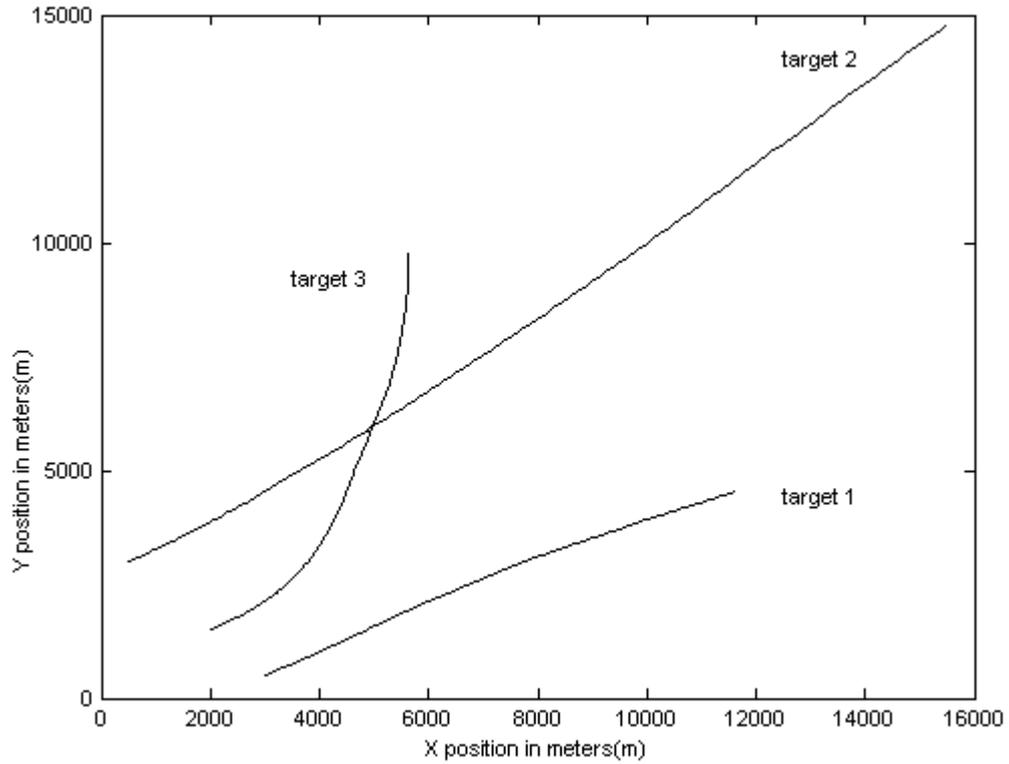


Fig. 5.1 : Trajectoires des 3 cibles

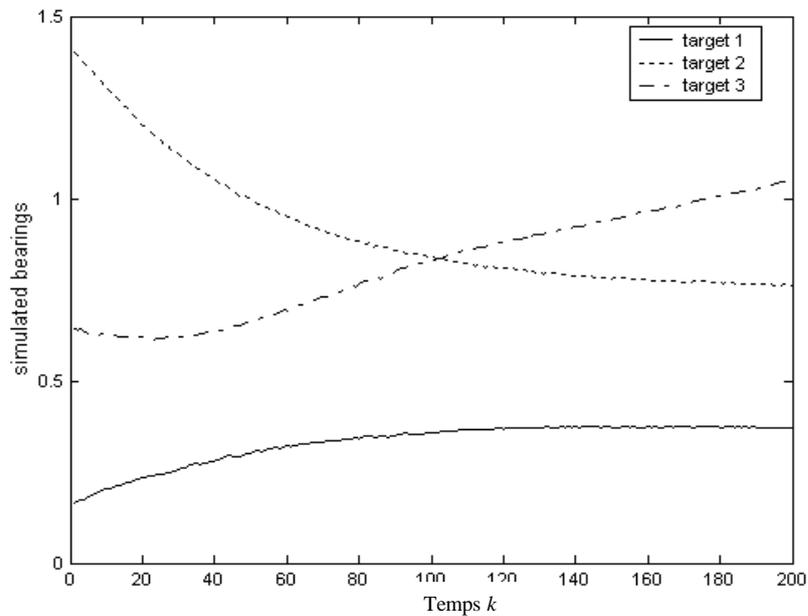


Fig. 5.2 : Mesures d'angles simulées des 3 cibles en radians

$$(X_0^1)_{moy} = \begin{pmatrix} 2800 \\ 11 \\ 600 \\ 4 \end{pmatrix}; \quad (X_0^2)_{moy} = \begin{pmatrix} 450 \\ 9 \\ 3200 \\ 4.5 \end{pmatrix}; \quad (X_0^3)_{moy} = \begin{pmatrix} 2100 \\ 11 \\ 1400 \\ 4 \\ 0.00323 \end{pmatrix}.$$

$$P_0^1 = P_0^2 = \begin{pmatrix} 100 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0.01 \end{pmatrix}, \quad P_0^3 = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix}$$

Les probabilités initiales des trois modèles de cibles pour chaque mesure sont :

$$\mu_i^1(0) = \frac{1}{3}; \quad \mu_i^2(0) = \frac{1}{3}; \quad \mu_i^3(0) = \frac{1}{3}; \quad i = 1, 2, 3.$$

Ses probabilités signifient qu'au départ tous les modèles de cibles possèdent les mêmes chances d'être choisis pour une mesure.

Pour évaluer les performances de l'algorithme NMMPF présenté dans l'algorithme 5.1, nous avons fait 100 exécutions monte Carlo. Pour chaque filtre particulière nous avons pris $N = 500$ particules avec rééchantillonnage adaptatif. Le seuil de rééchantillonnage dans chaque filtre particulière a été fixé à $N_{seuil} = 0.8N$.

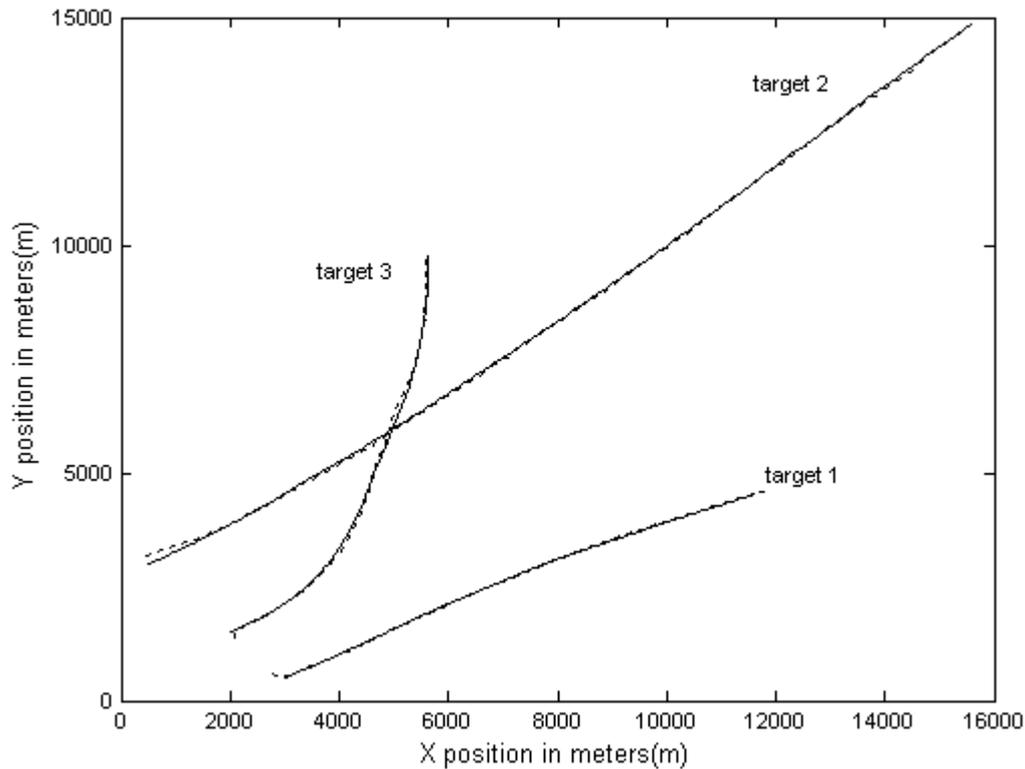


Fig. 5.3 : Trajectoires réelles et estimées des 3 cibles

Nous avons choisi comme mesure de performances de cet algorithme : les racines des erreurs quadratiques moyennes, dénommé root mean square error et dénoté RMSE en anglais, qui sont calculées selon (5.13).

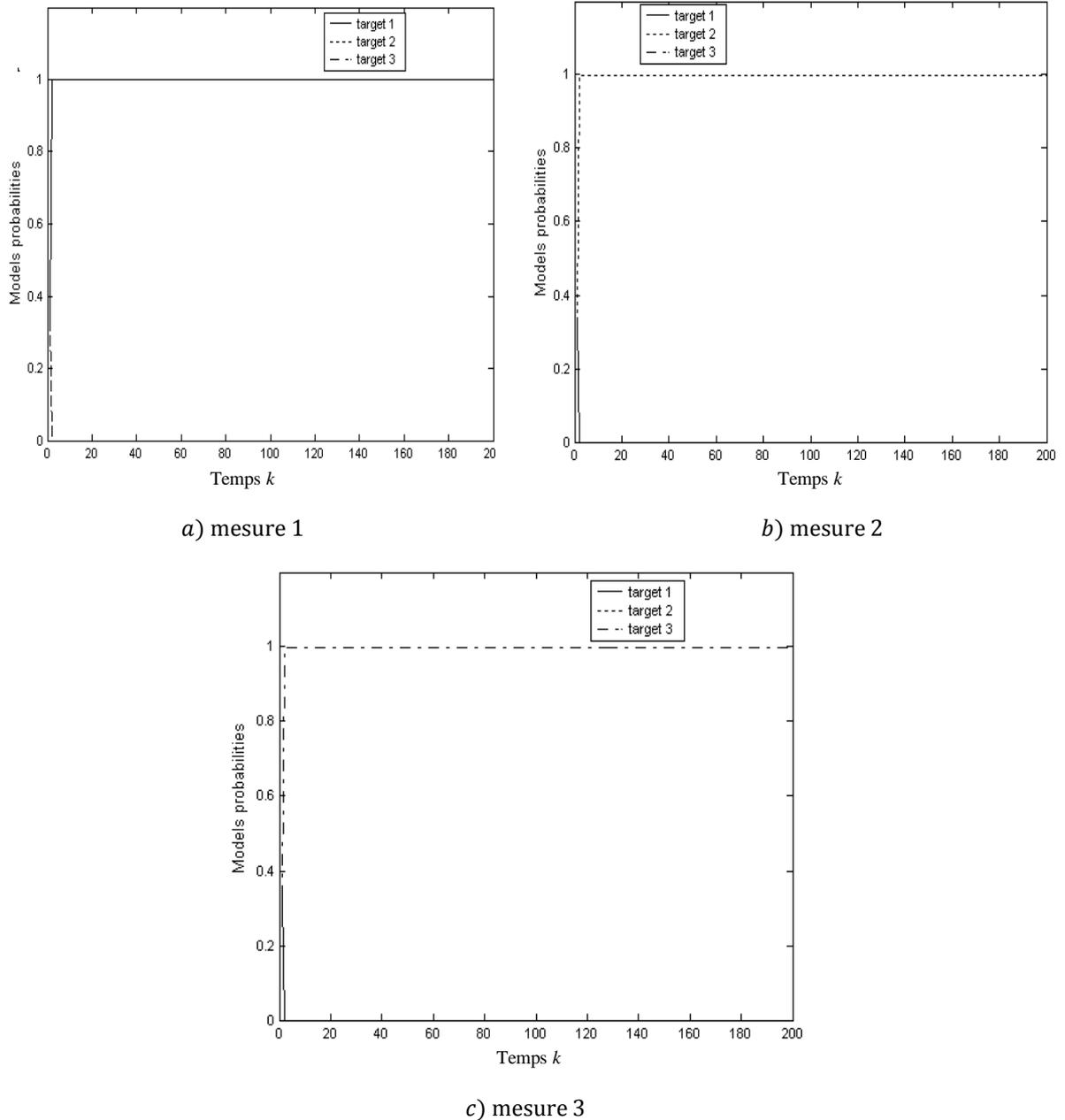


Fig. 5.4 : Probabilités des modèles (pour qu'une mesure $j = 1, 2, 3$ soit associée au modèle $i = 1, 2, 3$)

$$RMSE(X_j) = \sqrt{\frac{\sum_{k=1}^j (X_{réelle}(k) - X_{estimée}(k))^2}{j}} \quad \text{pour : } j = 1, \dots, fin \quad (5.13)$$

La figure 5.3 démontre que les trajectoires estimées et réelles (simulées) sont superposables et presque identiques pour les trois cibles. La figure 5.4 montre que l'algorithme NMMPF affecte chaque mesure ($j = 1, 2, 3$) à son propre objet cible et n'a besoin que de 1 à 3 échantillons.

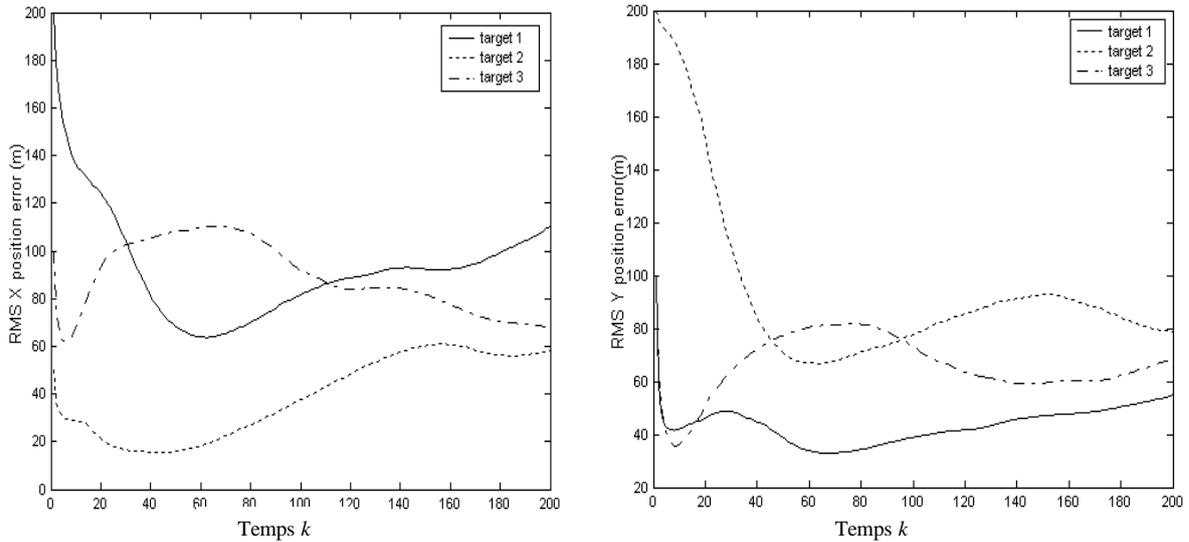


Fig. 5.5 : Racine de l'erreur quadratique moyenne (RMSE) de position (X, Y) pour les 3 cibles.

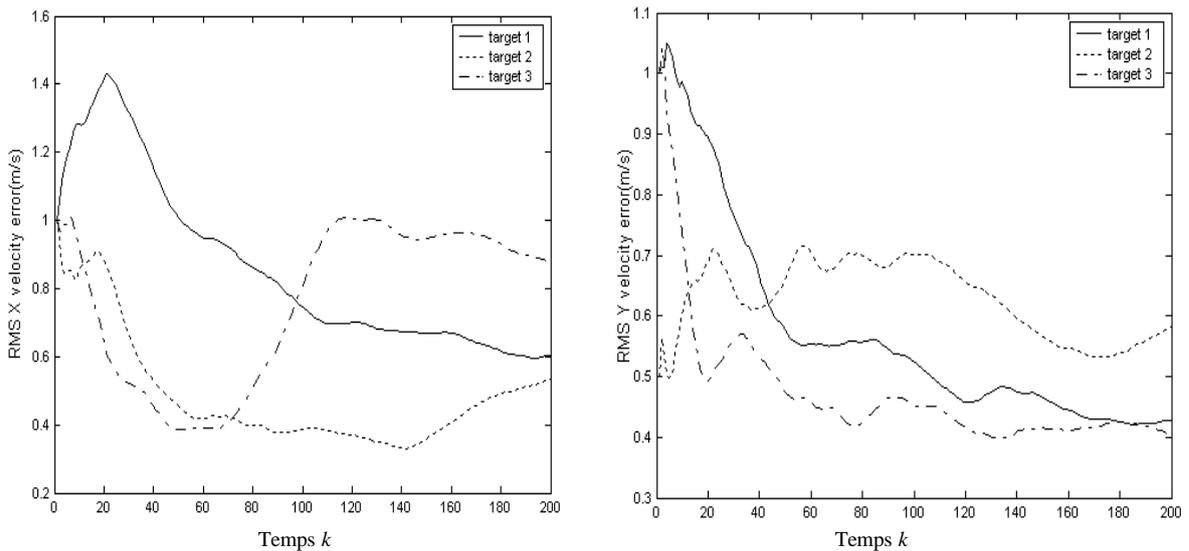


Fig. 5.6 : Racine de l'erreur quadratique moyenne (RMSE) de vitesse (v_x, v_y) pour les 3 cibles.

Les figures 5.5, 5.6 et 5.7 présentent les évolutions dans le temps de la racine de l'erreur quadratique moyenne RMSE, des positions (X, Y), des vitesses (v_x, v_y) et de la vitesse angulaire Ω . On peut remarquer que l'algorithme NMMPF dépiste précisément les

dynamiques des cibles. Ces RMSE faibles confirment les résultats tracés dans la figure 5.3.

A partir de ces résultats nous pouvons conclure que notre estimateur converge et affecte chaque mesure à son vrai modèle.

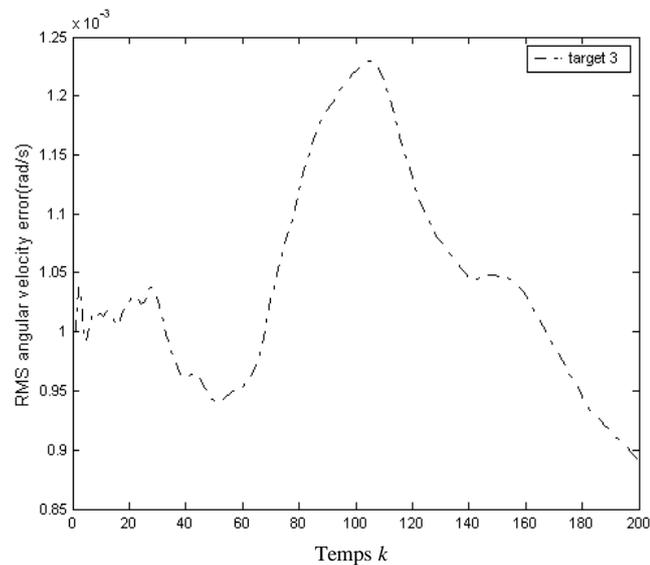


Fig. 5.7 : Racine de l'erreur quadratique moyenne (RMSE) de la vitesse angulaire Ω .

Concernant le temps de calcul, lorsque on exécute le NMMPF par un PC pentium IV, 3.40 GHz avec $N = 500$ particules, il prend 420 milli secondes pour un pas de temps k ou encore une itération..

5.2.2 Influence de l'augmentation du nombre de particules

L'augmentation du nombre de particules influe positivement sur l'erreur d'estimation, en d'autres termes, à chaque fois qu'on augmente le nombre de particules du filtre, l'estimation de la dynamique des cibles sera bonne, voir la figure 5.8 et 5.9. Par contre cette augmentation influe négativement sur le temps de calcul. Le tableau 5.1 indique les temps de calcul nécessaires pour une itération en fonction du nombre de particules.

Tab. 5.1 : Temps de calcul nécessaire pour une itération du NMMPF sur Pentium IV à 3.40 MHz

nombre de particules	500	1000	2000
une itération de NMMPF	420 ms	945 ms	2.3 s

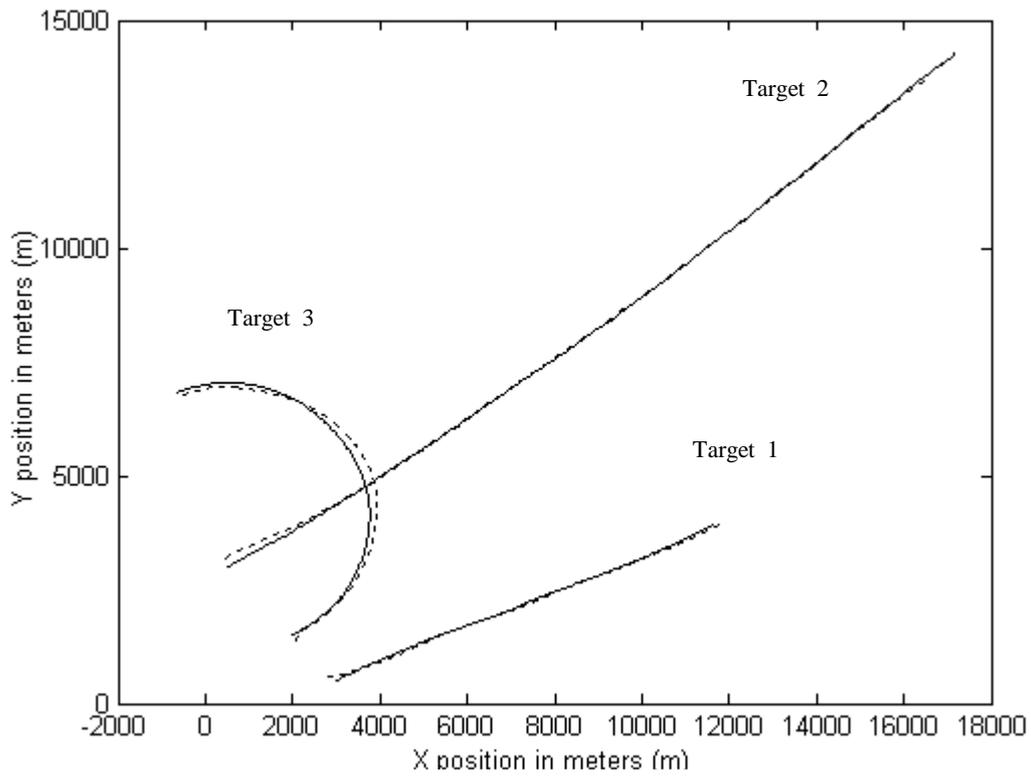


Fig. 5.8 : Trajectoires réelles et estimées des 3 cibles pour $N = 1000$ particules.

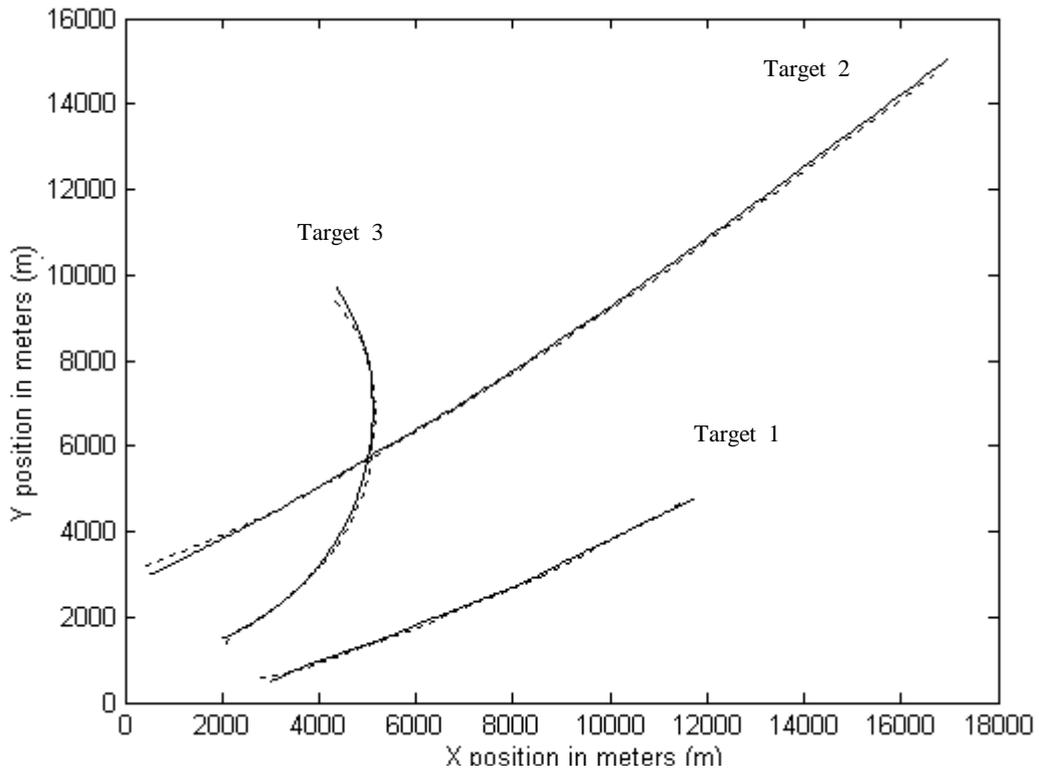


Fig. 5.9: Trajectoires réelles et estimées des 3 cibles pour $N = 2000$ particules

Le temps de calcul ne dépend pas seulement du nombre de particules utilisées, mais aussi de l'étape de rééchantillonnage car à chaque itération nous devons calculer le nombre de particules efficace N_{eff} . Si ce dernier est inférieur à un seuil N_{seuil} fixé au préalable, alors : nous devons effectuer une opération de rééchantillonnage à l'intérieur de cette itération. De ce fait, le temps de calcul d'une itération peut différer d'une itération à une autre.

Le temps de calcul d'une itération dans le tableau 5.1 est donc le temps moyen d'une itération, nous l'avons calculé comme suit :

$$\text{Temps moyen d'une itération} = \frac{\text{temps total de NMMPF}}{\text{nombre d'itération}}.$$

5.2.3 Influence de l'initialisation

Nous avons également expérimenté des réalisations de notre l'algorithme en biaisant l'initialisation en position. Par exemple :

$$(X_0^1)_{moy} = \begin{pmatrix} 3500 \\ 11 \\ 1000 \\ 4 \end{pmatrix}; \quad (X_0^2)_{moy} = \begin{pmatrix} 1000 \\ 9 \\ 2500 \\ 4.5 \end{pmatrix}; \quad (X_0^3)_{moy} = \begin{pmatrix} 2500 \\ 11 \\ 1000 \\ 4 \\ 0.00323 \end{pmatrix}.$$

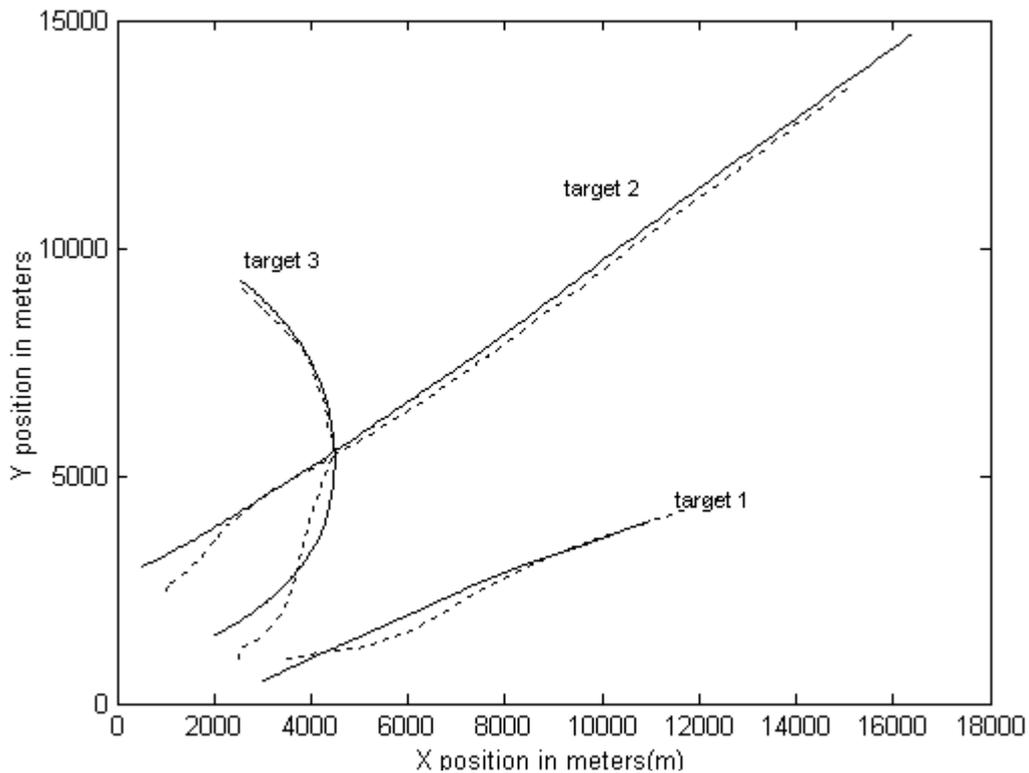


Fig. 5.10: Trajectoires réelles et estimées des 3 cibles avec initialisation biaisée pour $N = 500$ particules

La figure 5.10 montre que le NMMPF ne trouve pas les vraies trajectoires de cibles qu'après plusieurs pas, i.e., ne converge pas rapidement vers les vraies modèles. Cependant, des biais sur les vitesses donnent des résultats beaucoup moins satisfaisants. Comme pour les algorithmes déterministes classiques, l'initialisation reste un problème des méthodes particulières.

5.2.4 Remarque

Notons que, à notre connaissance, il n'existe pas dans la littérature des algorithmes qui traitent la poursuite multi-cibles non linéaire multi-modèles pour pouvoir les comparer avec le notre.

5.3 Conclusion

En ce qui concerne le NMMPF, plusieurs filtres particulières regroupés dans un banc de filtres permettent de représenter de manière indépendante les états de cibles. L'approche modèles multiples permet de gérer l'ensemble des filtres particulières et d'affecter l'ensemble des mesures ; chaque mesure à sa cible correspondante. Le secret de cette affectation repose sur l'estimation de la probabilité pour qu'une mesure j soit associée à la cible i . L'estimé final de NMMPF est la somme des estimés élémentaires pondérés par leurs probabilités.

Notre étude a porté sur plusieurs scénarios simulés mettant en œuvre des cibles faiblement manœuvrantes. La confrontation du NMMPF à un scénario possédant des modèles différents a permis de souligner le bon fonctionnement en ce qui concerne l'estimation de la probabilité a posteriori des associations. Nous avons choisi les RMSE comme mesure de performances de cet algorithme. D'après les RMSE tracées, nous pouvons dire que l'algorithme NMMPF est une solution pertinente pour des problèmes de poursuite multi-cibles dont les modèles d'états et/ou de mesures sont non linéaires avec des bruits non gaussiens.

Conclusion générale et perspectives

Dans ce travail de thèse, nous nous sommes intéressés à la poursuite non linéaire multi-cibles par des méthodes séquentielles de Monte Carlo ou encore filtres particulaires, motivées par de nombreuses applications. Dans le cadre Bayésien que nous nous sommes fixé, le problème de filtrage revient à l'estimation de la loi a posteriori de l'état conditionnellement aux mesures. Les approches classiques telles que les filtres de Kalman étendus et les filtres de Kalman unscented sont mises en défaut lorsque la loi a posteriori diffère trop de la loi gaussienne dont ils estiment les moments. Dans ce contexte, les méthodes particulaires fournissent une approximation discrète de la loi a posteriori particulièrement intéressante. Elle consiste en une somme pondérée de lois de Dirac centrées en des points générés aléatoirement et dénommés particules. La propagation des particules au cours du temps s'effectue selon le principe d'échantillonnage pondéré qui permet de les guider dans les régions de forte vraisemblance. Ces méthodes ont fait l'objet de nombreuses études depuis les années 90, sans pour autant avoir été étendues au cas de cibles multiples qui constituait le sujet de cette thèse.

Notre contribution sur ce sujet se décline en ces points:

- L'élaboration d'un algorithme non linéaire multi-cibles dénoté NMMPF, fondé sur la combinaison de l'approche modèles multiples et une structure particulière permettant l'estimation des associations a posteriori et la position de cibles.
- La mise en œuvre du NMMPF à la poursuite multi-cibles, fondée sur des mesures de mouvement.

La modélisation du filtrage multi-cibles issue du traitement du signal se donne un vecteur d'état résultant de la concaténation des états des différents objets ainsi qu'un vecteur de mesures dont chaque composante est issue d'un des objets ou elle est une fausse alarme. L'estimation des objets nécessite alors de résoudre le problème crucial de l'association des mesures aux objets. Un vecteur d'association est introduit pour modéliser cette association. Deux approches se distinguent alors selon que l'on suppose que l'association d'une mesure donnée dépend ou pas des associations des autres mesures. Sous l'hypothèse de

dépendance, les deux principaux algorithmes de poursuite multi-cibles sont le JPDAF et le MHT. Ils nécessitent l'énumération exhaustive des associations. Le PMHT en revanche suppose l'indépendance des associations et considère la vraisemblance comme une loi de mélange dont les paramètres sont estimés par un algorithme d'espérance-maximisation. L'introduction de méthodes particulières pour le filtrage multi-cibles permet de résumer les associations passées dans le nuage à l'instant précédent. Cependant, la mise à jour des poids nécessite l'évaluation de la vraisemblance du vecteur de mesures qui ne peut être atteinte que par sa décomposition selon les associations courantes possibles. La vraisemblance s'écrit alors comme une somme de vraisemblances selon l'association considérée et pondérée par la probabilité de l'association conditionnellement à l'état. L'algorithme NMMPF proposé se situe dans la catégorie des algorithmes supposant l'indépendance des associations. Il consiste à tenter d'estimer au mieux la probabilité de l'association conditionnellement à l'état par la probabilité de l'association conditionnellement aux mesures. Cette probabilité est atteinte par l'approche modèle multiple.

Le NMMPF a été validé dans le contexte de poursuite non linéaire multi-cibles. Les scénarios simulés mettent en évidence le bon fonctionnement de l'algorithme tant au niveau des probabilités des associations a posteriori que de l'estimation de la position des cibles. La souplesse des méthodes particulières nous a permis de les combiner avec l'approche modèles multiples afin de pouvoir estimer les probabilités d'associations.

De nombreuses perspectives se dégagent de ces travaux. Tout d'abord, la combinaison des filtres particuliers avec l'unscented Kalman filtre et de les mettre dans l'algorithme NMMPF à la place des filtres particuliers afin de poursuivre des cibles multiples manoeuvrantes.

D'autre part, il serait intéressant d'utiliser cet algorithme en analyse d'images afin de poursuivre des entités dans des séquences vidéos.

Enfin, d'un point de vue applicatif, une perspective naturelle de nos travaux en poursuite serait donc de confronter nos algorithmes à des scénarios réels.

Bibliographies

- [1] R. A. Fisher. On an absolute criterion for fitting frequency curves. *Messenger of Math*, 41,155. 1912
- [2] A. N. Kolmogoroff. Interpolation and Extrapolation Von Stationnaren Zufalligen Folgen. *Bulletin of The Academic sciences of U.S.S.R. Series in Mathimatics*, 5, pp.3-14. 1941.
- [3] N. Wiener. *Interpolation, Extrapolation and Smoothing of Stationnary Time Series*, Wiley, New York. 1948
- [4] N. Metropolis, S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44:335-341, 1949.
- [5] J. Von Neuman. Various techniques used in connection with random digits. *National Bureau of standards applied mathematics series*, 1951.
- [6] N. Metropolis, A. W. Rosenbluth, M. N Rosenbluth, A. H. Teller, E. Teller. Equations of state calculations by fast computing machines. *The Journal of Chemical Physics*, 1953.
- [7] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions ASME, Journal of Basic Engineering* .82, 35-45.1960.
- [8] J. E. Handschin. Monte Carlo techniques for prediction and filtering of nonlinear stochastic processes. *Automatica*, 6:555-563, 1970.
- [9] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 1970.
- [10] Y. Bar-Shalom, E. The. Tracking in a cluttered environment with probabilistic data association. *Proceedings of the 4th Symp. On nonlinear estimation theory and its applications*, 1973.
- [11] A. Gelb. *Applied Optimal Estimation*, The MIT Press, Cambridge 1974.
- [12] R. A. Singer, R. G. Sea, K. Housewright. Derivation and evaluation of improved tracking filters for use in dense multitarget environments. *IEEE Trans. on Information Theory*, 20(4):423-432, 1974.
- [13] Y. Bar-Shalom, E. The. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11:451-460, 1975.
- [14] A. P. Dempster, N. M. Laird, D. B. Rubin. Maximum likelihood. *Journal of Royal Statistical Society, Series B*, 39:1-38, 1977.
- [15] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on*

- Automation and Control, 24(6):84-90, 1979.
- [16] D. E. Knuth. The art of computer programming - Reading, MA, USA: Addison-Wesley, vol. 2, 1981.
- [17] V. E. Benes. Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics*, 5:65-92, 1981.
- [18] C. F. Wu. On the convergence of the EM algorithm. *Annals of Statistics*, 11 (1), 95-103, 1983.
- [19] M. Broniatowski, G. Celeux, J. Diebolt. Reconnaissance de mélanges de densités par un algorithme d'apprentissage probabiliste. *Data Analysis and Informatics*, pp. 359-374, 1983.
- [20] R. A. Redner, H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *Society for Industrial and Applied Mathematics*, 26(2), 1984.
- [21] Blackman, S. Samuel. *Multiple-Target Tracking with Radar Applications*. Norwood, MA: Artech House, 1986.
- [22] L. Devroye, *Non-uniform random variate generation*. New York: Springer-Verlag, 1986.
- [23] B. Ripley. *Stochastic simulation*. New York: John Wiley, 1987.
- [24] D. B. Rubin. Comment on "the calculation of posterior distributions by data augmentation" by M.A. Tanner and W.H. Wong. *Journal of the American Statistical Association*, 82:543-546, 1987.
- [25] Blom, A. P. Henk and Y. Bar-Shalom. The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients, *IEEE Transactions on Automatic Control*, 33 (Aug):780-783 (August 1988).
- [26] D. B. Rubin. Using the SIR algorithm to simulate posterior distributions, chap. Bayesian statistics 3, pp. 395-402. - Oxford University Press, Oxford, 1988.
- [27] Y. Bar-Shalom, E. The. Fortmann. *Tracking and data association*. Academic Press, 1988.
- [28] D. J. Salmond. Mixture reduction algorithms for target tracking in clutter. *SPIE Signal and Data Processing of small targets*, vol. 1305, pp. 434-445, 1990.
- [29] G. C. Wei, M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithm. *Journal of the American Statistical Association*, 85, 1990.
- [30] M. A. Tanner. *Tools for statistical inference. Lecture Notes in Statistics*. New-York, Springer Verlag, 1991.

- [31] A. F. M. Smith, A. E. Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):84-88, mai 1992.
- [32] A. Schrott. Feature-based camera-guided grasping by an eye in hand robot. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, volume 2, pages 1832-1837, Nice, France, May 1992.
- [33] P. Del Moral, G. Rigal, G. Salut. Estimation et commande optimale non-linéaire. *Rapport de Recherche n 2, Convention DRETDIGILOG-LAAS/CNRS, SM.MCY /685.92/ A,89.34.553.00.470.75.01, LAAS, 1992.*
- [34] A. Smith, G. Roberts. Bayesian computation via the Gibbs Sampler and related Markov chain Monte Carlo methods. *J. Royal Statist. Soc., Series B 55(1) :3-24, 1993.*
- [35] G. Chen. Approximate Kalman filtering series in *Approximations & Decompositions* C. K. Chui, ed., World Scientific Publishing Singapore/ New Jersey, 240. 1993.
- [36] J. Besag, P. J. Green. Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society, Series B, Methodological*, 55:25-27, 1993.
- [37] N. Gordon, D. Salmond, A. Smith. Novel approach to nonlinear/nonGaussian Bayesian state estimation. *IEE Proc.F, Radar and signal processing*, 140(2):107-113, April 1993.
- [38] S. P. Meyn, R. L. Tweedie. *Markov chain and stochastic stability*. London: Springer Verlag, 1993.
- [39] Y. Bar-Shalom and Xiao-Rong Li. *Estimation and Tracking: Principles, Techniques and Software*. Norwood, MA: Artech House, 1993.
- [40] A. Kong, J.S. Liu, W.R Wong. Sequential imputation method and Bayesian missing data problems. *J. Amer. Statist. Assoc.*, 89:278-288, 1994.
- [41] J. K. Uhlmann. Simultaneous map building and localization for real time applications. Technical report, University of Oxford, 1994. Transfer thesis
- [42] L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22(4):1701-1727, 1994.
- [43] R. L. Streit, T. E. Luginbuhl. Maximum likelihood method for probabilistic multi-hypothesis tracking. *Proceedings of SPIE International Symposium, Signal and Data Processing of Small Targets 1994*, vol. 2235, Orlando, FL, 5-7 April 1994.
- [44] J. S. Liu, R. Chen. Blind deconvolution via sequential imputation. *J. Amer. Statist. Assoc.*, 90(430):567-576, 1995.

- [45] S. Chib, E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327-335, 1995.
- [46] S. J. Julier, J. K. Uhlmann and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems, in *Proceedings of the American Control Conference*, pp.1628–1632, (Seattle, WA), 1995.
- [47] S. J. Julier, J. K. Uhlmann and H. F. Durrant-Whyte. A New Approach for the Nonlinear Transformation of Means and Covariances in Linear Filters. *IEEE Transactions on Automatic Control*, Accepted for publication as a Technical Note, 1995.
- [48] G. Celeux, D. Chauveau, J. Diebolt. Stochastic versions of the EM algorithm: an experimental study in the mixture case. *Journal of Statistical Computation and Simulation*, 1996.
- [49] 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.
- [50] M. Isard, A. Blake. Contour tracking by stochastic propagation of conditional density. *Lecture Notes in Computer Science*, 1064:343-349, 1996.
- [51] S. J. Julier and J. K. Uhlmann. A General Method for Approximating Nonlinear Transformations of Probability Distributions. tech. rep., Robotics Research Group, Department of Engineering Science, University of Oxford, 1996. Available: <http://www.robots.ox.ac.uk/siju>.
- [52] C. Berzuini, N. G. Best, W. R. Gilks, C. Larizza. Dynamic conditional independence models and Markov chain Monte Carlo methods. *Journal of the American Statistician*, 92(440):1403-1412, 1997.
- [53] H. Gauvrit. Extraction multi-pistes: approche probabiliste et approche combinatoire. PhD. Thesis, Univ. RennesI, 1997.
- [54] H. Gauvrit, J. P. Le Cadre, C. Jauffret. A formulation of multitarget tracking as an incomplete data problem. *IEEE Trans. on Aerospace and Electronic Systems*, 33(4):1242-1257, October 1997.
- [55] J. R. Norris. *Markov Chains*. Cambridge University Press, J.-C. 1997.
- [56] S. J. Julier and J. K. Uhlmann. A New Extension of the Kalman filter to Nonlinear Systems, in *Proceedings of the SPIE AeroSens International Symposium on Aerospace/Defense Sensing, Simulation and controls*, (Orlando, Florida), pp. 20-25, April, 1997.

- [57] A. Doucet. On sequential Simulation-Based Methods for Bayesian Filtering. Rapport de recherche, CUED/F-INFENG/TR 310, Signal Processing Group, Department of Engineering, University of Cambridge, 1998.
- [58] J. S. Liu, R. Chen. Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.*, 93:1032-1044, 1998.
- [59] M. Hürzeler, H. R. Künsch. Monte Carlo approximations for general state space models. *Journal of Computational and Graphical Statistics*, 7: 175-193, 1998.
- [60] P. Fearnhead. Sequential Monte-Carlo Methods in filter theory. PhD. Thesis, Department of Statistics, Oxford University, England, 1998.
- [61] C. Andrieu, J. F. G de Freitas, A. Doucet. Sequential MC for Bayesian model selection. *IEEE Higher Order Statistics Workshop*, pp. 130-134, Caesarea, Israel, 1999.
- [62] D. Crisan, M. Grunwald. Large deviation comparison of branching algorithms versus resampling algorithms: application to discrete time stochastic filtering. Rapport de recherche n: 9, Cambridge University, Statistical laboratory, 1999.
- [63] D. Fox, W. Burgard, S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:195-207, 1999.
- [64] H. Wang, T. Kirubarajan and Y. Bar-Shalom. Precision Large Scale Air Traffic Surveillance Using an IMM Estimator with Assignment. *IEEE Transaction on Aerospace and Electronic Systems*, AES-35(1), 255-266, 1999.
- [65] J. Carpenter, P. Clifford, P. Fearnhead. An improved particle filter for nonlinear problems. *IEE Proc.-Radar. Sonar Navig.*, 146(1), February 1999
- [66] M. K. Pitt, N. Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590-599, 1999.
- [67] N. Bergman. Recursive Bayesian estimation: navigation and tracking applications. PhD. Thesis, Linköping University, Sweden, 1999.
- [68] A. Doucet, S.J. Godsill, C. Andrieu. On sequential Monte-Carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197-208, 2000.
- [69] R. Van Der Merwe, A. Doucet, N. de Freitas, E. Wan. The unscented particle filter. Rapport de recherche, CUED/F-INFENG/TR 310, Cambridge University Engineering Department, 2000.
- [70] S. J. Julier. A new method for the nonlinear transformation of means and covariances in filters and estimators, *IEEE Trans. on Automatic Control* 45, pp. 477-482, Mars 2000.

- [71] C. Andrieu, A. Doucet, E. Puskaya. Sequential Monte Carlo methods for optimal filtering. chap. 4, pp. 79-95. Springer, 2001.
- [72] D. S. Salmond, N. Gordon. Partides and mixtures for tracking and guidance. Sequential Monte Carlo methods in practice, éd. par A. Doucet, N. De Freitas, N. Gordon, chap. 25, pp. 525-532. Springer, 2001
- [73] Skolnik, I. Merrill. Introduction to Radar Systems (Third Edition). New York, NY: McGraw-Hill, 2001.
- [74] Y. Iba. Population Monte Carlo algorithms. Transactions of the Japanese Society for Artificial Intelligence, 16(2):279-286, 2001.
- [75] Y. Bar-Shalom, X. Rong. Li, T. Kirubarajan. Estimation with applications to tracking and navigation, USA, 2001.
- [76] W. R. Gilks, C. Berzuini. Following a moving target. Monte Carlo inference for dynamic Bayesian models. Journal of the Royal Statistical Society - Series B Statistical Methodology, 63(1):127-146, 2001.
- [77] D. Crisan, A. Doucet. Survey of convergence results on particle filtering methods for practitioners. IEEE Trans. On Signal Processing, 50(3) :736-746, 2002.
- [78] J. Vermaak, C. Andrieu, A. Doucet, S. J. Godsill. Particle methods for Bayesian modelling and enhancement of speech signals, IEEE Transactions on Speech and Audio processing, 10(3): 173-185, 2002
- [79] Maybeck, S. Peter. Multiple Model Adaptive Estimation.” Lecture Notes. Air Force Institute of Technology, Wright-Patterson Air Force Base EENG768, OH, 2002.
- [80] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp. A tutorial on particle filters for online nonlinear /non-Gaussian Bayesian tracking. IEEE Trans. on Signal Processing, 50(2):174-188, 2002.
- [81] P. Willett, y. Ruan, R. Strei. PMHT: problems and some solutions. IEEE Transactions on Aerospace and Electronics, 38(3):738-754, 2002.
- [82] S. J. Julier. The scaled unscented transformation, in Proceedings of the American Control Conference, (Anchorage, AK), 2002.
- [83] C. Hue. Méthodes séquentielles de Monte Carlo pour le filtrage non linéaire multi-objets dans un environnement bruité. Applications au pistage multi cibles et à la trajectographie d’entités dans des séquences d’images 2D. Thèse de doctorat. Renne 1, France, 2003.
- [84] W. Jason. Flight Lieutenant, RAAF. Gaussian Mixture Reduction for Tracking

Multiple Maneuvering Targets in Clutter, Thesis: 2003.

- [85] M. S. Djouadi, A. Sebbagh, D. Berkani. A Nonlinear Algorithm for Maneuvering Target Visual-based Tracking: IEEE Proceeding of the Second International Conference on intelligent Sensing and Information Proceeding, ICISIP 2005, page 61-66, Chennai, India, 2005.
- [86] M. S. Djouadi, A. Sebbagh, D. Berkani. IMM based UKF and IMM base EKF algorithms for tracking highly manoeuvrable target, Archives of Control Sciences, Vol. 15 (LI), No, 1, 19-51, (2005).
- [87] A. Sebbagh, H. Tebbikh. Particle filtering for air craft tracking with bearings-only measurement, International Conference on systems and Information processing, ICSIP'09, May 2-4 2009, Algeria.
- [88] A. Sebbagh, H. Tebbikh. Nonlinear multiple model particle filters algorithm for tracking multiple targets. Archives of control sciences, Volume 21(LVII), No. 1, pages 37-60, 2011.

Annexe

Principe de Monte Carlo et génération de nombres aléatoires

Adopté pour la première fois dans les années 50 en physique statistique, le principe de Monte Carlo consiste à utiliser un échantillon discret, obtenu par simulation, pour approximer une distribution de probabilité que l'on ne sait pas calculer analytiquement. Typiquement, cet échantillon pourra être utilisé pour approximer des intégrales de grandes dimensions.

Cette approximation repose sur la loi des grands nombres. En effet, soient $(D, \mathcal{F}, \mathbb{P})$, un espace probabilisé, X une variable aléatoire définie sur cet espace, à valeurs dans \mathbb{R}^{n_x} et admettant une loi de densité $p(\cdot)$ et s^1, \dots, s^N un échantillon indépendant et identiquement distribué selon $p(\cdot)$. D'après la loi des grands nombres, la loi discrète $S^N = \frac{1}{N} \sum_{n=1}^N \delta_{s^n}$, où δ_s désigne la masse de Dirac centrée en s , converge vers la loi de densité $p(\cdot)$, i.e., pour toute fonction ϕ continue bornée sur \mathbb{R}^{n_x} :

$$\frac{1}{N} \sum_{n=1}^N \phi(s^n) \xrightarrow{N \rightarrow \infty} \mathbb{E}_p[\phi(x)] \quad (\text{A. 1})$$

Où \mathbb{E}_p désigne l'espérance prise par rapport à la densité $p(\cdot)$. Si de plus la variance de $\phi(x)$ est finie, alors le théorème central limite nous assure que le terme de gauche dans (A.1) est de variance proportionnelle à $1/N$.

Afin de mettre en œuvre le principe de Monte Carlo, il est donc nécessaire de pouvoir générer des nombres selon certaines lois. Au début du vingtième siècle, de tels nombres étaient fournis par des méthodes analogiques. Les ordinateurs ont désormais pris la relève et fournissent des suites déterministes de nombres, dont les propriétés statistiques sont similaires à celles des suites aléatoires visées. On les désigne par le terme de nombres *pseudo-aléatoires*. Dans ce chapitre, nous présentons, tout d'abord, une classe primordiale d'algorithmes permettant de simuler des nombres selon la loi uniforme sur l'intervalle

[0 1]. Afin de générer des nombres selon d'autres lois, trois méthodes non itératives sont ensuite exposées et illustrées. Nous décrivons enfin le principe des méthodes itératives de Monte Carlo par chaîne de Markov.

A.1 Génération de nombres pseudo-aléatoires selon la loi uniforme sur [0 1]

La génération de nombres pseudo-aléatoires selon la loi uniforme sur [0 1] est primordiale car elle constitue un outil pour générer ensuite des nombres pseudo-aléatoires selon des lois plus complexes.

Définition 1

Soit X un ensemble fini discret, qui représente par exemple l'ensemble fini des entiers-machines représentable par un ordinateur. Soient T une application de X dans X , $U : X \rightarrow [0 1]$ une application de normalisation et $(x_n)_{n \in \mathbb{N}}$ la suite obtenue en appliquant itérativement à partir d'une racine x_0 donnée, i.e., telle que $x_n = T(x_{n-1})$. Le système dynamique discret (X, T, U) est un générateur de nombres pseudo-aléatoires sur [0 1] si la suite $(U(x_n))_{n \in \mathbb{N}}$ a le même comportement statistique qu'une suite de variables indépendantes uniformément distribuées sur l'intervalle [0 1].

Du fait que X soit discret, l'ensemble $U(X)$ n'est qu'une partie finie discrète de [0 1]. Toute suite $(x_n)_{n \in \mathbb{N}}$ est donc périodique. On caractérisera l'application T par cette période. Les suites obtenues par un bon générateur algorithmique devront vérifier plusieurs propriétés statistiques telles que l'uniformité, l'indépendance, la reproductibilité et posséder une période élevée.

La fonction T est généralement une application de la forme $T_x = (ax + c) \bmod m$ où l'entier m est tel que $X = \{1, \dots, m - 1\}$. Les générateurs associés sont dénommés générateurs congruentiels affines. Afin de maximiser la période d'un tel générateur, on utilise la décomposition en facteurs premiers de m . Le générateur le plus utilisé est celui correspondant aux choix $c = 0$, $m = 2^{31} - 1$ et $a = 7^5 = 16807$. La période obtenue vaut alors $p = m - 1 = 2^{31} - 2$. On pourra se référer à [16] ou [23] pour la description de tels générateurs et de tests statistiques permettant la vérification indispensable de leurs bonnes propriétés statistiques. Ce choix est d'autant plus important que ces générateurs seront utilisés un très grand nombre de fois et qu'il s'est avéré que certaines méthodes proposées fournissent en réalité des générateurs défaillants.

A.2 Méthodes non itératives de génération d'autres lois

Pour obtenir des nombres distribués selon des lois autres que la loi uniforme sur $[0, 1]$, il existe plusieurs méthodes qui dépendent des propriétés et de la connaissance que l'on a de la loi de densité visée, dénotée $p(\cdot)$. Nous exposons dans un premier temps la méthode d'inversion qui repose sur l'hypothèse forte que la fonction de répartition soit inversible. Dans un second temps, nous nous intéresserons à la génération de nombres selon une loi de densité $p(\cdot)$ que l'on ne sait évaluer qu'à une constante de normalisation près, ce que réalisent la méthode d'acceptation-rejet et la méthode d'échantillonnage pondéré.

A.2.1 Méthode d'inversion

La méthode la plus simple est la méthode d'inversion de la fonction de répartition qui n'est applicable que si la fonction de répartition de la loi visée est inversible. Elle repose sur la proposition suivante:

Proposition 1

Soit X une variable aléatoire de fonction de répartition F_X inversible et U une variable aléatoire de distribution uniforme sur $[0, 1]$. La variable $F_X^{-1}(U)$ a alors la même loi que X .

Preuve :

En effet on a,

$$\mathbb{P}(F_X^{-1}(U) \leq x) = \mathbb{P}(U \leq F_X(x)) \triangleq F_X(x) \quad (\text{A. 2})$$

Si la loi X est discrète, sa fonction de répartition présente une discontinuité en chaque point de masse de la distribution. Après avoir simulé un nombre pseudo-aléatoire uniformément sur $[0, 1]$, on affecte à X la valeur telle que, pour $\epsilon > 0$, on ait:

$$\mathbb{P}(x - \epsilon) < u \leq \mathbb{P}(x) \quad (\text{A. 3})$$

Ce principe est illustré dans la figure A.1 dans le cas d'une loi discrète et l'algorithme est décrit en figure A.2. On y désigne par $U_{[0, 1]}$ la loi uniforme sur $[0, 1]$. Cette méthode est par exemple utilisée pour générer un échantillon selon la loi exponentielle selon l'algorithme décrit en figure A.3.

Il est cependant fréquent que la fonction de répartition de la loi d'intérêt $p(\cdot)$ soit non inversible. On trouvera dans [22] un exposé pour de nombreuses lois des méthodes qui leur sont applicables. Nous nous intéressons maintenant au cas où la loi $p(\cdot)$ n'est connue qu'à une constante de normalisation près, ce qui est souvent le cas en analyse d'images par

exemple. On écrit alors $p(x) = C_p \tilde{p}(x)$ où C_p est la constante inconnue et où l'on suppose savoir évaluer $\tilde{p}(x)$ pour tout x .

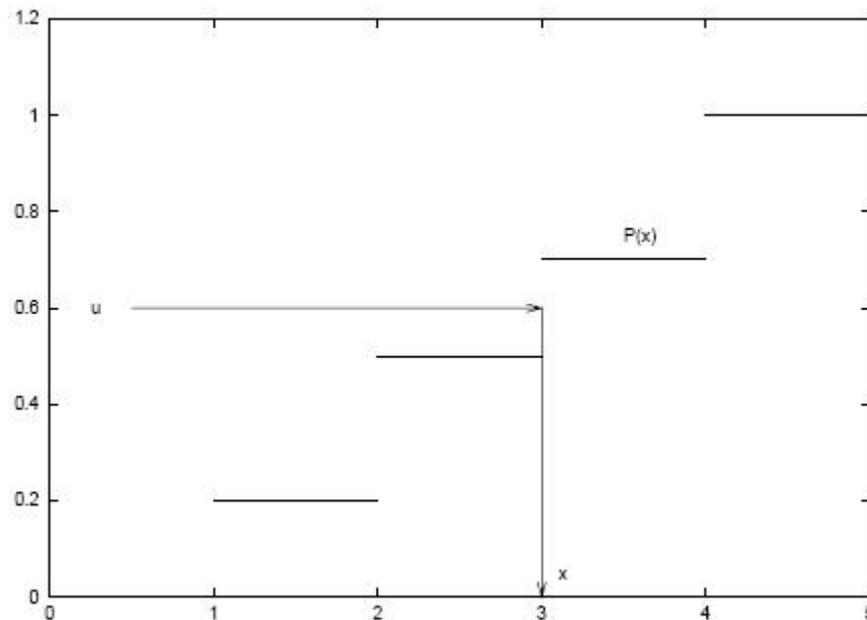


Fig. A.1 : Génération de nombre pseudo-aléatoires distribués selon une loi discrète par la méthode d'inversion de la fonction de répartition

Requiert : fonction de répartition F inversible, loi uniforme sur $[0, 1]$.

Retourne : s distribué selon la loi $p(\cdot)$ de fonction de répartition F .

1. Générer une réalisation u selon la loi uniforme sur l'intervalle $[0, 1]$, dénoté $U_{[0, 1]}$.
2. Retourner $s = F^{-1}(u)$.

Fig. A.2 : Méthode d'inversion

Requiert : $U_{[0, 1]}$.

Retourne : s distribué selon la loi $p(x) = \exp(-x)$.

1. Générer une réalisation u selon $U_{[0, 1]}$.
2. Retourner $s = -\log(u)$.

Fig. A.3 : Génération selon la loi exponentielle par la méthode d'inversion

Nous supposons dans la suite de ce paragraphe que l'on sait tirer des échantillons selon

une loi $g(\cdot)$ dont le support contient le support de $p(\cdot)$. Autrement dit, $g(x) = 0 \Rightarrow p(x) = 0$. On peut de la même façon ne connaître $g(\cdot)$ qu'à une constante de normalisation près et l'on note $g(x) = C_g \tilde{g}(x)$. Le ratio des deux densités $q(x) = \frac{p(x)}{g(x)}$, i.e., la dérivée de Radon-Nikodym, est bien définie et l'on suppose savoir évaluer le ratio non normalisé $\tilde{q}(x) = \frac{\tilde{p}(x)}{\tilde{g}(x)}$ en tout x . Les méthodes non itératives suivantes permettent de générer des réalisations indépendantes distribuées selon la loi $p(\cdot)$, à partir de réalisations de $\tilde{g}(\cdot)$.

A.2.2 Algorithme d'acceptation /rejet

Supposons dans un premier temps que l'on soit en mesure de borner supérieurement le ratio $\tilde{q}(x)$ par une constante c indépendante de x . L'algorithme d'acceptation/rejet consiste alors à suivre les étapes décrites en figure A.4 [5] :

Proposition 2

La loi des réalisations s obtenues en sortie de l'algorithme d'acceptation/rejet est la loi de densité $p(x)$.

Requiert : savoir échantillonner selon $\tilde{g}(\cdot)$ et $U_{[0,1]}$, connaître c tel que

$$c \geq \tilde{p}(x)/\tilde{g}(x) \text{ pour tout } x.$$

Retourne : s distribué selon la loi $p(\cdot)$.

1. Générer une réalisation s selon $\tilde{g}(\cdot)$.
2. Générer une réalisation u selon la loi uniforme sur l'intervalle $[0, 1]$.
3. Si $u < \tilde{q}(s)/c$ alors accepter la réalisation s , sinon le rejet et reprendre en 1

Fig. A.4 : Algorithme d'acceptation/rejet.

Preuve :

Soit \mathcal{S} un Borélien de $\mathcal{B}(\mathbb{R}^{n_x})$ et S la variable aléatoire associée aux réalisations obtenues par l'algorithme d'acceptation/rejet. On a alors:

$$\begin{aligned} \mathbb{P}(S \in \mathcal{S}, S \text{ acceptée}) &= \int_{\mathcal{S}} \int_0^1 \tilde{g}(s) \chi_{u \leq \frac{\tilde{q}(s)}{c}}(u) du ds \\ &= \int_{\mathcal{S}} \tilde{g}(s) \frac{\tilde{q}(s)}{c} ds \end{aligned} \quad (\text{A.4})$$

$$= \frac{1}{c} \int_{\mathcal{S}} \tilde{p}(s) ds$$

où χ désigne la fonction indicatrice. De la même façon on obtient que :

$$\mathbb{P}(S \text{ acceptée}) = \frac{1}{c} \int_{\mathbb{R}^{n_x}} \tilde{p}(s) ds = \frac{1}{c} \frac{1}{C_p}. \quad (\text{A.5})$$

Par conséquent:

$$\mathbb{P}(S \in \mathcal{S} | S \text{ acceptée}) = C_p \int_{\mathcal{S}} \tilde{p}(s) ds = \int_{\mathcal{S}} p(s) ds \quad (\text{A.6})$$

Et S est bien distribuée selon la loi de densité $p(\cdot)$.

D'après (1.5), la probabilité d'acceptation d'une réalisation selon $\tilde{g}(\cdot)$ vaut $(cC_p)^{-1}$ donc l'algorithme est d'autant plus efficace que la fonction $\tilde{g}(\cdot)$ est proche de la loi d'intérêt $\tilde{p}(\cdot)$.

Exemple 1 Supposons que l'on ne connaisse pas la constante de normalisation de la loi normale centrée réduite. Soit $\tilde{p}(x) = \exp(-x^2/2)$ et $\tilde{g}(x) = 1/(1+x^2)$. On peut alors majorer le ratio selon :

$$\frac{\tilde{p}(x)}{\tilde{g}(x)} = (1+x^2)\exp(-x^2/2) \leq 2/\sqrt{e} \quad (\text{A.7})$$

et utiliser l'algorithme d'acceptation/rejet correspondant pour générer des réalisations selon la loi centrée réduite.

La probabilité d'acceptation de l'algorithme d'acceptation/rejet associé vaut alors 0.66

A.2.3 Echantillonnage pondéré

On relâche maintenant l'hypothèse sur la connaissance de la borne c . L'échantillonnage pondéré, dénommé importance sampling et dénoté IS en anglais, consiste à suivre l'algorithme décrit en figure A.5. [24] et [26].

On a alors la proposition suivante:

Requiert : savoir échantillonner selon $\tilde{g}(\cdot)$, savoir évaluer $\tilde{q}(\cdot)$ en tout point.

Retourne : un échantillon (s^1, \dots, s^N) distribué selon $p(\cdot)$.

1. Générer N réalisations (s^1, \dots, s^N) selon $\tilde{g}(\cdot)$.

2. Affecter à chacune de ces réalisations le poids normalisé : $q(s^n) = \frac{\tilde{q}(s^n)}{\sum_{n=1}^N \tilde{q}(s^n)}$

Fig. A.5 : algorithme d'échantillonnage pondéré

Proposition 3

Si s^1, \dots, s^N et q^1, \dots, q^N sont obtenus par échantillonnage pondéré, alors la loi discrète $\sum_{n=1}^N q(s^n) \delta_{s^n}$, ou δ_s désigne la loi de Dirac centrée en s , converge en loi vers la loi de densité $p(\cdot)$ lorsque $N \rightarrow \infty$.

Preuve :

Soit \mathcal{S} un borélien de $\mathcal{B}(\mathbb{R}^{n_x})$ et S la variable aléatoire associée aux réalisations obtenus par l'algorithme d'échantillonnage pondéré. On a alors d'après la loi des grands nombres (A.1)

$$\begin{aligned} \mathbb{P}(S \in \mathcal{S}) &= \sum_{n=1}^N \chi_{\mathcal{S}}(s^n) \frac{\tilde{q}(s^n)}{\sum_{v=1}^N \tilde{q}(s^v)} = \frac{\frac{1}{N} \sum_{n=1}^N \chi_{\mathcal{S}}(s^n) \tilde{q}(s^n)}{\frac{1}{N} \sum_{n=1}^N \tilde{q}(s^n)} \\ &\xrightarrow{N \rightarrow \infty} \frac{\mathbb{E}_{\tilde{g}} \left(\frac{\tilde{p}(s)}{\tilde{g}(s)} \chi_{\mathcal{S}}(s) \right)}{\mathbb{E}_{\tilde{g}} \left(\frac{\tilde{p}(s)}{\tilde{g}(s)} \right)} = \mathbb{E}_p(\chi_{\mathcal{S}}(s)). \end{aligned} \quad (\text{A. 8})$$

A.2.4 Efficacité et taille efficace de l'échantillon

Soit ϕ une fonction intégrable par rapport à $p(\cdot)$ et $\mu = \mathbb{E}_p[\phi(X)]$ la quantité que l'on souhaite estimer. Supposons que l'on sache directement échantillonner selon la loi $p(\cdot)$ et définissons $\hat{\mu}_p^{N_1}(\phi)$ l'estimateur associé à un échantillon p^1, \dots, p^{N_1} de taille N_1 distribué selon $p(\cdot)$:

$$\hat{\mu}_p^{N_1}(\phi) = \frac{1}{N_1} \sum_{n=1}^{N_1} \phi(p^n) \quad (\text{A. 9})$$

Soit $\hat{\mu}_g^N(\phi)$ l'estimateur de $\mathbb{E}[\phi(X)]$ fourni par un échantillon s^1, \dots, s^N de taille N issue d'une autre méthode de Monte Carlo, par exemple l'une des deux méthodes précédentes. On définit alors l'efficacité de cette méthode de simulation par le ratio:

$$\frac{\text{var}_g \left(\hat{\mu}_g^N(\phi) \right)}{\text{var}_p \left(\hat{\mu}_p^{N_1}(\phi) \right)}. \quad (\text{A. 10})$$

La variance de l'estimateur direct $\hat{\mu}_p^{N_1}(\phi)$ est :

$$\text{var}_p \left(\hat{\mu}_p^{N_1}(\phi) \right) = \frac{\text{var}_p \phi(x)}{N_1}. \quad (\text{A. 11})$$

Des approximations seront souvent nécessaires pour évaluer la variance de l'estimateur $\hat{\mu}_g^N(\phi)$.

On définit ensuite la taille efficace de l'échantillon N_{eff} comme la taille que devrait avoir l'échantillon associé à $\hat{\mu}_p^{N_1}(\phi)$ pour que le ratio $\frac{var_g(\hat{\mu}_g^N(\phi))}{var_p(\hat{\mu}_p^{N_1}(\phi))}$ soit égal à un. Intéressons-nous à l'efficacité de la méthode d'échantillonnage pondéré. L'égalité suivante est montrée dans [60] :

$$var_g(\hat{\mu}_g^N(\phi)) = \frac{\mathbb{E}_g[(\phi(X) - \mu)^2 \tilde{q}(X)]}{N} + O(1/N^{3/2}). \quad (\text{A. 12})$$

Cette égalité est obtenue par linéarisation selon la méthode delta du ratio suivant, où X et Y sont deux sommes de variables aléatoires i.i.d. de taille N :

$$var\left(\frac{X}{Y}\right) = \frac{var(X)}{\mathbb{E}[Y]^2} + \frac{\mathbb{E}[X]^2 var(Y)}{\mathbb{E}[Y]^4} - 2 \frac{cov(X, Y)}{\mathbb{E}[Y]^3} + O(1/N^{3/2}). \quad (\text{A. 13})$$

L'approximation suivante est ensuite proposée dans [40] à condition que la fonction $\phi(x)$ varie peu avec x :

$$\mathbb{E}_g[(\phi(X) - \mu)^2 \tilde{q}(X)] \approx var_p(\phi(X)) \left(1 + var_g(\tilde{q}(X))\right) \quad (\text{A. 14})$$

Le principal avantage de cette approximation est qu'elle ne dépend plus de la fonction ϕ considérée.

Par définition, la taille efficace N_{eff} de l'échantillon tiré par échantillonnage pondéré utilisant l'approximation (A.14) est alors la solution en N_1 de l'équation:

$$\frac{var_p \phi(x) \left(1 + var_g(\tilde{q}(x))\right)}{N} = \frac{var_p \phi(x)}{N_1} \quad (\text{A. 15})$$

Soit :

$$N_{eff} = \frac{N}{1 + var_g(\tilde{q}(x))}. \quad (\text{A. 16})$$

A.3 Méthodes itératives : chaînes de Markov de Monte Carlo

Les méthodes de Monte Carlo itératives par chaînes de Markov consistent à générer une chaîne de Markov $(X_k)_{k \in \mathbb{N}}$ qui converge vers la loi d'intérêt $p(\cdot)$. Contrairement aux méthodes non itératives précédentes, les réalisations successives de telles méthodes sont par nature dépendantes. Nous rappelons en premier lieu quelques définitions et propriétés utiles des chaînes de Markov à valeurs sur un espace d'état continu. Nous exposons ensuite

un algorithme de Monte Carlo par chaîne de Markov, à savoir l'algorithme de Metropolis Hastings.

A.3.1 Rappel sur les chaînes de Markov

Définition 2

Soit $(X_k)_{k \in \mathbb{N}}$ un processus défini sur l'espace probabilisé $(D, \mathcal{F}, \mathbb{P})$ et à valeurs dans \mathbb{R}^{n_x} . Le processus $(X_k)_{k \in \mathbb{N}}$ est une chaîne de Markov d'ordre 1 si pour tout $k \in \mathbb{N}$ et tout borélien \mathcal{S} :

$$\mathbb{P}(X_{k+1} \in \mathcal{S} | X_0, \dots, X_k) = \mathbb{P}(X_{k+1} | X_k) \quad (\text{A.17})$$

La distribution de la chaîne est entièrement déterminée par la distribution initiale de X_0 et par son noyau de transition $P_k(x, \mathcal{S}) \triangleq \mathbb{P}(X_{k+1} \in \mathcal{S} | X_k = x)$ pour tout $k \in \mathbb{N}$, $x \in \mathbb{R}^{n_x}$ et tout borélien \mathcal{S} . Le noyau de transition vérifie les deux propriétés suivantes :

- Pour tout $k \in \mathbb{N}$ et tout $x \in \mathbb{R}^{n_x}$, $P_k(x, \cdot)$ est une probabilité.
- Pour tout $k \in \mathbb{N}$ et tout borélien \mathcal{S} , la fonction $P_k(\cdot, \mathcal{S})$ est mesurable et bornée.

La chaîne sera dite homogène si son noyau ne dépend pas de k

On suppose dans la suite que le noyau admet une densité et s'écrit donc $P_k(X_k, dx) = P_k(X_k, x)dx$

Définition 3

Une chaîne de Markov homogène est de distribution invariante $p(\cdot)$ si son noyau de transition vérifie :

$$p(\mathcal{S}) = \int_{\mathbb{R}^{n_x}} P(x, \mathcal{S})p(x)dx \quad (\text{A.18})$$

Pour toute partie mesurable \mathcal{S} de \mathbb{R}^{n_x}

Définition 4

Si l'on note $P^k(X_0, \mathcal{S}) \triangleq \mathbb{P}(X_k \in \mathcal{S} | X_0)$ la probabilité conditionnelle de X_k sachant X_0 , alors une distribution invariante $p(\cdot)$ est une distribution d'équilibre si elle vérifie pour $p(\cdot)$ presque tout x :

$$\lim_{k \rightarrow \infty} P^k(x, \mathcal{S}) = p(\mathcal{S}). \quad (\text{A.19})$$

On dit alors que la chaîne $(X_k)_{k \in \mathbb{N}}$ converge vers $p(\cdot)$. Les méthodes de Monte Carlo itératives par chaînes de Markov consistent à générer une chaîne de Markov $(X_k)_{k \in \mathbb{N}}$ de distribution invariante égale à $p(\cdot)$ et qui converge vers $p(\cdot)$ indépendamment de l'état

initial X_0 . Il faut pour cela trouver le noyau de transition adéquat de telle sorte que la chaîne construite converge vers $p(\cdot)$. Supposons que ce noyau s'exprime sous la forme:

$$P(x, dy) = g(x, y)dy + r(x)\delta_x(dy) \quad (\text{A. 20})$$

Où $g(x, x) = 0$. La probabilité que la chaîne reste en x vaut ainsi $r(x) = 1 - \int_{\mathbb{R}^{n_x}} g(x, y)dy$ et peut être non nulle si $\int_{\mathbb{R}^{n_x}} g(x, y)dy < 1$. On a alors la proposition suivante [42] :

Proposition 4

Si $g(x, y)$ vérifie la propriété d'équilibre détaillée :

$$p(x)g(x, y) = p(y)g(y, x) \quad (\text{A. 21})$$

Alors $p(\cdot)$ est une distribution invariante pour la chaîne de Markov associée au noyau de transition $P(\cdot, \cdot)$.

Pour que la chaîne converge vers $p(\cdot)$ sachant que $p(\cdot)$ est invariante pour $P(\cdot, \cdot)$, il est suffisant qu'elle vérifie les propriétés d'irréductibilité et d'apériodicité suivantes:

Définition 5

Une chaîne $(X_k)_{k \in \mathbb{N}}$ de distribution invariante $p(\cdot)$ est dite irréductible si elle admet une probabilité positive d'accéder à toute partie S de probabilité positive pour $p(\cdot)$.

Définition 6

Une chaîne $(X_k)_{k \in \mathbb{N}}$ est dite périodique s'il existe des parties de l'espace d'état qui ne peuvent être visitées qu'à des intervalles de temps périodiques. Sinon, elle est dite apériodique.

Une chaîne irréductible et apériodique est donc une chaîne telle que, si x et y sont de probabilité positive pour $p(\cdot)$, alors il existe une suite finie de mouvements de probabilité positive permettant de passer de x à y . De plus, la longueur de cette suite ne peut être contrainte à être un multiple d'un entier donné. Sous ces conditions de régularité, on a alors la proposition suivante:

Proposition 5

Si une chaîne est homogène $(X_k)_{k \in \mathbb{N}}$ de distribution invariante $p(\cdot)$ est irréductible et apériodique alors $p(\cdot)$ est l'unique distribution invariante. De plus $p(\cdot)$ est alors la distribution d'équilibre de la chaîne.

Pour une présentation détaillée des chaînes de Markov, on pourra par exemple se référer à [38], [55].

A.3.2 Algorithme de Metropolis-Hastings

L'algorithme de Metropolis-Hastings, initialement proposé dans [6] a été intensivement utilisé en physique avant de devenir populaire dans le domaine de la statistique, notamment grâce à la généralisation de [9]. La présentation suivante s'inspire de l'exposé remarquablement clair de l'algorithme qu'on trouvera dans [45]. On pourra également consulter [42] pour des aspects plus théoriques.

L'algorithme repose sur la propriété suivante, introduite dans le paragraphe précédent Soit $g_{des}(\cdot, \cdot)$ une densité telle que

$$\int g_{des}(x, y) dy = 1 \quad (\text{A. 22})$$

Si $g_{des}(\cdot, \cdot)$ satisfait directement la propriété d'équilibre détaillé (A.21), alors la chaîne de Markov de noyau P défini en (A.20) converge vers $p(\cdot)$ sous les hypothèses d'irréductibilité et d'apériodicité. Dans le cas général où la propriété d'équilibre détaillé n'est pas vérifiée, l'algorithme de Metropolis-Hastings effectue un rééquilibrage en favorisant certains mouvements de x vers y et en défavorisant d'autres par le principe d'acceptation-rejet. Supposons que pour un couple (x, y) on ait la relation:

$$p(x)g_{des}(x, y) > p(y)g_{des}(y, x). \quad (\text{A. 23})$$

Intuitivement, ce "déséquilibre" signifie que les mouvements de x vers y sont réalisés trop souvent et ceux de y vers x trop rarement. On peut alors réduire les mouvements de x vers y en ne les réalisant qu'avec la probabilité $\alpha(x, y)$. Si le mouvement n'est pas accepté, la chaîne demeure en x . On pose ainsi $g_{eq} = g_{des}(x, y)\alpha(x, y)$ et l'on va définir $\alpha(x, y)$ de telle sorte que g_{eq} vérifie la propriété d'équilibre détaillée et puisse déterminer le noyau de transition adéquat selon (A.20). Afin d'encourager les mouvements de y vers x , nous avons intérêt à prendre $\alpha(y, x)$ aussi élevée que possible, soit égal à 1. L'équation d'équilibre détaillé détermine alors uniquement $\alpha(x, y)$:

$$p(x)g_{des}(x, y)\alpha(x, y) = p(y)g_{des}(y, x)\alpha(y, x) = p(y)g_{des}(y, x) \quad (\text{A. 24})$$

d'où $\alpha(x, y) = \frac{p(y)g_{des}(y, x)}{p(x)g_{des}(x, y)}$.

Si l'inégalité (A.23) est inversée alors on obtient de même $\alpha(y, x) = \frac{p(y)g_{des}(y, x)}{p(x)g_{des}(x, y)}$

1. Finalement, en posant :

$$\alpha(x, y) = \begin{cases} \min \left\{ 1, \frac{p(y)g_{des}(y, x)}{p(x)g_{des}(x, y)} \right\} & \text{si } p(x)g_{des}(x, y) > 0; \\ 1 & \text{sinon} \end{cases} \quad (\text{A.25})$$

on est assuré que g_{eq} vérifie (A.21).

Le noyau de transition :

$$P(x, dy) = g_{des}(x, y)\alpha(x, y)dy + \left(1 - \int_{\mathbb{R}^n} g_{des}(x, y)\alpha(x, y)dy\right)\delta_x(dy) \quad (1.26)$$

admet alors $p(\cdot)$ comme distribution invariante. On peut remarquer que l'évaluation de $\alpha(x, y)$ ne nécessite pas de connaître la constante de normalisation de $p(\cdot)$, étant donné que cette distribution est présente au numérateur et au dénominateur. On peut ici remarquer une différence avec l'algorithme d'acceptation/rejet qui reconsidère un nouveau couple (s^*, u) lorsque le précédent s^* est rejeté. La figure A.6 résume les étapes de l'algorithme générique de Metropolis-Hastings.

Initialement, l'algorithme proposé dans [6] consistait à utiliser des fonctions $g_{des}(x, y)$ symétriques de telles sortes que la probabilité d'acceptation $\alpha(x, y)$ soit alors réduite à $p(y)/p(x)$.

Dans le cas particulier où $g_{des}(x, y) = g_{des}(y)$ [9], les candidats sont, en réalité, tirés selon une loi indépendante de x . La probabilité d'acceptation est alors de la forme:

$$q(x, y) = \min \left\{ 1, \frac{\tilde{q}(y)}{\tilde{q}(x)} \right\}, \quad (1.27)$$

1. Choisir un point initial arbitraire X_0 et poser $k = 0$.
2. Générer un candidat X^* selon $g_{des}(X_k, \cdot)$ et une réalisation u selon la loi uniforme sur $[0, 1]$.
3. Poser $X_{k+1} = X^*$ si $u \leq \alpha(X_k, X^*)$ et $X_{k+1} = X_k$ sinon, où la probabilité d'acceptation est donnée par :
$$\alpha(x, y) = \begin{cases} \min \left\{ 1, \frac{p(y)g_{des}(y, x)}{p(x)g_{des}(x, y)} \right\} & \text{si } p(x)g_{des}(x, y) > 0; \\ 1 & \text{sinon} \end{cases} \quad (\text{A.28})$$
4. Poser $k = k + 1$ et reprendre en 2.

Fig. A.6 : Algorithme de Metropolis- Hastings

où $\tilde{q}(x) = \frac{p(x)}{g_{des}(x)}$. Cette fonction $\tilde{q}(x)$ correspond ainsi à la fonction de poids intervenant dans l'échantillonnage pondéré selon $g_{des}(\cdot)$.

Comme toute méthode itérative de Monte Carlo, il est nécessaire de ne pas prendre en compte les premières réalisations, trop dépendantes de la condition initiale et qui constitue la période dite de transition. La convergence de la chaîne a lieu sous les hypothèses d'irréductibilité et d'apériodicité [34], [42]. La vitesse de convergence dépend alors intimement du choix de la fonction $g_{des}(\cdot)$. Pour autant, on ne sait pas de manière générale décider de la convergence effective de chaînes produites par un algorithme de type Metropolis-Hastings. Dans la pratique, plusieurs techniques sont envisageables: on peut générer plusieurs chaînes issues d'états initiaux différents et constituer un échantillon à partir des réalisations finales obtenues pour chaque chaîne au bout d'un nombre d'itérations fixé. Les réalisations obtenues sont alors censées être décorréélées, au prix du temps de calcul dépensé pour toutes les phases de transition initiales. Il est également possible de ne générer qu'une seule chaîne et de n'en conserver les réalisations que périodiquement afin de diminuer leur auto-corrélation.