Université d'Artois

Thèse de Doctorat

Suivi et Classification d'Objets Multiples : Contributions avec la Théorie des Fonctions de Croyance

Auteur :
Samir Hachour

Directeur :
Pr. François Delmotte

Thèse présentée en vue de l'obtention du grade de Docteur de :

Université d'Artois
Laboratoire de Génie Informatique et d'Automatique de l'Artois

Membres du jury :

M.	Didier Maquin	Professeur à l'université de Lorraine	président
M.	Thierry Denœux	Professeur à l'université de Technologie de Compiègne	rapporteur
M.	Arnaud Martin	Professeur à l'université de Rennes 1	rapporteur
M.	Jean-Charles Noyer	Professeur à l'université du littorale côte d'opale	examinateur
M.	François Delmotte	Professeur à l'université d'Artois	directeur
M.	David Mercier	Maître de conférences à l'université d'Artois	encadrant

Table des matières

Table des matières ii

Liste des figures vii

Liste des tableaux ix

Liste des acronymes xi

Liste des symboles xiii

Introduction Générale 1
Motivations . . . 1
Contribution de la thèse . . . 3

1 Suivi d'objets Multiples : état de l'art 7
1.1 Éléments du filtrage Bayésien . . . 8
1.1.1 Estimation linéaire . . . 10
1.1.1.1 Filtrage de Kalman . . . 11
1.1.2 Estimation non-linéaire . . . 13
1.1.2.1 Filtrage de Kalman étendu . . . 13
1.1.2.2 Filtrage particulaire . . . 15
1.1.3 Estimation multi-modale . . . 19
1.1.3.1 Modèles interactifs (*Interacting Multiple Model IMM*) . . . 20
1.1.3.2 Filtre de Rao-Blackwilize . . . 23
1.2 Association des observations aux objets . . . 24
1.2.1 Cas mono-objet . . . 25
1.2.1.1 Algorithme du plus proche voisin (*Nearest Neighbor NN*) . . . 25
1.2.1.2 Association de données probabiliste (*Probabilistic Data Association PDA*) . . . 26
1.2.2 Cas multi-objets . . . 26
1.2.2.1 Algorithme du plus proche voisin global (*Global Nearest Neighbor GNN*) . . . 28
1.2.2.2 Association de données probabiliste conjointe (*Joint Probabilistic Data Association JPDA*) . . . 30
1.2.2.3 Suivi Multi-Hypothèses (*Multi-Hypothesis Tracking MHT*) . . . 31
1.2.3 Suivi multi-objets sans association de données (*Probability Density Hypothesis PHD*) . . . 34

1.3 Gestion d'apparitions et de disparitions d'objets . . . 35
1.3.1 Test du ratio de vraisemblances . . . 36
1.3.2 Ratio de vraisemblances pour la confirmation d'objets . . . 36
1.4 Conclusion . . . 38

2 Classification à base de données cinématiques 39
2.1 Notions de base sur les fonctions de croyance . . . 40
2.2 Représentation de l'informations . . . 41
2.2.1 Information au niveau crédal . . . 42
2.2.1.1 Règles de combinaison . . . 42
2.2.1.2 Révision de connaissances . . . 43
2.2.2 Inférence crédale . . . 45
2.2.3 Information au niveau pignistique . . . 47
2.3 Solution pour le suivi et la classification de plusieurs objets à la fois . . . 47
2.3.1 Solution pour le suivi d'objets multiples . . . 48
2.3.2 Classification à base de données cinématiques . . . 53
2.3.2.1 Classifieur Bayésien . . . 53
2.3.2.2 Classifieur Crédal . . . 55
2.4 Application 1 : suivi et classification de cibles aériennes (classes constantes) . . . 57
2.4.1 Description . . . 57
2.4.2 Conditionnement de la connaissance . . . 60
2.4.3 Résultats de simulations . . . 61
2.5 Application 2 : suivi et classification de piétons (classes pouvant changer dans le temps)) . . . 64
2.5.1 Description . . . 64
2.5.2 Conditionnement de l'information . . . 66
2.5.3 Simulation et résultats . . . 67
2.6 Classification & fausses associations . . . 70
2.7 Conclusion . . . 71

3 Algorithmes d'association crédaux 73
3.1 Association des observations aux objets connus (association mono-capteur) . . . 74
3.1.1 Association non conflictuelle . . . 76
3.1.2 Association conflictuelle . . . 76
3.2 Association d'objets suivis par différents capteurs (association multi-capteurs) . . . 77
3.3 Description des algorithmes d'association crédale . . . 79
3.3.1 Méthode de Denœux et *al* [57, 67] : . . . 79
3.3.2 Méthode de Mercier et *al* [125] : . . . 81
3.3.3 Méthode de Lauffenburger et *al* [105] . . . 82
3.3.4 Méthode de Fayad et Hamadeh [69] : . . . 85
3.3.5 Méthode proposée par Dallil et *al* [46] : . . . 87
3.3.6 Solution proposée 1 : . . . 89
3.3.7 Solution proposée 2 : . . . 91
3.4 Mise en correspondance du paramètre γ propre à certaines solutions crédales et du paramètre λ de l'algorithme *GNN* . . . 93
3.5 Comparaisons dans un cadre d'association mono-capteur . . . 94

3.5.1 Apprentissage partiel des paramètres : scénario d'objets à trajectoires confondues . . . 94
3.5.2 Apprentissage partiel des paramètres : gestion de la réapparition d'objets . 96
3.5.3 Apprentissage complet des paramètres : objets à trajectoires confondues et gestion de réapparition . . . 97
3.5.3.1 Tests de robustesse . . . 99
3.5.4 Test sur des données de simulation issues de la littérature [24, 25] . . . 102
3.5.4.1 Apprentissage des paramètres et test de robustesse pour le scénario "*benchmark*" . . . 102
3.5.4.2 Test de robustesse sans apprentissage de paramètres . . . 103
3.5.4.3 Impacts des fausses décisions sur le suivi de trajectoires . . . 104
3.5.4.4 Comparaison des algorithmes en terme de complexité calculatoire 107
3.6 Conclusion . . . 108

4 Systèmes multi-capteurs pour le suivi et la classification d'objets 111
4.1 Aperçu de l'état de l'art concernant le suivi multi-objets et multi-capteurs . . . 113
4.2 Approche centralisée proposée . . . 115
4.2.1 Suivi local au niveau des capteurs . . . 115
4.2.2 Classification locale au niveau des capteurs . . . 116
4.2.3 Application de l'algorithme proposé 1 pour l'association multi-capteurs (track-to-track) . . . 116
4.2.3.1 Association multi-capteurs : utilisation des distances (information de base) . . . 117
4.2.3.2 Association multi-capteurs : intégration de l'information concernant les classes des objets (information additionnelle) . . . 117
4.2.3.3 Association multi-capteurs : résultat de simulation (avec/sans information additionnelle) . . . 119
4.2.4 Classification globale (fusion Bayésienne) . . . 119
4.2.4.1 Fusion conjonctive Bayésienne . . . 120
4.2.4.2 Autres règles de fusions Bayésiennes . . . 120
4.2.5 Classification globale (fusion crédale) . . . 120
4.2.6 Simulation et résultats . . . 121
4.2.7 Résultats de classification locale . . . 121
4.2.8 Résultats de classification globale . . . 124
4.3 Approche distribuée proposée . . . 125
4.3.1 Algorithme de consensus . . . 126
4.3.1.1 Algorithme de consensus asymptotique . . . 127
4.3.1.2 Algorithme de consensus en temps fini . . . 127
4.3.2 Suivi distribué basé sur l'algorithme de consensus en temps fini . . . 129
4.3.2.1 Indépendance des observations locales . . . 130
4.3.2.2 Phase de prédiction . . . 131
4.3.2.3 Échange et arrangement des prédictions . . . 132
4.3.2.4 Agrégation des observations locales . . . 133
4.3.2.5 Mise à jour de l'état des objets . . . 133
4.3.3 Classification distribuée basée sur l'algorithme de consensus . . . 134
4.3.4 Simulation et résultats . . . 135

4.3.4.1 Simulation 135
4.3.4.2 Résultats de simulations 136
4.4 Conclusion 139

Conclusion Générale **141**
Aboutissements 141
Perspectives 143

A Différentes dynamiques d'objets en mouvement **147**
A.1 Modèle de Singer [164] 147
A.2 Modèle de vitesse constante et modèle d'accélération constante 149
A.3 Modèle d'un objet en mouvement circulaire 150

Références bibliographiques **153**

Abstract **170**

Liste des figures

1 Quelques applications du suivi et de la classification d'objets. 2

1.1 Organigramme des méthodes principales dédiées au suivi d'objets. 8
1.2 Filtre de Kalman vs filtre particulaire. 19
1.3 Illustration du suivi mono-objet 25
1.4 Illustration du suivi multi-objets. 26
1.5 Génération d'hypothèses dans une méthode *MHT*. 33
1.6 Illustration de l'évolution de la fonction score d'un objet donné. 38

2.1 Avion de ligne escorté par deux avions de chasse. 39
2.2 Représentation de l'information dans le cadre des fonctions de croyance. 41
2.3 Solution pour le suivi de plusieurs objets à la fois. 49
2.4 Exemple de quatre objets manœuvrants. 51
2.5 Fonctions score des objets de l'exemple de la figure 2.4. 51
2.6 *IMM* classique vs *IMM* crédal [134]. 53
2.7 Organigramme du classifieur Bayésien. 54
2.8 Organigramme du classifieur crédal. 55
2.9 Les différents modes d'évolution et définition des classes pour l'application 1. . . . 59
2.10 Exemple de simulation : quatre objets manœuvrants. 62
2.11 Classification Bayésienne basée sur les comportements de cibles aériennes (application 1). 63
2.12 Classification Bayésienne vs classification crédale (application 1). 64
2.13 Les différents modes d'évolution et définition des classes pour l'application 2. . . . 65
2.14 Trajectoires de deux piétons sur le plan (x, y). 67
2.15 Classification crédale sans/avec affaiblissement de connaissances (application 2). . 68
2.16 Modifications du classifieur crédal permettant l'adaptation au changement de classe (application 2). 69
2.17 Fausse association entre les observations d'un avion de chasse et d'un avion de ligne. 70
2.18 Effet des fausses associations sur la classification. 71

3.1 Illustration du problème d'association 73
3.2 Association basée sur les distances dans un cas non conflictuel. 77
3.3 Association basée sur les distances dans un cas conflictuel. 77
3.4 Association basée sur plusieurs informations. 78
3.5 Association concernant une seule observation. 86
3.6 Illustration de la relation entre γ et λ. 93
3.7 Influence des paramètres μ, γ et λ sur les performances des algorithmes pour le suivi de deux trajectoires conflictuelles. 95

3.8 Influence des paramètres μ, γ et λ quant à la gestion de la disparition, la réapparition et l'apparition d'objets. . . . 96
3.9 Apprentissage des valeurs optimales des paramètres μ et γ. . . . 97
3.10 Influence des paramètres μ, γ et λ sur les performances des différents algorithmes (zone rouge : zone défavorable à la méthode proposée 1, zone verte : zone favorable à la méthode proposée 1). . . . 98
3.11 Robustesse par rapport aux erreurs de mesure. . . . 100
3.12 Test de robustesse combiné comparant la solution proposée 1 à la solution proposée par Denœux et *al.* . . . 101
3.13 Différence entre les taux de fausses décisions présentés par les figures 3.12(b) et 3.12(a) respectivement. . . . 101
3.14 Application des méthodes d'association au suivi d'objets manœuvrants. . . . 102
3.15 Apprentissage de paramètres et test de robustesse correspondant au scénario de la figure 3.14(a). . . . 103
3.16 Test sur un scénario dont les paramètres ne sont pas appris. . . . 104
3.17 Algorithmes d'association et suivi des trajectoires données par la figure 3.14(a). . . . 105
3.18 Estimation des trajectoires données par la figure 3.14(a) (cas d'une mauvaise calibration de paramètres). . . . 106
3.19 *EQM* sur l'estimation de trajectoires avec une mauvaise calibration des algorithmes d'association. . . . 106
3.20 Test de complexité calculatoire. . . . 107

4.1 Illustration d'une solution multi-capteurs centralisée. . . . 111
4.2 Illustration d'une solution multi-capteurs distribuée. . . . 112
4.3 Organigramme de la solution multi-capteurs centralisée proposée. . . . 116
4.4 Taux de fausses décisions obtenus pour des associations mono-information et multi-informations. . . . 119
4.5 Organigramme de la solution proposée réduite au cas de deux capteurs. . . . 121
4.6 Scénario de 3 objets manœuvrants observés par deux capteurs peu fiables. . . . 122
4.7 Résultats de classifications de l'objet 2 obtenus par un seul capteur fiable. . . . 122
4.8 Résultats de classification de l'objet 2 obtenus par un seul capteur non fiable. . . . 123
4.9 Classifications locales Bayésienne et crédale pour différents niveaux de fiabilité de capteur. . . . 123
4.10 Résultats de classifications Bayésiennes et crédales pour différents niveaux de fiabilité de capteurs. . . . 124
4.11 Comparaison entre les meilleures classifications globales Bayésienne et crédale. . . . 125
4.12 Organigramme de la solution distribuée proposée réduite au cas de deux capteurs. 126
4.13 Exemple simplifié de graphe de communication. . . . 128
4.14 Consensus asymptotique vs consensus en temps fini pour le calcul d'une valeur moyenne simplifiée. . . . 129
4.15 Étapes principales de l'algorithme de suivi distribué. . . . 131
4.16 Reconnaissance des comportements de véhicules sur une autoroute. . . . 136
4.17 Suivi d'objets avec et sans consensus. . . . 137
4.18 Reconnaissance des comportements de véhicules sur une autoroute. . . . 138
4.19 Résultats de classifications locales sans consensus. . . . 138
4.20 Classifications avec consensus. . . . 138

Liste des tableaux

1.1 Un exemple de matrice d'association : 3 objets connus $\{\overline{z}_1, \overline{z}_2, \overline{z}_3\}$ et 4 observations $\{z_1, z_2, z_3, z_4\}$. 29

2.1 Solution proposée pour le suivi de plusieurs objets à la fois. 50

3.1 Matrice des probabilités pignistiques. 91
3.2 Temps de calcul en secondes des algorithmes pour des problèmes d'association de dimension $n \times n$. 108

4.1 Phase de prédiction de l'algorithme de suivi distribué. 131
4.2 Phase de mise à jour de l'algorithme de suivi distribué. 134

Liste des acronymes

IMM	**I**nteracting **M**ultiple **M**odel
GPB	**G**eneralized **P**seudo-**B**ayesian
JMLM	**J**ump **M**arkov **L**inear **M**odel
NN	**N**earest **N**eighbor
GNN	**G**lobal **N**earest **N**eighbor
MHT	**M**ulti-**H**ypothesis **T**racking
PMHT	**P**robabilistic **M**ulti-**H**ypothesis **T**racking
PDA	**P**robabilistic **D**ata **A**ssociation
JPDA	**J**oint **PDA**
IPDA	**I**ntegrated **PDA**
IJPDA	**I**ntegrated **JPDA**
EKF	**E**xtended **K**alman **F**ilter
GKF	**G**eneralized **K**alman **F**ilter
FKD	**F**iltre de **K**alman **D**istribué
UKF	**U**scented **K**alman **F**ilter
UT	**U**scented **T**ransform
IEKF	**I**nvariant **E**xtended **K**alman **F**ilter
MCMC	**M**arkov **C**hain **M**onte **C**arlo
PF	**P**article **F**ilter
SIS	**S**equential **I**mportance **S**ampling
RSF	**R**andom **F**init **S**ets
PHD	**P**robability **H**ypothesis **D**ensity
SMC-PHD	**S**equential **M**onte **C**arlo-**PHD**
GM-PHD	**G**aussian **M**ixture-**PHD**

CPHD	**C**ardinal **PHD**
LR	**L**ikelihood **R**atio
LLR	**L**ogarithm **L**ikelihood **R**atio
SF	**S**core **F**unction
DS	**D**empster **S**hafer
MCT	**M**odèle de **C**royances **T**ransferables
BBA	**B**asic **B**elief **A**ssignment
EF	**E**léments **F**ocaux
CRC	**C**onjunctive **R**ule of **C**ombination
BT	**B**ayesian **T**heorem
GBT	**G**eneralized **B**ayesian **T**heorem
EQM	**E**rreur **Q**uadratique **M**oyenne

Liste des symboles

Symboles liés aux fonctions de croyance et aux probabilités

Ω	: cadre de discernement
ω	: singleton appartenant à Ω
A	: ensemble appartenant ou égal à Ω
$\emptyset$	: ensemble vide
2^Ω	: préférentiel de définition
m	: fonction de masse
m_s	: fonction de masse provenant de la source s
bel	: fonction de crédibilité
pl	: fonction de plausibilité
$Betp$	: probabilité pignistique
$P(h_i)$	: probabilité de l'hypothèse h_i
$P(h_i\|h_j)$	: probabilité conditionnelle de l'hypothèse h_i sachant l'hypothèse h_j
$p(x)$	: densité de probabilité de la variable x

Symboles liés au suivi et la classification

$\mathcal{X}$	: espace d'état
$x(k)$	: vecteur d'état à l'instant k
$\hat{x}(k)$	: vecteur d'état estimé à l'instant k
$\hat{x}(k\|k-1)$	: vecteur d'état prédit pour l'instant k
P	: matrice de covariance du vecteur d'état estimé
F	: matrice d'état
u	: entrée déterministe
G	: matrice d'entrée

w	: bruit d'état
Q	: matrice de covariance du bruit détat
$\mathcal{Z}$	: espace des observations
$z(k)$	: observation à l'instant k
$z(1:k)$	: observations cumulées jusqu'à l'instant k
$\overline{z}(k)$	: observation prédite pour l'instant k
$inov(k)$	: innovation : la différence entre l'observation prédite $\overline{z}(k)$ et l'observation réelle $z(k)$ à l'instant k
S	: matrice de covariance de l'observation prédite
H	: matrice d'observation
v	: bruit de mesure
R	: matrice de covariance du bruit de mesure
P_d	: probabilité de détection
β_{fa}	: densité de fausses alarmes
β_{no}	: densité de nouveaux objets
F_r	: matrice d'état correspondant au mode d'évolution r
$\mu^i(k)$	: probabilité d'être dans le mode d'évolution $r = i$ à l'instant k
$\mu^{i\|j}(k)$	: probabilité de passage du mode $r = j$ au mode $r = i$ à l'instant k
$\Pi = [\pi_{ij}]_{M\times M}$	: matrice de transition propre aux *IMMs*, elle contient les probabilités *a priori* de transition entre les modes d'évolution
M	: nombre de modes d'évolution
C	: ensemble de classes
B	: ensemble de comportements

Introduction Générale

Motivations

Les travaux de cette thèse rentrent dans le cadre du suivi et de la classification d'objets. La problématique en question représente l'une des tâches les plus complexes qui peuvent être confiées aux systèmes de notre ère. Des systèmes que l'évolution technologique ne cesse de doter d'une intelligence artificielle de plus en plus puissante et de moins en moins encombrante. Cette intelligence leur est nécessaire pour accomplir les tâches qui leurs sont confiées et explorer des environnements, parfois inconnus par l'homme. On parle alors de drones, de robots explorateurs de fonds marins ou encore de l'espace. Outre, le suivi et la classification d'objets représentent la tâche clé des systèmes de sécurité automatiques, militaires ou civils soient-ils.

Dans de tels systèmes, un objet peut correspondre à n'importe quel élément dans l'environnement considéré, ça peut être : un véhicule, un piéton, un avion, un obstacle ou autre. Certains des systèmes dont on parle sont illustrés par la figure 1.

Techniquement, le suivi et la classification d'objets consiste à propager un ensemble d'informations dans le temps. Généralement, des informations qui concernent l'état et/ou le type des objets. L'état d'un objet fait souvent référence à des données cinématiques comme la position, la vitesse, l'accélération ou autres. Le type de l'objet dépend de l'application considérée, cela peut représenter des bombardiers ou des avions de chasse dans un champs de bataille par exemple. Les premiers algorithmes de suivi et de classification d'objets ont justement été appliqués dans le domaine militaire dès l'apparition du radar. Depuis, les instruments de mesure et les applications de ce domaine se sont multipliés et les algorithmes sont devenus de plus en plus performants. Toutefois, face aux complexités pouvant être rencontrées, les performances des algorithmes en question demeurent limitées. Il convient de noter que la majeure partie des algorithmes auparavant développés est

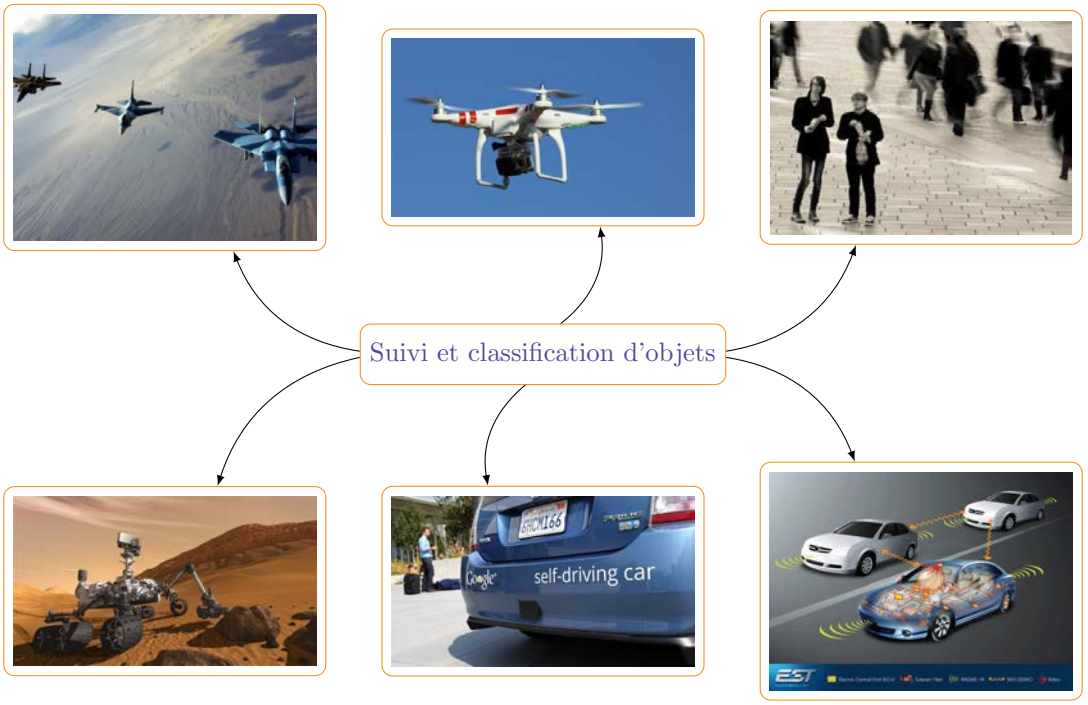


FIGURE 1: Quelques applications du suivi et de la classification d'objets.

basée sur le formalisme Bayésien, que ce soit pour la propagation ou pour la fusion d'informations. Étant le premier formalisme permettant de modéliser les imperfections, le formalisme Bayésien était utilisé pour remédier aux différentes complexités liées à la problématique considérée, notamment celles qui concernent les imperfections de mesure comme :

- les erreurs de mesure,
- et l'occultation des objets par rapport au positionnement des capteurs,

mais également les multiples complexités liées à l'environnement du suivi dont on peut citer :

- la multiplicité des objets,
- le comportement et les manœuvres aléatoires des objets,
- la gestion des apparitions et des disparitions d'objets,
- la présence de fausses alarmes (e.g. phénomènes d'ombrage),
- le risque de confondre les objets, notamment quand leurs trajectoires sont étroitement proches, et bien d'autres complexités.

Toutes les complexités venant d'être citées ainsi que d'autres engendrent des informations incertaines, imprécises, vagues et incomplètes. Toutes ces imperfections ne peuvent être gérées par le formalisme Bayésien à lui seul, lui qui est connu pouvoir gérer les incertitudes (quantifier la qualité de l'information) principalement. D'autres formalismes peuvent alors être proposés pour gérer les imperfections citées, notamment le formalisme des fonctions de croyance introduit par Shafer [162] dans son livre en 1976, et qui est apprécié pour sa capacité à gérer les imprécisions (plus d'une hypothèse possible en réponse à une question donnée) en plus des incertitudes [174].

Le travail de cette thèse explore l'idée d'utiliser le modèle des fonctions de croyance pour le suivi et la classification d'objets tout en faisant des comparaisons avec ce qui s'obtient en utilisant le formalisme Bayésien et d'autres travaux de la littérature qui se sont fixés le même objectif que nous. Le travail s'intéresse également au suivi et la classification multi-capteurs où des règles de fusion crédales et Bayésiennes sont mises en compétition. Les principales contributions de cette thèse en réponses à un certain nombre d'objectifs sont résumées dans la section suivante.

Contributions de la thèse

Dans cette section, nous rappelons les principaux objectifs auxquels le travail présenté s'est intéressé. Ensuite, nous résumons les principales contributions apportées.

Objectif N^o1 **:** le premier objectif de cette thèse porte sur l'estimation de plusieurs trajectoires à la fois. Dans le cadre Bayésien, le suivi d'une trajectoire est modélisé par la propagation d'une densité de probabilité multiple. La multiplicité correspond aux différentes manœuvres qui peuvent être effectuées par l'objet. Il va de soi que le suivi de plusieurs trajectoires revient à faire propager plusieurs densités de probabilités multiples dont le nombre n'est pas constant dans le temps tout en assurant la distinction entre les différentes densités de probabilités, et donc des objets suivis.

Objectif N^o2 **:** la phrase "distinction entre les densités de probabilités" est en soi une autre problématique, une problématique qui n'est pas des moindres dans la mesure où le nombre de densités de probabilités (objets) change aléatoirement dans le temps et que la distinction entre les densités de probabilités est irrésolue quand les objets sont étroitement proches : leurs observations risquent d'être confondues et ainsi engendrer une mise à jour erronée des densités de probabilités. Ce second objectif fait référence au problème d'association des observations aux objets connus.

Objectif N^o3 **:** les deux premiers objectifs concernent l'estimation d'état de plusieurs objets. L'état d'un objet étant constitué de ces données cinématiques, il est fourni à un étage de classification appelé à reconnaître le type des objets en se basant sur leurs comportements. L'objectif N^o3 consiste alors, à correctement classifier les objets tout en tenant compte de l'imperfection des informations à disposition.

Objectif N^o4 **:** le quatrième objectif concerne la fusion d'informations : fusionner les informations concernant le suivi et la classification d'objets, provenant de capteurs dont la fiabilité et/ou l'observabilité sont limitées.

Les principales contributions de cette thèse sont les suivantes :

Contribution N^o1 **:** en réponse aux objectifs 1, 2 et 3, une solution de suivi et de classification multi-objets est construite dans le chapitre 2 de cette thèse. La solution regroupe différents algorithmes dispersés dans la littérature. Elle utilise notamment l'un des algorithmes les plus performants dédiés à l'estimation des densités de probabilités multi-modales, c'est l'algorithme dit *Interacting Multiple Model (IMM)* basé sur le filtrage de Kalman. La résolution du problème d'association dans la solution de base proposée, dans l'algorithme 2.1 du chapitre 2, est assurée par l'algorithme dit *Global Nearest Neighbor (GNN)*. La gestion des fausses alarmes, des apparitions et des disparitions d'objets est faite à l'aide d'un test statistique introduit par Wald en 1945 et utilisé dans le domaine du suivi d'objets sous le nom de fonction score. Tous les algorithmes utilisés pour la construction de la solution peuvent être trouvés dans [10, 15, 21, 23]. La solution du suivi proposée dans 2.1 est exclusivement basée sur le formalisme Bayésien : la théorie de l'estimation est nettement plus avancée dans le cadre Bayésien comparé aux cadres des fonctions de croyance, de la logique floue ou encore des probabilités imprécises [15, 23]. Toutefois, Smets et Ristic ont montré que dans l'étape de classification, le classifieur crédal obtient de meilleures performances par rapport au classifieur Bayésien pour la classification d'une seule cible aérienne dont les classes sont imbriquées et constantes dans le temps [175]. Dans le chapitre 2 le résultat obtenu par Smets et Ristic est confirmé par la comparaison des deux classifieurs crédal et Bayésien dans un cadre multi-objets. Un second nouvel exemple de classification de piétons est proposé dans la section 2.5 du chapitre 2. Contrairement au premier exemple de cibles aériennes, les classes dans l'exemple des piétons ne sont pas imbriquées mais elles se chevauchent, elles ne sont également pas constantes dans le temps. L'exemple des piétons vise à approfondir l'étude du classifieur crédal. Nous montrons qu'une simple révision d'informations permet de s'adapter aux changements de classe.

Au niveau de la classification crédale dans un cadre multi-objets, la section 2.6 montre qu'une fausse association entre les observations d'objets de différents types peut engendrer une détérioration de leurs classifications respectives, ce qui motive davantage la résolution du problème d'association qui est initialement assuré par le *GNN* et repris dans le chapitre 3 avec des fonctions de croyance.

À coté de cette première contribution présentée dans le chapitre 2, la thèse présente également un état de l'art concernant les techniques les plus connues dans le domaine du suivi d'objets dans le chapitre 1. Des discussions et des comparaisons sont également avancées dans le même chapitre afin de justifier le choix des algorithmes qui ont construit la contribution du chapitre 2. Parmi les comparaisons faites dans le chapitre état de l'art, on trouve : filtrage de Kalman vs filtrage particulaire et *IMM* Bayésien vs *IMM* crédal présenté dans [134].

Contribution N^o2 **:** la contribution apportée par le chapitre 3 concerne l'association des observations aux objets connus. Le chapitre compare, à travers de multiples simulations réalisées sous Matlab, les algorithmes d'association les plus récents basés sur les fonctions de croyance. Tout comme l'algorithme *GNN*, l'ensemble des algorithmes étudiés dans ce chapitre sont mono-scan et déterministes : une décision d'association est prise à chaque pas de temps avec la supposition d'associer au plus une seule observation à chaque objet connu. Cela justifie le fait que les comparaisons n'ont pas intégré d'autres méthodes d'association multi-scan, probabilistes ou multi-scan et probabilistes à la fois, en l'occurrence les algorithmes : *Multi-Hypothesis Tracking (MHT)*, *Joint Probabilistic Data Association (JPDA)* et *Probabilistic MHT* qui représentent les algorithmes les plus performants dans le domaine du suivi d'objets dans le cadre Bayésien au prix d'une complexité calculatoire exponentielle [23]. Le chapitre est donc dédié à l'étude de l'apport des algorithmes d'association crédaux dans le cadre mono-scan et déterministe. Un des avantages des algorithmes crédaux est illustré dans [57], il concerne le fait que ces derniers présentent une facilité à résoudre le problème d'association en se basant sur plusieurs informations, ce qui peut être de grande utilité dans un cadre multi-capteurs par exemple.

La contribution majeure présentée dans le chapitre 3 est l'élaboration d'une nouvelle méthode d'association basée sur les fonctions de croyance. La méthode est nommée "proposition 1", elle se trouve être moins sensible à la calibration des paramètres et plus robustesse dans le cas d'erreurs de mesure extrêmes. Toutefois, il s'avère qu'elle soit plus complexe en terme de calcul par rapport à l'algorithme *GNN* et la méthode proposée par Denœux et *al* dans [57]. Une variante à la solution "proposition 1" nommée "proposition 2" est développée. C'est une variante qui n'atteint pas l'optimalité comme le font les solutions : proposition 1, algorithme *GNN* et la solution proposée

par Denœux et al, mais elle présente des performances intéressantes dans des environnements où l'apprentissage des paramètres n'est pas réalisable et/ou les incertitudes de mesure sont extrêmes.

Contribution N^o3 **:** les contributions présentées dans les chapitres 2 et 3 rentrent dans le cadre du suivi et de la classification multi-objets, mono-capteur. Les contributions présentées dans le chapitre 4 s'intéressent au suivi et à la classification multi-objets, multi-capteurs et donc à la redondance et la fusion d'informations qui représentent le principal remède aux imperfections de capteurs. Deux imperfections de capteurs sont principalement traitées. La première imperfection considère des capteurs peu fiables de par leur grande imprecision. Ces capteurs exécutent les algorithmes de suivi et de classification présentés dans le chapitre 2. Il a été montré que la non fiabilité des capteurs fait que la classification des objets soit détériorée, c'est pourquoi il a été proposé de fusionner les classifications locales à l'aide d'une architecture multi-capteurs centralisée. Les règles de fusion Bayésiennes et crédales sont mises en compétition afin d'obtenir le meilleur résultat de classification globale au niveau du centre de fusion.

La seconde imperfection de capteurs menant à la seconde contribution du chapitre 4 concerne des capteurs qui n'observent l'état des objets que partiellement. Ils ne peuvent donc pas reconstruire les trajectoires ni reconnaître les classes des objets par la simple execution des algorithmes de suivi et de classification présentés dans le chapitre 2. L'idée consiste à adopter une solution permettant aux capteurs d'agréger les informations des autres capteurs pour ainsi construire l'information complète concernant les trajectoires et les classes des objets tout en supposant que les informations partielles à disposition des différents capteurs sont complémentaires. La contribution dans cette solution n'est pas liée à l'utilisation des fonctions de croyance mais plutôt à l'insertion d'un algorithme de calcul distribué dans les deux phases de suivi et de classification. Il s'agit d'un algorithme de consensus qui permet à chaque capteur d'agréger les informations de tous les autres capteurs rien qu'en communiquant avec les capteurs qui lui sont immédiatement voisins, ce à travers quelques itérations. La solution de suivi et de classification résultante est dite distribuée, elle est simulée sur un exemple de surveillance du comportement de véhicules sur une autoroute.

Chapitre 1

Suivi d'objets Multiples : état de l'art

Le suivi d'objets est l'opération consistant à estimer l'état des objets à travers le temps. Cette opération repose essentiellement sur des techniques de filtrage assez largement développées dans le cadre Bayésien [23].

À défaut de ne trouver une solution suffisamment efficace qui tienne en compte de toutes les complexités pouvant être rencontrées face à un tel problème, ce dernier demeure un sujet de recherche très convoité. Parmi les complexités qui peuvent être rencontrées face à un tel problème, on trouve celles qui sont liées à l'environnement du suivi comme :

- la multiplicité des objets,
- les objets manœuvrants,
- les apparitions, les disparitions et les réapparitions aléatoires d'objets,
- la présence de fausses alarmes (e.g. phénomène d'ombrage),
- les trajectoires conflictuelles et autres.

En plus des complexités liées à l'environnement du suivi, on peut également citer celles qui sont liées aux imperfections des instruments de mesure, notamment :

- les erreurs de mesure,
- l'occultation d'objets par rapport au positionnement des capteurs,

- les confusions entre les objets suivis, et d'autres complexités.

Différentes suppositions sont en général adoptées pour la résolution du problème de suivi, en l'occurrence, la connaissance approximative des modèles d'évolution des objets, avec parfois les lois statistiques des variables d'état des objets. Les méthodes d'estimation les plus utilisées dans le domaine du suivi d'objets sont présentées et discutées dans ce chapitre.

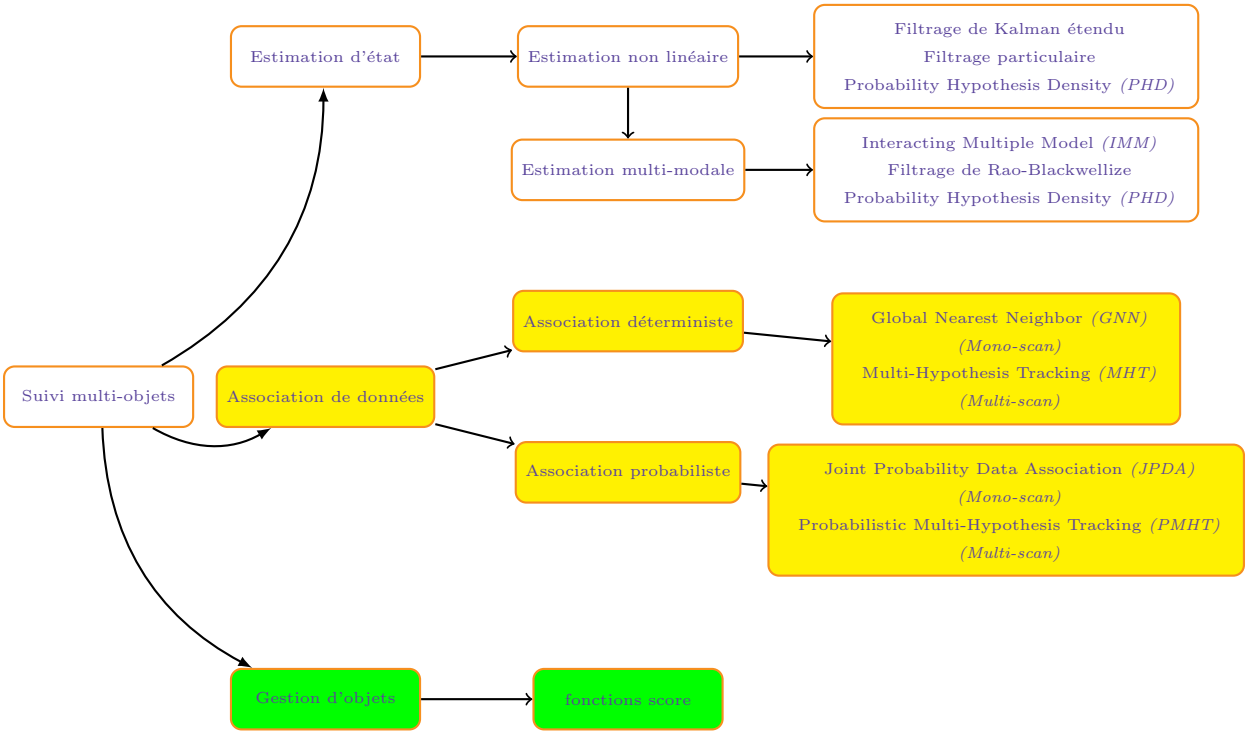


FIGURE 1.1: Organigramme des méthodes principales dédiées au suivi d'objets.

L'organigramme illustré par la figure 1.1 présente les principales approches décrites dans ce chapitre. En premier, les approches du filtrage Bayésien sont décrites dans la section 1.1. La section 1.2 illustre la différence entre le suivi d'un seul objet et le suivi de plusieurs objets, et présente les principales techniques chargées d'affecter l'observation ou les observations à l'objet ou aux objets suivis. La section 1.3 décrit une solution statistique, souvent utilisée pour gérer les fausses alarmes et la section 1.4 conclut le chapitre.

1.1 Éléments du filtrage Bayésien

La majorité des techniques de suivi mono-objet ou multi-objets sont basées sur le filtrage Bayésien [10, 23]. Le filtrage Bayésien n'est autre que l'estimation recursive de l'état des objets à base

d'observations acquises d'une manière séquentielle. L'état des objets ainsi que leurs observations sont considérés comme étant des variables aléatoires. Dans le cadre Bayésien, cela sous-entend une modélisation sous forme de densités de probabilités dont la propagation est basée sur l'inférence Bayésienne. Plus de précisions sont mentionnées dans ce qui suit.

Un objet est souvent représenté par un vecteur d'état $x(k)$ prenant des valeurs dans l'espace d'état noté $\mathcal{X} \subseteq \mathbb{R}^n$ où n représente la dimension du vecteur d'état. La dynamique des objets est représentée par un modèle à temps discret tel que le vecteur d'état à un pas de temps k peut être donné par :

$$x(k) = f(x(k-1), w(k-1)). \tag{1.1}$$

La fonction f dans l'équation (1.1) représente une transition de Markov, tel que l'état d'un objet à un instant donné n'est supposé dépendre que de son état à l'instant immédiatement précédent $x(k-1)$ et de l'erreur de modélisation $w(k-1)$. Alternativement cela peut être représenté par une densité de probabilité de transition notée :

$$p(x(k)|x(k-1)). \tag{1.2}$$

À chaque pas de temps k, des observations partielles du vecteur d'état $x(k)$ sont reçues, elles sont notées $z(k)$ et prennent des valeurs dans l'espace des observations $\mathcal{Z} \subseteq \mathbb{R}^m$ où m représente la dimension du vecteur d'observation. Le modèle d'observation peut être décrit de la manière suivante :

$$z(k) = h(x(k), v(k)), \tag{1.3}$$

où $v(k)$ représente l'erreur de mesure à l'instant k et h une fonction non linéaire, caractérisant le capteur, et liant le vecteur d'état $x(k)$ à l'observation $z(k)$.

Alternativement, une observation peut être représentée par une fonction de vraisemblance $p(.|.)$:

$$p(z(k)|x(k)), \tag{1.4}$$

qui représente la densité de probabilité que l'observation $z(k)$ soit générée par l'objet ayant comme vecteur d'état $x(k)$.

Le suivi d'un objet donné consiste à déterminer la densité de probabilité *a posteriori* de son vecteur d'état $x(k)$ en tenant compte de l'historique de ces observations noté : $z(1:k)$. Cette densité de probabilité $p(x(k)|z(1:k))$ peut être obtenue récursivement à l'aide du théorème de Bayes :

$$p(x(k)|z(1:k)) = \frac{p(z(k)|x(k))p(x(k)|z(1:k-1))}{\int_{x(k)} p(z(k)|x(k))p(x(k)|z(1:k-1))dx(k)}, \tag{1.5}$$

où

$$p(x(k)|z(1:k-1)) = \int_{x(k-1)} p(x(k)|x(k-1))p(x(k-1)|z(1:k-1))dx(k-1), \tag{1.6}$$

avec une densité de probabilité initiale $p(x(0)|z(0))$. L'équation (1.6) est également connue sous le nom de l'équation de Chapman-Kolmogorov pour un processus Markovien [75].

La détermination d'une estimation optimale du vecteur d'état notée $\hat{x}(k)$ revient à résoudre la densité de probabilité *a posteriori* $p(x(k)|z(1:k))$ selon un critère donné. Certains estimateurs Bayésiens tentent de minimiser la variance de l'erreur d'estimation et d'autres de maximiser la densité de probabilité *a posteriori*. Ces deux critères sont respectivement formalisés comme suit :

$$\hat{x}(k) = \min_{x(k)} \int x(k)p(x(k)|z(1:k))dx(k), \tag{1.7}$$

$$\hat{x}(k) = \underset{x(k)}{\operatorname{argmax}}\, p(x(k)|z(1:k)). \tag{1.8}$$

En minimisant un certain risque dit "risque de Bayes", les estimateurs assurent une convergence asymptotique vers le vecteur d'état réel [115]. Toutefois, l'aspect multidimensionnel de la densité de probabilité (1.5) nécessite le calcul d'intégrales multiples. Ainsi son calcul analytique est généralement intraitable et son estimation est souvent non implémentable si ce n'est pour des problèmes de dimension réduite.

Dans ce qui suit, quelques solutions dédiées à l'estimation de la densité de probabilité (1.5) sont décrites.

1.1.1 Estimation linéaire

Un système linéaire est un système qui suppose une évolution linéaire du vecteur d'état (voir l'équation (1.1)) et une relation linéaire entre le vecteur d'état et l'observation (voir l'équation (1.3)).

Le modèle de dynamique et le modèle d'observation peuvent alors être réécrits ainsi :

$$x(k) = Fx(k-1) + w(k-1), \tag{1.9}$$

$$z(k) = Hx(k) + v(k), \tag{1.10}$$

où F représente une matrice d'état de dimension $n \times n$, H une matrice d'observation de dimension $n \times m$ et $w(k-1)$ et $v(k)$ représentent respectivement le bruit d'état et le bruit de mesure qui sont considérés Gaussiens et dont les matrices de covariances sont respectivement notées $Q(k-1)$ et $R(k)$.

Comme cela peut être remarqué, en plus de l'hypothèse de linéarité, il est souvent supposé que le bruit d'état et le bruit de mesure sont de nature Gaussienne. Cela revient à supposer que les densités de probabilités du modèle d'état et du modèle d'observation sont également de nature Gaussienne :

$$p(x(k)|x(k-1)) = \mathcal{N}(x(k), Fx(k-1), Q(k-1)), \tag{1.11}$$

$$p(z(k)|x(k)) = \mathcal{N}(z(k), Hx(k), R(k)), \tag{1.12}$$

où $\mathcal{N}(var, moy, cov)$ représente une densité Gaussienne de la variable var, de moyenne moy et de covariance cov.

Dans le contexte des systèmes linéaires et Gaussiens, le filtre de Kalman est l'un des estimateurs les plus connus et utilisés pour le suivi d'objets. Il assure une estimation optimale de la densité de probabilité désirée.

1.1.1.1 Filtrage de Kalman

À chaque pas de temps k, le filtre de Kalman exécute deux étapes principales : une étape de prédiction et une étape de mise à jour [2, 96, 97, 163]. À partir de la densité de probabilité *a posteriori* calculée au pas de temps $k-1$:

$$p(x(k-1)|z(1:k-1)) = \mathcal{N}(x(k-1), \hat{x}(k-1), P(k-1)), \tag{1.13}$$

une prédiction pour le pas de temps en cours k est faite :

$$p(x(k)|z(1:k-1)) = \mathcal{N}(x(k), \hat{x}(k|k-1), P(k|k-1)), \tag{1.14}$$

où

$$\hat{x}(k|k-1) = F\hat{x}(k-1), \tag{1.15}$$

$$P(k|k-1) = FP(k-1)F' + Q(k-1), \tag{1.16}$$

avec F' représentant la matrice transposée de F.

À la réception d'une observation $z(k)$ au pas temps k, la densité de probabilité désirée peut être mise à jour de la façon suivante :

$$p(x(k)|z(1:k)) = \mathcal{N}(x(k), \hat{x}(k), P(k)), \tag{1.17}$$

où

$$\hat{x}(k) = \hat{x}(k|k-1) + K(k)(z(k) - H\hat{x}(k|k-1)), \tag{1.18}$$

$$P(k) = [I - K(k)H]P(k|k-1), \tag{1.19}$$

$$K(k) = P(k|k-1)H'S^{-1}(k), \tag{1.20}$$

$$S^{-1}(k) = HP(k|k-1)H' + R(k), \tag{1.21}$$

avec $K(k)$ représentant le gain du filtre de Kalman. La matrice $S(k)$ représente la covariance de l'erreur faite sur la prédiction de l'observation $\bar{z}(k) = H\hat{x}(k|k-1)$, elle est également appelée covariance de l'innovation sachant que l'innovation est donnée par : $z(k) - H\hat{x}(k|k-1)$.

L'avantage du filtre de Kalman est sa capacité à calculer une estimation optimale de la densité de probabilité d'intérêt. Son principal inconvénient est sa restriction aux systèmes linéaires et Gaussiens.

Des extensions du filtre de Kalman aux systèmes non linéaires existent, elles sont décrites dans ce qui suit avec d'autres types de filtres non linéaires.

1.1.2 Estimation non-linéaire

Généralement, la dynamique des objets ainsi que les modèles d'observation sont non linéaires. En terme de modélisation, cela se traduit par une fonction d'état f et une fonction d'observation h (voir les équations (1.1) et (1.3) respectivement) non linéaires.

Afin de permettre le suivi d'objets dans le cas de modèles non linéaires, deux techniques d'estimation phares étaient considérées dans la littérature [23]. Plus récent, une autre technique de filtrage non linéaire basée sur les ensembles finis aléatoires se rajoute à la littérature du domaine du suivi d'objet, on parle notamment du filtrage *PHD* [184]. Dans cette thèse, l'intérêt est porté sur les méthodes de filtrage classique, notamment les celles qui sont basées sur le filtrage de Kalman. Dans cette catégorie, on trouve le filtre de Kalman étendu (*Extended Kalman Filter EKF*) qui propose un développement limité de premier ordre des fonctions non linéaires, ce qui revient à linéariser le système pour ainsi pouvoir executer les équations de Kalman telles qu'elles sont. Le filtre de Kalman étendu et invariant (*Invariant Extended Kalman Filter IEKF*) combine l'avantage du *EKF* et les invariances possibles du système (e.g. symétrie). Le filtre de Kalman sans parfum (*Unscented Kalman Filter UKF*) est transversal entre cette première catégorie et la seconde catégorie, il propose un échantillonnage local et déterministe de la densité de probabilité désirée qui est propagée à l'aide des équations de Kalman.

La seconde technique d'estimation se passe de toute contrainte de linéarité. Elle consiste en un échantillonnage aléatoire massif visant à approximer la densité de probabilité d'intérêt. Cette technique se réfère aux méthodes Monte Carlo dont on trouve des implementations séquentielles utilisables dans le contexte du suivi d'objets.

1.1.2.1 Filtrage de Kalman étendu

Le filtre de Kalman étendu *EKF* [2, 81, 93] est une solution sous-optimale. Il propose une linéarisation de premier ordre des modèles d'état et/ou d'observation afin que ces derniers puissent être traités par les équations de Kalman. Le processus d'exécution du filtre de Kalman étendu pour un pas de temps k suit les étapes suivantes :

Connaissant la densité de probabilité au pas de temps $k-1$:

$$p(x(k-1)|z(1:k-1)) = \mathcal{N}(x(k-1), \hat{x}(k-1), P(k-1)), \tag{1.22}$$

une prédiction peut être faite de la manière suivante :

$$p(x(k)|z(1:k-1)) = \mathcal{N}(x(k), \hat{x}(k|k-1), P(k|k-1)), \tag{1.23}$$

où

$$\hat{x}(k|k-1) = f(\hat{x}(k-1), 0), \tag{1.24}$$

$$P(k|k-1) = FP(k-1)F' + Q(k-1), \tag{1.25}$$

$$F = \frac{\partial f(y,0)}{\partial y}|_{y=\hat{x}(k-1)}. \tag{1.26}$$

Le calcul de la densité de probabilité *a posteriori* (mise à jour) est effectué ainsi :

$$p(x(k)|z(1:k)) = \mathcal{N}(x(k), \hat{x}(k), P(k)), \tag{1.27}$$

où

$$\hat{x}(k) = \hat{x}(k|k-1) + K(k)(z(k) - h(\hat{x}(k|k-1), 0)), \tag{1.28}$$

$$P(k) = [I - K(k)H]P(k|k-1), \tag{1.29}$$

$$K(k) = P(k|k-1)H'S^{-1}(k), \tag{1.30}$$

$$S^{-1}(k) = HP(k|k-1)H' + R(k), \tag{1.31}$$

avec

$$H = \frac{\partial h(y,0)}{\partial y}|_{y=\hat{x}(k|k-1)}. \tag{1.32}$$

Le filtre de Kalman étendu n'est pas exact (sous-optimal) dans la mesure où il ne prend en considération que le premier terme du développement en série de Taylor des fonctions non linéaires. De ce fait, le filtre s'avère inefficace dans le cas de fonctions fortement non linéaires [94], ce qui peut correspondre à des dynamiques de manœuvres dans la réalité. De plus, comme cela peut être constaté sur les équations correspondantes, l'hypothèse de densités Gaussiennes est toujours maintenue. D'autres versions de filtres de Kalman sont développées, notamment le filtre de Kalman étendu invariant (*Invariant Extended Kalman Filter IEKF*) [28] qui propose de simplifier le calcul des fonctions dérivées du système en tenant compte des différentes invariances possibles, comme par exemple la symétrie (e.g. calculer la trajectoire d'une aile d'un avion suffirait pour déduire celle de l'autre aile). Dans le cas de fortes non linéarités, le filtre de Kalman sans parfum *UKF* est

utilisé. Il propose un échantillonnage déterministe autour de la moyenne de la densité de probabilité d'intérêt. L'échantillonnage est effectué à l'aide d'une technique appelée : transformation sans parfum (*Unscented Transform UT*). Les échantillons sont également appelés points sigma (*sigma points*). Ils sont propagés à l'aide des équations non linéaires du système et à chaque itération ils sont pondérés afin de reconstruire la moyenne et la covariance du vecteur d'état. Plus de détails concernant le *UKF* peuvent être trouvés dans [43, 66]. Contrairement, au filtre *UKF* qui propose un échantillonnage déterministe de la densité de probabilité $p(x(k)|z(1:k))$, les filtres particulaires proposent un échantillonnage aléatoire de cette densité de probabilité. Quelques éléments de base concernant les filtres particulaires et les méthodes d'estimation Monte Carlo *MC* sont donnés dans ce qui suit.

1.1.2.2 Filtrage particulaire

L'algorithme d'échantillonnage séquentiel par importance (*Sequential Importance Sampling SIS*), également connu par filtre particulaire (*Particle Filter PF*) dans [6, 34] représente la solution de base de la majorité des méthodes d'estimation dites Monte Carlo. Notamment le filtre *bootstrap* [80], filtre de condensation [113], approximation par particules interactives [52] et d'autres. L'objectif de base de ces algorithmes est l'implementation du filtre Bayésien avec une approche d'approximation Monte Carlo. Ils tentent d'approximer la densité de probabilité $p(x(k)|z(1:k))$ à l'aide d'un ensemble d'échantillons (particules) aléatoires munis de poids. L'évaluation séquentielle des poids permet de déterminer les échantillons approximant au mieux la densité de probabilité d'intérêt. Les méthodes d'estimation Monte Carlo en théorie n'ont aucune contrainte concernant la linéarité du système, toutefois, leurs performances dépendent de l'abondance de l'échantillonnage déployé. L'estimation est quasi-optimale quand le nombre d'échantillons tend vers l'infini. Afin de mieux comprendre le fonctionnement des filtres particulaires, une description succincte de l'algorithme de base *SIS* est donnée dans ce qui suit.

D'une manière générale, si on souhaite approximer une densité de probabilité quelconque $p(.)$ à l'aide de N échantillons (particules) indépendamment et identiquement distribués (i.i.d) notés $\{x^{(i)}\}_{i=1}^{N}$ et que ces derniers peuvent être générés à l'aide de la loi cible $p(.)$ (cas idéal) alors l'approximation se fait comme suit :

$$p(x) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}}(x). \tag{1.33}$$

La convergence d'une telle approximation dépend du nombre d'échantillons de telle sorte que pour toute fonction arbitraire integrable f, l'hypothèse suivante est vérifiée :

$$\frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \underset{N\to\infty}{\longrightarrow} \int f(x)p(x)dx. \tag{1.34}$$

L'approximation présentée par l'équation (1.33) est idéale dans la mesure où les échantillons sont générés à l'aide de la densité de probabilité cible qui est supposée connue.

En réalité la densité de probabilité cible $p(.)$ n'est pas connue, par conséquent, les échantillons $\{x^{(i)}\}_{i=1}^{N}$ sont générés selon une autre densité de probabilité $q(.)$ semblable à $p(.)$ [6]. La densité $q(.)$ est appelée : densité de proposition ou encore densité d'importance. Dans ce cas, l'approximation se fait de la manière suivante :

$$p(x) \approx \frac{1}{N}\sum_{i=1}^{N} \tilde{w}^i \delta_{x^{(i)}}(x), \tag{1.35}$$

où

$$\tilde{w}^i = \frac{w(x^{(i)})}{\sum_{j=1}^{N} w(x^{(j)})}, \tag{1.36}$$

et

$$w^{(i)} = w(x^{(i)}) = \frac{p(x^{(i)})}{q(x^{(i)})}, \tag{1.37}$$

avec

$$\sum_{i=1}^{N} \tilde{w}^i = 1. \tag{1.38}$$

Dans le cadre du suivi d'objets, l'objectif est d'approximer la densité de probabilité d'intérêt $p(x(k)|z(1:k))$ à l'instant k. L'application de l'algorithme *SIS* peut se faire de la manière suivante. Supposons qu'à l'instant $k-1$ la densité de probabilité *a priori* $p(x(k-1)|z(1:k-1))$ est représentée

par un ensemble de particules avec leurs poids correspondants $\{w^{(i)}(k-1), x^{(i)}(k-1)\}_{i=1}^{N}$:

$$p(x(k-1)|z(1:k-1)) \approx \frac{1}{N}\sum_{i=1}^{N} \tilde{w}^{i}(k-1)\delta_{x^{(i)}(k-1)}(x(k-1)). \tag{1.39}$$

En générant les nouveaux échantillons $\{w^{(i)}(k), x^{(i)}(k)\}_{i=1}^{N}$ de l'instant k à l'aide de la densité de proposition $q(.|x^{(i)}(k-1), z(k))$, l'approximation de la densité de probabilité *a posteriori* est effectuée ainsi :

$$p(x(k)|z(1:k)) \approx \frac{1}{N}\sum_{i=1}^{N} \tilde{w}^{i}(k)\delta_{x^{(i)}(k)}(x(k)), \tag{1.40}$$

où

$$\tilde{w}^{(i)}(k) = \frac{w^{(i)}(k)}{\sum_{j=1}^{N} w^{(i)}(k)}, \tag{1.41}$$

$$w^{(i)}(k) = w^{(i)}(k-1)\frac{p(z(k)|x^{(i)}(k))p(x^{(i)}(k)|x^{(i)}(k-1))}{q(x^{(i)}(k)|x^{(i)}(k-1), z(k))}, \tag{1.42}$$

avec $p(z(k)|x^{(i)}(k))$ et $p(x^{(i)}(k)|x^{(i)}(k-1))$ respectivement, la fonction de vraisemblance et la densité de probabilité de transition calculées pour chaque échantillon $x^{(i)}$.

Le principal inconvénient du filtre particulaire de base *SIS* est bien la dégénérescence des échantillons utilisés pour l'approximation. En effet, la propagation dans le temps des échantillons fait que la plupart d'entre eux se retrouvent avec des poids quasi-nuls avec une très faibles contribution à l'approximation finale, ce sont les échantillons générés loin de la densité de probabilité cible. Le peu d'échantillons dont les poids restent valables se retrouvent figés et incapables de suivre l'évolution de la densité cible [6]. La plupart des recherches en filtrage particulaire visent à contourner ce phénomène de dégénérescence. Deux pistes majeures sont suivies, la première repose sur le choix de la densité de proposition $q(.|x^{(i)}(k-1), z(k))$ tel que plus le choix de cette densité est proche de la densité cible, plus le nombre d'échantillons contribuant à l'approximation est important, ils sont placés dans l'espace d'état d'intérêt. La seconde piste concerne le ré-échantillonnage et la régénération des particules. Cette action consiste à éliminer les particules à faible contribution dans l'approximation et à dupliquer les particules à forte contribution.

- Choix de la densité de proposition $q(.|x^{(i)}(k-1), z(k))$: le meilleur choix de densité de proposition serait de prendre $q(x(k)|x^{(i)}(k-1), z(k)) = p(x(k)|x^{(i)}(k-1), z(k))$ [61]. Ce choix n'est pas si évident compte tenu de la difficulté à échantillonner depuis la densité de probabilité $p(x(k)|x^{(i)}(k-1), z(k))$ pouvant comporter des calculs fastidieux. Toutefois, cela reste possible dans un certain nombre de cas : le cas où la densité est séparable en une partie linéaire et une partie non linéaire par exemple [62, 100]. Le choix le plus souvent adopté pour la densité de proposition est : $q(x(k)|x^{(i)}(k-1), z(k)) = p(x(k)|x^{(i)}(k-1))$. La qualité de l'échantillonnage pouvant être généré par la densité $p(x(k)|x^{(i)}(k-1))$ peut être de moindre qualité par rapport au premier cas, cela est dû au fait que la densité de proposition $p(x(k)|x^{(i)}(k-1))$ ne tient pas en compte de l'information apportée par l'observation $z(k)$ [6].

- Ré-échantillonnages : la deuxième solution par laquelle le problème de dégénérescence peut être évité est le ré-échantillonnage. Cette solution vise à placer et replacer les particules en nombre suffisant dans la partie de l'espace d'état la plus vraisemblable afin d'assurer une meilleure approximation de la densité cible. Paradoxalement, cette technique peut ne pas fonctionner dans le cas de bruit d'état à variance faible, cela réduit l'espace de recherche et engendre une duplication de particules au même endroit [34]. L'usage de cette technique de ré-échantillonnage nécessite une optimisation dans la mesure où elle représente un dilemme entre la qualité d'approximation demandant un nombre important d'échantillons et la complexité calculatoire [111, 145].

Pour donner un ordre d'idée quant à la différence dans la complexité de calcul entre un filtre de Kalman et un filtre particulaire, une simulation d'un simple signal linéaire Gaussien est faite, elle est illustrée par la figure 1.2.

La figure 1.2 représente une illustration simplifiée du filtrage d'un signal linéaire et Gaussien dont la moyenne est égale à zero. L'objectif est de confirmer la conclusion qui met en accord l'ensemble des travaux portant sur l'estimation d'état [40, 152]. La conclusion en question suggère l'utilisation du filtrage de Kalman à la place de tout autre filtrage lorsque les conditions favorables à son application sont réunies. Dans le cadre de cette thèse nous nous somme intéressé à des modèles favorisant l'utilisation du filtrage de Kalman. Le filtrage particulaire pourrait être utilisé, mais cela au prix d'un grand besoin calculatoire : l'Erreur Quadratique Moyenne (EQM) du filtrage de Kalman sur l'exemple de la figure 1.2 est de 0.6057 tandis que l'EQM du filtrage particulaire est

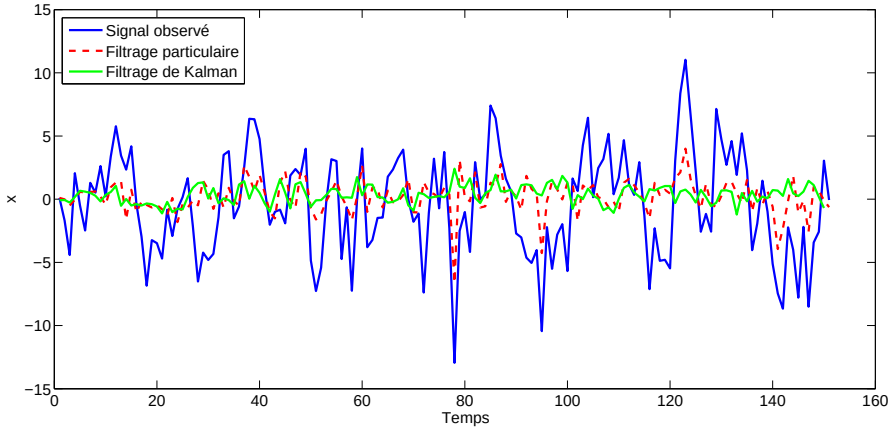


FIGURE 1.2: Filtre de Kalman vs filtre particulaire.

de 1.815 sachant que le nombre de particules utilisées est de 50 et que le temps de calcul qui peut être consommé par une seule particule est à peu près équivalent au temps de calcul qui peut être consommé par un filtre de Kalman [40, 118, 152].

1.1.3 Estimation multi-modale

Cette section s'intéresse à une question clé dans le cadre du suivi d'objets, en l'occurrence la représentation et l'estimation de l'état des objets effectuant des manœuvres au cours du suivi. Selon les manœuvres qui peuvent être effectuées, le problème peut s'avérer plus ou moins complexe. En effet, pour des objets dont l'évolution est quasi-linéaire avec de faibles manœuvres, un filtrage de Kalman avec réajustement de la matrice de covariance peut suffire pour assurer le suivi, cela fait partie de quelques méthodes pionnières présentées dans [23, chapitre 4]. Les méthodes d'estimation modernes s'intéressent aux plus cahoteuses des situations où les objets sont supposés pouvoir effectuer des manœuvres aléatoires de grandes complexités. À ce jour, aucun filtre ne peut estimer efficacement la densité de probabilité liée à ce genre de situations. Avec des ressources calculatoires illimitées cela pourrait être faisable à l'aide du filtrage particulaire à échantillonnage intensif. Cela étant irréalisable avec les moyens calculatoires de notre époque, des techniques d'estimation faisant des suppositions par rapport à la densité de probabilité d'intérêt sont adoptées. Dans ce rapport nous nous intéressons aux densités de probabilités multi-modales correspondant à des objets dont l'évolution est représentée par un modèle dit de Markov, linéaire à sauts (*Jump Markov Linear*

Model JMLM) donné par :

$$x(k) = F_{r(k)}x(k-1) + w_{r(k)}(k-1), \tag{1.43}$$

$$z(k) = H_{r(k)}x(k) + v_{r(k)}(k-1). \tag{1.44}$$

À la différence du modèle linéaire décrit par les équations (1.1) et (1.3), le modèle décrit par les équations (1.43) et (1.44) dépend du mode d'évolution $r(k)$ qui prend des valeurs dans $\{1, 2, ..., M\}$ où M représente le nombre de modes d'évolution possibles. L'évolution du paramètre $r(k)$ dans le temps est vue comme étant une chaîne de Markov dont les probabilités de transition sont données par la matrice $\Pi = [\pi_{ij}]_{M\times M}$ où $\pi_{ij} = P(r(k) = j|r(k-1) = i)$ représente la probabilité de passer du mode j au mode i, avec $\sum_{j=1}^{M} \pi_{ij} = 1$ pour tout $i \in \{1, 2, ..., M\}$.

Dans ce qui suit, une discussion des méthodes dédiées à la résolution de tels systèmes est donnée. Deux algorithmes des plus performants sont discutés : l'algorithme multi-modèles interactifs (*Interacting Multiple Model IMM*) qui est exclusivement basé sur le filtrage de Kalman et l'algorithme dit de *Rao-Blackwilize* qui fait appel au filtrage de Kalman et au filtrage particulaire à la fois.

1.1.3.1 Modèles interactifs (*Interacting Multiple Model IMM*)

L'algorithme *IMM* est une amélioration de l'algorithme Pseudo-Bayésien Généralisé (*Generalized Pseudo-Bayesian GPB*) [15]. L'*IMM* est l'algorithme d'estimation multi-modal le plus efficace en terme de complexité calculatoire [15], il propose de calculer la densité de probabilité *a posteriori* du suivi à l'aide d'une somme Gaussienne de M densités de probabilités calculées par des filtres de Kalman :

$$p(x(k)|z(1:k)) = \sum_{j=1}^{M} p(x(k)|r(k) = j, z(1:k))P(r(k) = j|z(1:k)), \tag{1.45}$$

où les densités de probabilités *a posteriori* au niveau des filtres de Kalman sont calculées à l'aide du théorème de Bayes :

$$p(x(k)|r(k) = j, z(1:k)) = \frac{p(z(k)|r(k) = j, x(k))}{p(z(k)|r(k) = j, z(1:k-1))} p(x(k)|r(k) = j, z(1:k-1)). \tag{1.46}$$

Les densités de probabilités *a priori* dans l'équation (1.46) sont calculées à l'aide du théorème des probabilités totales :

$$p(x(k)|r(k)=j,z(1:k-1)) = \sum_{i=1}^{M} p(x(k)|r(k)=j,r(k-1)=i,z(1:k-1)) \\ .P(r(k-1)=i|r(k)=j,z(1:k-1)),$$

où

$$p(x(k)|r(k)=j,r(k-1)=i,z(1:k-1)) = \int p(x(k)|r(k)=j,x(k-1),z(1:k-1)) \\ .p(x(k-1)|r(k-1)=i,z(1:k-1))dx(k-1),$$

et

$$P(r(k-1)=i|r(k)=j,z(1:k-1)) = \frac{P(r(k)=j|r(k-1)=i)P(r(k-1)=i|z(1:k-1))}{\sum_l^M P(r(k)=j|r(k-1)=l)P(r(k-1)=l|z(1:k-1))}. \tag{1.47}$$

Les équations (1.45) à (1.47) présentent la démarche théorique adoptée par un algorithme *IMM* pour résoudre la densité de probabilité d'intérêt. Cette démarche est exécutée en quatre étapes principales pour un pas de temps donné k et M filtres de Kalman :

Supposons que la densité *a posteriori* calculée à l'instant $k-1$ est Gaussienne :

$$P(x(k-1)|r(k-1)=i,z(1:k-1)) = \mathcal{N}(x(k-1);\hat{x}^i(k-1),P^i(k-1)), \tag{1.48}$$

et

$$P(r(k-1)=i|z(1:k-1)) = \mu^i(k-1). \tag{1.49}$$

Étape 1. Calcul des probabilités de mixage : les probabilités de transitions, entre les M modes d'évolution, basées sur les informations de l'instant $k-1$ sont calculées de la façon suivante :

$$\begin{aligned}\mu^{i|j}(k-1) &= P(r(k-1)=i|r(k)=j,z(1:k)) \\ &= \frac{P(r(k)=j|r(k-1)=i,z(1:k-1))P(r(k-1)=i|z(1:k-1))}{\sum_{l=1}^{M} P(r(k)=j|r(k-1)=l,z(1:k-1))P(r(k-1)=l|z(1:k-1))} \\ &= \frac{\pi_{ij}\mu^{i}(k-1)}{\sum_{l=1}^{M}\mu^{l}(k-1)}.\end{aligned}$$

Une fois les probabilités de mixage sont calculées, elles sont utilisées pour réinitialiser les estimations $\hat{x}^i(k-1)$ des M filtres de Kalman :

$$\hat{x}^{0j}(k-1) = \sum_{i=1}^{M} \hat{x}^i(k-1)\mu^{i|j}(k-1),\ \ j=1,...,M, \tag{1.50}$$

avec des matrices de covariances initiales données par :

$$\begin{aligned}P^{0j}(k-1) = \sum_{i=1}^{M} \mu^{i|j}(k-1)\{&P^i(k-1) \\ &+ [\hat{x}^i(k-1) - \hat{x}^{0j}(k-1)].[\hat{x}^i(k-1) - \hat{x}^{0j}(k-1)]'\} \\ &j=1,...,M.\end{aligned}$$

Étape 2. Filtrage : l'exécution de M filtres de Kalman permet de prédire la densité de probabilité :

$$P(x(k)|r(k)=j,z(1:k-1)) = \mathcal{N}(x(k);\hat{x}^j(k|k-1),P^j(k|k-1)), \tag{1.51}$$

ensuite, la mettre à jour à la réception des observations de l'instant k :

$$P(x(k)|r(k)=j,z(1:k)) = \mathcal{N}(x(k);\hat{x}^j(k),P^j(k)), \tag{1.52}$$

La phase de prédiction est assurée par les équations (1.15) et (1.16), et la phase de mise à jour par les équations (1.18) à (1.21).

Étape 3. Mise à jour des probabilités des modes :

L'information apportée par les observations, notamment les fonctions de vraisemblances $\Lambda^j(k) = p(z(k)|r(k)=j,z(1:k))$ de l'instant courant, permet de mettre à jour les probabilités liées aux

filtres de Kalman (aux modes d'évolution) :

$$\begin{aligned}\mu^j(k) &= P(r(k)=j|z(1:k)) \\ &= \frac{p(z(k)|r(k)=j,z(1:k-1))P(r(k)=j|z(1:k-1))}{\sum_{l=1}^{M} p(z(k)|r(k)=l,z(1:k-1))P(r(k)=l|z(1:k-1))} \\ &= \frac{\Lambda^j(k)\bar{\mu}^j(k-1)}{\sum_{l=1}^{M}\Lambda^l(k)\bar{\mu}^l(k-1)},\end{aligned}$$

où

$$\bar{\mu}^j(k-1) = \sum_{i=1}^{M} \pi_{ij}\mu^i(k-1), \tag{1.53}$$

avec des fonctions de vraisemblances supposées Gaussiennes :

$$\Lambda^j(k) = \mathcal{N}(z(k); \hat{x}^j(k|k-1), S^j(k)). \tag{1.54}$$

Étape 4. Combinaison des estimations : le vecteur d'état estimé ainsi que la matrice de covariance correspondante sont calculés par :

$$\hat{x}(k) = \sum_{j=1}^{M} \mu^j(k)\hat{x}^j(k), \tag{1.55}$$

$$P(k) = \sum_{j=1}^{M} \mu^j(k)\{P^j + [\hat{x}^j(k) - \hat{x}(k)][\hat{x}^j(k) - \hat{x}(k)]'\}. \tag{1.56}$$

À noter que l'étape 4 sert à fournir des quantités globales pour un usage externe, ces dernières ne sont pas nécessaires pour le suivi. L'*IMM* est un algorithme très largement utilisé dans le cadre du suivi d'objets, mais également dans d'autres domaines tel que le diagnostic des systèmes complexes [1, 191], et d'autres applications [120, 143, 159, 180].

1.1.3.2 Filtre de Rao-Blackwilize

Le filtre de Rao-Blackwilize combine le filtrage de Kalman et le filtrage particulaire [60]. Il s'intéresse aux densités de probabilités pouvant être séparées en une partie linéaire ou quasi-linéaire et une autre partie non linéaire. Le filtrage de la partie linéaire est assuré par des filtres de Kalman et le filtrage de la partie non linéaire est assuré par un filtrage particulaire. La technique de Rao-Blackwilize appliquée au filtrage Bayésien présente des avantages par rapport au filtrage

particulaire standard, qui en l'occurrence peut être incapable d'estimer une densité multi-modale en un temps raisonnable [60], cette technique peut également avoir un avantage par rapport à l'algorithme *IMM* concernant la qualité de l'estimation même si ces derniers utilisent un filtrage de Kalman. En effet, l'*IMM* n'approxime la densité multi-modale qu'avec un mélange Gaussien, tandis que le filtre de Rao-Blackwilize approxime la densité à l'aide d'un échantillonnage. Un nombre important de particules peut engendrer une qualité d'estimation meilleure. Cela au prix d'une complexité calculatoire considérable. À noter que les filtres de Kalman représentent les particules dans le filtre de Rao-Blackwilize [33]. Plusieurs extensions et améliorations ont été apportées dans [60, 78, 82, 83, 121].

1.2 Association des observations aux objets

Il peut être remarqué que les techniques de filtrage décrites estiment d'une manière explicite l'état de l'objet. En d'autres termes, une observation explicite est à chaque pas de temps attendue pour mettre à jour l'objet suivi. Dans cette section, nous nous intéressons aux méthodes dédiées à l'association des observations détectées à l'objet ou aux objets suivis. Cette étape est très complexe dans la mesure où elle est confrontée aux erreurs de prédictions faites par les filtres, des non-détections et les erreurs de mesure faites par les capteurs ainsi que d'autres complexités de l'environnement du suivi qui peut présenter des fausses alarmes, des observations fantômes (ombres) et autres.

Cette section présente quelques algorithmes de base ayant différentes stratégies quant à l'association des observations aux objets. Dans un premier temps, deux méthodologies d'association dans le cas mono-objet sont décrites : une solution déterministe (*Nearest Neighbor NN algorithm*) visant à mettre à jour le seul objet connu avec la plus proche observation parmi celles qui sont détectées, et une solution probabiliste mettant à jour l'objet connu à l'aide d'une somme pondérée de toutes les observations détectées (*Probability Data Association algorithm PDA*).

Les méthodes d'association les plus connues dans le cadre multi-objets sont également décrites. Notamment la méthode dite *Global Nearest Neighbor GNN* qui représente une généralisation de l'algorithme *NN* et l'algorithme dit *Joint Probabilistic Data Association JPDA* qui représente une généralisation de l'algorithme *PDA*. L'algorithme dit *Multi-Hypothesis Tracking MHT* pour le cas multi-objets est également décrit.

1.2.1 Cas mono-objet

Dans le cas de suivi d'un seul objet, plusieurs observations peuvent être détectées. Ces dernières peuvent être générées par l'objet suivi ou bien par du bruit (fausses alarmes, *clutter*). Le principe du suivi mono-objet est illustré par la figure 1.3 où l'état de l'objet est supposé prendre des valeurs dans l'ensemble {●, ●, ●}, où les boules de couleurs peuvent correspondre aux modes d'évolution dans l'*IMM* par exemple. La mise à jour séquentielle de l'état de l'objet à l'aide de l'observation ● est illustrée par la figure 1.3.

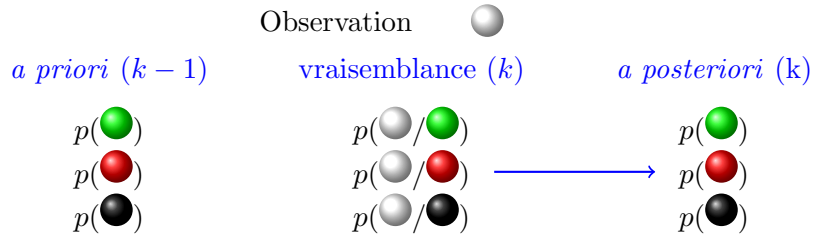


FIGURE 1.3: Illustration du suivi mono-objet

La question qui peut être posée dans le cadre du suivi mono-objet est : quelle observation choisir pour mettre à jour l'état de l'objet ?

Les méthodes suivantes apportent des réponses à cette question.

1.2.1.1 Algorithme du plus proche voisin (*Nearest Neighbor NN*)

L'algorithme du plus proche voisin *NN* représente la solution d'association la plus simple qui existe [23]. C'est une solution dédiée au suivi d'un seul objet dans un environnement pouvant contenir des fausses observations (fausses alarmes). L'algorithme *NN* est souvent accompagné d'une technique dite *gating*. Le *gating* consiste à éliminer toutes les observations éloignées de la position prédite pour l'objet suivi. L'algorithme *NN* consiste à choisir parmi les observations restantes celle qui est plus proche de la position prédite de l'objet. L'observation sélectionnée est utilisée pour mettre à jour le filtre assurant le suivi. La solution du plus proche voisin est l'une des premières solutions dédiées au suivi d'objet, elle est appréciée pour sa simplicité et sa facilité à implementer. Toutefois, elle présente des faiblesses quant au suivi dans des environnements denses en fausses alarmes [10, 14, 21]. Pour éviter des éventuelles fausses associations auxquelles

l'algorithme *NN* est confronté, l'algorithme *PDA* décrit une alternative prenant en compte toutes les observations lors de la mise à jour de l'état de l'objet.

1.2.1.2 Association de données probabiliste (*Probabilistic Data Association PDA*)

L'algorithme *PDA* propose de mettre à jour l'objet suivi à l'aide d'une somme pondérée de toutes les observations pouvant être associées à l'objet en question. Les observations très éloignées de l'objet sont éliminées par le *gating*. La solution PDA [10] est proposée dans l'espoir de remédier à d'éventuelles fausses associations découlant des décisions déterministes comme dans le cas de l'algorithme *NN*. En terme de calcul, cet algorithme est beaucoup plus complexe que l'algorithme *NN*. De plus, il ne présente aucune stratégie assurant le suivi d'un objet détecté par intermittence [10]. Plusieurs améliorations de l'algorithme *PDA* ont été proposées, notamment pour permettre la gestion des non détections d'objet dans la solution dite *Integrated PDA* [132]. Plusieurs extensions dédiées au suivi multi-objets sont proposées [38, 68, 133, 147, 160], la plus aboutie est la solution dite *Joint Probability Data Association JPDA* [37, 44, 73].

1.2.2 Cas multi-objets

Le suivi dans le cas multi-objets est plus complexe que dans le cas mono-objet. La complexité est essentiellement liée à l'ignorance concernant l'association des multiples observations détectées, à chaque instant, aux multiples objets suivis. Le problème est illustré par la figure 1.4 où on peut remarquer un conflit quant à l'association des deux observations détectées aux deux objets connus.

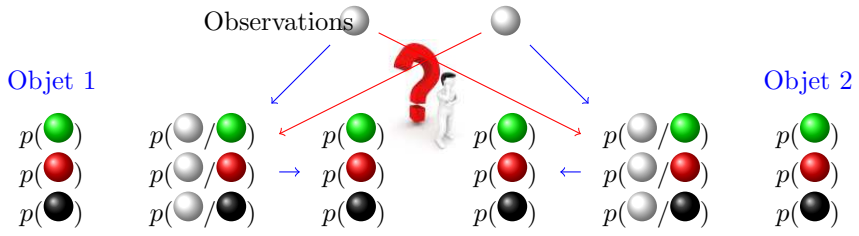


FIGURE 1.4: Illustration du suivi multi-objets.

Cette section donne un aperçu des techniques utilisées pour associer les observations aux objets. Ces dernières adoptent des stratégies d'association différentes afin de contourner les difficultés liées au problème d'association. Des difficultés relatives aux imprécisions et aux incertitudes de capteurs

et d'autres liées à la complexité du système, notamment, les situations conflictuelles de trajectoires confondues, l'occlusion des objets et d'autres difficultés.

Parmi les strategies utilisées on trouve :

- L'association déterministe : à chaque événement (objet connu) une seule hypothèse est associée (une observation où l'hypothèse de non détection). Telle est la stratégie adoptée par l'algorithme *GNN* et l'algorithme *MHT* par exemple.
- L'association non déterministe (probabiliste) : contrairement à la stratégie déterministe, la stratégie probabiliste propose d'associer à chaque événement plusieurs hypothèses, comme par exemple l'algorithme *PDA* et *JPDA* qui mettent à jour l'état des objets à l'aide de sommes pondérées de plusieurs observations.
- L'association mono-scan : les solutions sont dites mono-scan si à chaque pas de temps une décision d'association est prise. Tel est le cas des algorithmes *GNN*, et *JPDA* par exemple.
- L'association multi-scans : dans les situations conflictuelles, les approches multi-scans comme l'algorithme *MHT* proposent de différer la prise de décision.
- L'association multi-scans et probabiliste : l'algorithme *PMHT* est parmi les solutions les plus performantes dans le cadre du suivi d'objets multiples, il propose de retarder la prise de décision dans les situations conflictuelles et de mettre à jour les objets à l'aide de plusieurs observations afin d'éviter d'éventuelles fausses associations.
- Le suivi sans association de données : quelques méthodes comme le *PHD*, par exemple proposent de suivre les objets comme étant un ensemble, elles se passent de la nécessité d'associer les observations aux objets d'une manière explicite. Ces méthodes auraient l'inconvénient de ne pas garder l'identité des objets suivis.

Selon la stratégie d'association adoptée, les méthodes d'association existantes sont de complexités différentes, elles sont évidemment contraintes par le temps de calcul afin de permettre un suivi en temps réel par exemple. La section suivante donne quelques détails à propos des techniques d'association multi-objets.

1.2.2.1 Algorithme du plus proche voisin global (*Global Nearest Neighbor GNN*)

L'algorithme *GNN* représente une généralisation de l'algorithme *NN* au cas du suivi multi-objets [23]. L'idée est de minimiser la distance globale séparant les observations prédites notées par : $\overline{z}_i(k) = H\hat{x}_i(k|k-1)$ avec $i = 1, ..., n$, et les observations réelles $z_j(k)$ avec $j = 1, ..., m$ où n et m sont respectivement le nombre d'objets connus au pas de temps k et m le nombre d'observations détectées. Habituellement, la distance de Mahalanobis [122] est utilisée pour calculer la matrice d'association $D = [d_{i,j}]$. Le problème d'optimisation peut être formalisé de la manière suivante :

$$min \sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} r_{i,j}, \tag{1.57}$$

avec

$$\sum_{i}^{n+m} r_{i,j} \leq 1 \ , \tag{1.58}$$

$$\sum_{j}^{m} r_{i,j} = 1 \ , \tag{1.59}$$

$$r_{i,j} \in \{0, 1\}, \ \forall i \in \{1, \ldots, n+m\}, \forall j \in \{1, \ldots, m\}, \tag{1.60}$$

où $r_{i,j}$ est une variable auxiliaire représentant la relation entre $\overline{z}_i$ et z_j ($r_{i,j} = 1$ si z_j est associée à $\overline{z}_i$, $r_{i,j} = 0$ sinon).

Le problème d'association tel qu'il est posé par l'équation (1.57) et les contraintes (1.58) et (1.59) peut être résolu à l'aide de l'algorithme de Munkres [30, 130], de l'algorithme des enchères (*auction*) [20] ou bien de l'algorithme Hongrois [103] sachant que ce dernier est adapté aux problèmes dont le nombre d'éléments en lignes et en colonnes dans la matrice d'association sont égaux. Ces algorithmes de résolution sont optimaux et fournissent une solution en un temps polynomial qui croît linéairement avec l'augmentation du nombre d'hypothèses [23].

La contrainte exprimée par l'équation (1.58) signifie qu'un objet connu, dont l'observation prédite est $\overline{z}_i$, peut être associé à une observation réelle z_j ou n'être associé à aucune observation, ce qui représente la non détection de l'objet en question. La contrainte exprimée par l'équation (1.59) signifie qu'une observation z_j peut être associée à un objet connu, représenté par son observation prédite $\overline{z}_i$, ou bien un nouvel objet noté par $*_j$. La question qui se pose par rapport à la contrainte (1.59) est : comment fixer le paramètre (distance) permettant d'associer une observation z_j à un nouvel objet $*_j$ et non pas aux objets existants ?

Le problème d'association pour un exemple de trois objets connus dont les observations prédites sont $\overline{z}_1$, $\overline{z}_2$ et $\overline{z}_3$ et quatre observations z_1, z_2, z_3 et z_4 peut être illustré par la matrice d'association représentée par la Table 1.1.

TABLEAU 1.1: Un exemple de matrice d'association : 3 objets connus $\{\overline{z}_1, \overline{z}_2, \overline{z}_3\}$ et 4 observations $\{z_1, z_2, z_3, z_4\}$.

	z_1	z_2	z_3	z_4
$\overline{z}_1$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$d_{1,4}$
$\overline{z}_2$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$d_{2,4}$
$\overline{z}_3$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	$d_{3,4}$
$*_1$	λ	∞	∞	∞
$*_2$	∞	λ	∞	∞
$*_3$	∞	∞	λ	∞
$*_4$	∞	∞	∞	λ

Comme illustré par la Table 1.1, la prise en compte de l'apparition de nouveaux objets est conditionnée par le paramètre λ. Ce paramètre représente la distance d'apparition de telle sorte que si une observation z_j est éloignée d'une distance λ de tous les objets connus, elle est considérée comme étant un nouvel objet $*_j$. Le paramètre λ est tiré de la table de χ^2 (distance de Mahalanobis suit une distribution de χ^2 [122]) en se basant sur une information *a priori* $P(z_j|\overline{z}_i)$, $i = 1, 2, ..., n$, qui représente la probabilité qu'une observation détectée soit générée par un objet déjà existant.

$$P(d_{i,j} < \lambda) = P(z_j|\overline{z}_i). \tag{1.61}$$

L'équation (1.61) donne une solution permettant la détermination de la distance d'apparition λ à l'aide d'information *a priori*, cependant il est souvent plus adéquat de procéder par un apprentissage [23, 87].

Le *GNN* est un algorithme très largement utilisé en pratique vue sa simplicité et ces appréciables performances calculatoires. Toutefois, comme c'est un algorithme déterministe (associe une seule hypothèse à chaque événement) et mono-scan (une décision d'association est prise à chaque pas de temps), il présente quelques inconvénients quant au suivi d'objets dont les trajectoires sont étroites par exemple [10, 23]. Le *GNN* est également peu efficace dans les environnements de suivi à fortes densités de fausses alarmes, il fait souvent appel aux fenêtres glissantes ou à des tests statistiques pour la confirmation des nouveaux objets et la gestion de la disparition d'objets, cet aspect est discuté dans la section 1.3. Quelques améliorations de l'algorithme *GNN* ont été proposées dans

[101, 165]. Des solutions basées sur les fonctions de croyance ont récemment repris le concept de l'algorithme *GNN* [47, 57, 88, 125, 129]. L'idée est de profiter des avantages de l'algorithme *GNN* concernant le temps de calcul par exemple et d'assurer une meilleure modélisation de l'incertitude et de l'imprécision liées à des scenarios conflictuels tel que le cas d'objets à trajectoires confondues. L'apport des fonctions de croyance pour le problème d'association est étudié dans le chapitre 3.

1.2.2.2 Association de données probabiliste conjointe (*Joint Probabilistic Data Association JPDA*)

L'algorithme *JPDA* représente une extension de la solution *PDA* au cas multi-objets [10]. La solution consiste à calculer un ensemble de probabilités jointes liant les objets suivis et les observations détectées au pas de temps k. La méthode *JPDA* suppose que le nombre d'objets suivis est connu. La mise à jour de leurs vecteurs d'état se fait à l'aide des équations de Kalman.

Pour chaque objet connu $i \in \{1, 2, ..., n\}$ et m observations détectées, le calcul de la densité de probabilité *a posteriori* s'effectue de la façon suivante :

$$p(x_i(k)|z(1:k)) = \mathcal{N}(x_i(k); \hat{x}_i(k), P_i(k)), \tag{1.62}$$

où

$$\hat{x}_i(k) = \hat{x}_i(k|k-1) + K_i(k)\tilde{z}_i(k), \tag{1.63}$$

avec

$$\tilde{z}_i(k) = \sum_{j=1}^{m} \beta_{i,j}(k) inov_j(k), \tag{1.64}$$

$$inov_j(k) = \overline{z}_i(k) - z_j(k), \tag{1.65}$$

et $K_i(k)$ représentant le gain de Kalman et $\beta_{i,j}(k)$ la probabilité que l'observation $z_j(k)$ soit générée par $\hat{x}_i(k)$ qui est la même que $P(z_j|\overline{z}_i)$ utilisée dans l'équation (1.61). La matrice de covariance est mise à jour de la façon suivante :

$$P_i(k) = \beta_{i,0} P_1(k) + \tilde{P}_i(k), \tag{1.66}$$

où

$$P_1(k) = [I - K(k)H(k)]P(k|k-1), \tag{1.67}$$

représente la mise à jour standard du filtre de Kalman et $\beta_{i,0}$ représente la probabilité qu'aucune observation n'est détectée. Le complément $\tilde{P}_i(k)$ de l'équation (1.66) est calculé ainsi :

$$\tilde{P}_i(k) = K_i(k)[\sum_{j=1}^{m} \beta_{i,j}\bar{z}_j(k)\bar{z}'_j(k) - \bar{z}_j(k)\bar{z}'_j(k)]K'_i(k). \tag{1.68}$$

Plus de détails sur le calcul des poids $\beta_{i,j}$ peuvent être trouvés dans [74]. Une mise à jour optimale à l'aide de l'algorithme *JPDA* nécessite un calcul minutieux des poids $\beta_{i,j}$. La complexité de ce calcul évolue de manière exponentielle avec l'évolution du nombre d'hypothèses (objets et/ou observations) [23], cela rend la méthode irréalisable pour des applications en temps réel. Plusieurs alternatives sous-optimales de l'algorithme *JPDA* sont développées : certaines visent à améliorer le temps de calcul de la méthode [16, 71, 112, 157, 192] et d'autres complètent la méthode avec une estimation du nombre d'objets suivis à chaque pas de temps [131, 132], qui est supposé connu dans l'algorithme de base, et également intégrer une action multi-scans ou fenêtre glissante afin de permettre une gestion d'apparitions et de disparitions d'objets [23, 38, 156]. Des combinaisons de l'algorithme *JPDA* et l'algorithme *IMM* sont proposées dans [50, 91] afin de permettre le suivi d'objets effectuant des manœuvres.

1.2.2.3 Suivi Multi-Hypothèses (*Multi-Hypothesis Tracking MHT*)

L'association des observations aux objets connus, dans le cas des approches *MHT*, se fait d'une manière différée. En effet, selon la propagation du conflit dans l'environnement du suivi, ces approches peuvent prendre des décisions à chaque pas de temps dans le cas où le conflit est faible, elles peuvent également faire propager l'information et prendre une décision à la réception de données plus claires, dans le cas conflictuel. La méthode *MHT* est initialement introduite par Reid dans [151]. Depuis, plusieurs extensions ont été proposées [21, 49, 177].

La sortie d'un algorithme *MHT* est une liste d'hypothèses avec leurs probabilités correspondantes. La probabilité d'une hypothèse h_i sachant une observation z est calculée grâce au théorème de Bayes :

$$P(h_i|z) = \frac{\Lambda(z|h_i)}{\sum_i \Lambda(z|h_i)P(h_i)}, \tag{1.69}$$

où $P(h_i)$ représente la probabilité *a priori* de l'hypothèse h_i, $\Lambda(z|h_i)$ est la vraisemblance de la mesure z sachant l'hypothèse h_i et $P(h_i|z)$ représente la probabilité *a posteriori* de h_i après avoir perçu l'observation z.

Le théorème de Bayes dans l'équation (1.69) permet d'évaluer la probabilité de chaque hypothèse à chaque réception d'une nouvelle observation z. Toutefois, dans une méthode *MHT*, une hypothèse peut correspondre à différents événements dans l'environnement du suivi. Cependant le calcul des probabilités est différent d'une hypothèse à une autre, par exemple :

- l'observation reçue correspond à une fausse alarme (notée par 0 dans la figure 1.5), la probabilité de l'hypothèse correspondante se calcule par :

$$P(h_i) = \frac{\beta_{fa}(1-p_d)^n}{C} P(h_i^-), \tag{1.70}$$

  où β_{fa} est une information *a priori* représentant la densité de fausses alarmes, P_d la probabilité de détecter un nouvel objet dans l'environnement de suivi, $P(h_i^-)$ la probabilité que l'hypothèse h_i, avant la réception des observations, soit vraie. L'entier n représente le nombre d'objets connus et C est une constante de normalisation dépendant du volume observé [8].

- l'observation reçue j correspond à un objet connu i, la probabilité de l'hypothèse correspondante se calcule comme suit :

$$P(h_i) = \frac{d_{i,j}P_d(1-P_d)^n}{C} P(h_i^-), \tag{1.71}$$

  où $d_{i,j}$ représente la distance de Mahalanobis entre l'observation j et l'objet connu i.

- l'observation reçue peut également correspondre à l'apparition d'un nouvel objet (noté par $\star$ dans la figure 1.5), dans ce cas la probabilité de l'hypothèse correspondante se calcule par :

$$P(h_i) = \frac{\beta_{no}(1-P_d)^n}{C} P(h_i^-), \tag{1.72}$$

  où β_{no} représente une information *a priori* sur la densité de nouveaux objets.

Plus de détails quant aux calcul des probabilités des différentes hypothèses dans les méthodes *MHT* peuvent être trouvés dans [12].

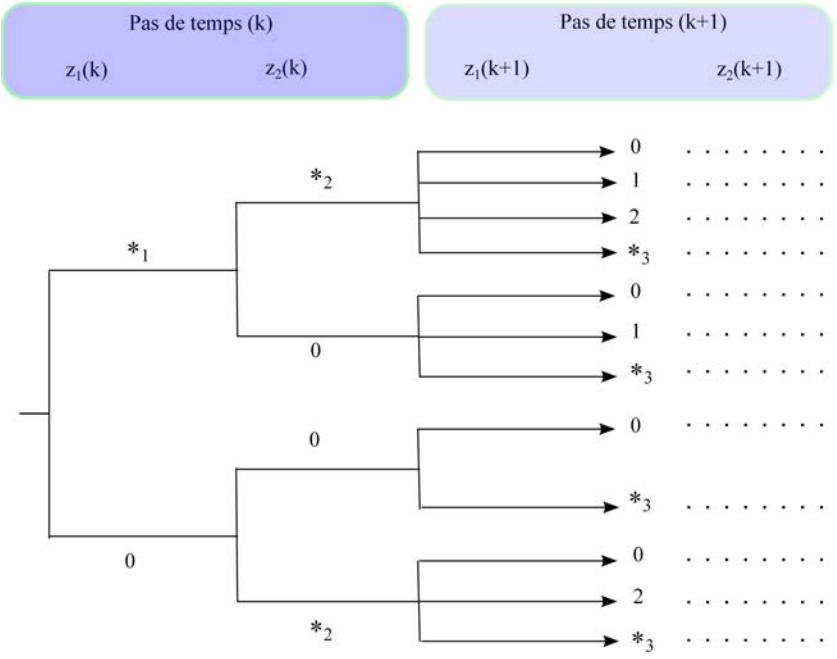


FIGURE 1.5: Génération d'hypothèses dans une méthode *MHT*.

L'implémentation directe d'une méthode *MHT* est quasiment impossible, dû à l'augmentation exponentielle du nombre d'hypothèses, comme cela est illustré par la figure 1.5. L'implémentation de telles méthodes, ne peut être réalisée que si on limitait le nombre d'hypothèses, ce qui peut être fait à l'aide de méthodes dites d'élagage (*pruning*). Plusieurs méthodes d'élagage ont été proposées [23], la plus simple consiste à éliminer les hypothèses dont les probabilités sont inférieures à un seuil donné. Les méthodes d'élagage sont souvent dépendantes de l'application considérée. Plusieurs variantes de la méthode *MHT* ont été proposées dans la literature, en l'occurrence la méthode *MHT* Probabiliste (Probabilistic Multi-Hypothesis Tracking *PMHT*) [49, 177]. La méthode *PMHT* est conçue afin d'améliorer le temps de calcul de la méthode *MHT* classique mais aussi afin d'être plus robuste quant à la gestion de scénarios conflictuels. Dans une étude plus récente [45], il a été démontré que les performances des deux méthodes sont équivalentes lorsque ces dernières sont comparées dans des conditions similaires.

1.2.3 Suivi multi-objets sans association de données (*Probability Density Hypothesis PHD*)

Contrairement au filtrage de Kalman et au filtrage particulaire qui font propager le premier et le second moments de la densité de probabilité d'un objet lors du suivi, le filtrage *PHD* ne fait propager que le premier moment de la densité correspondant à tous les objets. Cette méthode est introduite par Mahler [115] dans le cadre du suivi d'objets. Elle est basée sur la théorie des Ensembles Finis Aléatoires (*Random Finit Sets RFS*), elle a fait l'objet d'une recherche intensive durant ces dernières années [108, 109, 182–184]. La méthode *PHD*, permet de suivre les objets comme étant un ensemble, ce qui la rend indépendante de la phase d'association [184]. La technique *PHD* permet également d'estimer le nombre d'objets suivis, cela est présenté dans [114, 116, 154], cette faculté assure une certaine robustesse par rapport aux fausses alarmes.

Plusieurs implémentations sont proposées pour mettre en œuvre les filtres *PHD*. Certaines sont basées sur le filtrage particulaire [41, 86, 153, 182, 183, 188], elles sont également appelées méthodes *PHD* Monte Carlo séquentielles (*Sequential Monte Carlo Probability Hypothesis Density SMC-PHD*). Les questions qui se posent par rapport au filtrage particulaire concernant le temps de calcul ou le choix de la densité d'importance se posent également pour les méthodes *SMC-PHD*. D'autres implémentations sont indépendamment considérées, elles sont basées sur le filtrage de Kalman, ces méthodes sont généralement appelées *Gaussian Mixture Probability Hypothesis Density GM-PHD* [144, 181, 185] ou *GM-CPHD* pour les solutions tenant compte de la cardinalité (nombre d'objets) [77, 146, 178, 189]. Ces méthodes sont également limitées à la condition de Gaussienneté afin que l'application du filtrage de Kalman soit possible.

Les méthodes *PHD* donnent d'appréciables résultats concernant le suivi d'objets dans des environnements à forte densité de fausses alarmes. Il serait également moins complexe en terme de calcul par rapport à la méthode *MHT* [146, 178]. Toutefois, l'application de cette méthode pour des fins de classification à base de données cinématiques doit être développée. En effet, une telle classification nécessite une association explicite des observations aux objets suivis, ce qui permet de garder leurs identités. Cette faculté est inexistante dans la méthode *PHD* [181, 185].

Afin d'éviter la phase d'association pour le suivi multi-objets, l'article [18] présente une autre solution qui consiste à rassembler tous les vecteurs d'état de tous les objets suivis en un seul vecteur étendu. Cela engendre une modification des modèles d'état et d'observation quand cela s'avère possible. Les modèles étendus sont ensuite utilisés par le filtre *UKF* afin de permettre un

suivi de l'ensemble des objets. Cette technique vient d'émerger comme cela est signalé par les auteurs dans [18]. Elle est également limitée au suivi de deux objets uniquement. Son extension au cas de plusieurs objets s'avère difficile. De plus l'identité des objets n'est pas conservée [18], ce qui empêche une classification à base de données cinématiques.

1.3 Gestion d'apparitions et de disparitions d'objets

La robustesse aux fausses alarmes est un point important dans le domaine du suivi d'objets. Une fausse alarme peut être due à un bruit important, conditions d'observabilité réduites et d'autres phénomènes comme l'ombrage par exemple. Assurer une robustesse par rapport aux fausses alarmes revient à suivre leur évolution et les distinguer des objets d'intérêt. Les méthodes d'association à décision différée comme le *MHT* [22, 151] et le *PMHT* [45, 177] gèrent automatiquement les fausses alarmes, ces dernières se retrouvent généralement avec des probabilités de subsistance assez faibles, et les heuristiques d'élagage se chargent alors de les supprimer. La question se pose spécialement pour les méthodes d'association mono-scan comme le *GNN* et le *JPDA* par exemple. La version basique de l'algorithme *JPDA* ne présente pas de méthode explicite permettant de gérer les fausses alarmes. Toutefois, d'autres méthodes plus élaborées sont proposées comme par exemple : l'algorithme *IJPDA* [131–133] qui introduit une action intégrale permettant d'évaluer les probabilités d'association des objets au fil du temps, et ainsi supprimer ceux ayant des probabilités d'association faibles. D'autres variantes de l'algorithme *JPDA* proposent l'utilisation de fenêtres de temps pour faire propager les probabilités d'association [23, 38, 156].

La version de base de l'algorithme *GNN* ne gère également pas les fausses alarmes, la solution nécessite des algorithmes complémentaires comme par exemple l'estimation de la fréquence de détection des objets sur une fenêtre de temps ou bien l'utilisation de tests statistiques comme le test du rapport de vraisemblances des objets par exemple. Cette section décrit le test du rapport (ratio) de vraisemblances qui peut être utilisé avec les algorithmes d'association mono-scan et déterministes comme le *GNN* afin de permettre une certaine robustesse aux fausses alarmes.

1.3.1 Test du ratio de vraisemblances

Le test de Wald [23, 186] est un test statistique se présentant comme un rapport de vraisemblances (*Likelihood Ratio LR*) :

$$LR = \frac{p(D|h_1)p_0(h_1)}{p(D|h_0)p_0(h_0)} \triangleq \frac{P_T}{P_F}, \tag{1.73}$$

où h_1 et h_0 sont respectivement les hypothèses qu'une donnée D soit vraie ou fausse, avec respectivement les probabilités P_T et P_F. La quantité $p(D|h_i)$ représente la vraisemblance de l'événement D sachant que l'hypothèse h_i est vraie et $p_0(h_i)$ représente la probabilité *a priori* de l'hypothèse h_i.

Il est souvent plus adéquat de considérer le logarithme du ratio de vraisemblances (*Logarithm Likelihood Ratio LLR*) donné par :

$$LLR = ln[P_T/P_F], \tag{1.74}$$

ainsi la probabilité que l'événement D soit vrai peut être déduite :

$$P_T/P_F = \frac{P_T}{1 - P_T} = e^{LLR}, \tag{1.75}$$

$$P_T = \frac{e^{LLR}}{1 + e^{LLR}}. \tag{1.76}$$

Le résultat de l'équation (1.76) est intéressant dans la mesure où il peut aider à décider si un objet suivi est un objet d'intérêt ou bien seulement une fausse alarme.

1.3.2 Ratio de vraisemblances pour la confirmation d'objets

Dans un contexte de suivi, le test de vraisemblances peut être cumulé sur plusieurs pas de temps. Pour un bruit de mesure non corrélé et sur une période de temps $k = 1 : K$, le test peut se présenter ainsi :

$$LR(1:K) = L_0 \prod_{k=1}^{K} LR(k), \tag{1.77}$$

où $LR(k)$ représente le ratio de vraisemblances calculé à l'instant k et $L_0 = P_0(h_1)/P_0(h_0)$.

Le logarithme du ratio de vraisemblances cumulé (1.77) est simplement une somme, il est noté par $L(1:K)$ et donné ainsi :

$$L(1:K) = ln[LR(1:K)] = \sum_{k=1}^{K} LR(k). \tag{1.78}$$

La formule (1.78) peut être calculée récursivement de la manière suivante :

$$L(k) = L(k-1) + \Delta L(k), \tag{1.79}$$

avec

$$L(k) = \begin{cases} ln[1 - P_d] & \text{si l'objet n'est pas détecté,} \\ ln[\frac{P_d}{(2\pi)^{q/2}\beta_{FT}\sqrt{|S_t|}} - \frac{d_t^2}{2}] & \text{si l'objet est détecté,} \end{cases} \tag{1.80}$$

où P_d représente une probabilité de détection et β_{FT} la densité de fausses alarmes. Ce sont des informations sur le capteur et l'environnement du suivi qui sont supposées connues *a priori*. Plus de détails concernant l'utilisation du test de vraisemblances pour le suivi d'objets peuvent être trouvés dans [23, chapitre 2, pages 327-334]. Le test est calculé pour chaque objet t (confirmé ou non) suivi, il est également appelé : fonction score (*Score Function SF*). La prise de décision quant à la validité ou la non validité des objets se fait de la manière suivante :

$$\begin{cases} L_t(k) > T_c & \text{confirmation de l'objet suivi,} \\ L_t(k) < T_s & \text{suppression de l'objet suivi,} \\ T_s < L_t(k) < T_c & \text{doute concernant l'objet suivi.} \end{cases}$$

Les seuils T_c et T_s sont respectivement calculés par :

$$T_c = ln[\frac{1-\xi}{\sigma}], \quad T_s = ln[\frac{\xi}{1-\sigma}], \tag{1.81}$$

où ξ et σ sont des probabilités de fausses décisions supposées connues *a priori* : ξ représente la probabilité de confirmer de faux objets et σ la probabilité de supprimer de vrais objets.

La figure 1.6 illustre le processus de prise de décision pour une fonction score d'un objet donné.

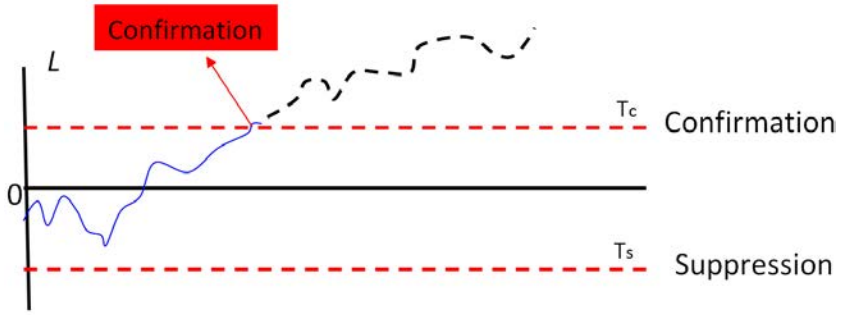


FIGURE 1.6: Illustration de l'évolution de la fonction score d'un objet donné.

1.4 Conclusion

Ce chapitre résume les différentes approches dédiées au suivi d'objets. Leurs avantages et inconvénients sont discutés, notamment dans la section 1.1.2.2 où une comparaison entre le filtrage de Kalman et le filtrage particulaire est donnée. À l'image de cette comparaison, toutes les approches décrites sont soumises aux dilemmes liés au suivi d'objets, notamment ceux qui sont liés à l'hypothèse de Gaussienneté ou de linéarité, à la prise en compte des complexités citées dans l'introduction et au final, au temps de calcul qui est également important pour les systèmes de suivi dont les applications sont souvent en temps réel.

Les différents algorithmes présentés dans ce chapitre ont aidé à construire une solution de suivi de plusieurs objets à la fois. Cette solution est décrite dans le chapitre 2. Elle a permis d'assurer le suivi d'objets dans toutes les simulations de cette thèse qui est essentiellement orientée vers la phase d'association et celle de la classification d'objets.

Chapitre 2

Classification à base de données cinématiques

Durant les 40 dernières années, une dizaine d'avions civils ont été abattus [63]. Le dernier événement de ce genre est survenu tout récemment, en juillet 2014, où un avion civil de la *Malaysia airlines* est abattu dans l'espace aérien ukrainien avec 298 personnes abord [7].

Il est peu probable que l'acte d'abattre un avion civil soit volontaire, mais la conséquence d'une erreur de classification faisant prendre les avions civils pour des avions militaires par exemple.

Ce chapitre traite de la classification d'objets en mouvement. Il présente des cas de classification avec un niveau d'ambiguïté important où il compare deux types de classifieurs : le classifieur Bayésien, qui est communément utilisé, et le classifieur crédal qui a été plus récemment proposé. De premières comparaisons entre ces deux classifieurs étaient développées dans la littérature [19, 155, 175], où l'intérêt est porté sur la classification d'un seul objet aérien en mouvement. Dans ce chapitre les classifieurs sont comparés dans un contexte multi-objets. Ajouté à l'ambiguïté liée à la classification de



FIGURE 2.1: Avion de ligne escorté par deux avions de chasse.

chaque objet, le risque de confusion entre les objets complexifie davantage le problème, notamment dans le cas où les trajectoires des objets sont étroitement proches, comme cela est illustré par la figure 2.1.

Deux exemples de classifications sont donnés. Le premier concerne la classification de cibles aériennes, c'est un exemple où les classes sont constantes dans le temps. Pour un tel exemple, il a été démontré dans la littérature que le classifieur crédal donne de meilleurs résultats par rapport au classifieur Bayésien [175], ce pour un seul objet suivi. Dans ce chapitre, ce résultat a été confirmé dans le cadre multi-objets. Par ailleurs, la question d'erreurs de classification, dues à la confusion entre les trajectoires est évoquée. Cette question est directement liée à l'algorithme mis en place pour suivre les objets, notamment, dans la phase chargée d'associer les multiples observations aux multiples objets connus. La solution proposée dans ce chapitre rassemble plusieurs algorithmes tirés de la littérature dédiée au suivi d'objets [15, 23]. Une partie de cette littérature a été présentée dans le chapitre 1.

Le second exemple de classification porte sur la reconnaissance des comportements des piétons. C'est un exemple où les objets (piétons) peuvent changer de classe. Cet exemple a permis de mettre en avant la souplesse des fonctions de croyance tel qu'une simple modification du classifieur crédal permet de s'adapter aux changements de classe. Le suivi des piétons est assuré par le même algorithme que dans le cas des cibles aériennes mais avec des modèles d'évolution adéquats.

La section 2.1 de ce chapitre reprend quelques notions de base sur les fonctions de croyance. La section 2.3 présente et positionne la solution proposée concernant le suivi de plusieurs objets à la fois. Les deux algorithmes de classification, crédal et Bayésien, sont décrits dans la section 2.3.2. L'exemple de suivi et de classification de plusieurs cibles aériennes est traité dans la section 2.4. L'exemple visant à reconnaître les comportements des piétons est exposé dans la section 2.5. La section 2.6 illustre la sensibilité de la classification crédale aux cas conflictuels d'objets à trajectoires confondues. Au final, une discussion concluant le chapitre est donnée dans la section 2.7.

2.1 Notions de base sur les fonctions de croyance

L'origine de la théorie des fonctions de croyance remonte aux travaux d'Arthur Dempster concernant la généralisation du formalisme Bayésien [53, 54]. L'aspect évidentiel, ou encore crédal, de la théorie est formalisé par Glenn Shafer [162] sous l'appellation : théorie des fonctions de croyance

qui est également connue par la théorie de Dempster-Shafer (DS). Dans les travaux de Philippe Smets [169, 174], la théorie est reprise sous le nom : Modèle de Croyances Transférables (MCT).

Cette section introduit les notions de base du MCT, qui ont contribué au développement du travail présenté dans ce manuscript.

2.2 Représentation de l'informations

Dans le cadre des fonctions de croyance, l'ensemble $\Omega = \{\omega_1, \omega_2, ...\}$ fait référence à un cadre de discernement, c'est l'ensemble de toutes les hypothèses possibles en réponse à une question donnée. Il est considéré exhaustif et ces éléments sont considérés exclusifs. À la différence du formalisme Bayésien qui distribue la croyance d'une source d'information (capteur, agent, ou autres.) sur les éléments ω_i en tant que singletons, le formalisme crédal est en mesure de distribuer la croyance sur les éléments ω_i ou bien sur des ensembles d'éléments $A \subseteq \Omega$. En effet, le principal avantage des fonctions de croyance est le fait de pouvoir préserver la connaissance sur des ensembles d'hypothèses, ce lorsque l'information reçue ne permet pas de les départager. Cela s'opère au niveau crédal (voir la figure 2.2) où l'on peut également propager et/ou combiner l'information. La fonction la plus souvent utilisée pour représenter l'information au niveau crédal est dite : fonction de masse, elle est également appelée *Basic Belief Assignment (BBA)*. À tout instant l'information peut être projetée sur les éléments singletons de Ω afin de prendre une décision, cela s'opère au niveau dit : pignistique (voir la figure 2.2).

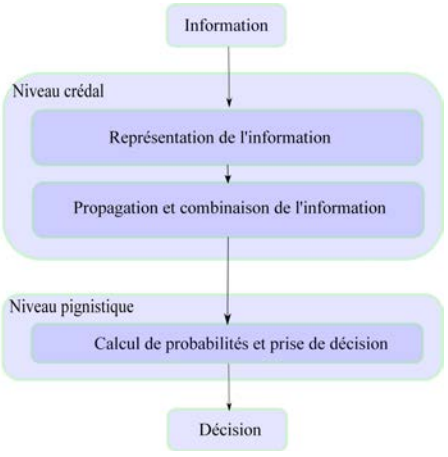


FIGURE 2.2: Représentation de l'information dans le cadre des fonctions de croyance.

Une fonction de masse $m : 2^{\Omega} \rightarrow [0, 1]$ sur Ω satisfait la condition :

$$\sum_{A \subseteq \Omega} m(A) = 1, \tag{2.1}$$

avec $A \subseteq \Omega$. Les sous-ensembles A de Ω tel que $m(A) > 0$ sont appelés : les éléments focaux (EF) de m. Outre la fonction de masse m, d'autres fonctions permettent de représenter la connaissance. On trouve notamment la fonction de crédibilité et la fonction de plausibilité qui sont définies par :

$$bel(A) = \sum_{B \subseteq A,\ B \neq \emptyset} m(B), \quad \forall A \subseteq \Omega, \tag{2.2}$$

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \quad \forall A \subseteq \Omega, \tag{2.3}$$

Les deux fonctions *bel* et *pl* offrent deux alternatives de représentation de l'information. Elles sont équivalentes et représentent des mesures de probabilités dans le cas où les éléments focaux sont des singletons, c'est-à-dire $m(A) > 0$ avec $|A| = 1$. En revanche, si les éléments focaux sont emboîtés $A_1 \subseteq A_2 \subseteq ...$, alors la fonction *pl* représente une mesure de possibilité et *bel* une mesure de nécessité. Cette souplesse dans la représentation de l'information permet d'être plus ou moins engagé dans la propagation, la combinaison et la prise de décision.

2.2.1 Information au niveau crédal

Au niveau crédal, la théorie des fonctions de croyance offre une panoplie d'opérations permettant de modifier l'information d'une manière dynamique. Parmi ces opérations, on trouve des règles de combinaison qui permettent de fusionner des fonctions de croyance qui s'expriment sur le même cadre de discernement. On trouve également des fonctions qui proposent de réviser une connaissance, l'affaiblir, la renforcer ou la projeter sur des cadres plus ou moins précis. Quelques uns des outils permettant le traitement de l'information dans le cadre crédal sont décrits dans ce qui suit.

2.2.1.1 Règles de combinaison

Deux règles de combinaison sont principalement utilisées dans cette thèse, elles sont décrites ici.

Règle de combinaison conjonctive : la combinaison conjonctive (*Conjunctive Rule of Combination CRC*) de deux fonctions de masses m_{s_1} et m_{s_2}, issues de deux sources indépendantes et

fiables s_1 et s_2, est définie ainsi :

$$m_{s_1 \textcircled{\cap} s_2}(A) = (m_{s_1} \textcircled{\cap} m_{s_2})(A) = \sum_{A_1 \cap A_2 = A} m_{s_1}(A_1) m_{s_2}(A_2), \tag{2.4}$$

où $A_1,\ A_2 \subseteq \Omega$.

Une version normalisant le poids $m_{s_1 \textcircled{\cap} s_2}(\emptyset)$ est décrite par l'équation (2.5), c'est la règle de *Dempster* :

$$m_{s_1 \textcircled{\cap} s_2}(A) = \frac{\sum_{A_1 \cap A_2 = A} m_{s_1}(A_1) m_{s_2}(A_2)}{1 - \sum_{A_1 \cap A_2 = \emptyset} m_{s_1}(A_1) m_{s_2}(A_2)}. \tag{2.5}$$

La combinaison conjonctive est très utilisée pour fusionner les informations dans le cadre des fonctions de croyance. Plusieurs autres versions avec différentes manières de normaliser le poids conflictuel $m_{s_1 \textcircled{\cap} s_2}(\emptyset)$ ont été développées [56, 64, 106, 173, 187].

Règle de combinaison disjonctive : la combinaison disjonctive de deux fonctions de masse m_{s_1} et m_{s_2}, issues de deux sources d'informations distinctes s_1 et s_2 dont une au moins est supposée fiable, est définie par :

$$m_{s_1 \textcircled{\cup} s_2}(A) = (m_{s_1} \textcircled{\cup} m_{s_2})(A) = \sum_{A_1 \cup A_2 = A} m_{s_1}(A_1) m_{s_2}(A_2), \tag{2.6}$$

où $A_1,\ A_2 \subseteq \Omega$.

Tout comme la règle de combinaison conjonctive, la règle disjonctive est commutative et associative. La règle conjonctive est souvent utilisée dans un environnement de sources fiables, elle transfert la croyance sur les éléments sur lesquels toutes les sources s'accordent. La combinaison disjonctive est plus prudente, elle favorise l'ignorance dans la mesure où elle est supposée être utilisée dans un environnement de sources incertaines [56, 168]. Toutefois, des règles de combinaison conjonctives prudentes ont également été introduites [55, 59, 95].

2.2.1.2 Révision de connaissances

Dans le cadre des fonctions de croyance, il est tout à fait possible de modifier la croyance attribuée à un événement donné. Le grossissement, le raffinement, la marginalisation, l'extension vide, le conditionnement, l'affaiblissement, le renforcement sont autant d'opérations qui peuvent être

utiles pour modifier l'information d'une manière dynamique et ainsi s'adapter aux changements de l'environnement. Certaines de ces opérations, comme le grossissement et le raffinement, consistent à changer le support de l'information, notamment, le cadre de discernement qui peut devenir plus ou moins précis. D'autres opérations, comme le conditionnement et l'affaiblissement par exemple, consistent à changer la croyance qui est attribuée aux événements, selon leur correlation ou la fiabilité des sources d'information.

Marginalisation et extension vide : la marginalisation consiste à exprimer une fonction de masse m^{Ω} sur un cadre de discernement plus grossier Θ. L'opération est symbolisée par une flèche orientée vers le bas $\downarrow$:

$$m^{\Omega\downarrow\Theta}(B) = \sum_{A\subseteq\Omega,\ A\downarrow\Theta=B} m^{\Omega}(A), \quad \forall B \subseteq \Theta, \tag{2.7}$$

L'extension vide est duale à la marginalisation, elle permet d'exprimer une fonction de masse m^{Θ} sur un cadre de discernment plus précis Ω. Cette opération est symbolisée par une flèche vers le haut $\uparrow$:

$$m^{\Theta\uparrow\Omega}(A) = \begin{cases} m^{\Theta}(B) & si\ B \subseteq A,\ B \subseteq \Omega, \\ 0 & sinon. \end{cases} \tag{2.8}$$

Affaiblissement et conditionnement : une fonction de masse m peut être modifiée selon le degré de fiabilité de sa source. La fonction de masse est inchangée si la source est fiable ou bien affaiblie si la source est estimée non fiable. Le degré d'affaiblissement, noté $\alpha \in [0 \quad 1]$, est égal à 0 si la source est complètement fiable, à 1 si la source est non fiable. La fonction de masse affaiblie est notée ${}^{\alpha}m$ et calculée par :

$$\begin{cases} {}^{\alpha}m(A) = (1-\alpha)m(A), \quad \forall A \subset \Omega, \\ {}^{\alpha}m(\Omega) = (1-\alpha)m(\Omega) + \alpha. \end{cases} \tag{2.9}$$

Ce mécanisme de correction proposé initialement par Shafer [162] affaiblit les éléments focaux $\{A|m(A) > 0\}$ d'une manière uniforme, un autre mécanisme de correction tenant compte du contexte est proposé par Mercier et *al* [126].

Une fonction de masse m conditionnée par un événement catégorique B est notée $m[B]$. La fonction de masse de l'événement B est dite catégorique si $m(B) = 1$, elle est plus simplement notée par

m_B. Le conditionnement de la fonction de masse m s'effectue par le moyen d'une combinaison conjonctive :

$$m[B] = m \textcircled{\cap} m_B, \quad B \in \Omega. \tag{2.10}$$

Renforcement vers un élément : il a été montré dans [124] qu'il est tout à fait possible de renforcer la croyance en un élément $A \subseteq \Omega$ si des informations additionnelles le justifient. En utilisant les fonctions de masse, cette opération est exprimée de la manière suivante :

$$^{\mu}m = (1 - \mu)\ m + \mu\ m_A, \tag{2.11}$$

où $\mu \in [0\ 1]$ représente le taux de renforcement, et m_A représente une fonction de masse catégorique sur l'ensemble A. En terme de plausibilités, l'operation de renforcement vers un ensemble A en détriment d'un autre ensemble B par exemple, s'exprime ainsi :

$$\begin{cases} ^{\mu}pl(A) & = \mu\ pl(A) + (1 - \mu), \\ ^{\mu}pl(B) & = \mu\ pl(B). \end{cases} \tag{2.12}$$

Cette opération signifie que l'ensemble $B \subseteq \Omega$ est affaibli en le multipliant par le paramètre $\mu \in [0\ 1]$ et en contre partie l'ensemble $A \subseteq \Omega$ est renforcé en rajoutant le terme $(1 - \mu)$ ce qui assure la conservation des fonctions pl. Ce mécanisme est utilisé dans le chapitre 3.

2.2.2 Inférence crédale

D'une manière générale, l'inférence signifie l'estimation d'événements à l'aide d'autres événements. En particulier, cela peut correspondre à l'estimation des paramètres d'un système à l'aide de l'ensemble des observations faites par un capteur par exemple. Cette notion a un solide fondement dans le cadre Bayésien et commence à être développée dans le cadre des fonctions de croyance, notamment, à l'aide du théorème de Bayes généralisé (*Generalized Bayes Theorem GBT*).

Théorème de Bayes Généralisée : dans le contexte d'estimation, le théorème de Bayes classique propose de calculer des probabilités *a posteriori* d'événements $\omega_i \in \Omega$ à l'aide de probabilités *a priori* et des mesures de vraisemblances notées $\Lambda(\omega_i|z)$ qui représentent les probabilités conditionnelles $P(z|\omega_i)$ où z représente l'observation (la mesure). Le calcul des probabilités est effectué de la manière suivante :

$$P(\omega_i|z) = \frac{\Lambda(\omega_i|z)P(\omega_i)}{\sum\limits_{\omega_j} \Lambda(\omega_j|z)P(\omega_j)} \ . \tag{2.13}$$

Quand aucune information initiale sur la distribution des probabilités *a priori* $P(\omega_j)$ est disponible, la distribution est choisie équiprobable.

Avec la même idée, Smets a introduit le *GBT* [168] dédié à l'inférence de l'information dans le cadre crédal. Le *GBT* permet de calculer une fonction de masse ou une fonction de plausibilité à l'aide des vraisemblances. La mesure de vraisemblance $P(z|\omega_i)$ correspond à une plausibilité conditionnelle $pl[\omega_i](z)$ dans le cadre crédal. À l'aide des plausibilités $pl[\omega_i](z)$, les fonctions de plausibilité et les fonctions de masse *a posteriori* des paramètres $\omega_i \in \Omega$ sont calculées ainsi :

$$pl[z](A) = 1 - \prod_{\omega_i \in A} (1 - pl[\omega_i](z)), \tag{2.14}$$

$$m[z](A) = \prod_{\omega_i \in A} pl[\omega_i](z)(1 - \prod_{\omega_i \in \overline{A}} (1 - pl[\omega_i](z)), \tag{2.15}$$

où $A \subseteq \Omega$. Quand une information *a priori* pl_0 ou m_0 est à disposition, elle est conjonctivement combinée avec la fonction de plausibilité ou la fonction de masse calculées respectivement par les équations (2.14) et (2.15). Dans le cas où aucune information *a priori* n'est disponible, il n'est pas nécessaire d'imposer une distribution initiale comme dans le cas Bayésien.

Étant donné les éventuelles erreurs de mesure, l'information sur les observations $pl[\omega_i](z)$ est choisie de manière à être la moins engagée possible [58, 168]. Elle obéit au principe d'engagement minimum.

Principe d'engagement minimum : le principe d'engagement minimum est utilisé quand il y a une ambiguïté quant aux degrés de croyance à accorder aux sous-ensembles $A \subseteq \Omega$. L'ambiguïté pourrait être levée si à chaque sous ensemble $A \subseteq \Omega$ on allouait la croyance la plus vague possible. Le principe d'engagement minimum est utilisé dans les opérations d'extension vide, *ballooning* et dans la définition des fonctions de plausibilités conditionnelles $pl[\omega_i](z)$ dans le *GBT*. En terme de plausibilités, le principe se traduit de la façon suivante : soient deux fonctions de plausibilités pl_1 et pl_2 s'exprimant sur le même sous-ensemble $A \in \Omega$. La plausibilité pl_1 est dite moins engagée que pl_2 si :

$$pl_1 > pl_2, \tag{2.16}$$

La plausibilité pl_1 n'est pas plus engagée que pl_2 si $pl_1 \geq pl_2$. Le choix de plausibilités égales aux vraisemblances dans le GBT est considéré comme étant le choix le moins engagé possible [175]. Le principe d'engagement minimum peut être formalisé pour les fonctions *bel* et m également [168].

2.2.3 Information au niveau pignistique

Soit $\mathcal{A} = a_1, a_2, ...$, l'ensemble des décisions qui peuvent être prises. La décision optimale a est celle qui minimise un certain risque noté $\rho(a)$:

$$\rho(a) = \sum_{w \in \Omega} L(a, \omega) P(\omega), \tag{2.17}$$

où la fonction $L(a, \omega)|\ L :\ \mathcal{A} \times \Omega \rightarrow \mathbb{R}$ représente la perte qui peut être engendrée par le choix de a sachant que la vérité est ω. La quantité $P(\omega)$ est une mesure de probabilité.

Dans le cadre des fonctions de croyance, des probabilités pignistiques sont souvent considérées pour la prise de décisions. Elles sont notées par $BetP$ et calculées de la manière suivante :

$$BetP(\omega) = \sum_{\omega \subseteq A,\ A \subseteq \Omega} \frac{m(A)}{|A|(1 - m(\emptyset))}, \quad \forall \omega \in \Omega, \tag{2.18}$$

où $|A|$ représente la cardinalité de l'ensemble A. La décision a ayant la probabilité pignistique maximum est celle qui minimise le risque $\rho(a)$. La transformation pignistique exprimée par l'équation (2.18) est justifiée dans [172]. Les probabilités pignistiques sont largement utilisées dans le cadre des fonctions de croyance, toutefois d'autres mécanismes de prise de décision existent, notamment le choix de la décision avec le maximum de plausibilité *pl* ou le maximum de crédibilité *bel* par exemple. Plus d'informations sur la prise de décision dans le cadre des fonctions de croyance peuvent être trouvées dans [42, 170].

2.3 Solution pour le suivi et la classification de plusieurs objets à la fois

Cette section propose une solution complète dédiée au suivi et à la classification d'objets manœuvrants [90]. La classification est basée sur les données cinématiques des objets : les données

cinématiques provenant des *IMMs* sont utilisées pour reconnaître les comportements ainsi que les classes des objets. À cet effet, deux types de classifications sont considérés : une classification Bayésienne et une classification crédale. La classification Bayésienne est initialement étudiée dans [155] pour la classification d'un seul objet aérien manœuvrant. L'exemple de classification considéré est particulier dans la mesure où les classes sont imbriquées, cela représente un niveau d'imprécision difficile à gérer à l'aide des probabilités classiques. Pour le même exemple de classification, il a été démontré dans [175] que le classifieur crédal est mieux adapté, il permet de mieux gérer l'imprécision liée à ce problème de classification.

Dans [90] et dans ce chapitre, les deux types de classifications sont étendues et comparées dans un cadre d'objets multiples. Notamment pour le cas d'objets aériens avec des classes imbriquées et constantes dans le temps et le cas d'un nouvel exemple de piétons avec des classes partiellement corrélées et changeantes dans le temps. Il est montré dans ce chapitre qu'une simple révision de l'information permet au classifieur crédal de s'adapter aux changements de classe. Les deux exemples de classifications utilisent une même solution concernant le suivi, qui est décrite dans la section 2.3.1. La section 2.3.2 décrit les deux classifieurs Bayésien et crédal. Les sections 2.4 et 2.5 sont respectivement dédiées aux deux exemples d'application concernant la classification d'objets aériens et la classification de piétons.

2.3.1 Solution pour le suivi d'objets multiples

La solution de suivi multi-objets proposée rassemble un certain nombre d'algorithmes auparavant développés, et répond à un bon nombre de difficultés liées au suivi d'objets multiples. Elle est constituée d'algorithmes *IMMs* pour suivre indépendamment plusieurs objets manœuvrants. L'association des observations aux objets connus est en premier temps assurée par un algorithme *GNN*, cet algorithme peut être remplacé par une des solutions étudiées dans le chapitre 3. Malgré son appréciable performance par rapport au temps de calcul (voir la section 3.3 du chapitre 3), le *GNN* souffre d'une incapacité à confirmer l'apparition d'objets et de gérer les fausses alarmes. Afin de remédier à ce problème, la solution proposée utilise des fonctions score. Une fonction score est un test de vraisemblance récursif qui permet de mesurer la validité des objets au cours du temps (voir la section 1.3 du chapitre1). La fonction score est croissante quand l'objet en question est fréquemment détecté, elle est décroissante si l'objet est peu fréquemment détecté ou non détecté. Les valeurs de la fonction score permettent de prendre des décisions quant à la confirmation ou la suppression d'objets.

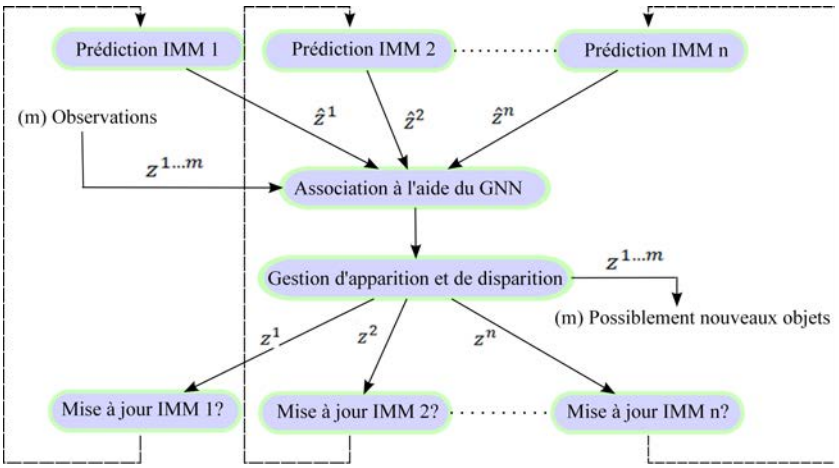


FIGURE 2.3: Solution pour le suivi de plusieurs objets à la fois.

La solution de suivi d'objets multiples est illustrée par la figure 2.3. Les algorithmes qui interviennent dans la solution sont décrits dans les sections 1.1.3, 1.2.2.1 et 1.3 du chapitre 1. La solution de suivi complète est résumée par l'algorithme 2.1.

Comme énoncé dans la section 1.1.3 du chapitre 1, un modèle de chaîne de Markov à sauts est souvent utilisé pour suivre les manœuvres des objets. Le modèle (1.43) est ici reformulé :

$$x(k) = Fx(k-1) + Gu_{m_i}(k) + w(k-1). \tag{2.19}$$

La variation de la connaissance *a priori* dans le modèle (2.19) est introduite par l'intermédiaire d'une entrée déterministe u_{m_i} qui dépend du mode d'évolution m_i où $i \in \{1, 2, ..., M\}$ avec M représentant le nombre de modes et G représentant la matrice d'entrée. Les observations sont supposées linéairement liées au vecteur d'état dans ce chapitre :

$$z(k) = Hx(k) + v(k), \tag{2.20}$$

où H représente la matrice d'observation et $v(k)$ représente un bruit de mesure supposé Gaussien avec une matrice de covariance notée R. La solution mise en place afin d'assurer le suivi de plusieurs objets à la fois est résumée par l'algorithme 2.1.

La matrice $\Pi = [\pi_{ij}]$ dans l'algorithme 2.1 contient les probabilités de transitions entre les différents modes de l'*IMM*. Les paramètres T_s et T_c représentent respectivement, les seuils de suppression et de confirmation d'objets, ils sont définis dans la section 1.3.2 du chapitre 1. Les paramètres P_d et

Initialisation :	
Pour chaque objet	$t \in \{1, 2, ..., n\}$
Pour chaque mode d'évolution	$i \in \{1, 2, ..., M\}$
Probabilités de mixage :	$\mu_t^{i\|j}(k-1) = [\pi_{ij}\mu_t^i(k-1)] / \sum_{l=1}^{M} \mu_t^l(k-1)$
Mixage des estimations :	$\hat{x}_t^{0j}(k-1) = \sum_{i=1}^{M} \hat{x}_t^i(k-1)\mu_t^{i\|j}(k-1)$
Mixage des covariances :	$P_t^{0j}(k-1) = \sum_{i=1}^{M} \mu_t^{i\|j}(k-1)\{P_t^i(k-1)$ $+[\hat{x}_t^i(k-1) - \hat{x}_t^{0j}(k-1)].[\hat{x}_t^i(k-1) - \hat{x}_t^{0j}(k-1)]'\}$
Prédiction :	
Pour chaque objet	$t \in \{1, 2, ..., n\}$
Pour chaque mode d'évolution	$i \in \{1, 2, ..., M\}$
Probabilités des modes :	$\overline{\mu}_t^i(k\|k-1) = \sum_{l=1}^{M} \pi_{li}\mu_t^l(k-1)$
Prédiction des estimations :	$\hat{x}_t^i(k\|k-1) = F\hat{x}_t^{0j}(k-1) + Gu$
Prédiction des covariances :	$P_t^i(k\|k-1) = FP_t^{0j}(k-1)F' + Q(k-1)$
Observation prédite :	$\overline{z}_t^i(k\|k-1) = H\hat{x}_t^i(k\|k-1)$
Covariance de l'observation :	$S_t^i(k\|k-1) = HP_t^i(k\|k-1)H' + R$
Quantités globales :	
Pour chaque objet	$t \in \{1, 2, ..., n\}$
Observations :	$\overline{z}_t(k\|k-1) = \sum_{i=1}^{M} \overline{\mu}^i(k\|k-1)\hat{z}_t^i(k\|k-1)$
Covariances d'observations :	$S_t(k\|k-1) = \sum_{i=1}^{M} \overline{\mu}^i(k\|k-1)S_t^i(k\|k-1)$
Phase d'association :	réception des observations $z_j(k),\ j \in \{1, 2, ..., m\}$
Pour chaque objet	$t \in \{1, 2, ..., n\}$
Matrice des distances $[d_{t,j}]$	$d_{t,j}(k) = (z_j(k) - \overline{z}_t(k\|k-1))' S_t(k\|k-1)^{-1}(z_j(k) - \overline{z}_t(k\|k-1))$
Résolution	algorithme *GNN*
Apparition et disparition :	
Pour chaque objet	$t \in \{1, 2, ..., n\}$
Fonctions score :	$L_t(k) = L_t(k-1) + \Delta L_t(k)$
	$L_t(k) = \begin{cases} ln[1 - P_D] & \text{si } t \text{ n'est pas détecté} \\ ln[\frac{P_D}{(2\pi)^{q/2}\beta_{FT}\sqrt{\|S_t\|}} - \frac{d_t^2}{2}] & \text{si } t \text{ est détecté} \end{cases}$
Décision :	$\begin{cases} L_t(k) > T_c & \text{confirmation de l'objet } t \\ L_t(k) < T_s & \text{suppression de l'objet } t \\ T_s < L_t(k) < T_c & \text{doute concernant l'objet } t \end{cases}$
Phase de mise à jour :	
Pour chaque objet	$t \in \{1, 2, ..., n\}$
Pour chaque mode d'évolution	$i \in \{1, 2, ..., M\}$
Gain de Kalman :	$K_t^i(k) = P_t^i(k\|k-1)H'(S_t(k\|k-1))^{-1}$
Estimations :	$\hat{x}_t(k) = \hat{x}_t^i(k\|k-1) + K_t^i(k)(z_j(k) - \overline{z}_t(k\|k-1))$
Covariances :	$P_t^i(k) = P_t^i(k\|k-1) - K_t^i(k)S_t(k\|k-1)K_t^i(k)'$
Vraisemblances des modes :	$\Lambda_t^i(k) = exp[-(1/2)d_{t,j}(k)] / \|2\pi S_t^i(k\|k-1)\|^{1/2}$
Probabilités des modes :	$\mu_t^i(k) = [\overline{\mu}_t^i(k\|k-1)\Lambda_t^i(k)] / \sum_{l=1}^{M} \overline{\mu}_l^i(k\|k-1)\Lambda_l^i(k)$

Tableau 2.1: Solution proposée pour le suivi de plusieurs objets à la fois.

β_{FT} représentent respectivement, la probabilité de détection et la densité de fausses alarmes. Ce sont des paramètres d'observation supposés connus.

Un exemple de simulation visant à tester l'algorithme du suivi multi-objets proposé ci-dessus est donné par la figure 2.4. On peut y voir l'évolution dans le temps de quatre objets différents. Il peut être remarqué que les objets peuvent apparaître, disparaître ou réapparaître d'une manière aléatoire. La figure 2.4 illustre l'évolution des fonctions score. Il peut être remarqué qu'une fonction score est croissante quand l'objet correspondant est détecté, elle est décroissante si l'objet en question n'est pas détecté. À l'aide des deux seuils T_c et T_s, les fonctions score gèrent la confirmation et la suppression d'objets. En effet, un objet non détecté n'est pas tout de suite supprimé, l'algorithme de suivi permet de prédire sa trajectoire avant que la fonction score concernée atteigne le seuil T_s. Les seuils T_c et T_s sont fixés selon la méthode décrite dans la section 1.3.2 du chapitre 1.

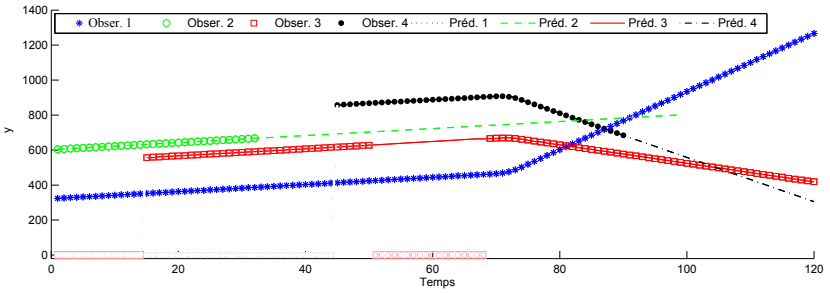


FIGURE 2.4: Exemple de quatre objets manœuvrants.

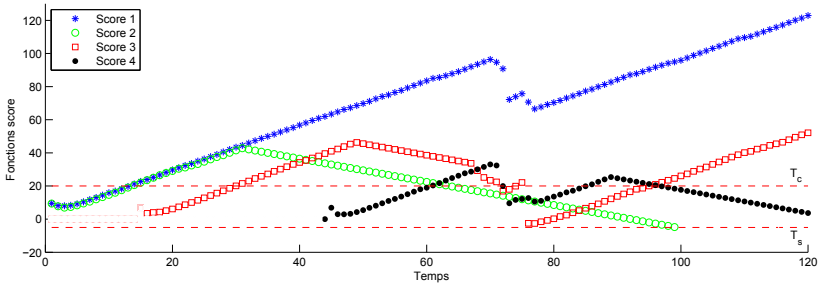


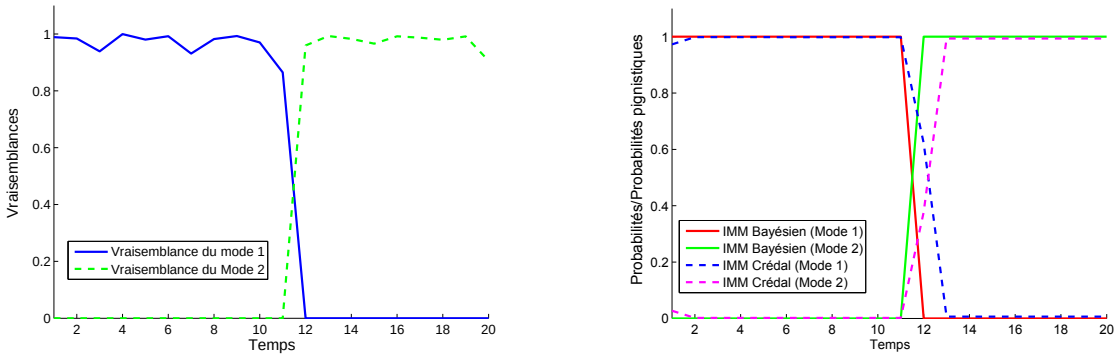
FIGURE 2.5: Fonctions score des objets de l'exemple de la figure 2.4.

L'objectif principal de l'utilisation des fonctions score dans la solution proposée est d'apporter une certaine robustesse par rapport aux fausses alarmes. Cet aspect est bien géré dans les méthodes *MHT* tel que les probabilités propagées aux cours du temps permettent de distinguer les faux des vrais objets : les faux objets (fausses alarmes) ont des probabilités assez faibles et leur suivi

est abandonné au bout d'un moment. Les méthodes *JPDA* ne présentent, généralement, pas de solution explicite permettant de gérer les fausses alarmes [23].

Il peut être remarqué que la solution de suivi présentée par l'algorithme 2.1 ne fait aucunement intervenir des fonctions de croyance. En effet, elle utilise des outils connus dans le cadre Bayésien, en l'occurrence, le filtrage de Kalman et l'algorithme *IMM* dont les performances sont considérablement appréciables dans le domaine de suivi d'objets, ce pour la complexité raisonnable et la qualité d'estimation qu'ils peuvent fournir. Un filtrage de Kalman Généralisé (*Generalized Kalman Filter GKF*), basé sur les fonctions de croyance, est introduit par Smets dans [65, 175]. La performance du *GKF* est optimale pour les systèmes linéaires et Gaussiens, elle est équivalente à la performance du filtre de Kalman classique en ce qui concerne la qualité d'estimation. Toutefois, la comparaison peut être complétée par une étude de complexité où le *GKF* utilisant des fonctions de croyance pourrait être plus complexe que le filtre de Kalman classique. En ce qui concerne le suivi d'objets manœuvrants qui nécessite une estimation adaptative, des méthodes multi-modales basées sur les fonctions de croyance ont été introduites, notamment dans [85] où les fonctions de croyance sont utilisées dans la phase de mise à jour des estimations et dans [135, 136] où la distribution de probabilités sur les modes d'évolution est remplacée par une fonction de masse qui est mise à jour d'une manière recursive. À chaque pas de temps, la fonction de masse est transformée en probabilités pignistiques afin d'assurer la phase de mixage par exemple. Pour un système à deux modes d'évolution dont les vraisemblances fournies par les filtres de Kalman sont données par la figure 2.6(a), les probabilités et les probabilités pignistiques des deux modes, données respectivement par l'*IMM* classique et l'*IMM* crédal, sont présentées par la figure 2.6(b). Par ailleurs, un aperçu concernant l'utilisation des fonctions de croyances et la théorie de Dezert-Smarandache ($DSmT$) [166, 167] pour le suivi d'objets est donné dans [26, 179].

Pour l'exemple simplifié de la figure 2.6, il peut être remarqué que l'*IMM* classique et l'*IMM* crédal obtiennent les mêmes probabilités de modes. Toutefois, un léger retard à l'adaptation peut être constaté pour l'*IMM* crédal au changement de mode d'évolution à l'instant 11. Les détails concernant l'*IMM* crédal peuvent être trouvés dans [134, chapitre 4, pages : 101-106]. Contrairement à l'*IMM* classique, l'*IMM* crédal n'est pas contraint de définir une distribution de probabilités initiale. Toutefois, il n'améliore pas pour autant les performances de l'*IMM* classique. L'intérêt quant à l'utilisation des fonctions de croyance dans les méthodes d'estimation reste à explorer. Ce chapitre est essentiellement dédié à l'aspect classification d'objets où le formalisme crédal est montré pouvoir surpasser le formalisme Bayésien dans certaines circonstances.



(a) Vraisemblance des modes d'évolution.

(b) Probabilités des modes d'évolution.

FIGURE 2.6: *IMM* classique vs *IMM* crédal [134].

2.3.2 Classification à base de données cinématiques

Cette section présente deux solutions de classification d'objets qui se basent sur les données cinématiques. La première solution utilise le théorème de Bayes BT classique est la seconde est basée sur les fonctions de croyance, elle utilise le théorème de Bayes généralisé GBT. La classification basée sur les données cinématiques consiste à définir un ensemble de comportements qu'on notera $B = \{b_1, b_2, ..., b_{nb}\}$ où nb représente le nombre de comportements. Une fois les comportements possibles des objets suivis connus, il ne reste qu'à déduire la classe de chaque objet, l'ensemble des classes étant noté $C = \{c_1, c_2, ..., c_{nc}\}$, où nc représente le nombre de classes.

2.3.2.1 Classifieur Bayésien

Le classifieur Bayésien est exécuté en deux étapes principales à chaque pas de temps, l'algorithme est illustré par la figure 2.7. La première étape consiste à calculer les probabilités *a posteriori* des comportements $b_i \in B$, celles-ci sont ensuite projetées sur l'espace des classes C en se basant sur des connaissances *a priori*.

Vraisemblances des comportements : les vraisemblances des comportements sont calculées à l'aide des vraisemblances des modes d'évolution issues des *IMMs*. Les comportements ne sont autre que des groupements (*clusters*) de modes d'évolution (e.g. un comportement de manœuvre complexe est le groupement de modes d'évolution permettant aux objets d'effectuer des mouvements complexes). On suppose que chaque comportement b_i regroupe M_{b_i} modes d'évolution tel que $M_{b_i} < M$ où M représente le nombre total de modes. Les vraisemblances des comportements

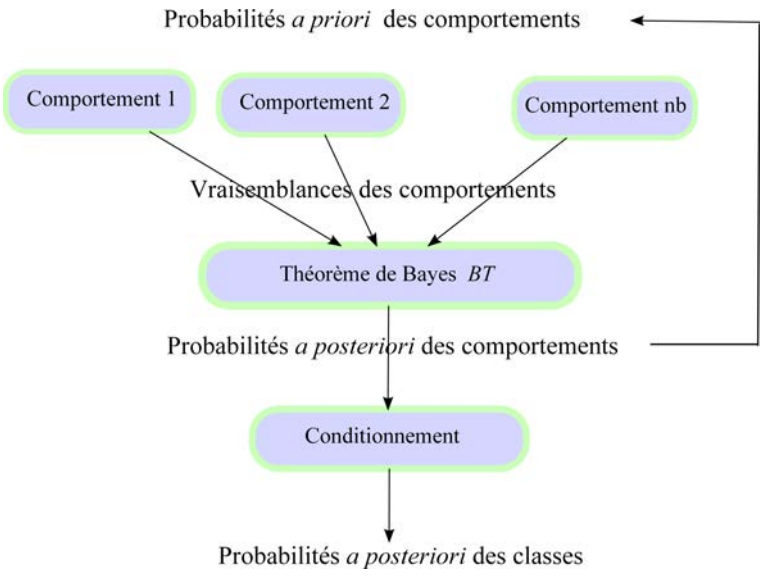


FIGURE 2.7: Organigramme du classifieur Bayésien.

$l_{b_i} = P(z(k)|b_i)$ sont alors déduites de la manière suivante :

$$l_{b_i}(k) = \sum_{j=1}^{M_{b_i}} \nu^j(k)\Lambda^j(k), \quad i \in \{1, 2, ..., , nb\}, \tag{2.21}$$

avec

$$\nu^j(k) = \frac{\mu^j(k)}{\sum_{l=1}^{M_{b_i}} \mu^l(k)}, \tag{2.22}$$

où $\Lambda^j(k) = P(z(k)|r(k) = m_j)$ et $\mu^j(k)$ représentent les vraisemblances et les probabilités des modes $m_j \in \{m_1, m_2, ..., m_M\}$ calculées dans la phase de mise à jour de l'algorithme de suivi présenté dans 2.1.

Probabilités *a posteriori* des comportements : les vraisemblances l_{b_i} combinées avec des probabilités *a priori* $P(b_i|z(1:k-1))$ des comportements, sont utilisées pour le calcul récursif des probabilités *a posteriori* $P(b_i|z(1:k))$ en utilisant la règle de Bayes :

$$P(b_i|z(1:k)) = \frac{l_{b_i} P(b_i|z(1:k-1))}{\sum_{j=1}^{nb} l_{b_j} P(b_j|z(1:k-1))} \ . \tag{2.23}$$

Une fois les probabilités *a posteriori* des comportements calculées, elles sont projetées sur l'espace des classes C. La projection est basée sur des connaissances *a priori* (e.g. un objet aérien ayant

un comportement de manœuvre complexe peut être de la classe des avions de chasse et non pas de la classe des avions de ligne par exemple). Les relations entre l'espace des comportements B et l'espace des classes C sont exprimées à l'aide de probabilités conditionnelles $P(c_i|b_j)$ contenues dans la matrice de transition T_1.

Probabilités *a posteriori* des classes : la distribution de probabilités *a posteriori* sur l'espace des classes notée $P(C)$ est calculée de la façon suivante :

$$P(C) = T_1 \times P(B), \tag{2.24}$$

où les probabilités conditionnelles contenues dans la matrice de transition T_1 dépendent du cadre d'application considéré. Plus d'informations seront données dans les deux exemples de simulations présentés dans ce chapitre.

2.3.2.2 Classifieur Crédal

Le classifieur crédal originalement introduit dans [175] est illustré par la figure 2.8.

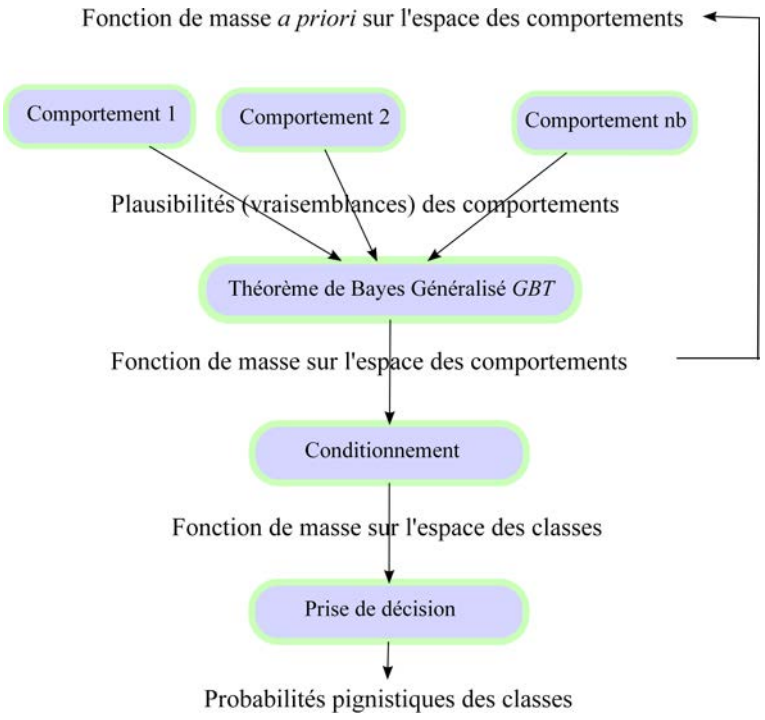


FIGURE 2.8: Organigramme du classifieur crédal.

Tout comme dans le cas Bayésien, le classifieur crédal est basé sur les vraisemblances $l_{b_j} = P(z(k)|b_j)$ de comportements b_j, qui sont vues comme étant des mesures de plausibilités conditionnelles $pl[b_j](z(k))$ dans le cadre crédal [3, 171, 175] :

$$pl[b_j](z(k)) = \sum_{j=1}^{M_{b_i}} \nu^j(k)\Lambda^j(k), \quad i \in \{1, 2, ..., , nb\}, \tag{2.25}$$

où les paramètres ν^j, $\forall j \in \{1, 2, ..., M_{b_i}\}$ sont calculés par l'équation (2.22).

Une fois les plausibilités des comportements sont calculées, une fonction de masse sur l'espace des comportements notée m^B est construite.

Fonction de masse sur l'espace des comportements : le calcul de la fonction de masse *a posteriori* des comportements $m^B[z(k)](b_i)$ à l'instant k est effectué en deux étapes principales. La première étape consiste à transformer les fonctions de plausibilités $pl[b_j](z(k))$ en une fonction de masse notée $\overline{m}^B[z(k)](b_i)$ à l'aide du GBT :

$$\overline{m}^B[z(k)](A) = \prod_{b_i \in A} pl[b_i](z(k))(1 - \prod_{b_i \in \overline{A}} (1 - pl[b_i](z(k))). \tag{2.26}$$

La seconde étape consiste à combiner conjonctivement la fonction de masse $\overline{m}^B(k)$ avec la fonction de masse *a posteriori* des comportements de l'instant précédent $m^B(k-1)$:

$$m^B(k) = \overline{m}^B(k) \textcircled{\cap} m^B(k-1). \tag{2.27}$$

À noter que le pas de temps k n'est pas un élément du cadre de discernement, il est utilisé dans l'équation (2.27) simplement pour exprimer la récursivité. La fonction de masse initiale $m^B(0)$ peut être considérée vide si aucune information *a priori* n'est disponible. Les fonctions de masse dans l'équation (2.27) sont combinées à l'aide de la règle de combinaison conjonctive (2.4), l'objectif étant la propagation de l'information à travers le temps.

Il peut être remarqué que contrairement au classifieur Bayésien où la croyance est exclusivement répartie sur les singletons $b_i \in B$, la croyance dans le cadre crédal est répartie sur des singletons $b_i \in A$ et des ensembles de singletons $A \subseteq B$. Cela représente un avantage considérable dans le sens où la croyance peut être transférée sur l'espace des classes C d'une manière plus précise que dans le cas Bayésien.

Fonction de masse sur l'espace des classes : les relations entre comportements et classes sont traduites par des fonctions de masse conditionnelles $m[D](A)$ tel que $A \subseteq C$ et $D \subseteq B$. Les fonctions de masse $m[D](A)$ sont rassemblées dans la matrice de transition notée T_2.

$$m^C = T_2 \times m^B. \tag{2.28}$$

La transition entre l'espace des comportements B et l'espace des classes C sera détaillée dans les exemples de simulations donnés dans les sections 2.4 et 2.5.

Prise de décision : afin de se prononcer sur les classes des objets suivis, les fonctions de masse *a posteriori* des classes $m^C(k)$ sont simplement transformées en probabilités pignistiques en utilisant l'équation (2.18).

Les deux classifieurs Bayésien et crédal sont testés sur un premier exemple de classification de cibles aériennes dans la section 2.4 où les classes sont supposées constantes dans le temps. Le classifieur crédal est ensuite confronté à une situation de classification plus complexe où les classes sont changeantes dans le temps. Cela est présenté dans la section 2.5 sur un exemple de classification de piétons. À noter que les classifieurs sont exécutés pour chaque objet $t \in \{1, 2, ..., n\}$. Par souci de lisibilité l'indice des objets t n'est pas utilisé dans la description des classifieurs.

2.4 Application 1 : suivi et classification de cibles aériennes (classes constantes)

2.4.1 Description

Cette section présente une comparaison des deux classifieurs crédal et Bayésien sur un exemple de suivi et de classification de plusieurs objets aériens. L'évolution des objets est supposée obéir au modèle donné par l'équation (2.19). Le vecteur d'état $x = [x\ \dot{x}\ y\ \dot{y}]$ représente la position et la vitesse des objets sur le plan (x, y). La matrice d'état F et la matrice d'entrée G représentent un modèle d'évolution en vitesse constante, elles sont définies ainsi :

$$F = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} (\Delta T)^2/2 & 0 \\ \Delta T & 0 \\ 0 & (\Delta T)^2/2 \\ 0 & \Delta T \end{bmatrix},$$

où ΔT est la période d'échantillonnage. Le modèle d'observation est donné par :

$$z(k) = Hx(k) + v(k), \tag{2.29}$$

où la matrice d'observation est :

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Le vecteur $u = [a_x \ a_y]'$, avec a_x : l'accélération selon l'axe x et a_y : l'accélération selon l'axe y, dans l'équation (2.19) représente un mode d'accélération $m_i \in \{m_1, m_2, ..., m_M\}$. L'exemple considéré dans cette section est initialement présenté dans [155] pour le suivi d'un seul objet. Ici le même exemple est repris dans un contexte de suivi multi-objets. Le suivi de chaque objet est assuré par un *IMM* constitué de 13 modes d'évolution $m_i \in \{m_1, m_2, ..., m_{13}\}$. Les modes d'évolution représentent différentes manœuvres plus ou moins complexes. La répartition des modes d'évolution sur l'ensemble des classes $C = \{$avion de ligne, bombardier, avion de chasse$\}$ considérées est donnée par la relation suivante :

$$-L_i \leq \{a_x, a_y\} \leq L_i \ , \tag{2.30}$$

avec $L_i = 0g, 2g$ et $4g$ qui sont respectivement les limites d'accélération pour les classes $c_1 = \{$avion de ligne$\}$, $c_2 = \{$bombardier$\}$ et $c_3 = \{$avion de chasse$\}$, où $g = 9.81 \ m/s^2$ représente l'accélération terrestre. La répartition des modèles sur les trois classes est illustrée par la figure 2.9.

Les connexions entre les modes dans la figure 2.9 représentent les éléments non nuls de la matrice de transition Π de l'*IMM*. Les éléments sur la diagonale de la matrice Π sont égaux à 0.9, le restant du poids est réparti sur les éléments non nuls de chaque ligne de sorte à ce que $\sum_i \pi_{i,j} = 1$.

En se basant sur des informations *a priori*, les différents modes d'évolution m_i peuvent être regroupés de manière à définir les comportements attendus des objets suivis. Les comportements de l'exemple considéré sont définis ainsi :

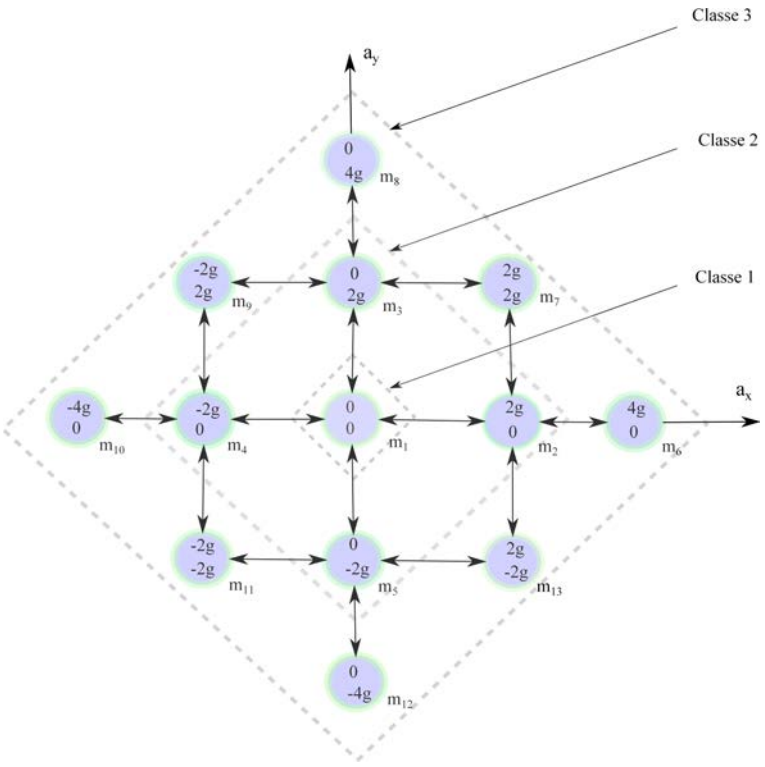


FIGURE 2.9: Les différents modes d'évolution et définition des classes pour l'application 1.

- Comportement 1 (b_1) : correspond au comportement d'évolution avec une vitesse constante uniquement (e.g. avions de ligne). L'ensemble des modes d'évolution appartenant à ce comportement est réduit à un seul élément : $\{m_1\}$, ce qui correspond au mode d'accélération : $u = [0\ 0]'$.

- Comportement 2 (b_2) : ce comportement concerne les objets pouvant évoluer en vitesse constante et effectuer des manœuvres moyennes (e.g. bombardiers). Ce comportement regroupe les 5 premiers modes $\{m_1, ..., m_5\}$ représentés par la figure 2.9, ce sont les modes dont l'accélération est limitée à $2g$.

- Comportement 3 (b_3) : c'est un comportement associé au objets pouvant effectuer tous les mouvements : évoluer en vitesse constante, effectuer des manœuvres de moyenne et de forte complexités. Ce comportement regroupe tous les modes d'évolution $\{m_1, ..., m_{13}\}$ de la figure 2.9.

Comme indiqué par les classifieurs présentés dans 2.3.2.1 et 2.3.2.2, une fois la connaissance est exprimée sur l'espace des comportements B, elle est projetée sur l'espace des classes C qui sont définies par :

- Classe 1 (c_1) : classe des avions de ligne.
- Classe 2 (c_2) : classe des bombardiers.
- Classe 3 (c_3) : classe des avions de chasse.

Le transfert de la connaissance entre l'espace des comportements B et l'espace des classes C est effectué à l'aide des équations (2.24) et (2.28) pour les classifieurs Bayésien et crédal respectivement. Les relations liant l'ensemble B et l'ensemble C sont les suivantes :

- Relation 1 : un objet ayant un comportement b_1 peut être un avion de ligne, un bombardier ou un avion de chasse. Tous peuvent évoluer en vitesse constante. Cette relation peut être traduite par : $b_1 = \{c_1, c_2, c_3\}$.
- Relation 2 : un objet ayant un comportement b_2, pouvant effectuer des manœuvres moyennes, peut être un bombardier ou un avion de chasse. Les avions de ligne sont supposés ne pas pouvoir effectuer de manœuvres. Cette relation peut être réécrite comme suit : $b_2 = \{c_2, c_3\}$.
- Relation 3 : un objet ayant un comportement b_3 ne peut être qu'un avion de chasse. Les avions de ligne et les bombardiers ne peuvent pas effectuer des manœuvres complexes. Cette relation peut être réécrite ainsi : $b_3 = \{c_3\}$.

2.4.2 Conditionnement de la connaissance

Pour le classifieur Bayésien, la matrice de transition figurant dans l'équation (2.24) est donnée par :

$$T_1 = \begin{bmatrix} 1/3 & 0 & 0 \\ 1/3 & 1/2 & 0 \\ 1/3 & 1/2 & 1 \end{bmatrix}. \tag{2.31}$$

La matrice T_1 traduit le conditionnement :

- Si $P(b_1) = 1 \Longrightarrow P(c_1|b_1) = \frac{1}{3},\ P(c_2|b_1) = \frac{1}{3},\ P(c_3|b_1) = \frac{1}{3}$.
- Si $P(b_2) = 1 \Longrightarrow P(c_1|b_2) = 0,\ P(c_2|b_2) = \frac{1}{2},\ P(c_3|b_2) = \frac{1}{2}$.
- Si $P(b_3) = 1 \Longrightarrow P(c_1|b_3) = 0,\ P(c_2|b_3) = 0,\ P(c_3|b_3) = 1$.

Le transfert d'information entre les espaces B et C est effectué de manière plus précise dans le cadre crédal. Le conditionnement de la connaissance est le suivant :

- $m[b_1](\{c_1, c_2, c_3\}) = 1$ (cf. Relation 1),
- $m[b_2](\{c_2, c_3\}) = 1$ (cf. Relation 2),
- $m[\{b_1, b_2\}](\{c_1, c_2, c_3\}) = 1$ (cf. Relations. 1 et 2),
- $m[b_3](c_3) = 1$ (cf. Relation 3),
- $m[\{b_1, b_3\}](\{c_1, c_2, c_3\}) = 1$ (cf. Relations 1 et 3),
- $m[\{b_2, b_3\}](\{c_2, c_3\}) = 1$ (cf. Relations 2 et 3),
- $m[B](\{c_1, c_2, c_3\}) = 1$ (cf. Relations 1, 2 et 3).

La matrice de transfert T_2 correspondante est de dimension $(2^3 = 8) \times (2^3 = 8)$, elle est donnée par :

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Le conditionnement ici présenté permet d'avoir une fonction de masse sur l'ensemble des classes C en utilisant l'équation (2.28).

2.4.3 Résultats de simulations

La figure 2.10 présente les observations et les estimations de quatre objets aériens sur le plan (x, y). Les objets apparaissent, disparaissent et effectuent des manœuvres aléatoires au cours de la simulation, ils sont suivis à l'aide de l'algorithme 2.1. À l'aide des deux classifieurs Bayésien et crédal, on souhaite également reconnaître les classes des objets parmi l'ensemble C.

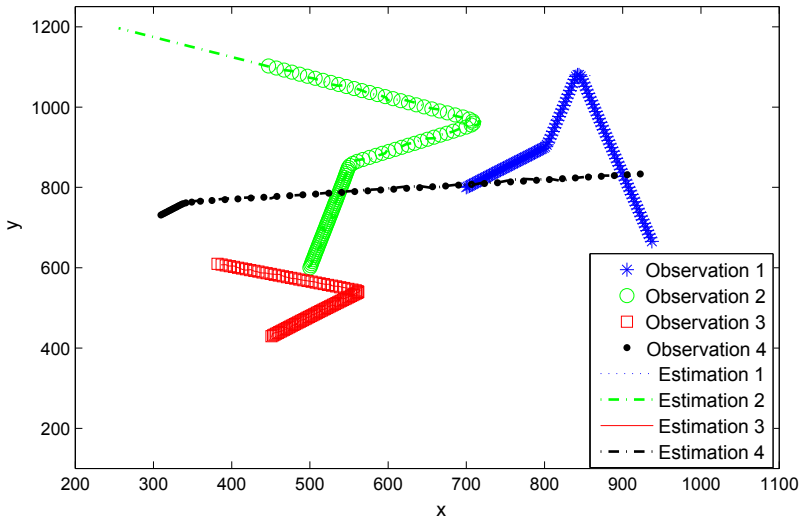


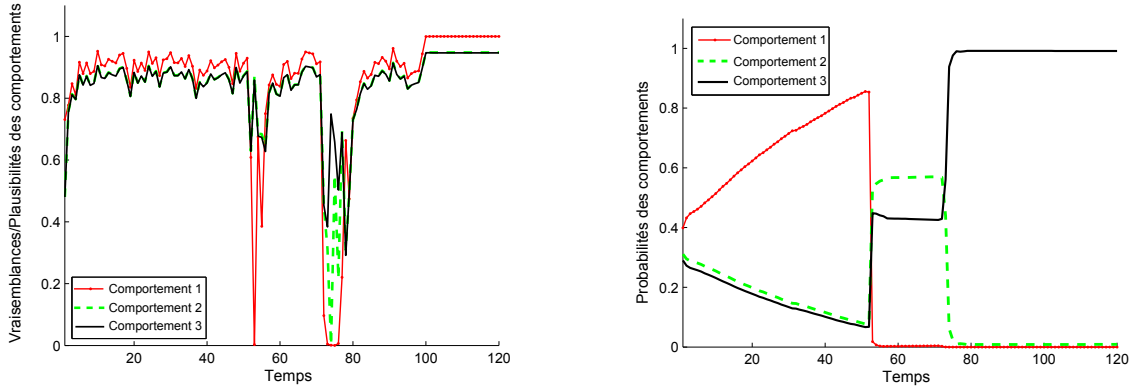
FIGURE 2.10: Exemple de simulation : quatre objets manœuvrants.

Dans ce qui suit, nous nous intéressons aux résultats de classification de l'objet 2. L'objet en question évolue principalement en vitesse constante, il effectue une manœuvre de complexité moyenne durant la période de temps $50 - 54$, et une manœuvre de forte complexité durant la période de temps $70 - 75$, et il disparaît au pas de temps 100.

Selon l'évolution observée pour l'objet 2, il est attendu des deux classifieurs d'être en doute parfait entre les trois classes durant la première phase du mouvement (avant la manœuvre moyenne) car tous les types d'avions peuvent évoluer en vitesse constante. Entre la première manœuvre à moyenne complexité et la seconde manœuvre à forte complexité, un doute parfait entre la classe 2 et la classe 3 est attendu. Après la manœuvre à forte complexité on attend à ce que l'objet 2 soit identifié comme étant un avion de chasse.

Les vraisemblances/plausibilités des comportements adoptés par l'objet 2 sont données par la figure 2.11(a). Dans [155] où le classifieur Bayésien a été étudié, la transition entre l'espace des comportements B et l'espace des classes C n'est pas considérée. Les comportements sont considérés directement comme étant les classes des objets $C = B$. Dès lors, le résultat de classification obtenu est donné par la figure 2.11(b).

Il peut être remarqué que l'objet 2 a bien été identifié comme étant un avion de chasse. Toutefois, le doute durant la première et la seconde phases du mouvement n'est pas bien géré. En effet, dans la première phase du mouvement, par exemple, où un doute parfait entre les trois classes est



(a) Vraisemblances des comportements.

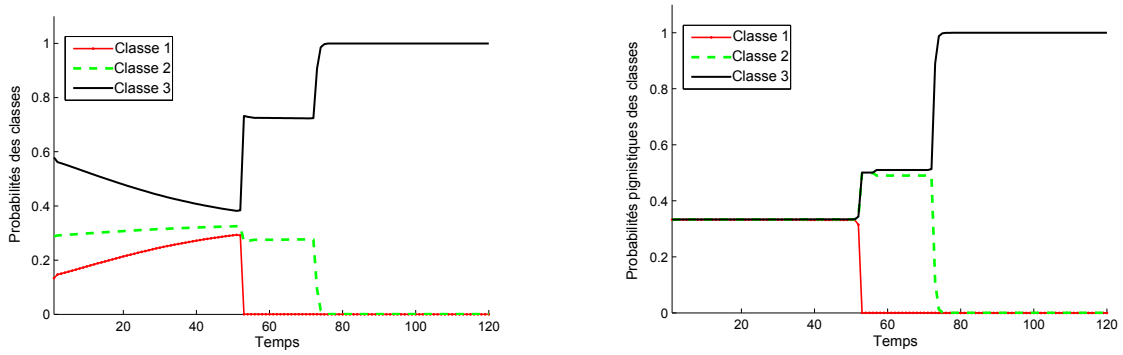
(b) Probabilités des comportements.

FIGURE 2.11: Classification Bayésienne basée sur les comportements de cibles aériennes (application 1).

attendu, le classifieur Bayésien a tendance à classer précipitamment l'objet comme étant un avion de ligne alors que c'est un avion de chasse, idem pour la seconde phase du mouvement où la classe 2 est favorisée. La cause principale de ce comportement est l'imprécision des vraisemblances : durant la première phase du mouvement par exemple, il peut être remarqué que la vraisemblance du comportment b_1 est légèrement supérieure aux vraisemblances des comportements b_2 et b_3, cela est dû au fait que les comportements sont imbriqués ($b_1 \subset b_2 \subset b_3$), et le fait que le comportement b_1 contient le nombre minimum de modes d'évolution, seulement le mode m_1 qui est justement le mode d'évolution de l'objet 2 durant la première phase du mouvement. De ce fait, le calcul de la vraisemblance de b_1 par l'équation (2.21) est moins influencé par d'autres modes d'évolution, contrairement aux comportements b_2 et b_3 qui sont respectivement composés de 5 et de 13 modes d'évolution et dont le calcul de la vraisemblance est donc moins pur pour une évolution en vitesse constante par exemple.

La projection des probabilités des comportements obtenues dans la figure 2.11(b) sur l'espace des classes C selon l'équation (2.24) donne le résultat présenté par la figure 2.12(a). Cela donne le résultat de la classification Bayésienne avec transfert d'information selon l'équation (2.24). Il peut être remarqué que même si l'objet 2 a fini par être classé comme étant un avion de chasse, les situations de doute restent irrésolues par le classifieur Bayésien. D'autre part, il peut être remarqué sur la figure 2.12(b) que le doute est mieux géré par le classifieur crédal, cela grâce à un transfert d'informations plus précis effectué en utilisant l'équation (2.28).

Les résultats présentés par la figure 2.12 représentent une moyenne sur 20 simulations, ils montrent que le classifieur crédal assure de meilleures performances dans les situations imprécises comme le



(a) Probabilités des classes.

(b) Probabilités pignistiques des classes.

FIGURE 2.12: Classification Bayésienne vs classification crédale (application 1).

cas de classes imbriquées ici considéré. À noter que l'utilisation de probabilités imprécises permet également de gérer ce genre de situations comme cela est démontré dans [19] pour la classification d'un seul objet au prix d'une complexité supplémentaire.

2.5 Application 2 : suivi et classification de piétons (classes pouvant changer dans le temps))

Ayant montré des propriétés intéressantes dans le cadre de classification où les classes sont constantes dans le temps, le classifieur crédal est, dans cette section, testé sur un cas où les classes sont changeantes dans le temps.

2.5.1 Description

Cette section propose de tester l'algorithme de classification crédal sur un exemple visant à reconnaître les comportements des piétons. Les piétons sont supposés évoluer selon des modèles de vitesse constante où les paramètres sont différents d'un modèle à un autre :

$$x(k) = A(s_x, s_y)x(k-1) + w(k-1) \ , \tag{2.32}$$

avec :

$$A(s_x, s_y) = \begin{bmatrix} 1 & s_x\Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & s_y\Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

où s_x et s_y représentent la vitesse en m/s selon l'axe x et l'axe y, respectivement. En fonction des paramètres s_x et s_y, un ensemble $\{m_1, m_2, ..., m_7\}$ de 7 modes d'évolution différents est défini. Les 7 modes sont exécutés par les *IMMs* chargés de suivre les différents objets. La répartition des modes sur les différents comportements considérés est illustrée par la figure 2.13 :

- Comportement 1 (b_1) correspondant au mode statique. Le comportement b_1 est constitué d'un seul mode d'évolution $\{m_1\}$, où m_1 correspond à la matrice d'état $A(s_x, s_y)$ avec $(s_x, s_y) = (0, 0)$.
- Comportement 2 (b_2) correspondant au comportement de marche. Le comportement b_2 est représenté par trois modes d'évolutions : $\{m_2, m_3, m_4\}$ correspondant à des modèles d'état avec les paramètres $(s_x, s_y) = \{(2, 0), (0, 2), (2, 2)\}$ respectivement.
- Comportement 3 (b_3) correspondant au comportement de course. Les modes constituant le comportement b_3 sont les suivants : $\{m_5, m_6, m_7\}$, ils correspondent aux paramètres $(s_x, s_y) = \{(6, 2), (2, 6), (6, 6)\}$ respectivement.

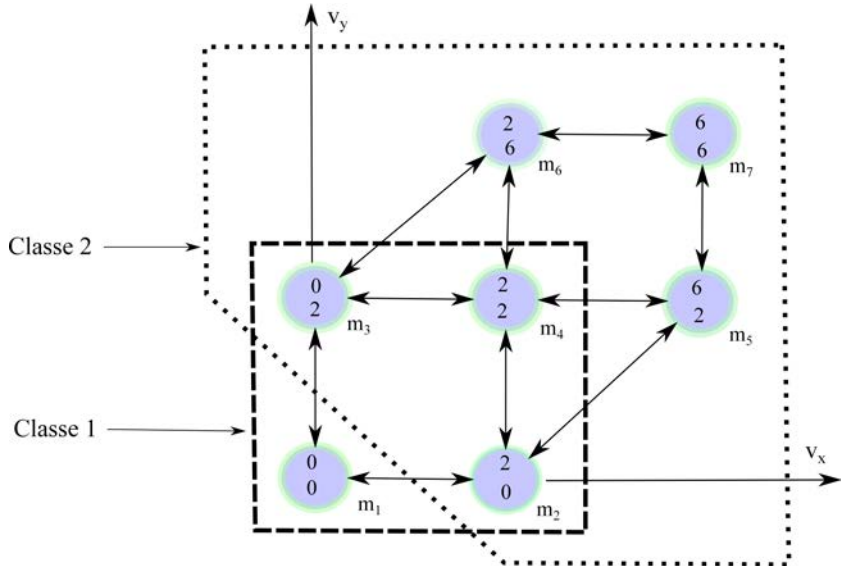


FIGURE 2.13: Les différents modes d'évolution et définition des classes pour l'application 2.

Contrairement au cas des cibles aériennes où les classes sont complètement imbriquées, dans cet exemple les classes se chevauchent mais ne sont pas imbriquées comme le montre la figure 2.13.

On se basant sur les comportements définis ci-haut, on souhaiterait reconnaître parmi les piétons, les promeneurs et les sportifs. L'ensemble des classes $C = \{c_1, c_2\}$ est donc donné par $C = \{\text{promeneur, sportif}\}$. Les relations entre les comportements et les classes sont les suivantes :

- Relation 1 : un promeneur peut marcher ou rester statique ($c_1 = \{b_1, b_2\}$).
- Relation 2 : un sportif peut marcher ou courir ($c_2 = \{b_2, b_3\}$).

Ces relations confirment le fait que les deux classes se chevauchent. Le sportif et le promeneur peuvent tous les deux adopter un comportement de marche, mais il est supposé qu'un sportif ne s'arrête pas et qu'un promeneur ne court pas.

2.5.2 Conditionnement de l'information

Le transfert de l'information entre l'espace des comportements B et l'espace des classes C dans cet exemple obéit aux relations 1 et 2. Les conditionnements permettant ce transfert sont donnés par :

- $m[b_1](c_1) = 1$,
- $m[b_2](\{c_1, c_2\}) = 1$,
- $m[\{b_1, b_2\}](\{c_1, c_2\}) = 1$,
- $m[b_3](c_2) = 1$,
- $m[\{b_1, b_3\}](\{c_1, c_2\}) = 1$,
- $m[\{b_2, b_3\}](\{c_1, c_2\}) = 1$,
- $m[\{b_1, b_2, b_3\}](\{c_1, c_2\}) = 1$.

Les conditionnements sont regroupés dans la matrice T_2 qui figure dans l'équation (2.28) du classifieur crédal :

$$T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{2.33}$$

Ce conditionnement permet d'obtenir une fonction de masse sur l'espace des classes C qui peut à son tour être transformée en probabilités pignistiques pour la prise de décision.

2.5.3 Simulation et résultats

Un exemple de suivi et de classification de deux piétons est donné dans cette section. Les deux piétons se déplacent sur l'espace (x, y). La figure 2.14 montre l'évolution dans le temps des deux piétons selon la direction y, leur évolution selon la direction x est identique.

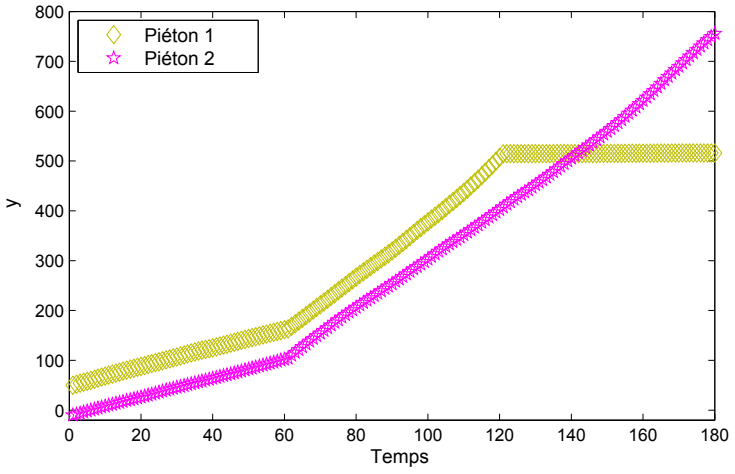
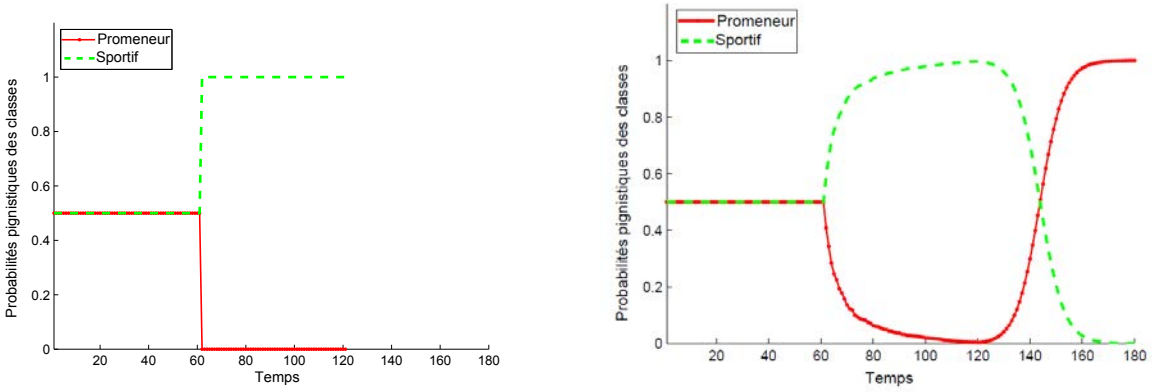


FIGURE 2.14: Trajectoires de deux piétons sur le plan (x, y).

Dans ce qui suit, on s'intéresse aux résultats de classification du piéton 1 qui commence son évolution avec un comportement de marche jusqu'au pas de temps 60 ensuite il effectue une course jusqu'au pas de temps 120 où il s'arrête. Selon ce scenario, illustré par la figure 2.14, un doute parfait entre les deux classes est attendu durant la phase de marche. On attend ensuite, que le piéton 1 soit classé comme étant un sportif durant la phase de course et qu'il soit classé comme étant un promeneur à son arrêt, ce qui signifie que l'objet a changé de classe.



(a) Classification crédale sans affaiblissement.

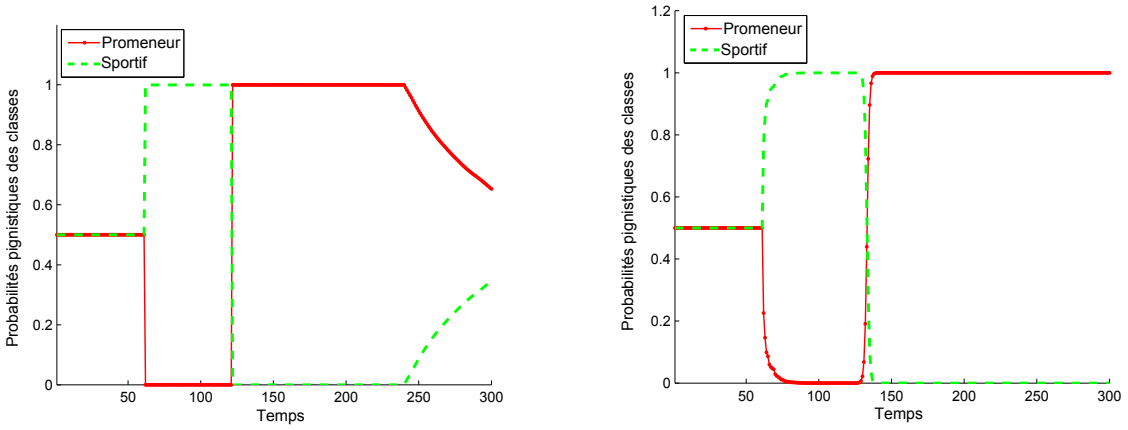
(b) Classification crédale avec affaiblissement.

FIGURE 2.15: Classification crédale sans/avec affaiblissement de connaissances (application 2).

La figure 2.15(a) montre le résultat de classification de l'algorithme crédal tel qu'il est présenté dans la section 2.3.2.2. Il peut être remarqué que les résultats de la classification durant la première et la deuxième phases du mouvement sont comme attendus, tel qu'un doute parfait est perçu durant la phase de marche et une classification en tant que sportif est atteinte durant la phase de course. Toutefois il peut être constaté qu'au changement de classe à l'instant 120, le classifieur crédal n'a pas pu s'adapter. En effet, les probabilités pignistiques des classes ne peuvent plus être calculées à cause d'un conflit total généré au changement de classe. Le conflit total est le résultat de la combinaison conjonctive de deux fonctions de masse antagonistes dans l'équation (2.27). En effet, à l'instant 120, la fonction de masse cumulée $m^B(k-1)$ a convergé vers la classe $c_2 = \{\text{sportif}\}$ et la fonction de masse instantanée $\overline{m}^B(k)$ croit que le piéton est un promeneur parce qu'il venait de s'arrêter au pas de temps $k = 120$.

Une solution simple qui consiste en l'affaiblissement de l'une des fonctions de masse antagonistes avant la combinaison (2.27) aide le classifieur crédal à s'adapter aux changements de classe. L'affaiblissement de la fonction de masse instantanée $\overline{m}^B$ avec un coefficient $\alpha = 0.3$ par exemple, permet au classifieur crédal de s'adapter au changement de classe, comme le montre la figure 2.15(b). D'autres coefficients d'affaiblissement peuvent également être utilisés, cela engendre une adaptation au changement de classe plus ou moins rapide. Toutefois, l'affaiblissement peut être introduit de manières différentes dans le classifieur crédal 2.3.2.2.

L'objectif de cette modification apportée au classifieur crédal est de réduire le conflit généré au changement de classe. L'affaiblissement peut être appliqué à la fonction de masse cumulée ou la fonction de masse instantanée avant la combinaison faite par l'équation (2.27). Les deux types



(a) Affaiblissement de l'information cumulée dans l'équation (2.27).

(b) Affaiblissement de l'information instantanée dans l'équation (2.27).

FIGURE 2.16: Modifications du classifieur crédal permettant l'adaptation au changement de classe (application 2).

d'affaiblissements peuvent conduire à des résultats différents. En effet, si par exemple le piéton 1 se remet à marcher après l'instant 180 (voir la figure 2.14), l'affaiblissement de la fonction de masse cumulée m^B (voir l'équation (2.27)) incite le classifieur à revenir en situation de doute entre les deux classes au fil du temps, comme le montre la figure 2.16(a). Cela peut être utile selon l'application considérée. D'un autre coté, l'affaiblissement de la fonction de masse instantanée $\overline{m}^B$ permet au classifieur de rester sur la décision prise auparavant, notamment, la classe $c_1 = \{\text{promeneur}\}$ dans le cas de l'exemple considéré dans cette section, comme le montre la figure 2.16(b).

Discussion : l'exemple présenté dans cette section montre une faculté intéressante pouvant être utilisée dans le cadre crédal, qui est celle de réviser ou modifier une connaissance afin de s'adapter à des situations de classification complexes, notamment la situation où les objets peuvent changer de classe. D'autres travaux ont introduit les fonctions de croyance dans des systèmes de suivi et de classification d'objets, notamment, dans [102] où les fonctions de croyance sont utilisées pour fusionner les données cinématiques des objets et les résultats de classification basés sur le son Doppler des objets où la classification est effectuée indépendamment du suivi. Dans certaines situations, il est montré dans [32], que le classifieur crédal ne fait pas mieux que le classifieur Bayésien et que ce dernier converge plus rapidement que le classifieur crédal. Dans [137], une solution de classification de véhicules militaires et civils est proposée, elle est basée sur la fusion Bayésienne des données cinématiques et la forme des objets, ou encore dans la référence [70] où les classifieurs Bayésien et

crédal sont appliqués à l'identification d'objets sous-marins. La situation de classification correspondant à la première application dans ce chapitre est étudiée dans [19] où le classifieur est basé sur des probabilités imprécises et l'intérêt est porté sur la classification d'un seul objet aérien. Dans [33], le classifieur crédal étudié dans ce chapitre est combiné avec un filtrage de Rao-Blackwilize qui assure le suivi d'un seul objet. D'autres méthodes de classification d'objets peuvent être trouvées dans [79, 104], elles sont basées sur l'inférence Bayésienne. Concernant l'aspect changement de classe, une solution crédale visant à reconnaître les comportements d'athlètes est proposée dans [148, 149], cette solution utilise la notion d'affaiblissement pour s'adapter au changements de classe.

2.6 Classification & fausses associations

Cette section reprend l'exemple de classification d'objets aériens présenté dans la section 2.4. Elle s'intéresse particulièrement à une situation de suivi conflictuel consistant à classifier des objets dont les trajectoires sont étroitement proches. L'objectif étant d'observer le comportement du classifieur crédal dans de telles situations.

L'évolution d'un avion de ligne escorté par un avion de chasse est considérée. Les trajectoires des deux avions sur le plan (x, y) sont illustrées par la figure 2.17 qui montre qu'à un certain moment $(k = 38)$ une fausse association entre les observations des deux objets est faite.

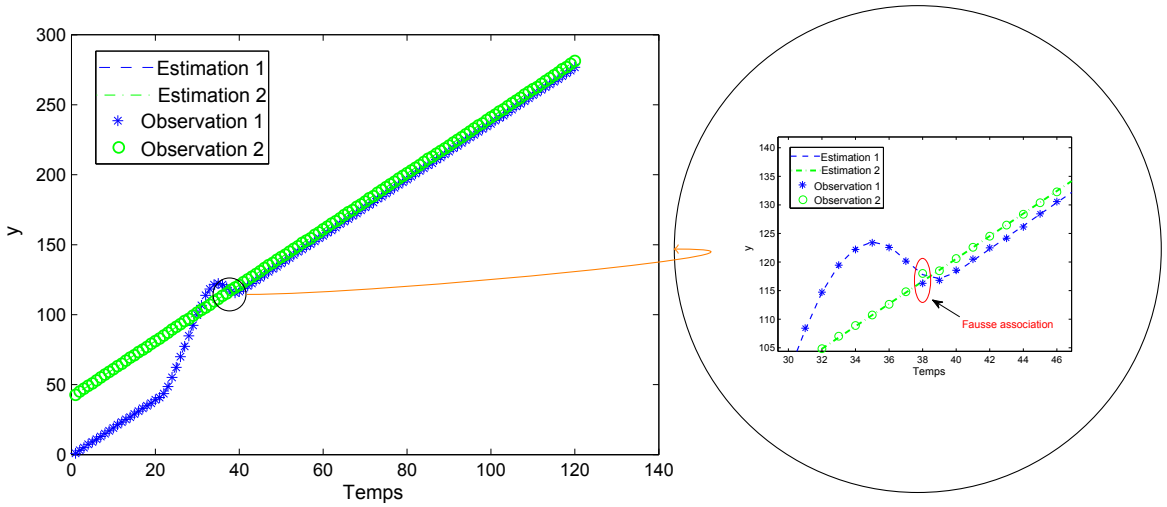
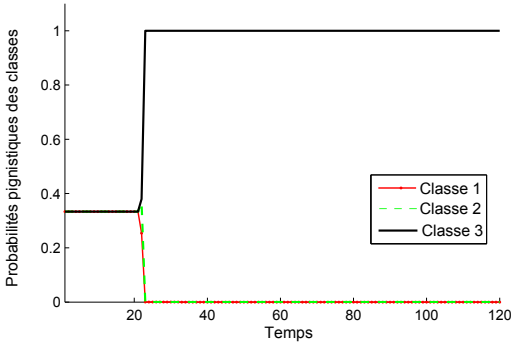
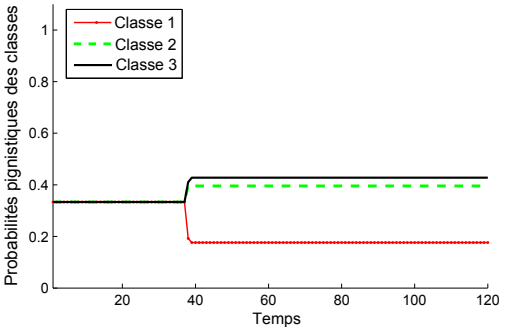


FIGURE 2.17: Fausse association entre les observations d'un avion de chasse et d'un avion de ligne.

La figure 2.18 montre que l'occurrence de telles fausses associations peut altérer la qualité de la classification, en l'occurrence la classification crédale dans cet exemple. En effet, il peut être remarqué sur la figure 2.18(a) que la classification concernant l'avion de chasse reste inchangée après l'occurrence de la fausse association, cela est lié au fait que les classes considérées dans la section 2.4 sont imbriquées et constantes dans le temps. D'autre part, il peut être remarqué sur la figure 2.18(b) que la classification concernant l'avion de ligne est modifiée par la fausse association. Cela est dû au fait que le classifieur (algorithme du suivi) a reçu une observation autre que celle attendue. En effet, d'après la figure 2.18(b) un quelconque organe de décision peut être induit en erreur en décidant que l'avion 2 est un avion de chasse ou bombardier alors que ce dernier n'a effectué aucune manœuvre, il est fort probable que ça soit un avion de ligne.



(a) Probabilités pignistiques (avion de chasse).



(b) Probabilités pignistiques (avion de ligne).

FIGURE 2.18: Effet des fausses associations sur la classification.

L'exemple présenté dans cette section montre que la qualité de la classification est liée à la qualité du suivi. En effet, la classification est basée sur les données cinématiques estimées par l'algorithme d'estimation. Le principal problème des algorithmes dédiés au suivi est bien le cas où les trajectoires des objets se confondent. Ce problème est traité dans le prochain chapitre.

2.7 Conclusion

Ce chapitre traite de la classification d'objets à base de leurs données cinématiques. Deux exemples de classification sont étudiés. Le premier concerne la classification de cibles aériennes avec un niveau d'ambiguïté assez élevé dans la mesure où les classes considérées sont imbriquées. Il a été démontré dans la littérature que dans de telles situations, le classifieur basé sur les fonctions de croyance donne de meilleurs résultats comparé au classifieur Bayésien qui est communément utilisé.

Ce résultat a été démontré dans un cadre de classification d'un seul objet. Dans ce chapitre, ce résultat est confirmé dans un cadre multi-objets. Toutefois, une limitation quant à son utilisation dans un cadre multi-objets est constatée, elle est liée à la phase d'association dans l'algorithme du suivi, la phase qui se charge d'attribuer une observation à chaque objet connu. Cette phase est assurée par un algorithme *GNN* dans la solution de suivi proposée. C'est un algorithme qui peut être remplacé par d'autres solutions plus élaborées comme le *MHT*, *JPDA*, *PMHT*, au prix d'une plus grande complexité calculatoire [23]. En restant dans le cadre mono-scan et déterministe, le problème d'association est abordé dans le chapitre 3 avec des fonctions de croyance.

Un second exemple de classification est donné dans ce chapitre, il concerne la classification de piétons et propose une situation où les classes se chevauchent et varient dans le temps. Il a été montré qu'un simple affaiblissement de connaissances permet de s'adapter aux changements de classe.

Chapitre 3

Algorithmes d'association crédaux

L'association séquentielle des observations aux objets connus est l'une des tâches les plus ardues relevant du domaine du suivi et de la classification d'objets [23]. En effet, la phase d'association est en charge d'attribuer, à chaque pas de temps, les observations reçues (mesurées par un capteur) aux objets déjà connus (pistes suivies par les filtres). La figure 3.1 illustre le problème pour un pas de temps donné. La complexité d'un tel problème est essentiellement liée à l'aspect incertain, imprécis et ambigu de l'environnement du suivi, et également aux imperfections des instruments de mesure. Dans la solution du suivi initiale présentée dans le chapitre 2, la phase d'association est assurée par l'algorithme *GNN* qui est un algorithme d'association mono-scan et déterministe.

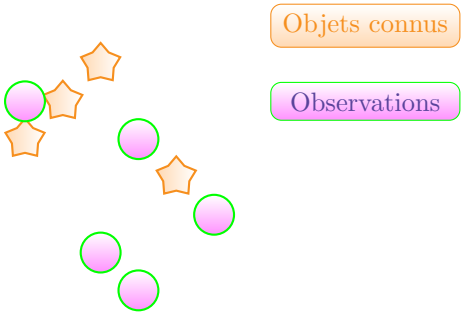


FIGURE 3.1: Illustration du problème d'association

L'algorithme *GNN* est loin de pouvoir répondre à toutes les exigences liées au problème d'association pour le suivi d'objets, ce chapitre considère d'autres solutions mono-scan et déterministes, basées sur les fonctions de croyance, qui visent à associer à chaque objet connu (observation) une hypothèse unique. À un objet connu, on peut associer une observation ou l'hypothèse de non détection. À une observation, on peut associer un objet déjà connu ou l'hypothèse d'apparition d'un nouvel objet. Pour l'algorithme *GNN*, décrit dans la section 1.2.2.1 du chapitre 1, cela est considéré comme étant un problème d'optimisation qui est résolu à l'aide de l'algorithme Hongrois [103], de l'algorithme

des enchères (*Auction*) [20] ou bien de l'algorithme de Munkres [30, 130]. Les algorithmes crédaux décrits dans ce chapitre proposent d'autres formalisations du problème d'association. Tout comme le *GNN*, ils nécessitent souvent l'usage d'un paramètre afin de permettre la gestion des apparitions et des disparitions d'objets. Toutefois, ils présentent une faculté intéressante quant à l'intégration de plusieurs informations (distances inter-objets, classes, formes et autres) dans un même problème d'association ce qui peut être utile dans un cadre multi-capteurs. Cette faculté d'association multi-informations n'est pas abordée dans ce chapitre, les algorithmes d'association crédale sont comparés à travers une panoplie de tests différents dans un cadre mono-capteur (association d'observations aux objets connus).

Les résultats présentés dans ce chapitre intègrent également les performances d'une nouvelle méthode d'association crédale à deux variantes : une variante optimale et une autre non optimale qui s'avèrent être de meilleure robustesse et par rapport à l'apprentissage paramétrique et par rapport aux erreurs d'estimation et/ou d'observation.

La section 3.1 de ce chapitre décrit la problématique liée à l'association des observations aux objets connus dans un cadre de suivi mono-capteur. La section 3.2 illustre le cas d'association multi-capteurs, là où plusieurs informations concernant les objets suivis peuvent être à disposition. La section 3.3 discutes et reprend les principes de base des algorithmes d'association crédale les plus récents : l'algorithme proposé par Denœux et *al* 3.3.1, l'algorithme proposé par Mercier et *al* 3.3.2, l'algorithme proposé par Lauffenburger et *al* 3.3.3, l'algorithme proposé par Fayad et Hamadeh 3.3.4, l'algorithme proposé par Dallil et *al* 3.3.5, ainsi qu'une nouvelle solution à deux variantes présentées respectivement dans 3.3.6 et 3.3.7. La section 3.4 met en relation l'algorithme *GNN* et certains algorithmes d'association crédale. Dans la section 3.5, le chapitre propose une panoplie de tests visant à comparer l'ensemble des algorithmes par rapport à leur optimalité, leur robustesse ainsi que leur complexité calculatoire. La section 3.6 conclue le chapitre.

3.1 Association des observations aux objets connus (association mono-capteur)

Les positions relatives des objets sont souvent la seule information dont les systèmes de suivi disposent. Dans ce cas, la formalisation du problème d'association se base sur les distances entre les observations réelles z_j, avec $j = \{1, 2, ..., m\}$ et les observations $\overline{z}_i$, $i = \{1, 2, ..., n\}$ prédites

par les *IMMs* chargés de suivre les objets connus. En plus de la prédiction des observations $\overline{z}_i$, les *IMMs* prédisent également les matrices de covariances associées S_i qui participent au calcul de la distance de Mahalanobis qui est utilisée tout au long de ce chapitre. On notera par O_i l'objet connu à qui correspond l'observation prédite $\overline{z}_i$.

Comme cela est décrit dans la section 1.2.2.1 du chapitre 1, l'association des observations z_j aux objets connus O_i est vue comme étant un problème d'optimisation où l'objectif est de minimiser la distance globale séparant les objects connus des observations :

$$min \sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} r_{i,j}, \tag{3.1}$$

où, $d_{i,j}$ représente la distance de Mahalanobis entre l'objet connu O_i et l'observation z_j, $r_{i,j}$ est une variable auxiliaire représentant la relation entre O_i et z_j ($r_{i,j} = 1$ si z_j est associée à O_i, $r_{i,j} = 0$ sinon). La gestion des apparitions et des disparitions d'objets est gérée en posant les contraintes suivantes :

$$\sum_{i}^{n+m} r_{i,j} \leq 1 \ , \tag{3.2}$$

$$\sum_{j}^{m} r_{i,j} \leq 1 \ , \tag{3.3}$$

$$r_{i,j} \in \{0,1\} \ . \forall i \in \{1,\ldots,n+m\}, \forall j \in \{1,\ldots,m\} \tag{3.4}$$

La contrainte (3.2) signifie qu'un objet existant peut être associé à une observation à l'instant k ou ne pas être associé, ce qui représente la non détection. La contrainte (3.2) signifie qu'une observation détectée à un instant k peut être associée à un objet existant, dans le cas échéant elle est considérée comme un nouvel objet.

Le problème, tel qu'il est formalisé par les équations (3.1), (3.2) et (3.3) correspond à la formalisation du *GNN*. La matrice d'association $D = [d_{i,j}]$ dans l'algorithme *GNN* contient des distances de Mahalanobis qui sont supposées suivre une distribution de χ^2 [122]. Le *GNN* considère qu'une observations z_j doit être associée à un objet connu ou bien à un nouvel objet noté $\star_j$, le paramètre ou la distance notée λ, liant l'observation z_j à l'hypothèse $\star_j$ est tiré de la table de χ^2 en se basant sur une connaissance *a priori* de l'environnement du suivi. Cette connaissance *a priori* est la

probabilité $P(z_j|O_i)$ qu'une observation détectée z_j soit générée par un objet connu O_i comme cela est décrit par l'équation (1.61) du chapitre 1.

Des formalisations différentes pour un objectif équivalent à celui du *GNN*, sont proposées par les algorithmes étudiés dans ce chapitre ainsi que par les deux nouvelles méthodes crédales proposées. Après la description des différents algorithmes, ces derniers sont comparés par rapport à leur optimalité, leur robustesse aux erreurs de mesure et/ou d'estimation et leur complexité calculatoire.

Dans les sections 3.1.1 et 3.1.2, on distingue ce que nous appelons un scénario conflictuel d'un scénario non conflictuel.

3.1.1 Association non conflictuelle

Notons par d_{opt} la distance minimale séparant les objets connus des observations, c'est la solution obtenue par l'algorithme Munkres ou l'algorithme Hongrois pour la résolution du problème d'optimisation posé par l'équation (3.1). La distance d_{opt} est fonction de la distance réelle entre les objets connus et les observations, qu'on notera d_r, elle est également fonction de la distance notée d_b occasionnée par le bruit de mesure et/ou les erreurs d'estimation :

$$d_{opt} = d_r + d_b. \tag{3.5}$$

Un scénario est considéré comme étant non conflictuel si $d_r \gg d_b$. Cela signifie que les distances réelles sont largement plus importantes que les modifications de distances causées par le bruit de mesure ou les erreurs d'estimation, il s'agit d'un scénario où les objets sont suffisamment espacés comme cela est illustré par la figure 3.2.

3.1.2 Association conflictuelle

D'une manière équivalente, on considère qu'un scénario est conflictuel si : $d_r \leq d_b$. Cela signifie que les objets sont suffisamment rapprochés au point que les distances qui les séparent sont considérablement modifiées par les erreurs de mesure et/ou les erreurs d'estimation, ce qui peut erroner la décision d'association. Un tel scénario est illustré par la figure 3.3 où l'observation z_1 de l'objet O_1 est associée par erreur à l'objet O_2 et l'observation z_2 de l'objet O_2 est associée par erreur à l'objet O_1. Sur la figure 3.3, $d_{opt} = d_{12} + d_{21} + d_{33}$.

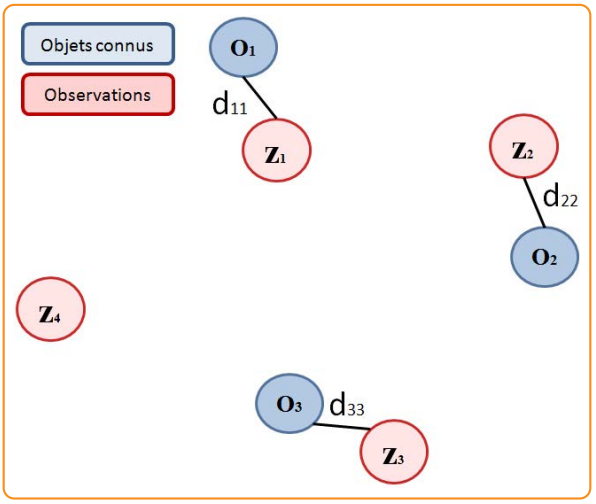


FIGURE 3.2: Association basée sur les distances dans un cas non conflictuel.

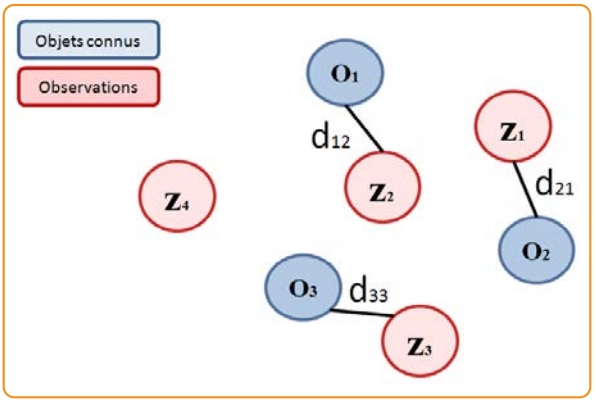


FIGURE 3.3: Association basée sur les distances dans un cas conflictuel.

D'une manière similaire, un scénario partiellement conflictuel peut être défini. C'est un scénario où certains objets sont éloignés et d'autres évoluent d'une manière étroite. Dans ce cas, la relation $d_r \leq d_b$ n'est valable que dans la partie conflictuelle du scénario.

3.2 Association d'objets suivis par différents capteurs (association multi-capteurs)

En présence d'informations supplémentaires comme : la vitesse, la classe et/ou la forme par exemple, en plus des distances séparant les objets estimés par deux capteurs différents, le conflit quant à leur association peut être réduit ou éliminé. Une illustration est donnée par la figure 3.4. Dans cet exemple, la connaissance de la forme de l'objet connu O_2 estimé à des positions différentes par les deux capteurs permet d'éviter l'erreur d'association qui est illustrée par la figure 3.3. La

question qui se pose est : comment intégrer des informations supplémentaires dans le problème d'association décrit par les équations (3.1), (3.2) et (3.3) ?

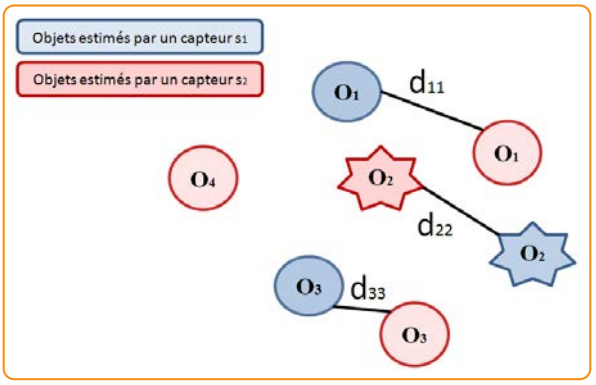


FIGURE 3.4: Association basée sur plusieurs informations.

Cette question est étudiée dans l'article [57], elle n'est cependant pas abordée dans ce chapitre où l'intérêt est porté à l'association mono-capteur, notamment l'association des observations aux objets suivis. L'association multi-capteurs est toutefois utilisée et illustrée dans la section 4.4(b) du chapitre 4.

La suite de ce chapitre a pour objectif de comparer les algorithmes d'association crédale les plus récents dans un cadre de suivi mono-capteur. La comparaison porte sur deux aspects fondamentaux dans le domaine du suivi d'objets, notamment :

- Performances dans les situations conflictuelles (voir la figure 3.3).
- Gestion des apparitions, des disparitions et des réapparitions d'objets.

Avant de passer à la comparaison, des descriptions succinctes des algorithmes sont proposées dans ce qui suit. L'idée vise à faire comprendre :

- Comment les distances sont-elles transformées en fonctions de croyance ?
- Comment les apparitions, les disparitions et les réapparitions d'objets sont-elle gérées ?
- Quels sont les principaux avantages et inconvénients des différentes solutions ?

3.3 Description des algorithmes d'association crédale

Dans la littérature, plusieurs solutions d'association basées sur des fonctions de croyance sont développées [47, 84, 123, 129, 158]. Cette section reprend les principes de base des algorithmes les plus récents et fournit un ensemble de remarques concernant leurs avantages et leurs inconvénients. Les algorithmes concernés sont décrits dans l'ordre suivant : l'algorithme proposé par Denœux et *al* 3.3.1, l'algorithme proposé par Mercier et *al* 3.3.2, l'algorithme proposé par Lauffenburger et *al* 3.3.3, l'algorithme proposé par Fayad et Hamadeh 3.3.4, l'algorithme proposé par Dallil et *al* 3.3.5 ainsi que deux versions d'un nouvel algorithme, présentées dans 3.3.6 et 3.3.7.

3.3.1 Méthode de Denœux et *al* [57, 67] :

La méthode proposée par Denœux et *al* dans [57, 67] transforme les distances entre observations et objets connus en fonctions de masse et considère la prise de décision comme étant un problème d'optimisation. Les fonctions de masse dans cette méthode s'expriment sur les relations $r_{i,j}$ concernant l'association de chaque objet connu O_i à une observation z_j, elles sont définies sur le cadre de discernement binaire $\Omega_{i,j} = \{0, 1\}$:

$$\begin{cases} m_{i,j}(\{1\}) = \alpha_{i,j}, & \text{poids affirmant l'association : } r_{i,j} = 1. \\ m_{i,j}(\{0\}) = \beta_{i,j}, & \text{poids réfutant l'association : } r_{i,j} = 0. \\ m_{i,j}(\{0,1\}) = 1 - \alpha_{i,j} - \beta_{i,j}, & \text{ignorance quant à l'association de } z_j \text{ à } O_i. \end{cases} \tag{3.6}$$

Disposant des distances $d_{i,j}$ comme information, les fonctions de masse sont calculées de la manière suivante :

$$\begin{cases} m_{i,j}(\{1\}) = \alpha_{i,j} = \sigma \exp(-\gamma d_{i,j}), \\ m_{i,j}(\{0\}) = \beta_{i,j} = \sigma(1 - \exp(-\gamma d_{i,j})), \\ m_{i,j}(\{0,1\}) = 1 - \alpha_{i,j} - \beta_{i,j} = 1 - \sigma, \end{cases} \tag{3.7}$$

où $d_{i,j}$ représente une distance entre l'objet connu O_i et l'observation z_j et γ un paramètre à fixer. Le paramètre σ est utilisé pour affaiblir l'information, selon la fiabilité de la source (capteur) [174].

L'ensemble des relations concernant l'association de O_i et z_j est noté R, et l'ensemble des relation affirmant leur association est noté $R_{i,j}$ et donné par :

$$R_{i,j} = \{r \in R | r_{i,j} = 1\}. \tag{3.8}$$

Les fonctions de masse $m_{i,j}$ initialement définies sur le cadre binaire $\Omega_{i,j} = \{0, 1\}$ sont étendues sur l'ensemble des relations R :

- la masse $m_{i,j}(\{1\}) = \alpha_{i,j}$ est transférée sur $R_{i,j}$.
- la masse $m_{i,j}(\{0\}) = \beta_{i,j}$ est transférée sur $\overline{R_{i,j}}$.
- la masse $m_{i,j}(\{0, 1\}) = 1 - \alpha_{i,j} - \beta_{i,j}$ est transférée sur R.

Le transfert des fonctions de masse se fait à l'aide de l'extension vide décrite dans la section 2.2.1.2 du chapitre 2. Les fonctions de masse $m_{i,j}$ sont ensuite transformées en fonctions de plausibilités $pl_{i,j}$ de la manière suivante :

$$pl_{i,j}(\{r\}) = (1 - \beta_{i,j})^{r_{i,j}}(1 - \alpha_{i,j})^{1-r_{i,j}}. \tag{3.9}$$

Une fois toutes les fonctions de plausibilités s'exprimant sur le couple (O_i, z_j) sont obtenues, elles sont combinées à l'aide la loi conjonctive décrite dans [57] :

$$pl(\{r\}) \propto \prod_{i,j}(1 - \beta_{i,j})^{r_{i,j}}(1 - \alpha_{i,j})^{1-r_{i,j}}. \tag{3.10}$$

Une fonction de plausibilité $pl(\{r\})$ est calculée pour tout $r \in R$. La fonction logarithme correspondante est donnée par :

$$\ln pl(\{r\}) \propto \sum_{i,j}[r_{i,j} \ln(1 - \beta_{i,j}) + (1 - r_{i,j}) \ln(1 - \alpha_{i,j})]. \tag{3.11}$$

Le calcul de la fonction logarithme dans l'équation (3.11) permet d'exprimer le problème comme étant un problème d'optimisation. Les paramètres $\alpha_{i,j}$ et $\beta_{i,j}$ sont considérés strictement inférieurs à 1. L'objectif est alors de retrouver la relation optimale r^* qui maximise les fonctions logarithme des plausibilités. Le problème pour n objets connus O_i et m observations z_j est décrit par :

$$max \sum_{i,j} w_{i,j} r_{i,j}, \quad i = \{1, ..., n\}, j = \{1, ..., m\}, \tag{3.12}$$

avec

$$\sum_{i}^{n} r_{i,j} \leq 1 \ , \tag{3.13}$$

$$\sum_{j}^{m} r_{i,j} \leq 1 \ , \tag{3.14}$$

$$r_{i,j} \in \{0, 1\}, \quad \forall i \in \{1, ..., n\}, \forall j \in \{1, ..., m\} \ , \tag{3.15}$$

et $w_{i,j} = \ln \frac{1-\beta_{i,j}}{1-\alpha_{i,j}}$. Comme dans le cas de l'algorithme *GNN*, ce problème est résolu à l'aide de l'algorithme Munkres [31], par exemple. L'algorithme proposé par Denœux et *al* est optimal quand le paramètre γ permettant de transformer les distances en fonctions de masse est optimal. Dans le cadre de la fusion multi-capteurs, il présente une faculté intéressante concernant l'association multi-informations (distances, classes, vitesses, et autres). Le temps de résolution de l'algorithme est nettement inférieur au temps de résolution des algorithmes d'association basés sur les fonctions de croyance comme par exemple la méthode proposée par Mercier et *al* [125], ce qui est confirmé dans l'étude qui va suivre.

3.3.2 Méthode de Mercier et *al* [125] :

Dans la méthode introduite par Mercier et *al* dans [125], les fonctions de masse $m_{i,j}$ sont également exprimées sur le cadre de discernement $\Omega_{i,j} = \{0, 1\}$ concernant l'association de l'observation z_j à l'objet connu O_i. Leur calcul est effectué comme décrit par le système d'équations (3.7).

L'association dans cette méthode peut s'effectuer dans deux sens différents : point de vue des objets connus (à quelle observation z_j l'objet connu O_i est-il associé ?) et le point de vue des observations (à quel objet connu O_i, l'observation z_j est-elle associée ?).

Selon le point de vue considéré, les masses de croyance $m_{i,j}$ sont étendues (extension vide décrite dans la section 2.2.1.2) à l'ensemble $\Theta_i = \{z_1, z_2, ..., z_m, \star\}$ contenant toutes les hypothèses concernant l'association de l'objet connu O_i, y compris l'hypothèse que O_i soit non détecté qui est notée $(\star)$. Si l'association se fait selon le point de vue des observations, les masses $m_{i,j}$ concernant l'observation z_j sont étendues à l'ensemble $\Theta_j = \{O_1, O_2, ..., O_n, \star\}$ où $(\star)$ représente l'hypothèse

que z_j soit un nouvel objet. On note par $m_j^{\Theta_i}$, l'extension de la fonction de masse $m_{i,j}$ de l'ensemble $\Omega_{i,j}$ à l'ensemble Θ_i.

Une fois les fonctions de masse sont exprimées sur le même cadre de discernment (Θ_i par exemple), elles sont combinées à l'aide de la règle de combinaison conjonctive (voir la section 2.2.1.1).

À l'issue de la combinaison conjonctive, pour chaque objet connu i, on obtient une fonction de masse m^{Θ_i}. La procédure est la même dans le cas où on s'intéresse au point de vue des observations.

En vue de prendre une décision quant à l'association des observations aux objets connus, les fonctions de masse sont transformées en probabilités pignistiques en utilisant l'équation (2.18).

Une fois, les probabilités pignistiques $Betp^{\Theta_i}$ de chaque objet connu $i \in \{1, 2, ..., n\}$ sont calculées, il en résulte une matrice de probabilités pignistiques. La décision d'association est celle qui maximise la probabilité pignistique jointe $Betp^{\Theta_1 \times \Theta_2, ... \times \Theta_n}$ et qui vérifie des contraintes d'association préalablement fixées, à savoir :

- chaque objet O_i doit être associé à une seule hypothèse : l'observation z_j ou bien $\star$ s'il est non détecté.
- chaque observation z_j doit être associée à une seule hypothèse : l'objet connu O_i ou bien $\star$ s'il s'agit d'un nouvel objet.

Comme cela peut être remarqué, le calcul des fonctions de masse dans la méthode proposée par Mercier et *al* s'effectue de la même manière que dans la méthode proposée par denœux et *al*, toutes les deux utilisent le paramètre γ afin de transformer les distances en fonctions de masse comme expliqué par le système d'équations (3.7). La différence principale est dans la phase de prise de décision qui nécessite dans la méthode de Mercier et *al* la recherche du maximum de la probabilité pignistique jointe ce qui la rend beaucoup plus complexe. Plus de détails concernant la comparaison des algorithmes en terme de complexité calculatoire sont présentés dans la section 3.5.4.4.

3.3.3 Méthode de Lauffenburger et *al* [105]

Au niveau crédal, la méthode proposée par Lauffenburger et *al* est équivalente à la méthode proposée par Mercier et *al*, toutefois, Lauffenburger et *al* proposent de considérer les deux points de vue (point de vue des objets connus et point de vue des observations) à la fois. Deux matrices

de probabilités pignistiques sont alors calculées au lieu d'une seule pour la méthode proposée par Mercier et *al*. Les deux matrices de probabilités pignistiques sont utilisées pour le calcul d'une matrice duale qui sert à la prise de décision.

Tout comme dans les méthodes de Denœux et *al* et Mercier et *al*, les fonctions de masse initiales $m_{i,j}$ sont exprimées sur le cadre de discernment local $\Omega_{i,j}$. Par la suite, elles sont étendues aux cadres de discernment $\Theta_i = \{z_1, z_2, ..., z_m, \star\}$ et $\Theta_j = \{O_1, O_2, ..., O_n, \star\}$ correspondant au point de vue des objets connus et au point de vue des observations, respectivement :

- $m_{i,j}(\{1\}) = \alpha_{i,j}$ est affecté à $\{z_j\} \in \Theta_i$.
- $m_{i,j}(\{0\}) = \beta_{i,j}$ est affecté à $\{z_1, ..., z_{j-1}, z_{j+1}..., z_m, \star\}$.
- $m_{i,j}(\{0,1\}) = 1 - \alpha_{i,j} - \beta_{i,j}$ est affecté à Θ_i ,

et

- $m_{i,j}(\{1\}) = \alpha_{i,j}$ est affecté à $\{O_i\} \in \Theta_j$.
- $m_{i,j}(\{0\}) = \beta_{i,j}$ est affecté à $\{O_1, ..., O_{i-1}, O_{i+1}..., O_n, \star\}$.
- $m_{i,j}(\{0,1\}) = 1 - \alpha_{i,j} - \beta_{i,j}$ est affecté à Θ_j.

De cette opération, il en résulte deux ensembles de fonctions de masse $m_j^{\Theta_i}$ et $m_i^{\Theta_j}$ exprimées sur Θ_i et Θ_j respectivement. Une fois obtenues, les fonctions de masses sont combinées à l'aide de la règle de combinaison conjonctive. Les fonctions de masse résultantes sont notées m^{Θ_i} et m^{Θ_j} :

$$m^{\Theta_i} = \cap_j m_j^{\Theta_i}, \tag{3.16}$$

$$m^{\Theta_j} = \cap_i m_i^{\Theta_j}. \tag{3.17}$$

Les fonctions de masse m^{Θ_i} et m^{Θ_j} sont transformées en probabilités pignistiques. Contrairement à la méthode proposée par Mercier et *al* où le poids accordé à l'ensemble vide $\emptyset$, à l'issue de la combinaison, est normalisé lors du calcul des probabilités pignistiques, cette méthode propose de conserver le poids en question. Le calcul des probabilités pignistiques est effectué de la façon suivante :

$$\begin{cases} BetP^{\Theta_i}(\{h_i\}) & = & \sum_{h_i \in A} \frac{m^{\Theta_i}(A)}{|A|} \\ P^{\Theta_i}(\emptyset) & = & m^{\Theta_i}(\emptyset) \end{cases} \tag{3.18}$$

$$\begin{cases} BetP^{\Theta_j}(\{h_i\}) & = & \sum_{h_i \in A} \frac{m^{\Theta_j}(A)}{|A|} \\ P^{\Theta_j}(\emptyset) & = & m^{\Theta_j}(\emptyset) \end{cases} \tag{3.19}$$

où $P^{\Theta_i}(\emptyset)$ et $P^{\Theta_j}(\emptyset)$ sont les poids accordés à l'ensemble vide ($\emptyset$), issus des combinaisons conjonctives exprimées par les équations (3.16), (3.17) respectivement.

Une fois les deux matrices de probabilités pignistiques $BetP^{\Theta_i}$ et $BetP^{\Theta_j}$ sont calculées, elles sont utilisées pour calculer une matrice dite duale. La matrice duale est calculée en effectuant le produit matriciel d'Hadamard :

$$BetP^{\Theta_i \times \Theta_j} = (BetP^{\Theta_i})' \circledast BetP^{\Theta_j}. \tag{3.20}$$

En utilisant les poids représentant le conflit (poids sur l'ensemble vide $\emptyset$), une matrice duale de conflit, notée M^{conf}, est également calculée de la manière suivante :

$$M^{conf} = (P^{\Theta_i})'.P^{\Theta_j},\ i = \{1, 2, ..., n\},\ j = \{1, 2, ..., m\}. \tag{3.21}$$

La matrice conflictuelle M^{conf} est transformée en deux vecteurs V_c^{conf} et V_r^{conf} de la façon suivante :

$$V_c^{conf} = \sum_{j=1}^{m} M^{conf}(., j), \tag{3.22}$$

$$V_r^{conf} = \sum_{i=1}^{n} M^{conf}(i, .). \tag{3.23}$$

La prise de décision est basée sur la matrice de probabilités pignistiques duale de dimension $(n \times m)$ et les vecteurs V_c^{conf} et V_r^{conf}, de longueurs n et m respectivement.

Pour chaque objet connu O_i correspondant à la ligne i de la matrice duale, l'hypothèse avec la probabilité pignistique maximale $\max_j BetP^{\Theta_i \times \Theta_j}(i, j)$ est retenue. L'objet O_i est associé à l'observation z_j, si et seulement si les conditions suivantes sont satisfaites :

- La probabilité pignistique maximale soit supérieure au poids conflictuel $V_c^{conf}(i)$:

$$\max_j BetP^{\Theta_i \times \Theta_j}(i,j) > V_c^{conf}(i).$$

- La probabilité pignistique maximale soit supérieure à un seuil Th fixé au préalable :

$$\max_j BetP^{\Theta_i \times \Theta_j}(i,j) > Th.$$

  Notons que dans l'article [105], Th est fixé à 0.1. Ce dernier n'est pas optimisé dans l'étude présenté dans ce chapitre, la même valeur $Th = 0.1$ est utilisée.

Si ces deux conditions sont satisfaites alors O_i est associé à z_j, sinon l'objet connu O_i est considéré comme étant non détecté. Dans [105], la méthode est testée sur des scénarios peu conflictuels et la solution semble donner de bons résultats. Dans ce chapitre, les algorithmes d'association sont testés sur des scénarios conflictuels. Cette solution s'avère ne pas être adaptée à ce genre de scénarios. Cela serait essentiellement dû à l'utilisation du poids accordé au conflit dans la prise de décision ou/et à la non optimisation du paramètre Th. Toutefois, la prise de décision locale évitant la recherche du maximum de la probabilité jointe peut s'avérer moins complexe que dans le cas de la méthode proposée par Mercier et *al.*

3.3.4 Méthode de Fayad et Hamadeh [69] :

La méthode de Fayad et Hamadeh décrite dans [69] propose un calcul différent des fonctions de masse et une autre stratégie de prise de décision. La méthode propose de calculer une fonction de masse pour chaque observation z_j dont les hypothèses d'association possibles sont $\Theta_j^\star = \{O_1, O_2, ..., O_n, \star\}$. Le calcul des poids de la fonction de masse se fait par normalisation des distances séparant l'observation des objets connus. L'élément $(\star)$ est vu comme étant un objet virtuel, tel que si une observation lui est associée, elle est considérée comme étant un nouvel objet ainsi donc la gestion des apparitions de nouveaux objets est permise. Pour illustrer la manière dont les fonctions de masse sont calculées, on considère l'exemple d'association donné par la figure 3.5 avec une seule observation. Supposons que $d_{1,1}$, $d_{2,1}$ et $d_{3,1}$ sont les distances séparant l'observation z_1 des objets connus O_1, O_2 et O_3 respectivement, et $d_{\star,1}$ est la distance d'apparition fixée au préalable. En premier temps, ces distances sont triées dans un ordre décroissant et utilisées pour calculer les poids M_i :

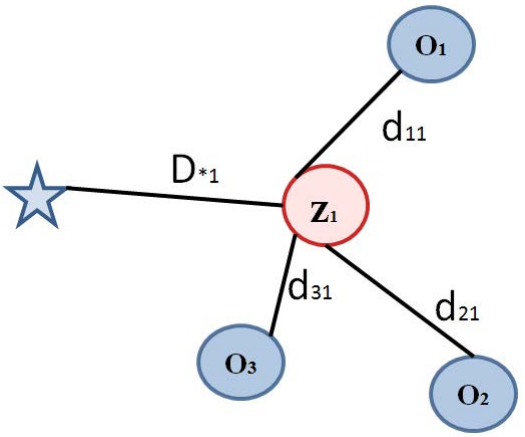


FIGURE 3.5: Association concernant une seule observation.

$$\begin{aligned} M_1 &= 1/d_{1,1}, \\ M_\star &= (1/d_{\star,1}) - M_1, \\ M_2 &= (1/d_{2,1}) - M_\star, \\ M_3 &= (1/d_{3,1}) - M_2. \end{aligned} \tag{3.24}$$

En utilisant les poids calculés dans (3.24), une fonction de masse m^{Θ_1} concernant l'observation z_1 est calculée :

$$\begin{aligned} m^{\Theta_1}(\{O_3\}) &= M_3/M, \\ m^{\Theta_1}(\{O_2, O_3\}) &= M_2/M, \\ m^{\Theta_1}(\{\star, O_1, O_2\}) &= M_\star/M, \\ m^{\Theta_1}(\{\star, O_1, O_2, O_3\}) &= M_1/M, \end{aligned} \tag{3.25}$$

où M est un facteur de normalisation : $M = M_\star + M_1 + M_2 + M_3$.

Une fonction de masse m^{Θ_j} est calculée pour toutes les observations z_j avec $j = \{1, 2, ..., m\}$.

Les fonctions de masse sont ensuite étendues au cadre de discernement joint de toutes les observations en adoptant une combinaison de connaissances particulière illustrée par l'exemple suivant : considérons un exemple de trois objets connus O_1, O_2 et O_3 et deux observations z_1, z_2 détectées. Deux fonctions de masse m^{Θ_1} et m^{Θ_2} sont calculées pour les observations z_1 et z_2 comme illustré par les équations (3.25). Les fonctions de masse sont ensuite exprimées sur le cadre de discernement joint $\Theta_{1\times 2}$:

$$\Theta_{1\times 2} = \left\{ \ (\star,\star),(\star,O_1),(\star,O_2),(O_1,\star),(O_1,O_2),(O_2,\star),(O_2,O_1) \ \right\}, \tag{3.26}$$

où la proposition $(\star, O_1)$ par exemple, signifie que l'observation z_1 est considérée comme un nouvel objet et l'observation z_2 est associée à l'objet connu O_1.

La fonction de masse $m^{\Theta_{1\times 2}}$ est le résultat de la combinaison des fonctions de masse m^{Θ_1} et m^{Θ_2} à l'aide de l'opérateur min (minimum). En voici par exemple, comment est ce que le poids accordé à la proposition $\{\star,\star\}$ du cadre $\Theta_{1\times 2}$ est obtenu :

$$m^{\Theta_{1\times 2}}(\{\star,\star\}) = \min(m^{\Theta_1}(\{\star\}), m^{\Theta_2}(\{\star\})), \tag{3.27}$$

$$m^{\Theta_{1\times 2}}(\{(\star,\star),(\star,O_1)\}) = \min(m^{\Theta_1}(\{\star\}), m^{\Theta_2}(\{\star,O_1\})). \tag{3.28}$$

Le même processus est répété pour toutes les autres propositions du cadre $\Theta_{1\times 2}$.

Afin de normaliser la fonction de masse $m^{\Theta_{1\times 2}}$, tous ces éléments $m^{\Theta_{1\times 2}}(A)$ avec $A \subseteq \Theta_{1\times 2}$ sont divisés par : $\sum_{A\subseteq\Theta_{1\times 2}} m^{\Theta_{1\times 2}}(A)$.

Afin de prendre une décision d'association, la fonction de masse jointe est transformée en probabilités pignistiques comme décrit dans la section 2.2.3 du chapitre 2. L'hypothèse $A \subseteq \Theta_{1\times 2}$ la plus probable est considérée comme étant la solution au problème d'association posé.

À noter que dans cette solution la distance $d_{\star,j}$ correspond au paramètre λ de l'algorithme *GNN* : c'est la distance d'apparition. Il est attendu que la méthode soit de grande complexité calculatoire notamment pour un nombre d'objets et/ou d'observations important dans la mesure où elle cherche à calculer et prendre une décision sur des fonctions de masse jointes.

3.3.5 Méthode proposée par Dallil et *al* [46] :

Tout comme dans les méthodes de Denœux et *al* et Mercier et *al* les fonctions de masse initiales dans la méthode proposée par Dallil et *al* sont exprimées sur le cadre de discernement binaire $\Omega_{i,j} = \{0,1\}$ pour chaque objet connu O_i et observation z_j.

Dans cette approche les poids confirmant les associations notés $\alpha_{i,j}$ dans la méthode de Denœux et *al* ne sont autres que des vraisemblances normalisées, calculées comme suit :

$$\alpha_{i,j} = \frac{G_{i,j}}{\sum_{j}^{m} G_{i,j}} \ , \tag{3.29}$$

où $G_{i,j}$ est une mesure de vraisemblance :

$$G_{i,j} = \frac{exp[-d_{i,j}^2/2]}{\sqrt{(2\pi)^q |S_i|}}, \quad i \in \{1, ..., n\} \ , \tag{3.30}$$

et S_i représente la matrice de covariance associée à l'objet connu O_i et q représente la dimension du vecteur observé z_j.

Les fonctions de masse initiales sont alors données par :

$$\begin{cases} m_{i,j}(\{1\}) &= \alpha_{i,j}, \\ m_{i,j}(\{0\}) &= 1 - \alpha_{i,j} \ = \ \beta_{i,j}, \end{cases} \tag{3.31}$$

Selon le point de vue choisi, les fonctions de masse sont conjonctivement combinées. La combinaison selon le point de vue des objets connus, par exemple, est effectuée de la manière suivante :

$$M_{i,j} = m_{i,j}(\{1\}) \times \prod_{l=1, l \neq j}^{m} m_{i,l}(\{0\}). \tag{3.32}$$

Le poids conflictuel issu de la combinaison est calculé ainsi :

$$M_{i,\emptyset} = 1 - \sum_{j=1}^{m} M_{i,j}. \tag{3.33}$$

La quantité $M_{i,j}$ représentant la croyance globale concernant l'association de l'objet connu O_i et l'observation z_j, est exprimée sur le cadre de discernement Θ_i concernant l'objet connu O_i. Les différents poids sont ensuite étendus au cadre de discernement joint $\Theta_{1,1} \times ... \times \Theta_{n,m}$ concernant tous les objets et toutes les observations. Une fois exprimées sur le même cadre de discernement, les fonctions de masse sont conjonctivement combinées. Il en résulte une fonction de masse jointe notée m_T :

$$m_T(A) = \prod_{i \in A, j \in A} \frac{M_{i,j}}{1 - M_{i,\emptyset}}, \tag{3.34}$$

où A est une proposition du cadre de discernement joint $\Theta_{1,1} \times ... \times \Theta_{n,m}$. La proposition $A = \{(O_1, z_1), (O_i, z_j), (O_n, z_m)\}$ par exemple, signifie que les observations z_1, z_j et z_m sont générées par les objets connus O_1, O_i et O_n, respectivement.

Pour la phase de prise de décision, la proposition $A \in \Theta_{1,1} \times ... \times \Theta_{n,m}$ la plus crédible est alors considérée comme étant la solution au problème d'association posé.

Le nouveauté dans la méthode proposée par Dallil et *al* est le fait de calculer les fonctions de masse à partir des vraisemblances, seulement leur extension sur un cadre de discernement joint engendre une grande complexité calculatoire. De plus la méthode ne présente aucune stratégie pour gérer les apparitions de nouveaux objets [46].

3.3.6 Solution proposée 1 :

La méthode proposée repose sur le *GBT*, elle construit les fonctions de masse concernant les associations à partir de vraisemblances. Elle s'inspire des travaux de Smets [171] sur les fonctions de croyances appliquées aux variables réelles ainsi que les travaux d'Appriou [3, 5], où les vraisemblances sont interprétées comme étant des fonctions de plausibilités. Ce sont les fonctions les moins spécifiques qui permettent de respecter le principe d'engagement minimum. Les fonctions de plausibilités (vraisemblances) représentent les données en entrée de l'algorithme d'association proposé, elles sont exprimées sur l'ensemble des observations $z_j,\ j \in \{1, \ldots, m\}$ et peuvent être notées par $pl[O_i](z_j)$ ce qui répondrait à la question : à quelle objet connu O_i l'observation z_j est associée ? Dans ce qui suit, la notation $pl[O_i](z_j)$ est simplifiée à $pl_i(\{j\})$ et le calcul des plausibilité s'effectue ainsi :

$$pl_i(\{j\}) = G_{i,j}(\mu\ d_{i,j}), \quad \forall i \in \{1, \ldots, n\}, \tag{3.35}$$

où $G_{i,j}$ est une mesure de vraisemblance et $\mu \in [0, 1]$ est un paramètre qui pondère les distances de Mahalanobis $d_{i,j}$. Dans le cas de distributions Gaussiennes, les plausibilités peuvent être calculées par l'équation (3.30). Les simulations de ce chapitre considère des exemples de distributions Gaussiennes mais cela n'est point une limite à la méthode proposée. Peu importe la distribution des variables tant que des vraisemblances peuvent être calculées.

La plausibilité maximale qu'une observation z_j soit associée à un des n objets connus correspond à : $\max_{i=1,...,n}(pl_i(\{j\})) \leq 1$. Si l'observation en question est réellement générée par l'un des objets connus alors $\max_{i=1,...,n}(pl_j(\{i\})) \approx 1$ sinon le maximum de plausibilité est nettement inférieur à 1 ($\max_{i=1,...,n}(pl_j(\{i\})) < 1$) ce qui correspondrait à un cadre de discernement non exhaustif. Dans ce cas l'observation z_j correspond éventuellement à un nouvel objet $\star_j$. La plausibilité de cette hypothèse peut alors être calculée par :

$$pl_{\star_j}(\{j\}) = 1 - \max_{i=1,...,n}(pl_i(\{j\})). \tag{3.36}$$

À l'aide des plausibilités calculées par les équations (3.35) et (3.36), une fonction de masse m est construite. La fonction de masse s'exprime sur l'ensemble $\Theta_j = \{O_1, ..., O_n, \star_j\}$ et peut être notée par $m[z_j](h_i)$ tel que $h_i \in \Theta_j$. La notation est simplifiée à $m_j(A)$ où j fait référence aux observations z_j et A à un ensemble où un sous-ensemble de Θ_j. Les fonctions de masse sont obtenues à l'aide du *GBT* :

$$m_j(A) = \prod_{hi \in A} Pl_{hi}(\{j\}) \prod_{hi \in \bar{A}} (1 - Pl_{hi}(\{j\})), \quad \forall A \subseteq \Theta_j. \tag{3.37}$$

Une fois toutes les fonctions de masse m_j sont calculées, elles sont transformées en probabilités pignistiques $BetP_j$ comme décrit dans la section 2.2.3 du chapitre 2. L'objectif final est alors de choisir la relation jointe avec le maximum de probabilité pignistique, ce qui est formalisé de la manière suivante :

$$max \sum_{i,j} BetP_j(\{i\}) r_{i,j}, \quad i = \{1, \ldots, n+m\}, j = \{1, \ldots, m\}, \tag{3.38}$$

en respectant les contraintes suivantes :

$$\sum_{i}^{n+m} r_{i,j} \leq 1 \ , \tag{3.39}$$

$$\sum_{j}^{m} r_{i,j} = 1 \ , \tag{3.40}$$

$$r_{i,j} \in \{0, 1\} \ . \forall i \in \{1, \ldots, n+m\}, \forall j \in \{1, \ldots, m\}. \tag{3.41}$$

Tout comme dans la méthode de Denœux et *al* (voir la section 3.3.1), ce problème d'optimisation est résolu à l'aide de l'algorithme des enchères ou l'algorithme de Munkres [31], par exemple.

La contrainte exprimée par (3.39) signifie qu'un objet connu peut recevoir ou ne pas recevoir d'observation. Dans le cas de non réception d'observation, l'objet est considéré comme étant non détecté. La contrainte (3.40) signifie qu'une observation peut être associée à un objet connu, dans le cas contraire, elle est considérée comme étant un nouvel objet.

Le problème d'association à l'étape décisionnelle, pour l'exemple présenté par la figure 3.3 peut être traduit par le tableau 3.1.

TABLEAU 3.1: Matrice des probabilités pignistiques.

	z_1	z_2	z_3	z_4
O_1	$BetP_{1,1}$	$BetP_{1,2}$	$BetP_{1,3}$	$BetP_{1,4}$
O_2	$BetP_{2,1}$	$BetP_{2,2}$	$BetP_{2,3}$	$BetP_{2,4}$
O_3	$BetP_{3,1}$	$BetP_{3,2}$	$BetP_{3,3}$	$BetP_{3,4}$
$\star_1$	$BetP_{\star_1,1}$	0	0	0
$\star_2$	0	$BetP_{\star_2,2}$	0	0
$\star_3$	0	0	$BetP_{\star_3,3}$	0
$\star_4$	0	0	0	$BetP_{\star_4,4}$

3.3.7 Solution proposée 2 :

La seconde méthode proposée et quasi-similaire à celle présentée dans la section 3.3.6. À la différence de la première, celle-ci utilise une distance de Mahalanobis pure pour le calcul des vraisemblances et effectue un transfert d'informations au calcul des plausibilités. Le transfert d'information est simplement un renforcement des éléments du cadre de discernement en détriment de l'élément ($\star$). Les plausibilités se calculent alors par :

$$pl_i(\{j\}) = G_{i,j}(d_{i,j}), \quad \forall i \in \{1, \ldots, n\}, \tag{3.42}$$

avec,

$$pl_{\star_j}(\{j\}) = 1 - \max_{i=1,\ldots,n} (pl_i(\{j\})), \tag{3.43}$$

et le paramètre μ manquant par rapport à la formule (3.35) intervient cette fois dans le renforcement des plausibilités $pl_i(\{j\})$ favorisant l'association de l'observation z_j aux objets connus O_i, ce qui se fait de la manière suivante :

$$^{\mu}pl_i(\{j\}) = \mu \times pl_i(\{j\}) + (1-\mu), \quad \forall i \in \{1,\ldots,n\}, \tag{3.44}$$

$$^{\mu}pl_{\star_j}(\{j\}) = \mu \times pl_{\star_j}(\{j\}). \tag{3.45}$$

Les équations (3.44) et (3.45) expriment un renforcement d'hypothèses tel qu'il est introduit dans [124] et repris par l'équation (2.11) du chapitre 2. Dans le cadre du suivi d'objets dans cette solution, ce concept est utilisé afin de renforcer les hypothèses favorisant l'association des observations z_j aux objets connus O_i. Cette action est motivée par l'effet des erreurs de prédiction des objets connus et les erreurs d'observation qui lorsqu'elles sont importantes peuvent conduire à des décisions erronées, notamment les décisions associant les observations d'objets connus à de nouveaux objets. De ce fait, le paramètre μ dans les équations (3.44), (3.45) représenterait le poids qu'aurait engendré les erreurs de mesure et de prédiction, il est utilisé dans cette solution comme information *a priori* dont l'objectif est d'apporter plus de robustesse. À noter que dans un cadre idéaliste où les erreurs de mesure et de prédiction sont négligeables, la solution sans renforcement où les plausibilités sont calculées à l'aides des équations (3.42) et (3.43) simplement, serait optimale mais surtout indépendante de tout paramètre.

Le calcul des fonctions de masse et la prise de décision dans cette méthode se font exactement comme dans la méthode proposée 1 : le calcul des fonctions de masse se fait à l'aide de l'équation (3.37) et la prise de décision se fait sur les probabilités pignistiques comme cela est exprimé par le tableau 3.1 par exemple.

Alternativement, l'affaiblissement de la plausibilité $pl_{\star_j}(\{j\})$ de la manière suivante peut être envisagé :

$$pl_{\star_j}(\{j\}) = \mu(1 - \max_{i=1,\ldots,n}(pl_i(\{j\}))).$$

L'affaiblissement étant motivé par les erreurs d'estimation ou de mesure qui conduiraient au fait de considérer que les observations des objets existants correspondent à de nouveaux objets. L'optimisation du paramètre μ dans le cas de cette altérnative est difficile dans la mesure où on agit sur les plausibilités, contrairement au cas de la solution proposée 1 où on agit directement sur les distances. Tout comme la solution proposée 2, c'est une solution qui n'atteint pas l'optimalité, toutefois des valeurs faibles du paramètre μ donnent de bons résultats. Afin d'éviter la confusion avec les deux solutions proposées, les résultats de cette altérnative ne sont pas présentés dans cette thèse.

L'originalité dans les deux solutions proposées et le fait de pouvoir calculer d'une manière dynamique les poids accordés à l'hypothèse ($\star$) notamment, grâce à l'équation (3.36).

3.4 Mise en correspondance du paramètre γ propre à certaines solutions crédales et du paramètre λ de l'algorithme *GNN*

Cette section illustre une relation entre l'algorithme *GNN* et l'algorithme proposé par Fayad et Hamadeh qui dépendent du paramètre λ et les algorithmes crédaux qui dépendent du paramètre γ permettant de transformer les distances en fonctions de masse (voir le système d'équations (3.7)).

Le paramètre λ dans l'algorithme *GNN* correspond à la distance de détection. Ce paramètre peut être mis en relation avec le paramètre γ dans l'ensemble d'équations (3.7). Pour ce faire, il suffit de choisir un paramètre γ tel que la distance de détection des algorithmes crédaux soit égale à λ sachant que la distance de détection pour les méthodes crédales satisfait la relation $\beta_{i,j} = \alpha_{i,j}$. Cette relation représente le cas où le poids confirmant une association donnée est égal au poids qui la réfute.

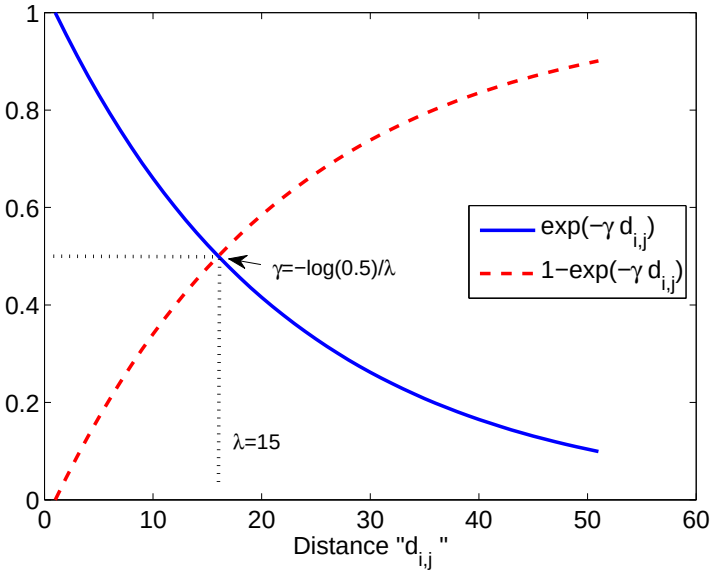


FIGURE 3.6: Illustration de la relation entre γ et λ.

La figure 3.6 représente l'évolution du poids approuvant l'association de z_j à O_i : $\alpha_{i,j} = \exp(-\gamma d_{i,j})$, et le poids réfutant l'association : $\beta_{i,j} = 1 - \exp(-\gamma d_{i,j})$, en fonction de la distance $d_{i,j}$.

Si λ est choisi comme étant la distance de détection, on peut poser $\exp(-\gamma\lambda) = 1 - \exp(-\gamma\lambda)$ et déduire la valeur du paramètre γ permettant d'avoir une telle distance de détection. D'où :

$$\gamma = \frac{-\log(0.5)}{\lambda}. \tag{3.46}$$

La relation exprimée par l'équation (3.46) montre que les méthodes crédales gèrent l'apparition d'objets à l'aide d'un seuil γ équivalent à la détermination de la distance d'apparition λ dans la solution *GNN*.

3.5 Comparaisons dans un cadre d'association mono-capteur

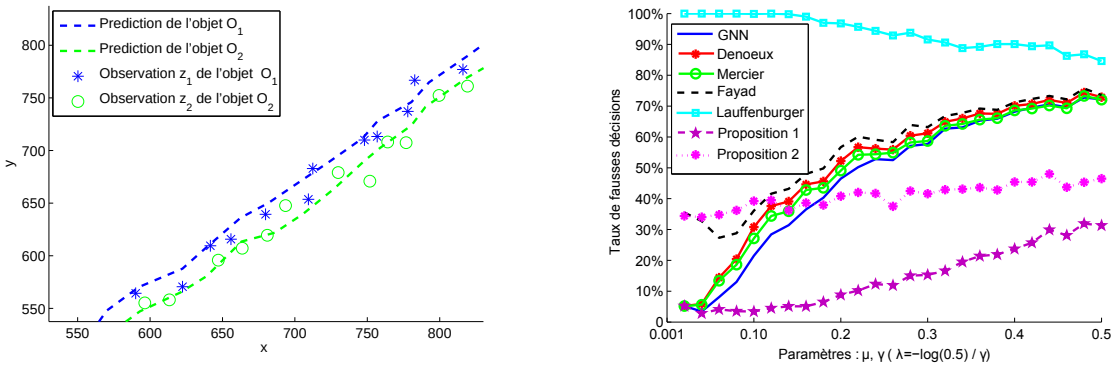
À travers les descriptions précédentes des algorithmes, il a pu être remarqué que tous les algorithmes dépendent d'au moins un paramètre. Les premières simulations présentées dans les sections 3.5.1, 3.5.2 et 3.5.3 ont pour objectif l'apprentissage des paramètres optimaux μ, γ et λ des différents algorithmes ainsi que l'observation de leurs sensibilités. L'étude de sensibilité est à chaque fois accompagnée d'une étude de robustesse à travers une multitude de simulations. Toutes les simulations d'apprentissage de paramètres ou d'études de robustesse représentent des moyennes sur 20 réalisations différentes.

3.5.1 Apprentissage partiel des paramètres : scénario d'objets à trajectoires confondues

Le premier scénario donné par la figure 3.7(a) représente un cas de suivi conflictuel réduit à deux objets. La figure 3.7(b) mesure le taux de fausses décisions pouvant être commises sur un tel scénario : *Taux de fausses décisions = (nombre de fois où les observations sont confondues + nombre de fois où les observations sont associées à de nouveaux objets à tort)/nombre de décisions totales.*

Les remarques pouvant être faites sur les résultats donnés par la figure 3.7(b) sont les suivantes :

- le taux de fausses décisions minimum dépend du rapprochement des deux objets ce qui correspondrait au nombre de fois où les observations sont confondues. Le rapprochement étant constant durant toute la durée du suivi, on admet que l'augmentation des erreurs de



(a) Scénario à deux trajectoires conflictuelles.

(b) Taux de fausses décisions.

FIGURE 3.7: Influence des paramètres μ, γ et λ sur les performances des algorithmes pour le suivi de deux trajectoires conflictuelles.

décisions, avec la variation des paramètres μ, γ et λ, est dû à l'association des observations des deux objets existants à de nouveaux objets.

- l'évolution des taux de fausses décisions des algorithmes dépendant du paramètre γ et des algorithmes dépendant du paramètre λ est similaire, cela est dû au fait que γ est pris égal à $-log(0.5)/\lambda$ où λ est la distance d'apparition de l'algorithme *GNN*. Il peut être remarqué qu'une augmentation du paramètre γ, correspondant à une diminution de la distance d'apparition λ, ce qui engendre une augmentation des erreurs de décision associant les observations des deux objets à de nouveaux objets. Cette remarque confirme la relation établie entre les paramètres γ et λ dans l'équation (3.46).

- il peut être remarqué également que la solution proposée 1 est également dépendante du paramètre μ, toutefois, elle est nettement moins sensible que les solutions dépendant du paramètre λ ou γ.

- la solution proposée 2 n'atteint pas la performance optimale qui est dans cette simulation à peu près égale 5% de fausses décisions, toutefois elle présente une certaine indépendance par rapport au paramètre μ utilisé pour le transfert d'information dans les équations (3.44) et (3.45).

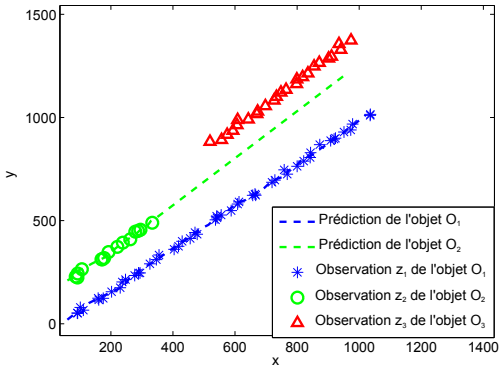
- étant un scénario conflictuel (trajectoires confondues), le conflit généré à la combinaison des fonctions de masse dans les méthodes crédales proposées par Mercier et *al*, Lauffenburger et *al* est important. N'ayant pas était normalisé comme dans la méthode de Mercier et *al*, mais utilisé pour la prise de décision dans la méthode de Lauffenburger et *al*, ce conflit conduit à un taux de fausses décisions très important comme le montre la figure 3.7(b).

- à noter que la solution proposée par Dallil et *al* présente un taux de fausses décisions optimal pour le scénario 3.7(a). Cette solution ne gère pas l'apparition d'objets, donc elle ne risque pas d'associer les observations des objets existants à de nouveaux objets.

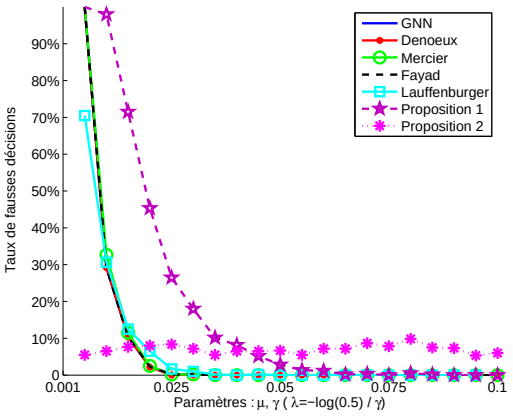
- la simulation présentée dans cette section incite à choisir des paramètres μ et γ très petits pour assurer une performance optimale. La simulation présentée dans la section suivante va à l'encontre de cette conclusion, elle montre que le choix de paramètres μ et γ très petits conduit à un autre type de fausses décisions.

3.5.2 Apprentissage partiel des paramètres : gestion de la réapparition d'objets

Cette section présente un scénario particulier concernant la gestion de la réapparition d'objets. En effet, comme illustré par la figure 3.8(a), l'objet O_2 a cessé d'être détecté, sa trajectoire continue toutefois à être prédite par l'algorithme *IMM*. À un instant donné, un nouvel objet O_3 apparaît a proximité de la trajectoire prédite de l'objet O_2. Le but de cette simulation est de calculer le nombre de fois où les observations du nouvel objet O_3 sont associées à tort à l'objet O_2. Dans cette simulation : le *Taux de fausses décisions = nombre de fois où les observations de l'objet O_3 sont associées aux prédictions de l'objet O_2/nombre d'associations possibles des observations de l'objet O_3.*



(a) Scenario illustrant la disparition et l'apparition d'objets.



(b) Taux de fausses décisions.

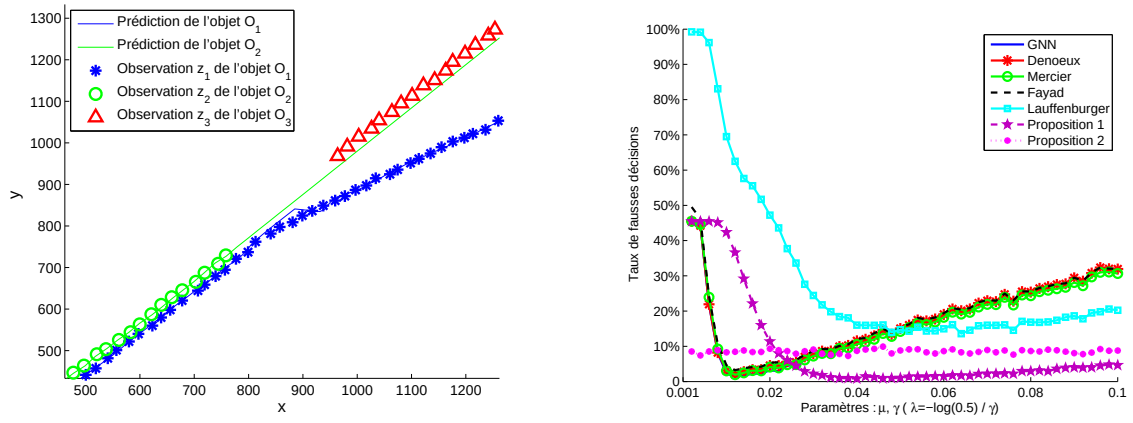
FIGURE 3.8: Influence des paramètres μ, γ et λ quant à la gestion de la disparition, la réapparition et l'apparition d'objets.

Comme attendu, les résultats présentés par la figure 3.8(b) remettent en question le choix de petites valeurs pour les paramètres μ et γ, ce qui correspond au choix de larges distances d'apparition λ.

En effet, les résultats montrent qu'un tel choix induit en erreur les algorithmes d'association, tel que les observations z_3 du nouvel objet O_3 sont par erreur, associées aux prédictions de l'objet O_2. Bien qu'il soit rare d'être confronté à ce type d'erreurs dans les systèmes de suivi, il faut en tenir compte dans l'apprentissage des paramètres. Il peut être remarqué que l'algorithme proposé 1 est un peu plus sensible que les autres algorithmes à ce type d'erreurs. La solution proposée 2 est quant à elle à performance constante sans pour autant atteindre l'optimalité qui est dans ce cas de 0% d'erreurs.

3.5.3 Apprentissage complet des paramètres : objets à trajectoires confondues et gestion de réapparition

La simulation présentée dans cette section tient compte des deux types d'erreurs considérées dans les sections 3.5.1 et 3.5.2. Elle vise à apprendre finement les paramètres des différents algorithmes. La performance est mesurée ainsi : *Taux de fausses décisions = (nombre de fois où les observations des objets O_1 et O_2 sont confondues + nombre de fois où les observations des objets existants O_1 et O_2 sont associées à de nouveaux objets + nombre de fois où les observations du nouvel objet O_3 sont associées aux prédictions de l'objet O_2)/ nombre de décisions totales.*



(a) Scenario conflictuel avec apparition et disparition d'objets.

(b) Taux de fausses décisions.

FIGURE 3.9: Apprentissage des valeurs optimales des paramètres μ et γ.

Les résultats présentés par la figure 3.9(b) montrent qu'en effet les paramètres μ, γ doivent être plus finement appris afin de permettre une meilleure performance pour les algorithmes correspondants. Certains algorithmes atteignent une performance optimale qui est dans cette simulation à peu près

égale à 2%. À noter que la performance optimale dépend des paramètres du système, notamment la distance séparant les trajectoires de l'objet O_1 et l'objet O_2 par exemple.

Discussion des résultats présentés par la figure 3.9(b) : les résultats présentés par la figure 3.9(b) et illustrés par la figure 3.10 montrent l'apprentissage des paramètres μ et γ correspondant au scénario de la figure 3.9(a). Un certain nombre de remarques en découlent :

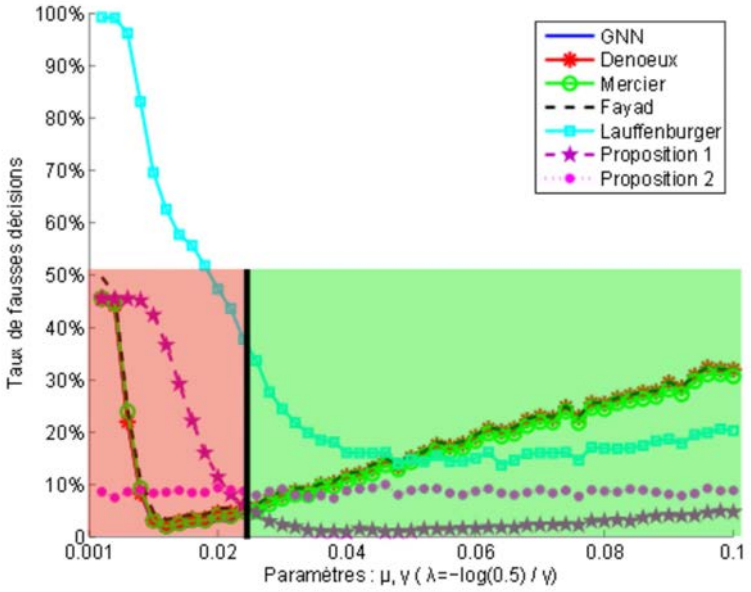


FIGURE 3.10: Influence des paramètres μ, γ et λ sur les performances des différents algorithmes (zone rouge : zone défavorable à la méthode proposée 1, zone verte : zone favorable à la méthode proposée 1).

- la même variation est adoptée pour le paramètre μ et le paramètre γ ($\lambda = -log(0.5)/\gamma$), c'est une variation entre 0 et 0.1, c'est la zone où les algorithmes d'association présentent les meilleures résultats concernant le système de la figure 3.9(a).

- certains algorithmes d'association présentent des performances optimales pour certaines valeurs de leurs paramètres respectifs. C'est le cas notamment de l'algorithme proposé par Denœux et *al*, Mercier et *al*, l'algorithme de Fayad et Hamadeh ainsi que l'algorithme *GNN*, ces algorithmes ont une performance quasi-similaire ce qui confirme à nouveau la relation (3.46).

- la solution proposée 1 atteint également une performance optimale et ce pour une variation du paramètre μ plus large comparée à la variation de la zone d'optimalité du paramètre γ concernant les autres algorithmes crédaux. À en déduire que la solution proposée 1 est beaucoup moins sensible à la variation de son paramètre, comme le montre la figure 3.10.

- on remarque toutefois que la zone de variation des paramètres γ et μ se divise en deux zones représentées par deux couleurs différentes dans la figure 3.10. Dans la zone rouge où les erreurs concernent la gestion de réapparition (voir la figure 3.8), la solution proposée 1 se trouve être plus sensible. Un choix hasardeux des paramètres μ et γ dans cette zone pour le système 3.9(a) va en défaveur de la solution proposée 1. Cependant, un choix hasardeux des paramètres μ et γ dans la zone verte permet à la solution proposée 1 d'être bien plus robuste que les autres solutions d'association.

- à remarquer que la solution proposée 2 n'atteint pas la performance optimale mais elle présente une certaine constance par rapport à la variation du paramètre μ ce qui peut être intéressant dans certaines situations. En l'occurrence, les situations où l'apprentissage paramétrique ne peut pas être effectué.

- la non optimalité de la solution proposée 2 serait due au fait que l'information de base (distances) est modifiée à un plus haut niveau, notamment au calcul des plausibilités, contrairement à la solution proposée 1 ou la solution proposée par Denœux et *al* qui modifient directement les distances on les multipliant par les coefficients μ et γ respectivement. À noter que les paramètres μ et γ ont une influence linéaire sur les distances de sorte à ce que, si : $\textit{distance 1} < \textit{distance 2}$ alors : $\textit{paramètre} \times \textit{distance 1} < \textit{paramètre} \times \textit{distance 2}$.

Les études relevant du domaine du suivi d'objets ont rarement abordé le type d'erreurs concernant la réapparition d'objets [16, 23]. Cette situation se trouve être particulière et rare dans les systèmes de suivi. De ce fait, la zone rouge illustrée dans la figure 3.10 est susceptible d'être très mince, voir inexistante dans les situations réelles du suivi d'objets et cela va évidemment au profit de la solution proposée 1. L'utilisation de cette dernière solution est alors recommandée lorsque les paramètres optimaux sont difficiles ou impossibles à définir.

3.5.3.1 Tests de robustesse

Cette section propose un test de robustesse, par rapport aux erreurs de mesure, sur le scénario de la figure 3.9(a) en utilisant les paramètres optimaux appris dans la figure 3.9(b). Les paramètres optimaux sont ceux qui donnent des taux de fausses décisions minimaux. Les résultats du test sont présentés par la figure 3.11.

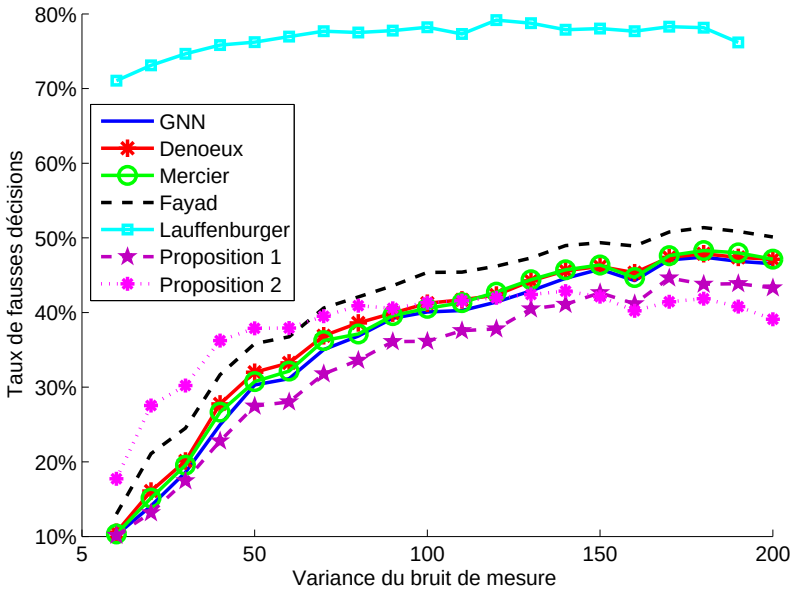


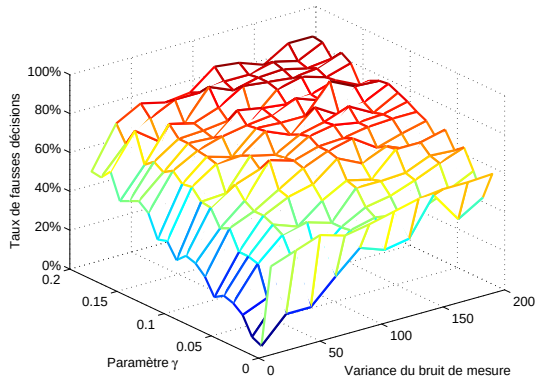
FIGURE 3.11: Robustesse par rapport aux erreurs de mesure.

Il est à remarquer que l'augmentation des erreurs de mesure, modélisée dans la simulation par une augmentation de la variance du bruit de mesure, engendre une augmentation dans les taux de fausses décisions de tous les algorithmes testés, comme le montre la figure 3.11. Il est à remarquer également que la solution proposée 1 présente une meilleure robustesse, et la solution proposée 2 se voit moins robuste pour les variances de bruit nominales, ce qui est dû au fait que c'est une solution non optimale, mais elle présente une meilleure robustesse pour des variances de bruit extrêmes, très supérieures aux variances de bruit nominales qui ont aidé à l'apprentissage des paramètres.

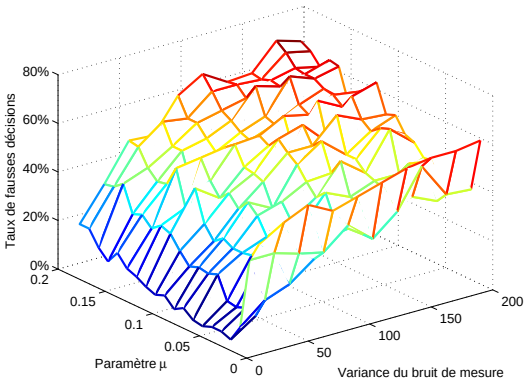
Afin de confirmer la robustesse par rapport à l'apprentissage paramétrique ainsi qu'aux erreurs de mesure de la solution proposée 1, un autre test est effectué, il compare la méthode proposée 1 avec la méthode proposée par Denœux et *al.* Les résultats obtenus sont portés sur les figures 3.12(a) et 3.12(b) respectivement pour la solution proposée par Denœux et *al* et la solution proposée 1.

Le taux de fausses décisions moyen présenté par la figure 3.12(b) est inférieur au taux de fausses décisions moyen présentés par la figure 3.12(a), cela confirme l'avantage de la solution proposée 1 et sa robustesse par rapport à l'apprentissage paramétrique et par rapport aux erreurs de mesure.

La figure 3.13 représente la différence entre le taux de fausses décisions de la méthode proposée 1 et le taux de fausses décisions de la méthode proposée par Denœud et *al.* Il peut être remarqué que ce test confirme les résultats obtenu précédemment, notamment le fait que la méthode proposée



(a) Robustesse au paramètre γ et au bruit de mesure de la solution de Denœux et *al.*



(b) Robustesse au paramètre μ et au bruit de mesure de la solution proposée 1.

FIGURE 3.12: Test de robustesse combiné comparant la solution proposée 1 à la solution proposée par Denœux et *al.*

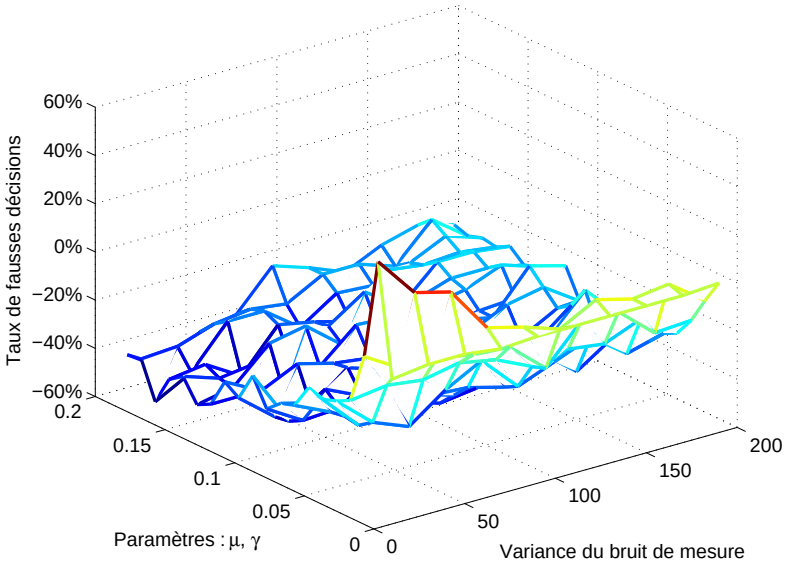
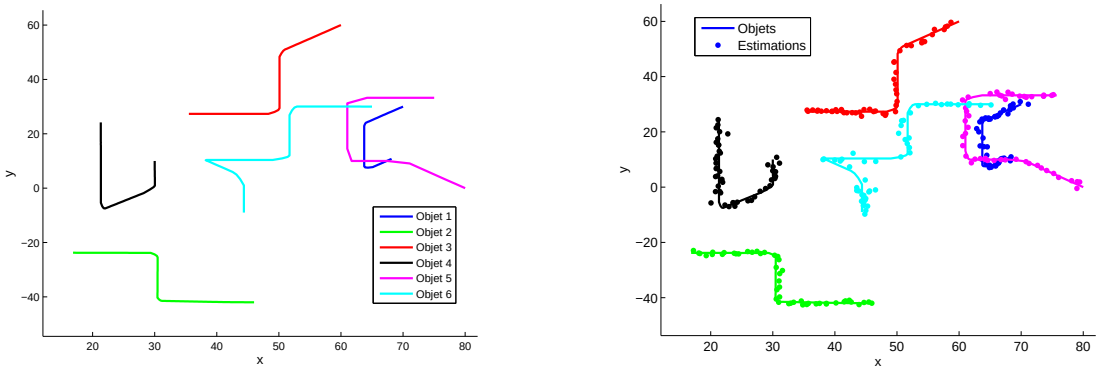


FIGURE 3.13: Différence entre les taux de fausses décisions présentés par les figures 3.12(b) et 3.12(a) respectivement.

soit moins robuste pour un choix de paramètres dans la zone rouge représentée dans la figure 3.10 mais bien plus pour tous les autres choix de paramètres.

3.5.4 Test sur des données de simulation issues de la littérature [24, 25]

Combinés à des algorithmes *IMM*, trois algorithmes d'association sont testés sur un scénario de suivi de six trajectoires d'objets manœuvrants. La simulation concerne un scénario de référence (*benchmark*) dédié au tests des algorithmes de suivi d'objets. Le scénario est initialement proposé dans [24, 25] pour essentiellement tester les méthodes du type multi-scan d'où la simulation de fausses alarmes. Ces dernières ne sont pas intégrées dans la simulation réalisée dans cette section dans la mesure où les *IMMs* sont combinés à des méthodes d'association mono-scan. L'intérêt est porté sur la capacité à estimer les trajectoires proposées qui sont illustrées par la figure 3.14(a). À noter que les trajectoires sont initialement proposées en trois dimensions [24, 25], dans cette section nous ne considérons que leur projection sur le plan (x, y).



(a) Trajectoires d'objets manœuvrants.

(b) Estimation des trajectoires présentées par la figure 3.14(a).

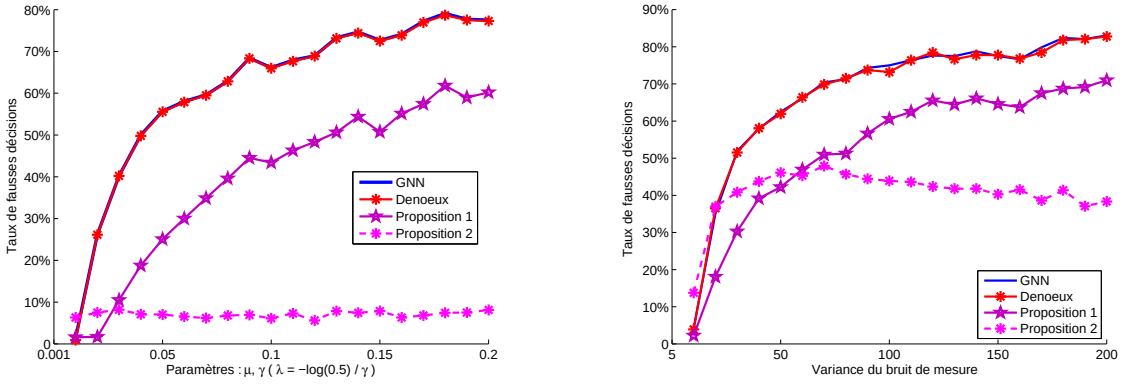
FIGURE 3.14: Application des méthodes d'association au suivi d'objets manœuvrants.

La figure 3.14(b) montre les trajectoires des six objets ainsi que les trajectoires estimées. Ce sont les estimations obtenues à l'aide des algorithmes *IMMs* combinés avec l'un des algorithmes d'association proposés, l'algorithme proposé par Denœux et *al* ou l'algorithme *GNN* avec des paramètres fixés adéquatement pour que les trajectoires puissent être suivies correctement en premier temps.

3.5.4.1 Apprentissage des paramètres et test de robustesse pour le scénario "*benchmark*"

Afin d'apprendre les paramètres optimaux permettant d'assurer un suivi correct des trajectoires, une simulation faisant varier les paramètres μ, γ est effectuée. Le résultat de l'apprentissage est

donné par la figure 3.15(b). Il peut y être remarqué que les paramètres optimaux sont les plus petites valeurs de μ et γ. Il peut être remarqué également que la méthode proposée 1 est moins sensible à la variation de son paramètre que la solution proposée par Denœux et *al* et l'algorithme *GNN*, ce qui confirme les résultats obtenus dans la section précédente. La solution proposée 2 présente une indifférence par rapport à la variation du paramètre μ, toutefois, contrairement aux autres méthodes, elle n'atteint pas l'optimalité.



(a) Apprentissage des paramètres μ et γ pour le scénario de la figure 3.14(a).

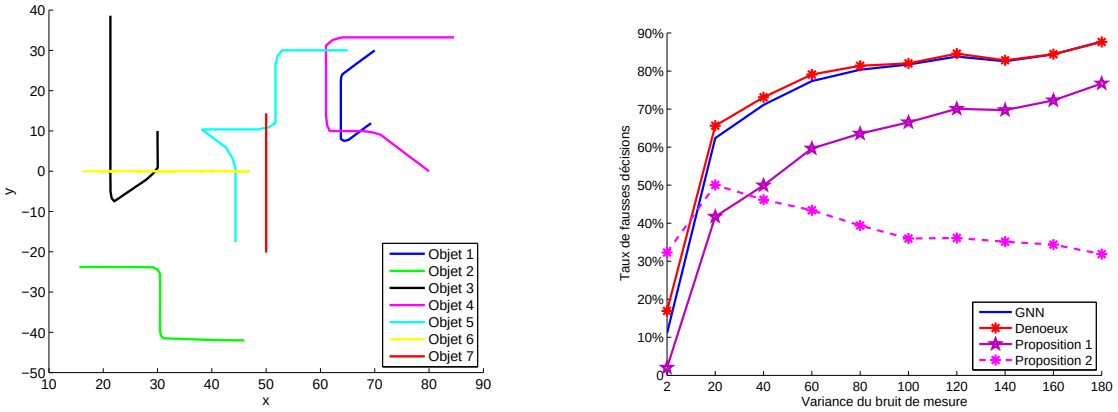
(b) Test de robustesse utilisant les paramètres μ et γ optimaux.

FIGURE 3.15: Apprentissage de paramètres et test de robustesse correspondant au scénario de la figure 3.14(a).

En utilisant les paramètres optimaux appris dans la figure 3.15(b), un test de robustesse consistant à faire varier les erreurs de mesure est réalisé. Le résultat obtenu est donné par la figure 3.15(b), il confirme que dans le cas où les variances de bruit ne sont pas apprises, la solution proposée 1 est plus robuste que les solutions optimales *GNN* et de Denœux et *al*. La figure confirme également l'aspect non optimale de la solution proposée 2 pour des variances de bruit nominales (proches de celles qui ont permis l'apprentissage des paramètres), mais elle présente une meilleure performance que toutes les autres solutions dans le cas de variances de bruit extrêmes non apprises.

3.5.4.2 Test de robustesse sans apprentissage de paramètres

Cette section reprend les paramètres appris pour le scénario 3.14(a) dans la figure 3.15(b) et les utilisent pour réaliser un second test de robustesse sur le scénario donné par la figure 3.16(a) qui est légèrement différent du scénario 3.14(a) de par le nombre d'objets dans la scène par exemple. Dans la réalité, le nombre d'objets ne peut pas être connu *a priori*.



(a) Scénario de 7 cibles.

(b) Taux de fausses décisions.

FIGURE 3.16: Test sur un scénario dont les paramètres ne sont pas appris.

Les résultats donnés par la figure 3.16(b) confirment les conclusions faites par rapport à la robustesse des deux solutions proposées. Les résultats montrent même que les algorithmes d'association perdent leur optimalité même pour de basses variances de bruit, là où il peut être remarqué que la solution proposée 1 présente le meilleur taux de fausses décisions. Cela peut être un avantage pour des applications dont l'environnement de suivi est inconnu.

3.5.4.3 Impacts des fausses décisions sur le suivi de trajectoires

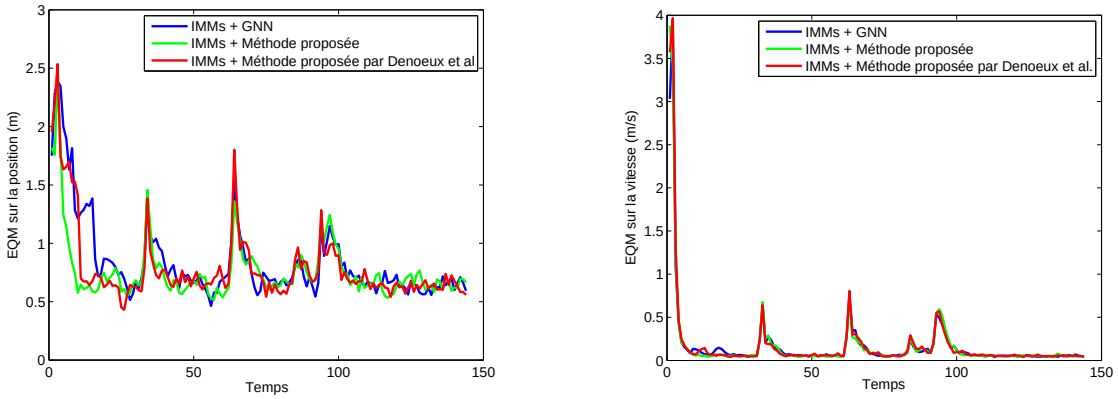
Quand les paramètres des algorithmes d'association sont optimalement appris, l'Erreur Quadratique Moyenne EQM de la position concernant l'estimation des trajectoires présentées par la figure 3.14(a) est donnée par la figure 3.17(a) et l'EQM sur l'estimation de la vitesse des objets est donnée par la figure 3.17(b). L'EQM sur l'estimation de la position est calculée ainsi : $(H_p\ \hat{x} - H_p\ x)' \times (H_p\ \hat{x} - H_p\ x)$ où x est le vecteur d'état réel et $\hat{x}$ son estimation avec H_p donnée par :

$$H_p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

L'EQM sur l'estimation de la vitesse est calculée de la même manière mais en remplaçant la matrice H_p par la matrice H_v :

$$H_v = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Les algorithmes d'estimation sont simulés d'une manière indépendante sur 20 réalisations de bruit de mesure différentes. Les résultats de la figure 3.17 représentent alors une moyenne sur 20 simulations et sur les six trajectoires. Les *EQMs* présentées par les figures 3.17(a) et 3.17(b) correspondent à des algorithmes d'association bien calibrés combinés à des algorithmes *IMMs*.



(a) *EQM* sur l'estimation de la position.

(b) *EQM* sur l'estimation de la vitesse.

FIGURE 3.17: Algorithmes d'association et suivi des trajectoires données par la figure 3.14(a).

Les résultats présentés par la figure 3.17 montrent que les performances des algorithmes sont équivalentes. Il peut également être remarqué que l'*EQM* de position et de vitesse sont plus importantes lors des manœuvres, ce qui peut surprendre les algorithmes d'association. En effet, cela peut conduire à une erreur de décision au niveau des algorithmes d'association ainsi à la perte d'une ou de plusieurs trajectoires suivies comme illustré par la figure 3.18.

Suite à la perte d'une ou de plusieurs trajectoires, l'*EQM* de position diverge comme le montre la figure 3.19(a) : les positions estimées sont loins des trajectoires réelles. Par contre, il peut ne pas y avoir une grande différence entre les vitesses réelles et les vitesses estimées, comme le montre la figure 3.19(b) illustrant l'*EQM* faite sur l'estimation de la vitesse.

À noter que pour la simulation dont les résultats sont donnés par la figure 3.19, les algorithmes sont simulés sur la même réalisation de bruit d'où la superposition des courbes. L'intérêt est de tester les algorithmes sur exactement le même scénario tel qu'il est donné par la figure 3.18, avec les mêmes erreurs de mesure (même réalisation de bruit).

La perte de trajectoires illustrée par la figure 3.18 engendrant les erreurs d'estimation données par la figure 3.19 est ce qu'il peut être obtenu lorsque les algorithmes d'association ne sont pas bien calibrés. Cependant, l'algorithme d'association proposé 1 est à peu près 20% moins exposé à ce

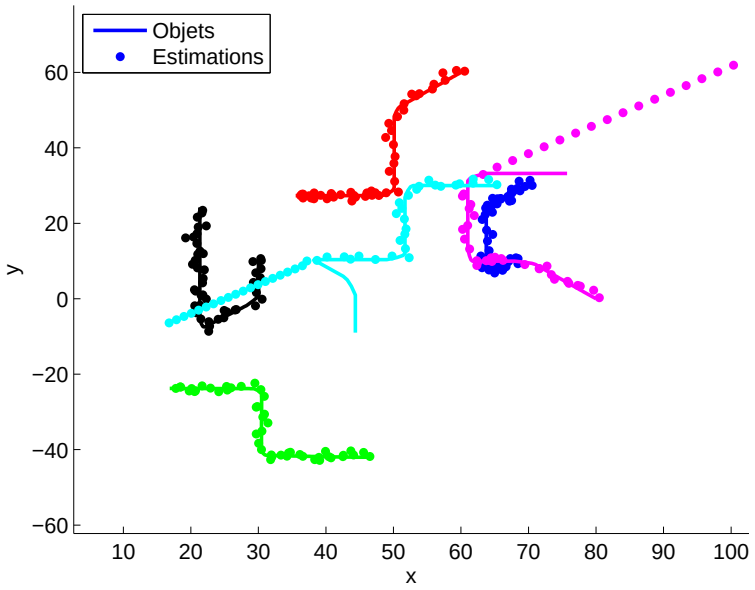
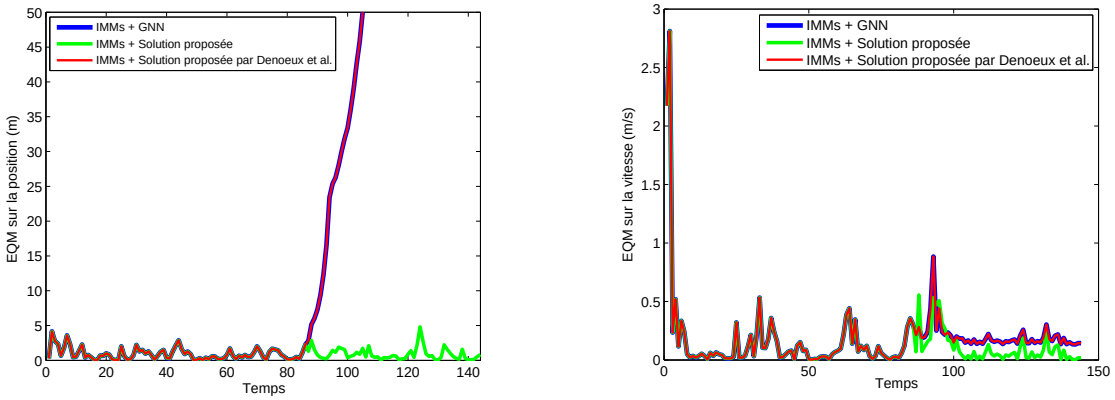


FIGURE 3.18: Estimation des trajectoires données par la figure 3.14(a) (cas d'une mauvaise calibration de paramètres).



(a) EQM sur l'estimation de la position.

(b) EQM sur l'estimation de la vitesse.

FIGURE 3.19: *EQM* sur l'estimation de trajectoires avec une mauvaise calibration des algorithmes d'association.

problème comme le montrent les tests de robustesse donnés par les figures 3.15(b) et 3.16(b). Le gain est plus important pour la solution proposée 2, mais seulement pour des variances de bruit extrêmes.

Les tests effectués dans cette section justifient la robustesse avérée des solutions proposées. Toutefois, il se trouve qu'elles soient moins performantes en terme de capacité calculatoire lorsqu'on les compare à la solution proposée par Denœux et *al* et l'algorithme *GNN*.

3.5.4.4 Comparaison des algorithmes en terme de complexité calculatoire

Cette section donne une idée quant à la complexité calculatoire des algorithmes étudiés dans ce chapitre. Les temps d'exécution des algorithmes pour des problèmes d'association réduits, dont le nombre d'objets est croissant, sont donnés par la figure 3.20.

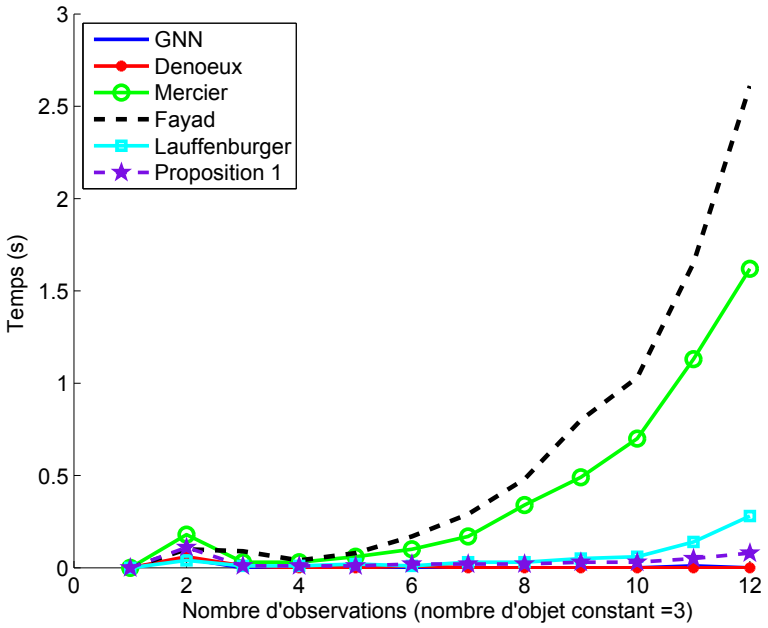


FIGURE 3.20: Test de complexité calculatoire.

La figure 3.20 montre que l'algorithme *GNN* et l'algorithme de Denœux et *al* présentent les temps de calcul les plus faibles, ils sont négligeables pour les instances présentées. Le temps de calcul est un peu plus important pour la solution proposée 1. La résolution du problème d'association à l'aide de l'algorithme de Munkres, qui est de complexité polynomiale, fait que ces solutions soient parmi les moins complexes et ont d'appréciables facultés calculatoires. Il peut être remarqué que l'algorithme proposé par Fayad et Hamadeh est le plus complexe. Le tableau 3.2 donne une idée sur la complexité de calcul des algorithmes pour des instances plus grandes que celles considérées dans la figure 3.20.

Les simulations présentées dans cette section donnent une idée de la complexité calculatoire des algorithmes étudiés : l'algorithme *GNN* et l'algorithme de Denœux et *al* sont les moins complexes en terme du temps de calcul, ils auraient une complexité qui se réduit à la complexité de l'algorithme de Munkres qui est de l'ordre $O(n^3)$ [130]. En plus de la complexité de résolution à l'aide de l'algorithme de Munkres, la complexité du *GBT* s'ajoute pour la solution proposée, elle serait

TABLEAU 3.2: Temps de calcul en secondes des algorithmes pour des problèmes d'association de dimension $n \times n$.

	GNN	Denœux et *al*	Proposition	Lauffenburger et *al*	Mercier et *al*	Fayad et *al*
$n = 4$	0.039	0.058	0.172	0.114	0.243	0.285
$n = 7$	0.041	0.061	0.225	0.175	643.965	683.966
$n = 10$	0.047	0.068	0.286	0.604	$+2h$	$+2h$
$n = 13$	0.051	0.070	0.808	4.907	$+2h$	$+2h$
$n = 16$	0.055	0.071	6.863	85.423	$+2h$	$+2h$
$n = 19$	0.05	0.072	58.776	$\approx 1h$	$+2h$	$+2h$
$n = 22$	0.058	0.075	689.101	$+2h$	$+2h$	$+2h$
$n = 25$	0.063	0.076	$\approx 1h$	$+2h$	$+2h$	$+2h$

polynomiale d'ordre 4 vu le nombre d'opérations à effectuer dans l'équation (3.37). La complexité de la solution proposée serait alors de l'ordre $O(n^4)+O(n^3) \approx O(n^4)$. D'après les résultats obtenus dans cette section, on pourrait déduire que la complexité de la solution proposée par Lauffenburger et *al* soit polynomiale d'ordre supérieur à 4 et que les solutions proposées par Mercier et *al* et Fayad et Hamadeh auraient une complexité exponentielle.

3.6 Conclusion

Ce chapitre décrit quelques algorithmes d'association basés sur les fonctions de croyance. Ces algorithmes accordent, à chaque pas de temps et d'une manière explicite, une hypothèse bien définie à chaque observation détectée ou objet connu. Ils sont donc mono-scan et déterministes dont la logique est équivalente à celle de l'algorithme *GNN*. Une description élaborée des différents algorithmes est fournie. En plus de la description des algorithmes existants, la description d'une nouvelle méthode à deux variantes est rajoutée. L'étude de ce chapitre vise, à observer l'influence des paramètres sur les performances des différents algorithmes, ce pour un problème d'association mono-capteur. Il a été observé que les paramètres des algorithmes sont en relation directe avec l'apparition de nouveaux objets au cours du suivi. En effet, les paramètres des algorithmes aident à modéliser la distance d'apparition de telle sorte que si une observation donnée est détectée au delà de cette distance par rapport aux objets existants, elle est considérée comme étant un nouvel objet. À noter toutefois que la solution proposée par Dallil et *al* ne dépend pas de paramètres, mais ne gère pas l'apparition d'objets également.

À l'issue de l'étude effectuée dans ce chapitre, deux conclusions importantes peuvent être faites :

- les solutions d'associations proposées présentent de meilleures performances quant à leur application dans des environnements incertains. Notamment le cas où il est difficile ou impossible de calibrer les algorithmes et le cas où les erreurs de mesure sont supérieures à celles qui ont aidé à la calibration des algorithmes ce qui peut en réalité correspondre à une calibration faite à l'aide d'un capteur neuf. Cependant, avec le temps, la fiabilité du capteur diminue ainsi, à défaut d'en être robuste à cette situation comme le cas de la solution proposée ou plus, une nouvelle calibration s'impose.

- comme le montre l'étude de complexité, l'utilisation de la solution proposée pour des applications en temps réel peut être compromise par le nombre d'objets dans la scène. La solution est significativement lente lorsque le nombre d'objets suivis dépasse 20.

Chapitre 4

Systèmes multi-capteurs pour le suivi et la classification d'objets

Les travaux présentés dans les chapitres 2 et 3 sont dédiés au suivi et à la classification d'objets multiples dans un cadre mono-capteur. Peu importe l'efficacité des algorithmes de suivi et/ou de classification considérés, la performance de ces derniers reste conditionnée par la fiabilité du capteur percevant les objets. Une fiabilité parfaite n'étant jamais assurée, il est souvent nécessaire de recourir à des approches multi-capteurs. Des stratégies de fusions peuvent alors corriger et/ou compléter les informations des capteurs dont la fiabilité et/ou l'observabilité sont limitées.

Après avoir donné un bref aperçu des stratégies multi-capteurs pour le suivi d'objets, ce chapitre propose deux solutions généralisant les algorithmes de suivi et de classification présentés dans le chapitre 2 au cadre multi-capteurs.

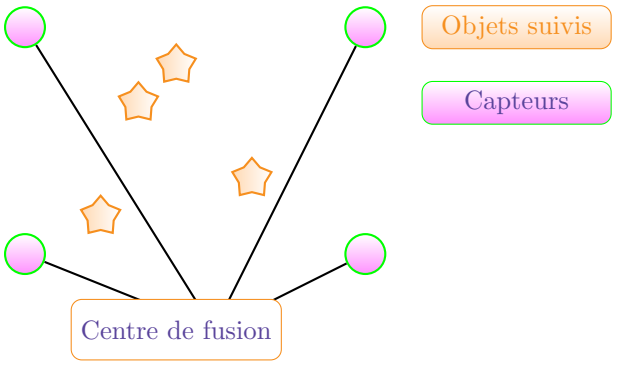


FIGURE 4.1: Illustration d'une solution multi-capteurs centralisée.

La première solution multi-capteurs proposée est dite centralisée, elle reprend les mêmes algorithmes de suivi et de classification d'objets présentés dans le chapitre 2 dans un contexte de capteurs non fiables. L'objectif est alors d'acheminer les classifications locales des capteurs à un centre de fusion en vue de construire

une classification globale de meilleure fiabilité. Les deux types de classifications Bayésienne et crédale sont considérés et plusieurs règles de fusions sont testées. Cette architecture centralisée a permis également de tester l'algorithme d'association proposé dans la section 3.3.6 dans un cadre multi-capteurs. En effet, il est nécessaire que les capteurs aient un consensus sur les objets suivis pour ainsi fusionner leurs classifications locales. Une illustration simplifiée d'une architecture multi-capteurs centralisée est donnée par la figure 4.1.

Pour des systèmes nécessitant une multitude de capteurs, les approches de fusions centralisées sont toutefois jugées limitées de par leur non extensibilité [139, 150]. En effet, si on se propose de surveiller une zone spatialement large, d'autres capteurs doivent être rajoutés au système de surveillance. Dès lors, la communication de ces derniers avec le centre de fusion devient très coûteuse, voir irréalisable. Il est donc nécessaire de décentraliser le système de surveillance en question.

La seconde partie de ce chapitre propose une solution de suivi et de classification distribuée, une solution où les capteurs n'observent que partiellement l'état des objets et n'ont pas à communiquer avec un quelconque centre de fusion pour construire l'information complète. En effet, chaque capteur dans cette solution ne communique qu'avec les autres capteurs qui lui sont immédiatement voisins. Les communications entre capteurs visent à faire propager des informations incomplètes dans le réseau de capteurs et ainsi construire une information complète au niveau de chaque capteur. Dans la solution proposée dans la section 4.3, cela est rendu possible par l'intégration d'un algorithme de consensus dans les phases de suivi et de classification présentées dans le chapitre 2. Cette solution ne fait pas intervenir des fonctions de croyance, seule le classifieur Bayésien est considéré. Une illustration simplifiée d'une solution de suivi et de classification multi-capteurs distribuée est donnée par la figure 4.2.

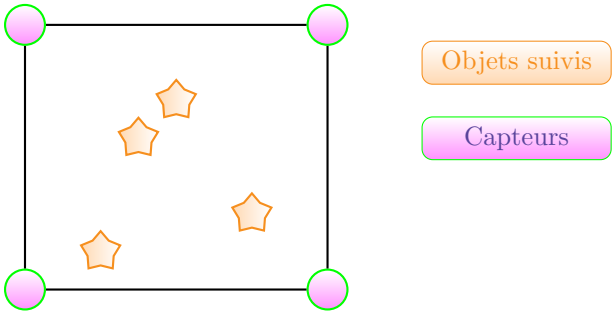


FIGURE 4.2: Illustration d'une solution multi-capteurs distribuée.

Le reste de ce chapitre est principalement divisé en quatre sections. La section 4.1 donne un aperçu de la littérature traitant du suivi et de la classification multi-capteurs. La section 4.2 détaille la solution multi-capteurs centralisée proposée et la section 4.3 présente la solution distribuée proposée. Quelques conclusions et perspectives sont données dans la section 4.4.

4.1 Aperçu de l'état de l'art concernant le suivi multi-objets et multi-capteurs

Cette section présente un aperçu des techniques multi-capteurs dédiées au suivi et/ou à la classification d'objets présentées dans la littérature [8, 11, 127, 150]. Principalement, trois catégories d'approches multi-capteurs peuvent être distinguées :

- L'approche centralisée : la plupart des approches multi-capteurs dédiées au suivi d'objets sont centralisées [11, 72, 110, 127], tout comme la solution proposée dans la section 4.2. Les approches centralisées sont en général limitées à la surveillance de zones peu étendues. La surveillance de zones étendues nécessite le déploiement de capteurs supplémentaires dont les communications avec le centre de fusion deviennent coûteuses, voir impossibles. De plus le fonctionnement des approches centralisées dépend du fonctionnement du centre de fusion. Une panne dans ce dernier engendre une paralysie totale du système de survaillance.

- L'approche décentralisée : les approches décentralisées ne dépendent pas de centre de fusion, chaque capteur représente un centre de fusion. Si un capteur tombe en panne le système de surveillance continue sa fonction avec les capteurs restants. Toutefois, les approches décentralisées nécessitent des réseaux complètement connectés : chaque capteur communique avec tous les autres capteurs du réseau. Par conséquent, ces approches sont également limitées à la surveillance de zones peu étendues à cause des coûts ou l'irréalisabilité des communications entre capteurs. Quelques exemples d'approches décentralisées appliquées au suivi d'objets peuvent être trouvées dans [35, 128, 150].

- L'approche distribuée : les approches distribuées sont proposées afin d'éviter les limites des approches centralisées et décentralisées. Cette solution ne dépend pas de centre de fusion et les capteurs ne communiquent qu'avec un nombre réduit de capteurs dans le réseau, souvent les plus proches, ce qui assure une extensibilité du système de surveillance. Quelques approches de fusions distribuées peuvent être trouvées dans [29, 39, 107, 138, 161]. Peu d'entre elles sont dédiées au suivi d'objets. Ce chapitre propose une solution basée sur le filtrage de Kalman distribué dans la section 4.3.

Comme cela est évoqué, la plupart des approches multi-capteurs dédiées au suivi multi-objets sont centralisées et visent à fusionner les estimations faites par différents capteurs [4, 8, 12–14, 92]. Dans

le domaine du suivi d'objets, la classification est un problème subordonné induit par le problème du suivi dans la mesure où la classification est généralement effectuée à l'aide des données estimées. Une illustration concernant la fusion des estimations $\hat{x}_1$ et $\hat{x}_2$ provenant respectivement de deux capteurs 1 et 2 tel qu'elle est adoptée dans [8] peut se présenter ainsi :

$$\hat{x} = \hat{x}_1 + \alpha(\hat{x}_1 + \hat{x}_2), \tag{4.1}$$

avec une erreur d'estimation globale donnée par :

$$e_{\hat{x}} = e_{\hat{x}_1} + \alpha(e_{\hat{x}_1} + e_{\hat{x}_2}). \tag{4.2}$$

L'objectif de la fusion étant de choisir le paramètre α tel que l'erreur d'estimation globale $e_{\hat{x}}$ soit minimale [8, 9]. En terme de variance d'erreur d'estimation, le problème de fusion est donné ainsi :

$$\sigma^2 = E\{e_{\hat{x}}^2\} = \sigma_1^2 + 2\alpha E\{e_{\hat{x}_1} e_{\hat{x}_2}\} + 2\alpha\sigma_1^2 + \alpha^2 E\{(e_{\hat{x}_1} e_{\hat{x}_2})^2\}, \tag{4.3}$$

où

$$\sigma_1^2 = E\{e_{\hat{x}_1}^2\},\ \sigma_2^2 = E\{e_{\hat{x}_2}^2\}, \tag{4.4}$$

avec l'opérateur $E\{.\}$ représentant l'espérance mathématique.

Les termes $E\{e_{\hat{x}_1} e_{\hat{x}_2}\}$ et $E\{(e_{\hat{x}_1} e_{\hat{x}_2})^2\}$ représentent les inter-correlations entre les deux estimations $\hat{x}_1$ et $\hat{x}_2$ des capteurs 1 et 2. Dans le cas où les estimations sont supposées complètement indépendantes, ce qui revient à négliger les termes (4.4), le paramètre α minimisant l'erreur d'estimation globale 4.2 peut être donné ainsi :

$$\alpha = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}, \tag{4.5}$$

ce qui donne une estimation globale :

$$\hat{x} = \hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}(\hat{x}_2 - \hat{x}_1) = \frac{\sigma_1^2 \hat{x}_2 + \sigma_2^2 \hat{x}_1}{\sigma_1^2 + \sigma_2^2}, \tag{4.6}$$

et une variance globale :

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}. \tag{4.7}$$

Le gain ou la perte d'information dans ce processus de fusion dépend de la valeur du rapport $\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}$ [8, 9, 48]. À noter que la supposition concernant l'indépendance des estimations n'est pas vérifiée dans la mesure où même les estimations faites par des capteurs indépendants peuvent être dépendantes [17, 127] : même si le bruit de mesure est indépendant entre les capteurs, le bruit d'état reste le même, ce qui fait que les estimations faites par les différents capteurs sont dépendantes. Une des solutions tenant compte de cette dépendance entre les estimations propose de réinitialiser à chaque pas de temps les estimations locales des capteurs à l'aides des estimations globales calculées au niveau du centre de fusion [17]. Cette question de dépendance occupe un bon nombre de chercheurs en fusion multi-capteurs pour le suivi d'objets [127, 176]. D'autres recherches dans le domaine s'intéressent à la non synchronisation des estimations provenant des différents capteurs [117, 190].

Dans ce chapitre, l'intérêt principal est porté sur la classification collaborative des objets. La première solution décrite dans la section 4.2, ne fusionne pas les estimations locales des capteurs, elle propose de fusionner de différentes manières les classifications locales. La seconde approches décrites dans la section 4.3 propose de fusionner les observations faites localement au niveau des capteurs ainsi que leurs classifications locales. Dans la seconde approche dite distribuée, la fusion se fait au niveau de chaque capteur, ce à l'aide de l'algorithme de consensus qui est un algorithme de fusion distribué décrit dans la section 4.3.1.2.

4.2 Approche centralisée proposée

La figure 4.3 représente l'organigramme de l'approche de classification multi-capteurs proposée. Les capteurs étant considérés peu fiables, ils assurent un suivi et une classification locale d'un certain nombre d'objets. La solution propose de fusionner les classifications locales des objets qui sont communément suivis afin d'avoir une classification globale de meilleure qualité. Le consensus sur les objets suivis est assuré par une étape d'association appelée également *track-to-track assignment* utilisant l'algorithme proposé dans la section 3.3.6 du chapitre 3.

4.2.1 Suivi local au niveau des capteurs

Chaque capteur exécute l'algorithme 2.1, décrit dans le chapitre 2, afin d'assurer le suivi de plusieurs objets à la fois. À chaque pas de temps k, chaque capteur $s \in \{1, 2, ..., S\}$ fournit un rapport

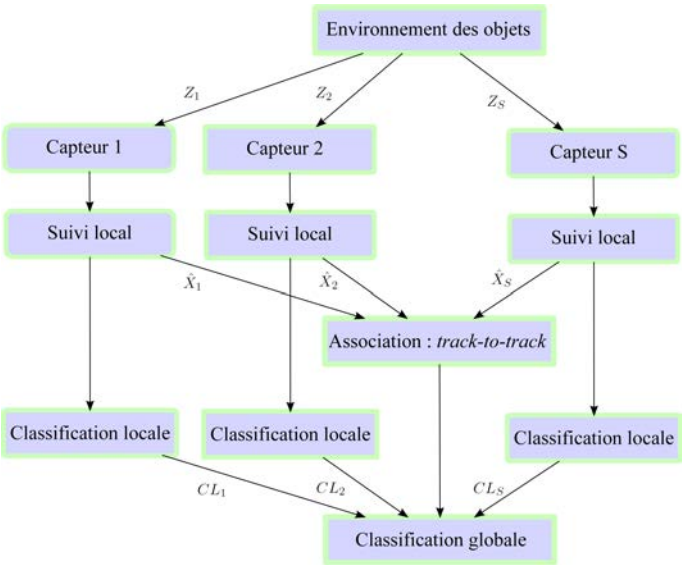


FIGURE 4.3: Organigramme de la solution multi-capteurs centralisée proposée.

$\hat{X}_s = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_{n_s}\}$ contenant les estimations des n_s objets suivis par le capteur s en ayant perçu l'ensemble des observations noté par : $Z_s = \{z_1, z_2, ..., z_{m_s}\}$, comme illustré par la figure 4.3.

4.2.2 Classification locale au niveau des capteurs

Pour un objectif de comparaison, les deux types de classifieurs présentés dans le chapitre 2 sont exécutés au niveau de chaque capteur.

Dans le cas de la classification Bayésienne, chaque capteur s fournit un rapport noté CL_s contenant n_s distributions de probabilités $CL_s = \{P(O_1|c_j), ..., P(O_{n_s}|c_j)\}$ concernant les classifications des n_s objets suivis O_i parmi les nc classes définies $c_j \in C$ où C représente l'ensemble des classes.

Dans le cas de la classification crédale utilisant l'algorithme 2.3.2.2, chaque capteur s fournit un rapport de classifications locales noté $CL_s = \{m(O_1|A), ..., m(O_{n_s}|A)\}$ contenant n_s fonctions de masse concernant la classification des n_s objets suivis O_i parmi les nc classes définies, avec $A \subseteq C$.

4.2.3 Application de l'algorithme proposé 1 pour l'association multi-capteurs (track-to-track)

Cette section montre l'intérêt d'utiliser une solution crédale dans le cadre de l'association multi-capteurs. Elle décrit la manière dont les informations de base (les distances entre les objets) sont

utilisées et la manière dont une information additionnelle (les classes des objets) peut être intégrée sachant que la technique est initialement présentée dans [57]. Un résultat de simulation est par ailleurs présenté afin d'illustrer l'avantage de pouvoir résoudre le problème d'association à l'aide de plusieurs informations.

4.2.3.1 Association multi-capteurs : utilisation des distances (information de base)

L'étape d'association entre les estimations faites par les capteurs est nécessaire, elle permet de reconnaître les objets qui sont communément suivis par les différents capteurs afin de fusionner leurs classifications locales comme illustré par la figure 4.3.

L'étape d'association *track-to-track* qui est usuellement effectuée à l'aide de l'algorithme *GNN* [23] où la matrice des distances entre objets estimés par deux capteurs i et j peut être calculée ainsi :

$$\begin{aligned} D_{t,l}(k) = (\hat{x}_{i,t}(k|k-1) - \hat{x}_{j,l}(k|k-1))'(P_{i,t}(k|k-1) \\ +P_{j,l}(k|k-1))^{-1}(\hat{x}_{i,t}(k|k-1) - \hat{x}_{j,l}(k|k-1)), \end{aligned} \tag{4.8}$$

où $D_{t,l}$ représente la distance de Mahalanobis entre le vecteur d'état $\hat{x}_{i,t}(k|k-1)$ de l'objet O_t estimé par le capteur i et le vecteur d'état $\hat{x}_{j,l}(k|k-1))$ de l'objet O_l estimé par le capteur j. Dans l'algorithme *GNN*, la matrice des distances est résolue à l'aide de l'algorithme de Munkres. Les distances peuvent également être transformées en vraisemblances et le problème peut alors être résolu à l'aide de la solution d'association proposée dans la section 3.3.6 du chapitre 3. L'utilisation de cette dernière est recommandée dans la mesure où elle se trouve être moins sensible à la définition des paramètres et plus robuste par rapport aux erreurs d'estimation. En plus de cette avantage, la méthode proposée permet d'intégrer des informations additionnelles afin d'améliorer la performance d'association comme étudié dans [57].

4.2.3.2 Association multi-capteurs : intégration de l'information concernant les classes des objets (information additionnelle)

Dans le cas de l'architecture proposée dans cette section, la situation est justement favorable à l'integration de l'information concernant les classes des objets dans la phase d'association. L'intégration de cette information peut se faire de la manière suivante :

- supposons que $m_{i,t}$ soit la fonction de masse concernant l'association de l'objet O_t estimé par le capteur i. Elle est calculée à l'aide de l'équation (3.37) et exprimée sur l'ensemble $O_j^\star$ des objets estimés par le capteur j.

- supposons que $m_{i,t}^C$ et $m_{j,l}^C$ soient deux fonctions de masse provenant des deux capteurs i et j, concernant la classification des objets O_t et O_l respectivement. Elles sont exprimées sur l'ensemble des classes C et calculées comme expliqué dans la section 2.3.2.2 du chapitre 2. La combinaison conjonctive selon l'équation (2.4) du chapitre 2 donne un poids $\kappa = m(\emptyset) \approx 0$ si les objets O_t et O_l sont de la même classe, autrement $\kappa = m(\emptyset) > 0$.

- une fonction de masse m^Ω se prononçant sur l'ensemble $\Omega = \{0, 1\}$ peut alors être construite à l'aide du poids κ :

$$\begin{cases} m_{t,l}^\Omega(\{0\}) = \kappa & \text{poids réfutant l'association de } t \text{ à } l, \\ m_{t,l}^\Omega(\{0,1\}) = 1-\kappa & \text{ignorance concernant l'association de } t \text{ à } l. \end{cases} \tag{4.9}$$

  Les deux équations dans (4.9) signifient que le poids κ est utilisé pour réfuter l'association de l'objet O_t estimé par le capteur i et l'objet O_l estimé par le capteur j parce qui'ils seraient de classes différentes selon la valeur du poids κ. La fonction de masse $m_{t,l}^\Omega$ est ensuite étendue à l'ensemble $O_j^\star$ de sorte que le poids sur l'élément $\{0\}$ soit affecté à l'ensemble $O_j^\star - \{O_l\}$ et le poids sur l'ensemble $\{0, 1\}$ soit affecté à l'ensemble $O_j^\star$.

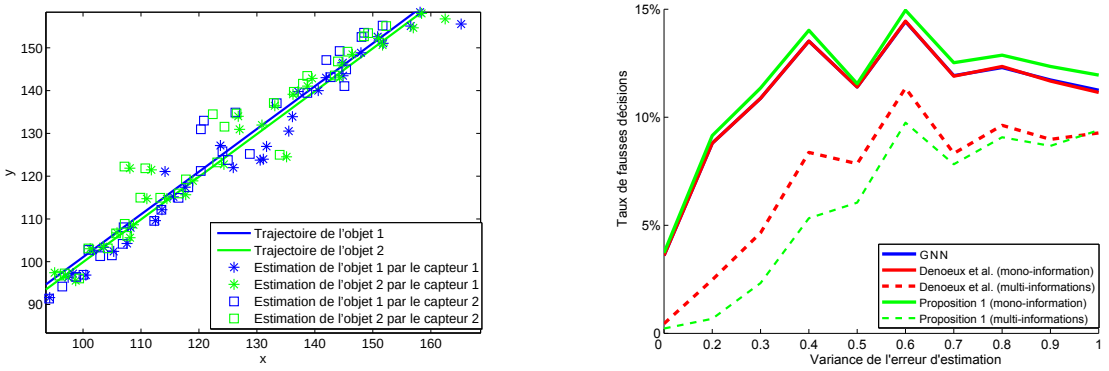
  le résultat est une fonction de masse qui peut être notée : ${}^\kappa m_{i,t}$ et qui s'exprime sur l'ensemble $O_j^\star$.

- au final les deux fonctions de masse $m_{i,t}$ et ${}^\kappa m_{i,t}$ qui s'expriment sur le même ensemble $O_j^\star$ peuvent être combinées conjonctivement. L'opération peut être répétées pour tous les objets $O_t \in O_i^\star$ estimés par le capteur i. Une fois que toutes les fonctions de masse sont calculées et combinées, elles sont utilisées dans le processus de prise de décision de la solution d'association proposée 1 (voir la section 3.3.6 du chapitre 3).

La stratégie décrite ci-haut concernant l'intégration des résultats de classifications des objets dans le problème d'association est proposée dans [57], elle peut être appliquée à toute autre information qui est en mesure d'être exprimée par une fonction de masse portant sur le même ensemble d'objets.

4.2.3.3 Association multi-capteurs : résultat de simulation (avec/sans information additionnelle)

La figure 4.4(a) présente un exemple simplifié qui consiste à associer les estimations de deux objets qui sont communément suivis par deux capteurs. Les estimations du capteur 1 sont représentées par des étoiles et des carrées bleus et les estimations du capteur 2 sont représentées par des étoiles et des carrés verts dans la figure 4.4(a).



(a) Scénario de deux trajectoires confondues avec leurs estimations faites par deux capteurs différents.

(b) Taux de fausses décision avec et sans information additionnelle.

FIGURE 4.4: Taux de fausses décisions obtenus pour des associations mono-information et multi-informations.

Les résultats portés sur la figure 4.4(b) montrent que la solution proposée 1 et la solution proposée par Denœux et *al* voient leurs taux de fausses décisions diminuer lorsque les informations concernant les classes des objets sont intégrées dans le problème d'association. D'un autre coté le taux de fausses décisions reste inchangé pour l'algorithme *GNN* dans la mesure où ce dernier ne présente aucune stratégie permettant d'intégrer l'information concernant les classes des objets.

Une fois que le consensus est atteint par rapport aux objets qui sont communément suivis par les différents capteurs, leurs classifications locales sont combinées au niveau du centre de fusion.

4.2.4 Classification globale (fusion Bayésienne)

Différentes règles de fusion Bayésienne sont répertoriées et comparées dans [27, 119]. Cette section reprend quelques-unes de ces règles, notamment la règle de fusion conjonctive et d'autres règles basées sur les opérateurs min, max, moyenne et autres. Les règles de fusion sont testées dans

un cadre de classification collaborative de cibles aériennes dont les classes sont imbriquées (voir l'application 1 dans la section 2.4 du chapitre 2).

4.2.4.1 Fusion conjonctive Bayésienne

La combinaison conjonctive de deux distributions de probabilités P_1 et P_2 issues de deux capteurs 1 et 2 supposés indépendants, est donnée par :

$$P_{1\cap 2}(O_i|c_j) = \frac{P_1(O_i|c_j)P_2(O_i|c_j)}{\sum_{c_l \in C} P_1(O_i|c_l)P_2(O_i|c_l)}, \tag{4.10}$$

où $P_{1\cap 2}(O_i|c_j)$ représente la probabilité que l'objet O_i soit de classe c_j, obtenue à l'issue de la combinaison.

4.2.4.2 Autres règles de fusions Bayésiennes

Les distributions de probabilités locales peuvent également être combinées à l'aide d'autres opérateurs de combinaison [27] :

$$P_{1,2}(O_i|c_j) = \Psi(P_1(O_i|c_j), P_2(O_i|c_j)), \tag{4.11}$$

où Ψ peut représenter les opérateurs : moyenne, minimum, maximum ou autres [27]. À noter que la combinaison disjonctive crédale n'a pas d'équivalent dans le cadre Bayésien [27].

4.2.5 Classification globale (fusion crédale)

Deux règles de fusions crédales sont testées, il s'agit de la règle de combinaison conjonctive décrite par l'équation (2.4) du chapitre 2 et la règle de fusion disjonctive décrite par l'équation (2.6). Généralement la combinaison conjonctive est utilisée dans le cas où les sources d'informations sont supposées indépendantes et fiables. La combinaison disjonctive est généralement utilisée dans le cas où les sources sont supposées indépendantes et au moins une d'elles est fiable [174].

À noter que la classification globale, dans la solution proposée, n'est effectuée que pour les objets O_i qui sont suivis par plus d'un seul capteur. Pour les objets qui ne sont suivis que par un seul capteur par exemple, leurs classifications locales sont conservées. Cela est valable pour les deux types de classifieurs : Bayésien et crédal.

4.2.6 Simulation et résultats

Cette section s'intéresse à l'exemple de classification de cibles aériennes présenté dans la section 2.4 du chapitre 2. Le suivi et la classification des cibles aériennes sont effectués par un ensemble de capteurs peu fiables, ce qui est modélisé par des bruits de mesure extrêmes. L'objectif est de fusionner leurs classifications locales en vue d'aboutir à une classification globale de meilleure qualité. Par souci de simplicité, on ne s'intéresse qu'à la fusion de données provenant de deux capteurs comme illustré par la figure 4.5.

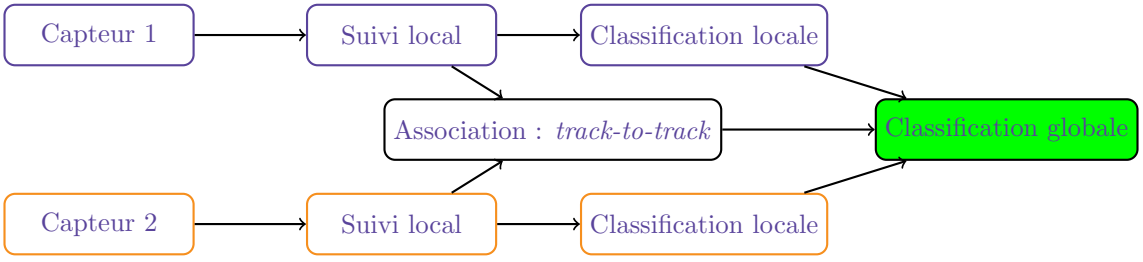


FIGURE 4.5: Organigramme de la solution proposée réduite au cas de deux capteurs.

4.2.7 Résultats de classification locale

La figure 4.6 montre les trajectoires de trois objets aériens suivis par deux capteurs. Les objets évoluent selon les modèles décrits dans la section 2.4 du chapitre 2. Ils peuvent être classés dans l'ensemble de classes $C = \{$avion de ligne, bombardier, avion de chasse$\}$ selon leurs capacités de manœuvre. Dans ce qui suit, on ne s'intéresse qu'aux résultats de classification de l'objet 2 qui évolue principalement en vitesse constante et effectue deux manœuvres : une première manœuvre d'accélération moyenne et une seconde manœuvre d'accélération forte. La classification attendue de l'objet 2 est : un doute entre les trois classes avant la manœuvre moyenne, un doute entre la deuxième et la troisième classes avant la seconde manœuvre forte et une classification finale comme étant un avion de chasse après la manœuvre forte.

La figure 4.7 présente les résultats de classification obtenus par un capteur fiable. Les résultats sont analysés dans la section 2.4 du chapitre 2. Dans cette section on s'intéresse au cas où le capteur est peu fiable (les capteurs sont peu fiables). La non fiabilité des capteurs est simulée par un bruit de mesure de grande variance dans l'équation (2.20) du chapitre 2.

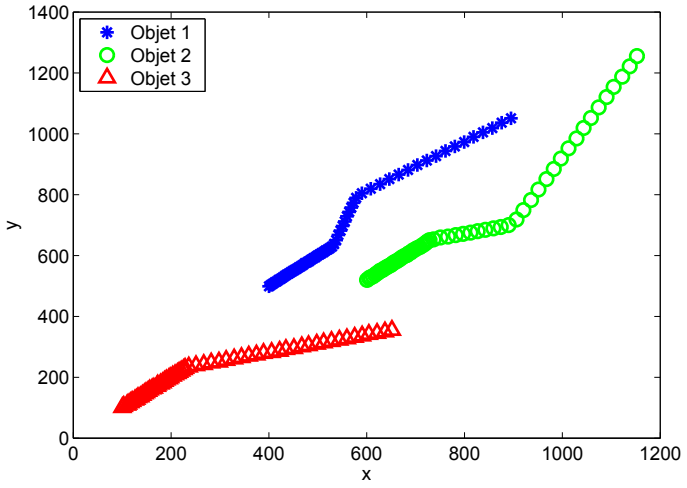
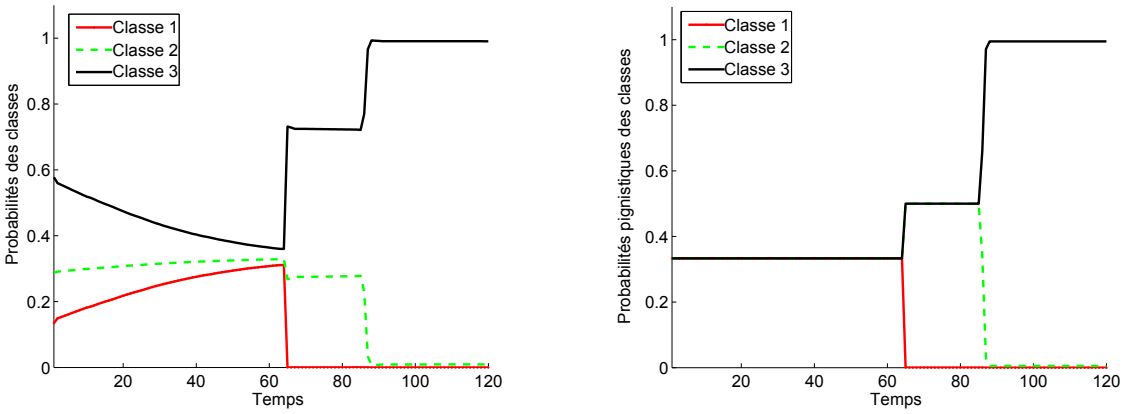


FIGURE 4.6: Scénario de 3 objets manœuvrants observés par deux capteurs peu fiables.



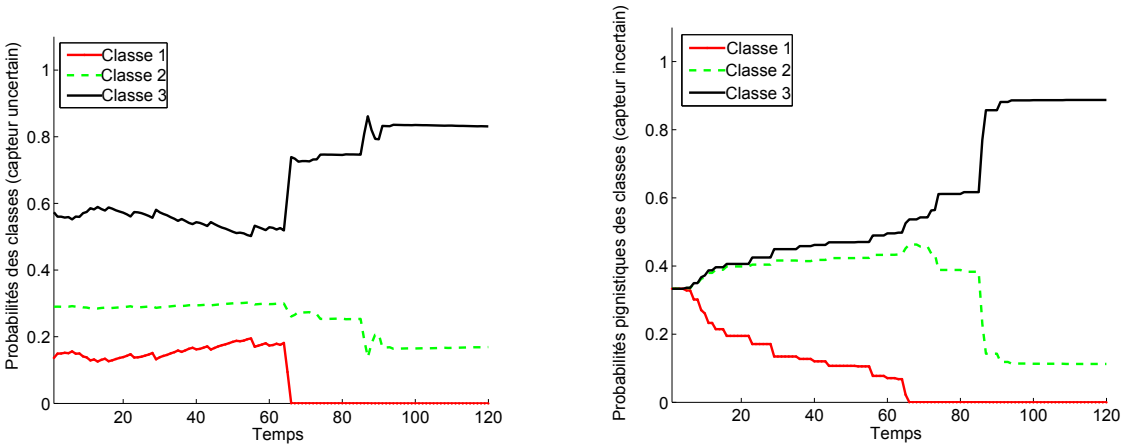
(a) Classification Bayésienne locale.

(b) Classification crédale locale.

FIGURE 4.7: Résultats de classifications de l'objet 2 obtenus par un seul capteur fiable.

La figure 4.8 montre que les erreurs de mesure engendrent la détérioration des résultats de classification. Il peut être remarqué que les probabilités des classes divergent d'avantage du résultat de classification attendu, qui est par exemple un doute parfait entre les trois classes dans la première phase du mouvement. Les classifieurs sont induits en erreur à cause des importantes erreurs de mesure qui sont considérées comme étant des manœuvres, ce qui peut conduire à des erreurs de décision.

La divergence dans la classification locale est mesurée pour plusieurs niveaux d'incertitude. Elle correspond à l'Erreur Quadratique Moyenne de classification calculée par : $EQM = (\hat{BetP} - BetP)'(\hat{BetP} - BetP)$ dans le cas de la classification crédale où $\hat{BetP}$ représente la distribution de probabilités pignistiques attendue et $BetP$ représente la distribution de probabilités pignistiques



(a) Classification Bayésienne locale.

(b) Classification crédale locale.

FIGURE 4.8: Résultats de classification de l'objet 2 obtenus par un seul capteur non fiable.

calculée par le classifieur crédal. La même quantité est calculée par : $EQM = (\hat{P} - P)'(\hat{P} - P)$ pour le cas de la classification Bayésienne où $\hat{P}$ représente la distribution de probabilités attendue et P représente la distribution de probabilités calculée par le classifieur Bayésien. L'EQM de classifications correspondant au données d'un seul capteur pour différents niveaux d'incertitude est présentée par la figure 4.9.

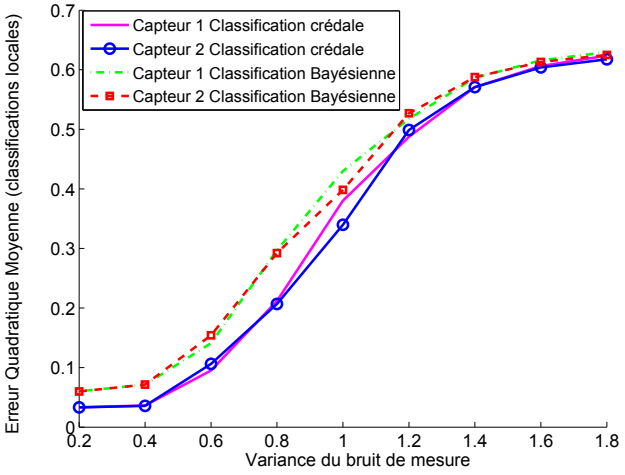
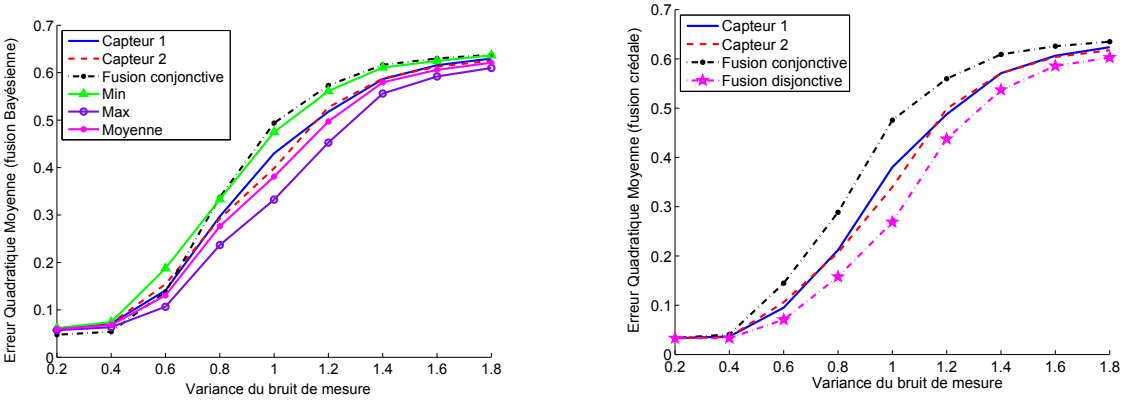


FIGURE 4.9: Classifications locales Bayésienne et crédale pour différents niveaux de fiabilité de capteur.

Le résultat présenté par la figure 4.9 confirme l'avantage de la classification crédale par rapport à la classification Bayésienne. Une saturation de performances est également remarquable pour un bruit de mesure important. Sachant que ces résultats sont obtenus au niveau locale, la classification vise à les améliorer en adoptant une loi de fusion adéquate.

4.2.8 Résultats de classification globale

Cette section s'intéresse à la fusion des classifications locales. La figure 4.10(a) présente l'ensemble des résultats de classifications Bayésiennes obtenus localement par les deux capteurs 1 et 2 ainsi que les résultats globaux obtenus à l'issue de la fusion conjonctive (voir l'équation (4.10)) et à l'aide des opérations : moyenne, min et max (voir l'équation (4.10)). Il peut être remarqué que le meilleur résultat de classification Bayésienne est obtenu à l'aide de l'opérateur max.



(a) Résultats, locaux et globaux, de la classification Bayésienne.

(b) Résultats, locaux et globaux, de la classification crédale.

FIGURE 4.10: Résultats de classifications Bayésiennes et crédales pour différents niveaux de fiabilité de capteurs.

Concernant les résultats de la classification crédale, il peut être constaté par la figure 4.10(b) que la fusion disjonctive fournit le meilleur résultat de classification comparé aux résultats obtenus localement par les capteurs et au résultat obtenu à l'aide de la fusion conjonctive. Compte tenu de l'aspect ambigu de la classification considérée, ces résultats peuvent être expliqués ainsi :

- L'aspect engagé de la fusion conjonctive est inadapté au problème de classification considéré où le doute est souvent préférable. En effet, si on considère par exemple une distribution de croyance : $m_1(\{c_2, c_3\}) = 0.4$ et $m_1(\{c_1, c_2, c_3\}) = 0.6$ fournie par le capteur 1 et une distribution : $m_2(\{c_2, c_3\}) = 0.5$ et $m_2(\{c_1, c_2, c_3\}) = 0.5$ fournie par le capteur 2, le résultat de la fusion conjonctive est : $m_{1\textcircled{\cap}2}(\{c_2, c_3\}) = 0.7$ et $m_{1\textcircled{\cap}2}(\{c_1, c_2, c_3\}) = 0.3$, cela montre que l'ensemble spécifique $\{c_2, c_3\}$ est favorisé par rapport au doute or dans l'exemple de classification considéré, comme le montre la figure 4.8(b), l'engagement sur les ensembles spécifiques dans des phases où on est censé avoir un doute parfait, est causé par les erreurs dont on souhaite minimiser l'effet.

- D'une autre part, la fusion disjonctive des fonctions de masse provenant des capteurs 1 et 2 donne : $m_{1\textcircled{\cup}2}(\{c_2, c_3\}) = 0.2$ et $m_{1\textcircled{\cup}2}(\{c_1, c_2, c_3\}) = 0.8$ ce qui montre que le doute $\{c_1, c_2, c_3\}$ est largement favorisé par rapport à l'ensemble spécifique $\{c_2, c_3\}$ ce qui explique l'aspect prudent de la fusion disjonctive assurant le meilleur résultat dans le cadre de la classification considérée. À noter que d'autres règles de fusion peuvent également être considérées, notamment les règles de fusion conjonctives prudentes [55, 59, 95].

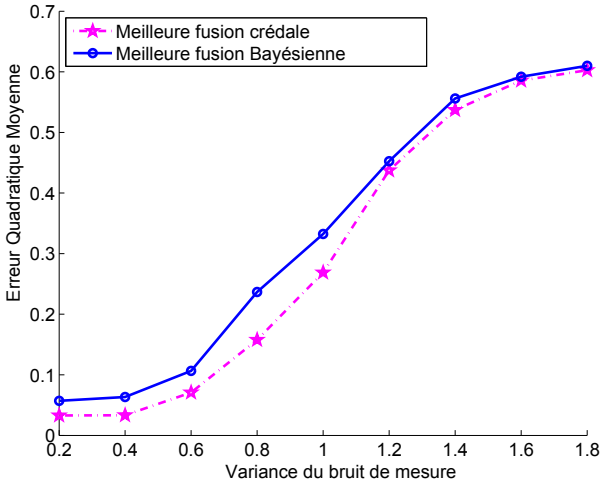


FIGURE 4.11: Comparaison entre les meilleures classifications globales Bayésienne et crédale.

La figure 4.11 reprend le meilleur résultat Bayésien obtenu à l'aide de l'opération max sur les classifications locales, et le meilleur résultat crédal obtenu à l'aide de la fusion disjonctive des classifications locales. Cette figure confirme l'avantage de la classification crédale par rapport à la classification Bayésienne dans le cas du problème de classification considéré, y compris dans le cadre de la fusion multi-capteurs.

4.3 Approche distribuée proposée

Cette section propose une solution de suivi et de classification multi-objets, multi-capteurs dite distribuée. La solution concerne des capteurs ayant de différents points de vues sur un ensemble d'objets, cela peut être le cas par exemple d'une scène surveillée par plusieurs caméras posées à des angles différents et ayant une observabilité partielle de l'état des objets. L'idée de la solution est de construire au niveau de chaque capteur l'information complète concernant l'état et les classes des objets. Les capteurs sont supposés pouvoir communiquer entre eux, ils sont considérés comme

étant des nœuds dans un graphe où les arcs représentent les communications entre les capteurs. La contribution de la solution consiste en l'insertion d'un algorithme de consensus dans les phases de suivi et de classification des objets comme illustré par la figure 4.12. L'algorithme de consensus permet de calculer la valeur moyenne des données en entrée en un certain nombre d'itérations où ces dernières représentent les échanges d'informations dans le réseau de capteurs [51, 76]. La particularité de la solution proposée dans ce chapitre est l'utilisation d'un algorithme de consensus pouvant calculer la valeur moyenne en un temps fini [98, 99] contrairement à d'autres travaux qui utilisent un algorithme de consensus dont le calcul de la moyenne est asymptotique [36, 140, 142]. À notre connaissance la technique de consensus est peu utilisée pour le suivi d'objets [39, 141, 161] et moins pour la classification d'objets.

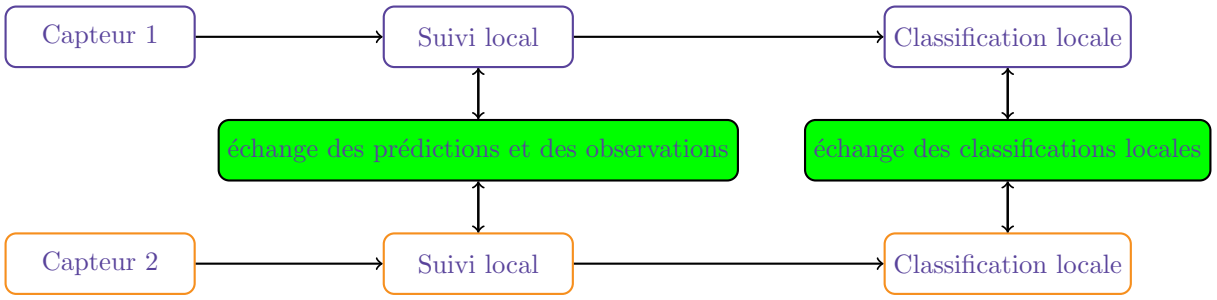


FIGURE 4.12: Organigramme de la solution distribuée proposée réduite au cas de deux capteurs.

Dans ce qui suit, une description succincte de l'algorithme de consensus asymptotique et de l'algorithme de consensus en temps fini est donnée. Ensuite, les solutions de suivi et de classification distribuées sont présentées ainsi que les résultats de leur application à la reconnaissance des comportements de véhicules sur une autoroute.

4.3.1 Algorithme de consensus

L'algorithme de consensus est une solution distribuée qui calcule la valeur moyenne de données en entrée provenant d'un ensemble de capteurs connectés en réseau. L'algorithme est embarqué avec chaque capteur, ce qui permet de s'en passer du centre de fusion et ainsi assurer l'extensibilité du système. L'aggregation des données au niveau des capteurs se fait à travers quelques itérations, ce qui correspond aux échanges de données dans le réseau. La moyenne est calculée au fur et à mesure que les données parviennent aux capteurs ce qui évite leur sauvegarde ainsi épargner la mémoire des capteurs. Dans ce qui suit, deux types d'algorithmes de consensus sont brièvement décrits : l'algorithme de consensus asymptotique et l'algorithme de consensus en temps fini.

4.3.1.1 Algorithme de consensus asymptotique

L'objectif de l'algorithme de consensus est de calculer une valeur moyenne tel qu'il est illustré par l'équation (4.12).

$$ave(y(0)) = \frac{1}{S}\sum_{s=1}^{S} y_i(0), \tag{4.12}$$

où "ave" symbolise la fonction qui calcule la valeur moyenne des données en entrée $y_s(0)$ avec $s = 1, ..., S$ où S représente le nombre de sources (de capteurs). La valeur moyenne dans l'équation (4.12) peut être calculée d'une manière distribuée en utilisant l'équation recursive suivante :

$$y(t+1) = Py(t), \tag{4.13}$$

où t représente les itérations de l'algorithme, $y(t) = [y_1(t)\ y_2(t)\ \dots\ y_S(t)]'$ et P représente une matrice de transition double stochastique de dimension $(S \times S)$ tel que : $(P_{i,j} \geq 0,\ \sum_{i=1}^{S} P_{i,j} = 1,\ \sum_{j=1}^{S} P_{i,j} = 1,)$. La matrice P est compatible avec le graphe de communication ce qui signifie que : $P_{i,j} > 0$ si le nœud i est connecté au nœud j, $P_{i,j} = 0$ sinon. La matrice P est constante lorsque le graphe de communication est fixe dans le temps, elle satisfait la condition suivante dans le cas du consensus asymptotique [76, 140] :

$$\lim_{t\to\infty} P = \frac{1}{S}\mathbf{11}, \tag{4.14}$$

où $\mathbf{11}$ représente une matrice unitaire de dimension $(S \times S)$. Des méthodes permettant de paramétrer la matrice P peuvent être trouvées dans [51].

4.3.1.2 Algorithme de consensus en temps fini

L'algorithme de consensus en temps fini présenté dans [98, 99] calcule la valeur moyenne (4.12) en un nombre fini d'itérations. Au lieu d'utiliser une seule matrice de transition P dans l'équation (4.13) l'algorithme utilise un ensemble de matrices $Q(t)$, une matrice pour chaque itération $t = 1, ..., L$ où L représente le nombre de valeurs propres distinctes de la matrice Laplacienne du graphe de communication.

$$\prod_{t=1}^{L} Q(t) = \frac{1}{N}\mathbf{11}. \tag{4.15}$$

L'équation (4.15) pose la condition concernant les matrices $Q(t)$: leur produit est censé calculer la valeur moyenne exacte au bout de L itérations. Les matrices $Q(t)$ sont compatibles avec le graphe de communication, plus d'informations quant à leur calcul peuvent être trouvées dans [98, 99].

Algorithme de consensus asymptotique vs algorithme de consensus en temps fini : ce paragraphe vise à comparer l'algorithme de consensus asymptotique et l'algorithme de consensus en temps fini sur un exemple simplifié de graphe de communication représenté par la figure 4.13.

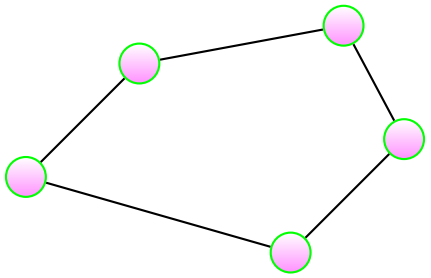


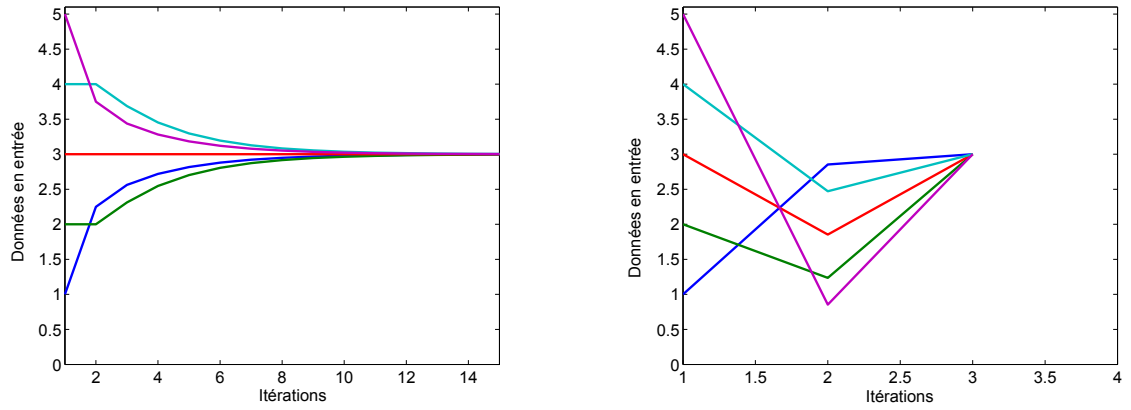
FIGURE 4.13: Exemple simplifié de graphe de communication.

On s'intéresse au calcul de la valeur moyenne des éléments d'un vecteur $y(0) = [1\ 2\ 3\ 4\ 5]$ où chaque élément provient d'un nœud distinct du graphe 4.13. La matrice P et la matrice Laplacienne Lp compatibles avec le graphe 4.13 sont :

$$P = \begin{vmatrix} 0.5 & 0.25 & 0 & 0 & 0.25 \\ 0.25 & 0.5 & 0.25 & 0 & 0 \\ 0 & 0.25 & 0.5 & 0.25 & 0 \\ 0 & 0 & 0.25 & 0.5 & 0.25 \\ 0.25 & 0 & 0 & 0.25 & 0.5 \end{vmatrix}, \quad Lp = \begin{vmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{vmatrix}.$$

Le calcul de la valeur moyenne à travers les itérations de l'algorithme de consensus asymptotique est illustré par la figure 4.14(a) et son calcul à l'aide de l'algorithme de consensus en temps fini est illustré par la figure 4.14(b).

Il peut être remarqué sur la figure 4.14 que l'algorithme de consensus en temps fini calcule la valeur moyenne du vecteur $y(0) = [1\ 2\ 3\ 4\ 5]$ en 2 itérations tandis qu'une valeur moyenne acceptable n'est atteinte qu'après environ 8 itérations pour l'algorithme asymptotique. De plus, ce dernier nécessite de fixer un critère d'arrêt dont dépend la finesse de la valeur moyenne calculée. Dans le cadre du suivi d'objets, cela se répercute sur la qualité d'estimation des trajectoires (voir l'algorithme 4.2). Quelques solutions de suivi d'objets basées sur l'algorithme de consensus asymptotique peuvent



(a) Calcul de la valeur moyenne à l'aide du consensus asymptotique.

(b) Calcul de la valeur moyenne à l'aide du consensus en temps fini.

FIGURE 4.14: Consensus asymptotique vs consensus en temps fini pour le calcul d'une valeur moyenne simplifiée.

être trouvées dans [39, 141, 161]. À notre connaissance l'algorithme de consensus en temps fini n'a pas été appliqué au suivi d'objets si ce n'est dans [89].

4.3.2 Suivi distribué basé sur l'algorithme de consensus en temps fini

Tout comme dans la section 2.3.1 du chapitre 2, le modèle d'évolution des objets (véhicules) est un modèle de Markov à sauts. Le modèle d'état est le modèle d'observation sont ainsi rappelés :

$$x(k) = Fx(k-1) + Gu_{m_i}(k) + w(k-1), \tag{4.16}$$

où u_{m_i} est une entrée déterministe et les modes $m_i \in \{m_1, m_2, ..., m_M\}$ représentent les différentes manœuvres sur la plan (x, y).

Le modèle d'observation d'un capteur s est donné par :

$$z_s(k) = H_s x(k) + v_s(k), \tag{4.17}$$

où H_s représente la matrice d'observation et $v_s(k)$ le bruit de mesure qui est supposé Gaussien avec une matrice de covariance R_s. L'indice s mentionne que ce sont des grandeurs locales, propres au capteur $s \in \{1, ..., S\}$ où S représente le nombre de capteurs.

L'application de la solution distribuée nécessite une indépendance des observations faites par les capteurs.

4.3.2.1 Indépendance des observations locales

L'indépendance des observations est exprimée par les équivalences suivantes concernant les vecteurs d'observations :

$$z(k) = [z_1^{'}(k)\ z_2^{'}(k)\ ...\ z_S^{'}(k)]^{'}, \tag{4.18}$$

les bruits de capteurs :

$$v(k) = [v_1^{'}(k)\ v_2^{'}(k)\ ...\ v_S^{'}(k)]^{'}, \tag{4.19}$$

les matrices de covariances correspondantes :

$$R = [R_1\ R_2\ ...\ R_S], \tag{4.20}$$

et finalement les matrices d'observations qui sont supposées complémentaires :

$$H = [H_1^{'}\ H_2^{'}\ ...\ H_S^{'}]^{'}. \tag{4.21}$$

Les paramètres : $z(k)$, $v(k)$, R, H correspondent à un modèle d'observation global qu'aurait eu un capteur imaginaire pouvant centraliser toute l'information. Les suppositions concernant les modèles d'observation locaux permettent de poser l'équivalence suivante [139] :

$$\begin{cases} H^{'}R^{-1}H = \sum_{s=1}^{S} H_s^{'}R_s^{-1}H_s \\ H^{'}R^{-1}z(k) = \sum_{s=1}^{N} H_s^{'}R_s^{-1}z_s(k) \end{cases} \tag{4.22}$$

L'équivalence exprimée par (4.22) est nécessaire à la mise à jour des filtres de Kalman assurant le suivi des objets. Ces derniers sont appelés : Filtres de Kalman Distribués *FKD* dans [139] lorsqu'ils sont combinés à des algorithmes de consensus.

Le suivi au niveau des capteurs illustré par l'étape "suivi local" dans la figure 4.12 est effectué en plusieurs étapes qui sont à leur tour illustrées par la figure 4.15 et résumées dans ce qui suivi.

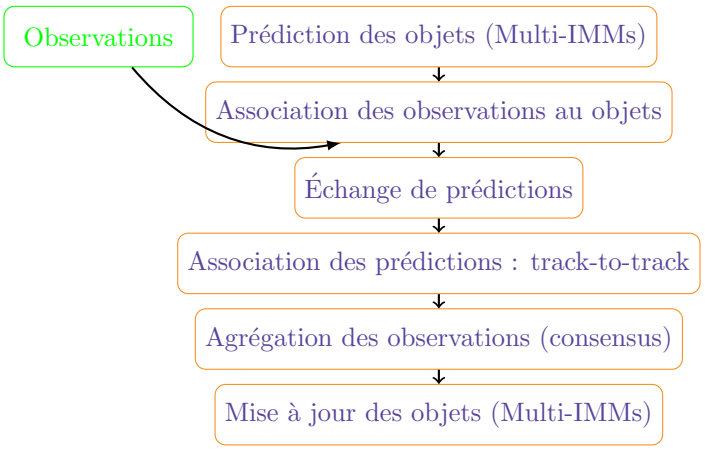


FIGURE 4.15: Étapes principales de l'algorithme de suivi distribué.

4.3.2.2 Phase de prédiction

La phase de prédiction est détaillée dans l'algorithme 4.1.

Initialisation :	
Pour chaque capteur	$s \in \{1, 2, ..., S\}$
Pour chaque objet	$t \in \{1, 2, ..., n_s\}$
Pour chaque mode d'évolution	$i \in \{1, 2, ..., M\}$
Vecteurs états à $k-1$:	$\hat{x}^i_{s,t}(k-1)$
Matrices de covariances à $k-1$:	$P^i_{s,t}(k-1)$
Probabilités des modes prédites :	$\overline{\mu}^i_{s,t}(k\|k-1)$
Prédiction :	
Pour chaque capteur	$s \in \{1, 2, ..., S\}$
Pour chaque objet	$t \in \{1, 2, ..., n_s\}$
Pour chaque mode d'évolution	$i \in \{1, 2, ..., M\}$
Prédiction des estimations :	$\hat{x}^i_{s,t}(k\|k-1) = F\hat{x}^i_{s,t}(k-1) + Gu(m_i)$
Prédiction des covariances :	$P^i_{s,t}(k\|k-1) = FP^i_t(k-1)F' + Q(k-1)$
Observation prédite :	$\overline{z}^i_{s,t}(k\|k-1) = H_s\hat{x}^i_{s,t}(k\|k-1)$
Covariance de l'observation :	$S^i_{s,t}(k\|k-1) = H_sP^i_{s,t}(k\|k-1)H'_s + R_s$
Quantités globales :	
Pour chaque capteur	$s \in \{1, 2, ..., S\}$
Pour chaque objet	$t \in \{1, 2, ..., n_s\}$
Vecteurs d'état globaux :	$\hat{x}_{s,t}(k\|k-1) = \sum_{i=1}^{M} \overline{\mu}^i_{s,t}(k\|k-1)\hat{x}^i_t(k\|k-1)$
Covariances d'état :	$P_{s,t}(k\|k-1) = \sum_{i=1}^{M} \overline{\mu}^i(k\|k-1)P^i_{s,t}(k\|k-1)$
Observations globales :	$\overline{z}_{s,t}(k\|k-1) = \sum_{i=1}^{M} \overline{\mu}^i_{s,t}(k\|k-1)\overline{z}^i_t(k\|k-1)$
Covariances des observations :	$S_{s,t}(k\|k-1) = \sum_{i=1}^{M} \overline{\mu}^i_{s,t}(k\|k-1)S^i_{s,t}(k\|k-1)$

TABLEAU 4.1: Phase de prédiction de l'algorithme de suivi distribué.

Avant la réception des observations $z_{s,l}(k)$ au pas de temps k, chaque capteur aura prédit les observations $\overline{z}_{s,t}$ des objets connus avec $l \in \{1, ..., m_s\}$ et $t \in \{1, ..., n_s\}$ où m_s est le nombre d'observations réelles détectées par le capteur s et n_s le nombre d'observations prédites par le

même capteur. Les observations réelles sont associées aux observations prédites à l'aide du *GNN* ou un algorithme d'association parmi ceux présentés dans le chapitre 3.

À l'issue de la phase d'association, chaque capteur s est en mesure de fournir les données suivantes :

- L'ensemble des vecteurs d'état prédits : $\hat{X}_s(k|k-1) = \{\hat{x}_{s,1}(k|k-1), \hat{x}_{s,2}(k|k-1), ..., \hat{x}_{s,n_s}(k|k-1)\}$.
- L'ensemble des matrices de covariances prédites : $P_s(k|k-1) = \{P_{s,1}(k|k-1), P_{s,2}(k|k-1), ..., P_{s,n_s}(k|k-1)\}$.
- Les indices de détections : $r_s(k) = \{r_{s,1}(k), r_{s,2}(k), ..., r_{i,n_s}(k)\}$, où $r_{s,t}(k) = 1$ si l'objet t est détecté par le capteur s à l'instant k, $r_{s,t}(k) = 0$ sinon.
- Les informations sur les observations locales : $\nu_s(k) = \{\nu_s^1(k), \nu_s^2(k), ..., \nu_s^{n_s}(k)\}$, $\upsilon_s(k) = \{\upsilon_s^1(k), \upsilon_s^2(k), ..., \upsilon_s^{n_s}(k)\}$ qui sont calculées par :

$$\nu_s^t(k) = r_{s,t}(k)(H_s^{'} R_s^{-1} H_s), \tag{4.23}$$

$$\upsilon_s^t(k) = r_s^t(k)(H_s^{'} R_s^{-1} z_{s,t}(k)). \tag{4.24}$$

Une fois toutes les quantités locales sont calculées au niveau des capteurs, elles sont communiquées aux autres capteurs. Cette phase de communication s'effectue en deux étapes principales : échange et arrangement des prédictions et agrégation des observations locales.

4.3.2.3 Échange et arrangement des prédictions

Cette étape vise à ordonner les objets suivis de la même manière au niveau de tous les capteurs. Elle nécessite la communication et la sauvegarde des données : l'ensemble des prédictions $\hat{X}_i(k|k-1)$, l'ensemble des covariances correspondantes $P_i(k|k-1)$ ainsi que l'ensemble des indices de détections $r_s(k)$. L'execution de l'un des algorithmes d'association décrit dans la chapitre 3 permet d'ordonner les objets de la même manière au niveau de tous les capteurs. Toutefois, l'algorithme d'association doit être exécuté pour chaque paire de données provenant de deux capteurs. Cette étape est la plus fastidieuse de l'algorithme de suivi distribué. La complexité n'est pas étudiée dans ce chapitre, l'intérêt est porté sur l'apport de la solution globale en terme de suivi et de classification

collaboratifs. À noter qu'à l'issue de cette opération tous les capteurs connaissent le même nombre d'objets n qu'il soient communément détectés, partiellement détectés ou non détectés.

4.3.2.4 Agrégation des observations locales

Cette étape vise à calculer des observations globales selon l'équivalence (4.22). La sauvegarde des observations locales au niveau des capteurs n'est pas nécessaire, ces dernières sont agrégées au fur et à mesure qu'elles sont propagées dans le réseau de capteurs grâce à l'algorithme de consensus en temps fini présenté dans la section 4.3.1.2. Les calculs effectués par l'algorithme de consensus sont donnés par :

$$\nu^t(k) = Nb^t ave(\{\nu_s^t(k)\}_{s=1}^{Nb^t}), \tag{4.25}$$

$$\upsilon^t(k) = Nb^t ave(\{\upsilon_s^t(k)\}_{s=1}^{Nb^t}), \tag{4.26}$$

où $Nb^t = \sum_{s=1}^{S} r_s^t(k)$ représente le nombre de capteurs ayant détecté l'objet t à l'instant k.

En utilisant les quantités globales calculées dans les équations (4.25) et (4.26), la mise à jour complète de l'état des objets peut être effectuée, cela passe par la mise à jour des filtres de Kalman au niveau des *IMMs*.

4.3.2.5 Mise à jour de l'état des objets

La mise à jour de l'état des objets est effectuée selon l'algorithme 4.2.

Il peut être remarqué que les équations des filtres de Kalman utilisées dans l'algorithme 4.2 sont différentes de celles utilisées dans l'algorithme 2.1 du chapitre 2. Elles correspondent aux filtres de Kalman dits informationnels utilisés dans [140]. À l'aide de l'algorithme de consensus, chaque capteur centralise toute l'information concernant l'état des objets suivis, détectés ou non détectés. Comme dans la solution 2.1 du chapitre 2 la gestion des apparitions et des disparitions d'objets est gérée par les fonctions score.

Données nécessaires :	
Pour chaque capteur	$s \in \{1, 2, ..., S\}$
Pour chaque objet	$t \in \{1, 2, ..., n\}$
Pour chaque mode d'évolution	$i \in \{1, 2, ..., M\}$
Vecteurs d'état prédits :	$\hat{x}^i_{s,t}(k\|k-1)$
Matrices de covariances prédites :	$P^i_{s,t}(k\|k-1)$
Quantités globales :	$\nu^t,\ \upsilon^t$
Probabilités des modes (algorithme 2.1) :	$\mu^i_{s,t}(k)$
Phase de mise à jour :	
Pour chaque capteur	$s \in \{1, 2, ..., S\}$
Pour chaque objet	$t \in \{1, 2, ..., n\}$
Pour chaque mode d'évolution	$i \in \{1, 2, ..., M\}$
Matrices de covariances à l'instant k :	$(P^i_{s,t}(k))^{-1} = (P^i_{s,t}(k\|k-1))^{-1} + \nu^t.$
Vecteurs d'état à l'instant k :	$\hat{x}_{s,t}(k) = P^i_{s,t}(k)[(P^i_{s,t}(k\|k-1))^{-1}\hat{x}^i_{s,t}(k\|k-1) + \upsilon^t].$
Quantités globales :	(usage externe)
Pour chaque capteur	$s \in \{1, 2, ..., S\}$
Pour chaque objet	$t \in \{1, 2, ..., n_s\}$
Vecteurs d'état globaux :	$\hat{x}_{s,t}(k) = \sum_{i=1}^{M} \mu^i_{s,t}(k)\hat{x}^i_{s,t}(k)$
Covariances d'état :	$P_{s,t}(k) = \sum_{i=1}^{M} \mu^i_{s,t}(k)P^i_{s,t}(k)$

TABLEAU 4.2: Phase de mise à jour de l'algorithme de suivi distribué.

4.3.3 Classification distribuée basée sur l'algorithme de consensus

La classification distribuée des objets nécessite l'échange de vraisemblances des classes qui sont calculées d'une manière similaire au calcul des vraisemblances des comportements dans la section 2.3.2.1 du chapitre 2. Chaque capteur envoie un ensemble de vraisemblances : $\lambda_s(k) = \{\lambda_{s,1}(k), \lambda_{s,2}(k), ..., \lambda_{s,n}(k)\}$ sachant que l'objectif est de classer les n objets suivis parmi l'ensemble $\{c_1,\ c_2,\ ...,\ c_{nc}\}$ de nc classes. Une vraisemblance $\lambda_{s,}(k)$ n'est qu'une probabilité conditionnelle qui peut être notée par : $\lambda_{s,t}(\overline{z}_t(k)|c_j)$, elle correspond à la probabilité que l'objet connu O_t représenté par son observation prédite $\overline{z}_t$ à l'instant k, soit de la classe c_j. L'agrégation des vraisemblances locales à l'aide de l'algorithme de consensus permet de calculer des vraisemblances globales au niveau des capteurs ainsi procéder au calcul des probabilités *a posteriori* des classes. Le processus permettant cette opération est décrit ainsi :

Selon [141], les probabilités *a posteriori*, à l'instant k, des classes $c_j \in \{c_1,\ c_2,\ ...,\ c_{nc}\}$ peuvent être calculées par :

$$p(c_j|\overline{z}_t(k)) = \alpha p(c_j|\overline{z}_t(k-1)) \prod_{s=1}^{S} \lambda_{s,t}(\overline{z}_t(k)|c_j), \tag{4.27}$$

où α est un paramètre de normalisation. La validité de l'équation (4.27) nécessite l'indépendance des capteurs.

L'idée est de calculer la vraisemblance globale de l'objet O_t notée W_t à l'aide de l'algorithme de consensus :

$$W_t = \prod_{s=1}^{S} \lambda_{s,t}(z_t(k)|c_j), \tag{4.28}$$

Pour ce faire le produit des vraisemblances locales W_t est transformé en une somme de logarithmes des vraisemblances locales notée W'_t :

$$W'_t = \frac{1}{S} log(W_t) = \frac{1}{S} \sum_{s=1}^{S} log(\lambda_{s,t}(\overline{z}_t(k)|c_j)). \tag{4.29}$$

La quantité $W'_t = ave(\{log(\lambda_{s,t}(\overline{z}_t(k)|c_j))\}_{s=1}^{S})$ peut alors être calculée d'une manière distribuée à l'aide de l'algorithme de consensus en temps fini. Une fois cela est fait, la vraisemblance globale est déduite : $W_t = exp(SW'_t)$.

Les vraisemblances globales W_t sont alors utilisées pour mettre à jour les probabilité *a posteriori* des classes dans l'équation (4.27).

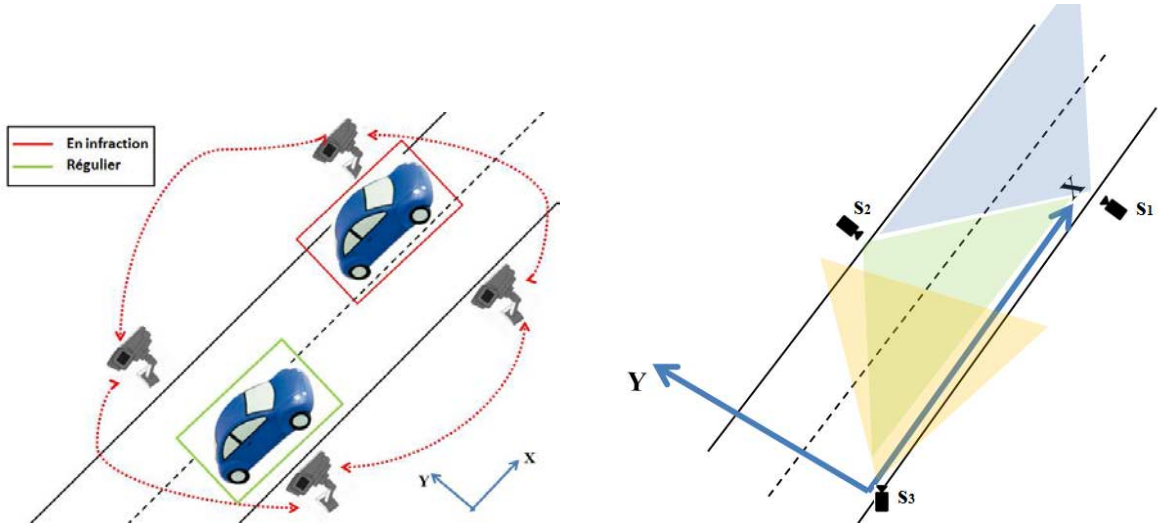
4.3.4 Simulation et résultats

Cette section présente une simulation visant à reconnaître le comportement des véhicules sur une autoroute. Deux résultats de simulation sont donnés, le premier concerne le suivi des trajectoires des véhicules et le second concerne leurs classifications.

4.3.4.1 Simulation

Un exemple de suivi de véhicules sur une autoroute est proposé. Les véhicules sont supposés évoluer selon le modèle (4.16) avec des paramètres adéquats à l'évolution de véhicules routiers. Le vecteur d'état $x = [x\ \dot{x}\ y\ \dot{y}]'$ représente la position et la vitesse d'un véhicule sur le plan (x, y). L'entrée déterministe dans (4.16) modélise les différents modes d'accélérations. Ces derniers sont séparés en deux groupes : le groupe des accélérations autorisées correspondant à un comportement d'évolution régulier et le groupe des accélérations non autorisées ce qui correspond à un comportement d'infraction. L'évolution régulière et l'évolution en infraction sont les comportements qu'on

souhaite reconnaître dans la phase de classification. L'exemple de simulation est illustré par la figure 4.16(a).



(a) Simulation proposée. (b) Dispositions des capteurs.

FIGURE 4.16: Reconnaissance des comportements de véhicules sur une autoroute.

La figure 4.16(b) représente le système d'observation mis en place afin de surveiller le comportement des véhicules sur un tronçon d'autoroute. Trois capteurs ayant des observations partielles de la scène sont mis en place. Le capteur 1 et 2 ont une observabilité redondante sur l'axe x du tronçon d'autoroute et le capteur 3 observe l'axe y. Les observations obéissent au modèle (4.17) où la matrice H_s pour les capteurs 1 et 2 est donnée par :

$$H_1 = H_2 = [1\ 0\ 0\ 0], \tag{4.30}$$

et la matrice H_3 est donnée par :

$$H_3 = [0\ 0\ 1\ 0]. \tag{4.31}$$

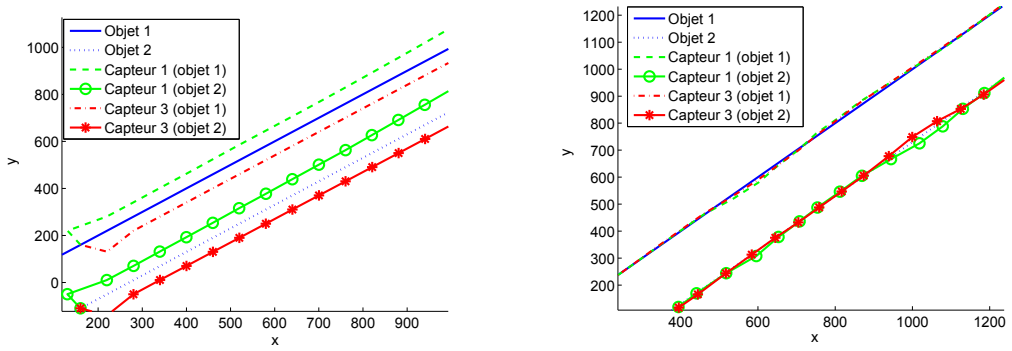
Il peut être constaté par ces modèles d'observations qu'aucun capteur n'a une information complète sur l'état des objets d'où l'intérêt de l'approche collaborative proposée dont les résultats de simulations sont présentés dans la section suivante.

4.3.4.2 Résultats de simulations

Cette section présente deux résultats de simulations, le premier résultat présenté par la figure 4.17 concerne le suivi distribué des véhicules, il compare les résultats de suivi qui peuvent être obtenus

localement par les capteurs et le résultat de suivi avec application de la solution collaborative. Le second résultat donné par la figure 4.19 présente une comparaison entre la classification locale pouvant être obtenue au niveau des capteurs et le résultat de la classification collaborative.

Résultats du suivi collaboratif :



(a) Suivi sans consensus.

(b) Suivi avec consensus.

FIGURE 4.17: Suivi d'objets avec et sans consensus.

La figure 4.17(a) présente deux trajectoires simples de deux objets et les résultats du suivi obtenus localement par les capteurs 1 et 3. Le résultat du suivi du capteur 2 est redondant à celui du capteur 1. Il peut être remarqué que l'aspect partiel des informations à disposition des capteurs fait que leurs estimations des trajectoires sont faussées. La figure 4.17(b) montre que le suivi collaboratif permet de compléter l'information à disposition des capteurs ainsi assurer un suivi correct des trajectoires.

Résultats de la classification collaborative : un second exemple de simulation est présenté dans la figure 4.18, il simule les trajectoires de deux véhicules sur une autoroute. La simulation s'étend sur 100 pas de temps. L'objet 2 excède la vitesse maximale autorisée à partir du pas de temps 60, comme illustré par la figure 4.18.

Dans ce qui suit on s'intéresse à la classification de l'objet 2. Les résultats de classification obtenus localement par les capteurs 1 et 3 sont respectivement donnés par les figures 4.19(a) et 4.19(b).

Il peut être remarqué que le capteur 1 surveillant l'axe x a pu détecter l'infraction du véhicule dans la mesure où l'accélération irrégulière est effectuée dans l'axe x. D'autre part, le capteur 3 ayant une visibilité sur l'axe y de l'autoroute n'a pas pu détecté l'infraction. L'application de l'algorithme de classification distribuée décrit dans la section 4.3.3 permet d'assurer une classification complète au niveau de tous les capteurs.

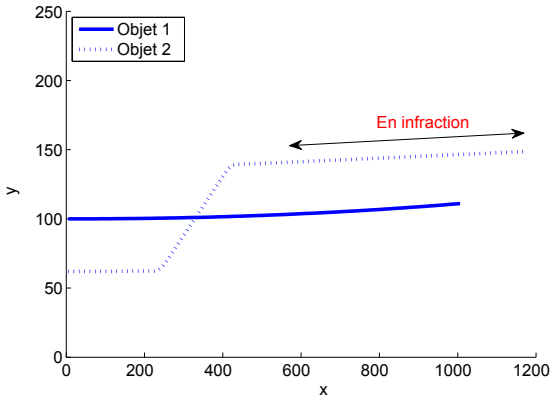
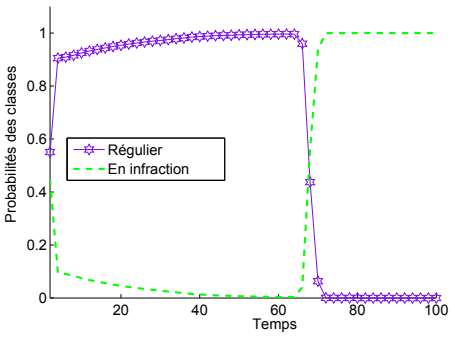
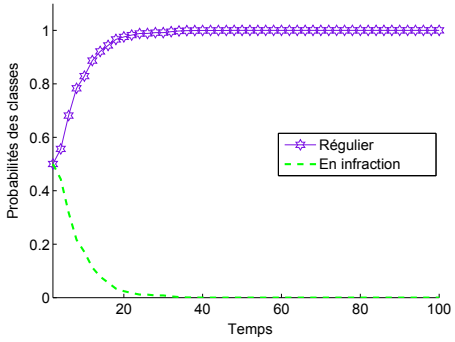


FIGURE 4.18: Reconnaissance des comportements de véhicules sur une autoroute.



(a) Classification locale de l'objet 2 par le capteur 1.



(b) Classification locale de l'objet 2 par le capteur 3.

FIGURE 4.19: Résultats de classifications locales sans consensus.

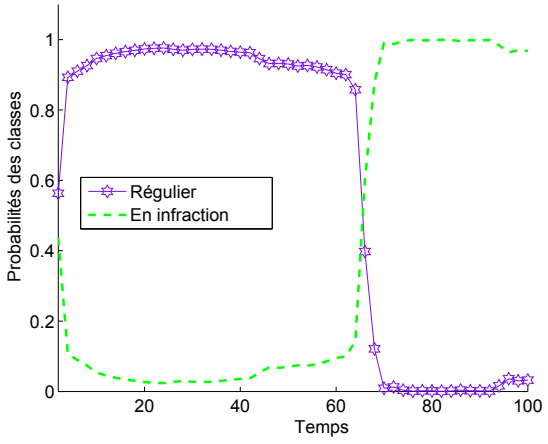


FIGURE 4.20: Classifications avec consensus.

La figure 4.20 montre le résultat de classification obtenu par les trois capteurs à l'issue de l'application de l'algorithme de classification distribuée. Il peut être constaté que l'infraction commise par l'objet

2 est détectée par les trois capteurs.

4.4 Conclusion

Deux solutions multi-capteurs, de suivi et de classification d'objets sont présentées dans ce chapitre. La première est une solution centralisée, elle vise à assurer une robustesse par rapport aux incertitudes des mesures. L'étude présentée consiste à tester et comparer plusieurs stratégies de fusions Bayésiennes et crédales sur un exemple de classification imprécis où les classes sont imbriquées. Cet exemple montre un avantage quant à l'utilisation de la loi de fusion disjonctive crédale. L'aspect conservateur de la fusion disjonctive permet d'avoir de bons résultats dans les cas de grandes imprécisions, ce qui confirme les études théoriques sur les fonctions de croyance [174]. La mise en pratique des solutions de surveillance centralisées est toutefois contrainte à des systèmes à dimensions spatiales limitées dans la mesure où l'ajout de capteurs au système engendre des communications très coûteuses, voir irréalisables, avec le centre de fusion.

La deuxième solution multi-capteurs est dite distribuée. C'est une solution qui a plusieurs avantages, elle permet entre autre de construire une information complète à partir d'informations partielles, à condition que ces dernières soient complémentaires. Un autre avantage d'importance est l'extensibilité, tel qu'il est possible de déployer autant de capteurs que nécessaire sans avoir à effectuer des communications de longues distances, les capteurs ne communiquent qu'avec leurs plus proches voisins. Toutefois, l'application de la solution proposée en temps réel peut être compromise par la lourdeur en terme de calcul de l'étape visant à échanger et ordonner les prédictions des capteurs. Le calcul distribué de cette étape peut faire objet de futures études ainsi que l'extension de la solution globale au cadre crédal.

Conclusion Générale

Le travail présenté dans cette thèse fait partie des études qui s'intéressent à la question : que peuvent apporter les fonctions de croyance au domaine du suivi et de la classification d'objets ?

Les principales contributions ainsi que les perspectives à court et à long termes du travail présenté sont résumées dans ce qui suit.

Aboutissements

Dans le chapitre 1, cette thèse présente un état de l'art des méthodes les plus abouties relevant du domaine de suivi d'objets. L'état de l'art est accompagné de discussions et de comparaisons visant à justifier les techniques utilisées dans l'ensemble des travaux présentés.

Le chapitre 2 propose une solution de suivi multi-objets consistant en une combinaison de plusieurs algorithmes dispersés dans la littérature, notamment les *IMMs*, le *GNN* et les fonctions score. Elle est construite en vue d'estimer les données cinématiques de plusieurs objets tout en tenant compte des principales difficultés concernant leur environnement. Les données cinématiques estimées représentent les données en entrée des algorithmes de classification crédal et Bayésien qui sont comparés à travers des simulations sous Matlab. L'avantage du classifieur crédal par rapport au classifieur Bayésien est alors confirmé dans un cadre multi-objets pour le problème de classification considéré, notamment le cas où les classes sont imbriquées et constantes dans le temps. Ceci étant la première contribution du chapitre 2, la seconde contribution consiste à tester le classifieur crédal sur un nouvel exemple visant à reconnaître les comportements des piétons où les classes dans ce cas-ci se chevauchent et varient dans le temps. Il a été démontré qu'une simple révision de croyances concernant les classes permet au classifieur crédal de s'adapter au changements de

classe des objets. L'application du classifieur crédal dans un cadre multi-objets est toutefois conditionnée par l'algorithme mis en place pour suivre les objets, plus particulièrement par la phase d'association. En effet, comme le montre la section 2.6, il va de soi que si les observations des objets sont confondues, leurs classifications seront altérées, d'où la nécessité de revoir la phase d'association qui est en l'occurrence, étudiée dans le chapitre 3.

Le chapitre 3 présente alors l'étude des méthodes d'association basées sur les fonctions de croyance. Sa contribution est double : en premier temps, il reprend les principes de base des algorithmes d'association basés sur les fonctions de croyance les plus récents, incluant l'algorithme de référence *GNN*, et les compare à travers plusieurs simulations contenant les principales complexités concernant les trajectoires conflictuelles et la gestion des apparitions et des disparitions d'objets. La gestion des apparitions et des disparitions d'objets nécessite souvent la définition d'un paramètre qui est connu dans l'algorithme *GNN* par la distance d'apparition. Une relation entre le paramètre propre au *GNN* et le paramètre propre aux méthodes crédales est par ailleurs formalisée dans la section 3.4. La seconde et majeure contribution du chapitre est la proposition d'une nouvelle méthode à deux variantes qui gère les apparitions et les disparitions d'objets différemment. Au lieu de définir un seuil quelconque duquel depend la décision d'association ou non association, la méthode proposée calcule d'une manière dynamique les poids qui seraient accordés aux hypothèses que des observations données soient de nouveaux objets. Ces derniers prennent en compte les erreurs d'estimation et dote la méthode d'une certaine robustesse prouvée à travers de différentes simulations et benchmark, ce au prix d'un surplus de complexité calculatoire par rapport à l'algorithme *GNN* et l'algorithme proposé par Denœux et *al.* Le chapitre 3 teste les différents algorithmes d'association à travers une panoplie de simulations réalisées sous Matlab, il compare leurs performances par rapport : à la dépendance paramétrique, la sensibilité aux erreurs d'estimation et/ou de mesure et la complexité calculatoire.

Les contributions des chapitres 2 et 3 portent essentiellement sur le suivi et la classification multi-objets, mono-capteur. Le chapitre 4 s'intéresse aux solutions multi-capteurs et donc à la fusion d'informations, il en propose deux. La première solution présentée dans la section 4.2 est une extension de la solution de suivi et de classification présentée dans le chapitre 2 au cadre multi-capteurs. Un environment de capteurs incertains est considéré et l'objectif est alors de comparer des règles de fusion Bayésiennes et crédales visant à assurer une meilleure classification globale des objets. Il s'est avéré que la règle crédale disjonctive est plus adaptée au cadre de l'application étudiée. Une seconde solution multi-capteurs est proposée dans la section 4.3. Cette solution ne fait pas

intervenir de fonctions de croyance mais un algorithme de consensus qui a permis de distribuer les calculs des algorithmes de suivi et de classification présentés dans le chapitre 2. Dans cette solution aucun des capteurs n'observe complètement l'état des objets. Le calcul distribué permet à chaque capteur de construire l'information complète concernant l'état des objets par le simple échange d'informations avec les capteurs qui lui sont voisins, ce à travers les itérations de l'algorithme de consensus. La solution proposée prouve son efficacité sur un exemple qui vise à reconnaître le comportement des véhicules sur une autoroute, ce qui est présenté dans la section 4.3.4.

Perspectives

La principale perspective autour du travail présenté dans le chapitre 2 et 3 est l'élaboration d'une technique d'association multi-scan basé sur la solution proposée dans la section 3.3.6 du chapitre 3. Comme cela est évoqué dans le chapitre sur l'état de l'art, la technique multi-scan permet d'apporter une robustesse par rapport aux fausses alarmes (e.g. ombrage, fausses détections et autres). Cette technique fait la force des méthodes Bayésiennes les plus abouties comme l'algorithme *MHT* ou l'algorithme *PMHT*, nous estimons que les futures méthodes multi-scan crédales pourront apporter un plus par rapport aux imprécisions liées à ce problème de fausses alarmes, mais également dans les situations conflictuelles telles qu'elles sont décrites dans la section 3.1.2 du chapitre 3. La solution apportée dans la littérature à ce problème de trajectoires conflictuelles est bien l'action probabiliste comme dans les approches *JPDA* et *PMHT*, cette action probabiliste évite les décisions d'association déterministes qui sont sujettes à des fausses associations lorsque les trajectoires se confondent. Comme cela était évoqué, ces méthodes proposent de mettre à jour les objets suivis à l'aide d'une somme pondérée de toutes les observations candidates. Cette action est tout à fait possible en utilisant les méthodes d'association crédales actuelles décrites dans le chapitre 3, il suffirait de mettre à jour les objets avec des sommes pondérées des observations où les poids seraient les probabilités pignistiques calculées dans le tableau 3.1 du chapitre 3 par exemple. Dans le cas où on voudrait prendre des décisions déterministes, les approches multi-scan crédales pourraient mieux traiter ce problème de trajectoires conflictuelles par rapport aux approches multi-scan Bayésiennes qui ne font que propager le conflit sans le situer. En effet, dans le cas des méthodes d'association crédales, il est tout à fait possible de distinguer les trajectoires conflictuelles des trajectoires qui ne le sont pas, cela à travers les poids accordés aux ensembles et aux sous-ensembles. Par exemple : si le problème est d'associer une observation z_j à trois objets existants O_1, O_2 et O_3 et qu'un

poids majoritaire est accordé à l'ensemble $\{O_1, O_2\}$ dans la fonction de masse, on saurait que les trajectoires conflictuelles sont celles des objets O_1, O_2 et que celle de l'objet O_3 n'intervient pas dans le conflit, elle peut alors être mise jours d'une manière déterministe. En résumé, cet exemple illustre le problème du conflit partiel évoqué dans la section 3.1.2 du chapitre 3. Sa résolution à l'aide des fonctions de croyance est intuitive alors que dans le cadre Bayésien il faudrait faire appel à des heuristiques qui testeraient les probabilités pour ainsi se prononcer sur les trajectoires qui sont conflictuelles et celles qui ne le sont pas. L'objectif bien sûr est de pouvoir prendre des décisions d'association pour les trajectoires non conflictuelles et de ne s'abstenir que pour les trajectoires conflictuelles.

Sur le plan multi-capteurs, deux perspectives à court terme peuvent être envisagées pour la solution distribuée présentée dans la section 4.3. La première concerne la généralisation de l'étage de classification au cadre des fonctions de croyance. Cette opération peut se faire sans encombre, il suffirait de considérer les vraisemblances calculées d'une manière distribuée (voir la section 4.3.3 du chapitre 4) comme étant des plausibilités et procéder à la classification crédale à l'aide du *GBT*, comme cela est présenté dans le chapitre 2, pour ainsi bénéficier des avantages du classifieur crédal. La seconde perspective concernant la solution multi-capteurs distribuée est bien la phase d'association (*track-to-track*) qui actuellement associe les prédictions faites par les capteurs deux à deux. Cette opération est fastidieuse, elle l'est encore plus avec un nombre de capteurs croissant. Il serait intéressant d'adopter une solution d'association globale traitant toutes les données des capteurs à la fois, on fait ici allusion à des méthodes de *clustring* comme par exemple les méthodes *KNN*, *SVM* ou autres, qui pourraient évidemment traiter les imperfections des distances entre estimations mais également gérer les apparitions et les disparitions d'objets.

Les perspectives venant d'être citées ainsi que les tests des algorithmes présentés dans cette thèse sur des données réelles motivent des travaux à court terme. Des travaux de plus long terme peuvent également être envisagés, ils concerneront la problématique du suivi en elle même qui est actuellement quasi-exclusivement basée sur le formalisme Bayésien. Smets a formalisé dans ces travaux un cadre de fonctions de croyance sur les nombres réels, il en a ainsi pu proposer une version généralisée du filtrage de Kalman par exemple. Il en est hélas avéré que le filtre de Kalman généralisé n'améliore pas le filtre de Kalman classique [175]. D'autres travaux ont également tenté d'aborder le problème du suivi avec des fonctions de croyance [134], cependant aucune méthode n'a pu surpasser les méthodes Bayésiennes. Le challenge pour les fonctions de croyance dans le cadre du suivi multi-objets reste alors entier, il consisterait par exemple :

- à dépasser les contraintes de Gaussienneté et de linéarité adoptées pour le filtrage de Kalman optimal, sachant que pour ce faire les méthodes Bayésiennes font recours aux méthodes d'échantillonnage Monte Carlo.
- Éventuellement relaxer les connaissances *a priori* sur les modèles d'évolution des objets.

D'une manière plus générale, la perspective en question concerne le développement des travaux qui étaient entamés par Smets sur l'inférence d'informations dans un cadre crédal. Smets se basait sur l'interprétation probabiliste des variables aléatoires, à travers le calcul de vraisemblances notamment, comme par exemple dans le cas du *GBT*. D'autres manières d'interpréter les variables aléatoires pourront offrir un cadre d'application digne de la souplesse du formalisme crédal.

Annexe A

Différentes dynamiques d'objets en mouvement

Cette section récapitule les modèles qui sont souvent utilisés pour imiter l'évolution des objets. À défaut de ne pouvoir élaborer un seul modèle exhaustif tenant en compte de tous les mouvements possibles, plusieurs modèles sont développés. Ces modèles sont issus d'équations liées à la dynamique des objets (vitesse constante, accélération, virage et autres comportements). Ces modèles représentent des informations *a priori* utilisées par les algorithmes de filtrage (e.g. Kalman dans le cas mono-modèle, *IMM* dans le cas multi-modèles) afin de suivre les trajectoires d'intérêt. Quelques exemples de modèles sont décrits :

A.1 Modèle de Singer [164]

Le vecteur d'état $x = [x\ v\ a]'$ dans le modèle de Singer est composé de la position (x), la vitesse (v) et l'accélération (a) de l'objet en mouvement. L'évolution de l'accélération dans ce modèle est vue comme étant un processus aléatoire de premier ordre qui peut être donné par :

$$a(k) = \rho_m a(k-1) + \sigma_m \sqrt{1-\rho_m^2} w(k-1), \tag{A.1}$$

avec $\rho_m = e^{-\beta \Delta T}$, $\beta = 1/\tau_m$ où τ_m et σ_m représentent respectivement la constante de temps et l'écart-type de l'erreur de modélisation w.

La matrice d'état qui en découle de cette modélisation est la suivante :

$$F = \begin{pmatrix} 1 & \Delta T & \frac{1}{\beta^2}(-1+\beta\ \Delta T+\rho_m) \\ 0 & 1 & \frac{1}{\beta}(-1+\rho_m) \\ 0 & 0 & \rho_m \end{pmatrix}$$

où ΔT représente la période d'échantillonnage. Dans le cas où cette dernière est inférieure à la constante de temps du mouvement ($\Delta T < \tau_m$), l'approximation suivante peut être faite [23] :

$$F = \begin{pmatrix} 1 & \Delta T & \frac{(\Delta T)^2}{2} \\ 0 & 1 & \Delta T(1-\frac{\Delta T}{2\tau_m}) \\ 0 & 0 & \rho_m \end{pmatrix}$$

La matrice de covariance Q du bruit d'état est donnée par :

$$Q = \frac{2\sigma_m^2}{\tau_m} \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{pmatrix},$$

où

$q_{11} = \frac{1}{2\beta^5}[1 - e^{-2\beta\ \Delta T} + 2\beta\ \Delta T + \frac{2\beta^2\ \Delta T}{3} - 2\beta^2(\Delta T)^2 - 4\beta\ \Delta T e^{-\beta\ \Delta T}]$,

$q_{12} = \frac{1}{2\beta^4}[e^{-2\beta\ \Delta T} + 1 - e^{-\beta\ \Delta T} + 2\beta\ \Delta T e^{-\beta\ \Delta T} - 2\beta\ \Delta T + \beta^2\ (\Delta T)^2]$,

$q_{13} = \frac{1}{2\beta^3}[1 - e^{2\beta\ \Delta T} + 2\beta\ \Delta T e^{-2\beta\ \Delta T}]$,

$q_{22} = \frac{1}{2\beta^3}[4e^{-\beta\ \Delta T} - 3 - 2e^{-2\beta\ \Delta T} + 2\beta\ \Delta T]$,

$q_{23} = \frac{1}{2\beta^2}[e^{-2\beta\ \Delta T} + 1 - 2e^{-\beta\ \Delta T}]$,

$q_{33} = \frac{1}{2\beta}[1 - 2e^{-2\beta\ \Delta T}]$,

Plusieurs simplifications du modèle décrit ci-dessus peuvent être considérées. Notamment, dans le cas où la période d'échantillonnage est négligeable par rapport à la constante de temps du mouvement ($\Delta T << \tau_m$), dans ce cas la matrice d'état F et la matrice de covariance Q deviennent :

$$\lim_{\substack{\beta \\ \Delta T \to 0}} F = \begin{pmatrix} 1 & \Delta T & \frac{(\Delta T)^2}{2} \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{pmatrix}, \tag{A.2}$$

$$\lim_{\substack{\beta \\ \Delta T \to 0}} Q = \frac{2\sigma_m^2}{\tau_m} \begin{pmatrix} \frac{(\Delta T)^5}{20} & \frac{(\Delta T)^4}{8} & \frac{(\Delta T)^3}{6} \\ \frac{(\Delta T)^4}{8} & \frac{(\Delta T)^3}{3} & \frac{(\Delta T)^2}{2} \\ \frac{(\Delta T)^3}{6} & \frac{(\Delta T)^2}{2} & (\Delta T) \end{pmatrix}. \tag{A.3}$$

Dans le cas d'une période d'échantillonnage très grande devant la constante de temps du mouvement ($\Delta T >> \tau_m$), l'estimation de l'accélération ne peut pas être obtenue. Le modèle d'état est alors réduit à l'évolution de la position et la vitesse de l'objet. La matrice d'état F et la matrice de covariance Q sont alors données par :

$$F = \begin{pmatrix} 1 & \Delta T \\ 0 & 1 \end{pmatrix}, \tag{A.4}$$

$$Q = 2\sigma_m^2 \tau_m \begin{pmatrix} \frac{(\Delta T)^3}{3} & \frac{(\Delta T)^2}{2} \\ \frac{(\Delta T)^2}{2} & (\Delta T) \end{pmatrix}.$$

L'expérimentation montre que l'estimation correcte de l'accélération dans le modèle de Singer dans l'equation (A.2) n'est obtenue que si la vitesse de l'objet est mesurée. Autrement, si on ne mesure que la position de l'objet, la période d'échantillonnage ΔT doit être au moins dix fois inférieure à la constante de temps du mouvement τ_m pour permettre une estimation correcte de l'accélération. Dans le cas contraire, il est préférable d'utiliser le modèle (A.4) tenant compte de la position et la vitesse uniquement [23].

A.2 Modèle de vitesse constante et modèle d'accélération constante

Le modèle de vitesse constante obtenu par la simplification du modèle de Singer dans l'équation (A.4) peut être confirmé en prenant la variation de la vitesse comme étant un bruit blanc (bruit borné) [23]. La matrice d'état de ce modèle est similaire à celle donnée par l'équation (A.4) et la matrice de covariance est donnée par :

$$Q = q \begin{pmatrix} \frac{(\Delta T)^3}{3} & \frac{(\Delta T)^2}{2} \\ \frac{(\Delta T)^2}{2} & (\Delta T) \end{pmatrix} \tag{A.5}$$

La matrice de covariance rejoint la modélisation de Singer où il suffit par identification de prendre $q = 2\sigma_m^2 \tau_m$ [23].

De la même manière, le modèle d'accélération constante peut être confirmé en considérant la variation de l'accélération comme étant un bruit blanc. Le modèle d'état qui en découle de cette considération rejoint le modèle de Singer donné par les équations (A.2) et (A.2) respectivement pour la matrice d'état et la matrice de covariance lorsque la constante de temps du mouvement est considérée largement supérieure à la période d'échantillonnage ($\Delta T << \tau_m$).

Le modèle de vitesse constante est le modèle d'accélération constante sont utilisés pour simuler l'évolution des objets dans les travaux présentés dans ce manuscrit. Toutefois, les algorithmes présentés sont indépendants des modèles d'état utilisés. Il peuvent être utilisés avec d'autres modèles. Quelques modèles représentant le movement circulaire des objets sont donnés dans ce qui suit.

A.3 Modèle d'un objet en mouvement circulaire

Un des modèles décrivant le mouvement circulaire d'un objet considère l'évolution de la position et la vitesse (v_x, v_y) sur le plan (x, y) et l'évolution de la vitesse angulaire ω. Le vecteur d'état est alors donné par :

$$x = [x, y, v_x, v_y, \omega]'.$$

Les paramètres contenus dans le vecteur d'état x évoluent selon les équations :

$$x(k+1) = x(k) + \Delta T[SW v_x(k) + CW v_y(k)], \tag{A.6}$$

$$y(k+1) = y(k) + \Delta T[SW v_x(k) + CW v_y(k)], \tag{A.7}$$

où,

$$SW = \frac{\sin \omega\ \Delta T}{\omega\ \Delta T},\ CW = \frac{1 - \cos \omega\ \Delta T}{\omega\ \Delta T}.$$

L'évolution de la vitesse s'effectue comme suit :

$$v_x(k+1) = v_x(k) \cos \omega\ \Delta T - v_y(k) \sin \omega\ \Delta T, \tag{A.8}$$

$$v_y(k+1) = v_x(k) \sin \omega\ \Delta T - v_y(k) \cos \omega\ \Delta T, \tag{A.9}$$

la vitesse angulaire est considérée constante :

$$\omega(k+1) = \omega(k).$$

Il peut être remarqué que le modèle d'état qui en résulte est non linéaire :

$$x(k+1) = f(x(k)). \tag{A.10}$$

La linéarisation du modèle d'état au voisinage de l'état estimé $\hat{x}(k)$ à l'instant k permet de mettre le modèle sous forme de matrice d'état F executable par les filtres de Kalman :

$$F = \frac{\delta f}{\delta x}|_{\hat{x}(k)} = \begin{pmatrix} 1 & (\sin\omega\ \Delta T)/\omega & 0 & -(1-\cos\omega\ \Delta T)/\omega & F_{15} \\ 0 & \cos\omega\ \Delta T & 0 & -\sin\omega\ \Delta T & F_{25} \\ 0 & (1-\cos\omega\ \Delta T)/\omega & 1 & (\sin\omega\ \Delta T)/\omega & F_{35} \\ 0 & \sin\omega\ \Delta T & 0 & \cos\omega\ \Delta T & F_{45} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}_{\hat{x}(k)}. \tag{A.11}$$

Les équations (A.6, A.7, A.8, A.9) permettent d'obtenir les paramètres :

$$F_{15} = \frac{\delta x}{\delta \omega}|_{\hat{x}(k)},\ F_{25} = \frac{\delta v_x}{\delta \omega}|_{\hat{x}(k)},$$

$$F_{35} = \frac{\delta y}{\delta \omega}|_{\hat{x}(k)},\ F_{45} = \frac{\delta v_y}{\delta \omega}|_{\hat{x}(k)}.$$

La matrice de covariance correspondante est donnée par :

$$Q = q \begin{pmatrix} B & 0_{2\times 2} & 0_{2\times 1} \\ 0_{2\times 2} & B & 0_{2\times 1} \\ 0_{1\times 2} & 0_{1\times 2} & \sigma_\omega^2\ (\Delta T)^2/q \end{pmatrix}, \tag{A.12}$$

où q et σ_ω^2 représentent respectivement l'erreur de modélisation de l'accélération et de la vitesse angulaire ω. Le paramètre B est donné par :

$$B = \begin{pmatrix} \frac{(\Delta T)^4}{4} & \frac{(\Delta T)^3}{2} \\ \frac{(\Delta T)^3}{2} & (\Delta T)^2 \end{pmatrix}. \tag{A.13}$$

Ce modèle est représentatif d'un mouvement circulaire uniforme où la vitesse angulaire est supposée constante est l'accélération quasi-nulle, elle est modélisée par un bruit blanc de covariance q.

L'extension des modèles présentés dans cette section à l'espace tri-dimensionnel ainsi que d'autres modèles d'évolution d'objets peuvent être trouvés dans [23, chapitre 4].

Références bibliographiques

[1] M. Adam Medina. *Diagnostic de défauts des systèmes à représentation multi-modèle linéaire invariant dans le temps*. PhD thesis, Nancy 1, 2004.

[2] B.D. Anderson and J.B. Moore. *Optimal filtering*. Courier Dover Publications, 2012.

[3] A. Appriou. Probabilités et incertitude en fusion de données multi-senseurs, in french. *Revue Scientifique et Technique de la Défense*, 11:27–40, 1991.

[4] A. Appriou. Multiple signal tracking processes. *Aerospace science and technology*, 1(3):165–178, 1997.

[5] A. Appriou. Multisensor data fusion in situation assessment processes. In *Qualitative and quantitative practical reasoning*, pages 1–15. Springer, 1997.

[6] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[7] P. Avril. Ukraine : le mh17 s'est disloqué en vol, Figaro : 09 septembre 2014.

[8] A.M.A. Aziz. New data fusion algorithms for distributed multi-sensor multi-target environments. Technical report, DTIC Document, 1999.

[9] Y. Bar-Shalom. On the track-to-track correlation problem. *IEEE Transactions on Automatic Control*, 1981.

[10] Y. Bar-Shalom. *Tracking and data association*. Academic Press Professional, 1987.

[11] Y. Bar-Shalom. Multitarget-multisensor tracking: advanced applications. *Norwood, MA, Artech House*, 1, 1990.

[12] Y. Bar-Shalom and W.D. Blair. *Multitarget/Multisensor Tracking: Applications and Advances*. Artech House, Boston u.a., September 2000.

[13] Y. Bar-Shalom and H. Chen. Multisensor track-to-track association for tracks with dependent errors. In *43rd IEEE Conference on Decision and Control*, volume 3, pages 2674–2679. IEEE, 2004.

[14] Y. Bar-Shalom and X-R Li. Multitarget-multisensor tracking: principles and techniques. *Storrs, CT: University of Connecticut*, 1995.

[15] Y. Bar-Shalom, X.R. Li, T. Kirubarajan, and J. Wiley. *Estimation with applications to tracking and navigation*. Wiley Library, 2001.

[16] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451–460, 1975.

[17] Y. Bar-Shalom, P.K. Willett, and X. Tian. Tracking and data fusion. *A Handbook of Algorithms*, 2011.

[18] M. Baum and U-D. Hanebeck. Association-free tracking of two closely spaced targets. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 62–67. IEEE, 2010.

[19] A. Benavoli and B. Ristic. Classification with imprecise likelihoods: A comparison of TBM, random set and imprecise probability approach. In *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8. IEEE, 2011.

[20] D.P. Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of operations research*, 14(1):105–123, 1988.

[21] S.S. Blackman. *Multiple-target tracking with radar applications*. Artech House Radar Library, 1986.

[22] S.S. Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004.

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

[24] W-D. Blair and G-A. Watson. Benchmark problem for radar resource allocation and tracking maneuvering targets in the presence of ECM. Technical report, DTIC Document, 1996.

[25] W-D. Blair, G-A. Watson, T. Kirubarajan, and Y. Bar-Shalom. Benchmark for radar allocation and tracking in ecm. *IEEE Transactions on Aerospace and Electronic Systems*, 34(4):1097–1114, 1998.

[26] E. Blasch, J. Dezert, and B. Pannetier. Overview of Dempster-Shafer and belief function tracking methods. In *SPIE Defense, Security, and Sensing*, pages 874506–874506. International Society for Optics and Photonics, 2013.

[27] I. Bloch. Information combination operators for data fusion: a comparative review with classification. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 26(1):52–67, 1996.

[28] S. Bonnabel, P. Martin, and E. Salaun. Invariant extended kalman filter: theory and application to a velocity-aided attitude estimation problem. In *Proceedings of the 48th IEEE Conference on Decision and Control CDC/CCC*, pages 1297–1304. IEEE, 2009.

[29] M. Bou Farah, D. Mercier, É. Lefèvre, and F. Delmotte. A high-level application using belief functions for exchanging and managing uncertain events on the road in vehicular ad hoc networks. *annals of telecommunications-annales des télécommunications*, 69(3-4):185–199, 2014.

[30] F. Bourgeois and J-C. Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12):802–804, 1971.

[31] F. Bourgeois and J-C. Lassalle. An extension of the Munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12):802–804, 1971.

[32] D.M. Buede and P. Girardi. A target identification comparison of bayesian and Dempster-Shafer multisensor fusion. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 27(5):569–577, 1997.

[33] F. Caron, B. Ristic, E. Duflos, and P. Vanheeghe. Least committed basic belief density induced by a multivariate gaussian: Formulation with applications. *International Journal of Approximate Reasoning*, 48(2):419–436, 2008.

[34] James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7, 1999.

[35] D-W. Casbeer. *Decentralized Estimation Using Information Consensus Filters With a Multi-Static UAV Radar Tracking System.* PhD thesis, PhD thesis, Brigham Young University, 2009.

[36] D-W. Casbeer and R. Beard. Distributed information filtering using consensus filters. In *American Control Conference*, pages 1882–1887. IEEE, 2009.

[37] K-C. Chang and Y. Bar-Shalom. Joint probabilistic data association for multitarget tracking with possibly unresolved measurements and maneuvers. *IEEE Transactions on Automatic Control*, 29(7):585–594, 1984.

[38] J. Chen, H. Leung, T. Lo, J. Litva, and M. Blanchette. A modified probabilistic data association filter in a real clutter environment. *IEEE Transactions on Aerospace and Electronic Systems*, 32(1):300–313, 1996.

[39] L. Chen, M. Cetin, and A-S. Willsky. Distributed data association for multi-target tracking in sensor networks. 2005.

[40] Z. Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69, 2003.

[41] D.E. Clark and J. Bell. Convergence results for the particle PHD filter. *IEEE Transactions on Signal Processing*, 54(7):2652–2661, 2006.

[42] B.R. Cobb and P.P. Prakash. On the plausibility transformation method for translating belief function models to probability models. *International Journal of Approximate Reasoning*, 41(3):314–330, 2006.

[43] J.L. Crassidis and F. L. Markley. Unscented filtering for spacecraft attitude estimation. *Journal of guidance, control, and dynamics*, 26(4):536–542, 2003.

[44] D.F. Crouse, Y. ak Bar-Shalom, P. Willett, and L. Svensson. The JPDAF in practical systems: Computation and snake oil. In *SPIE Defense, Security, and Sensing*, pages 769813–769813. International Society for Optics and Photonics, 2010.

[45] D.F. Crouse, M. Guerriero, and P. Willett. A critical look at the pmht. *J. Adv. Inf. Fusion*, 4(2):93–116, 2009.

[46] A. Dallil, M. Oussalah, and A. Ouldali. Evidential data association filter. In *13th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, pages 209–217, 2010.

[47] J. Daniel and J-P. Lauffenburger. Multi-object association decision algorithms with belief functions. In *Proceedings of the 15th International Conference on Information Fusion*, pages 669–676. IEEE, 2012.

[48] F. Daum. Tracking and data fusion: Handbook of algorithms (bar-shalom, y., et al; 2011)[book review]. *Aerospace and Electronic Systems Magazine*, 27(12):34–35, 2012.

[49] S.J. Davey. *Extensions to the probabilistic multi-hypothesis tracker for improved data association.* PhD thesis, The University of Adelaide, 2003.

[50] M. De Feo, A. Graziano, R. Miglioli, and A. Farina. IMMJPDA versus MHT and kalman filter with NN correlation: performance comparison. In *IEE Proceedings-Radar, Sonar and Navigation*, volume 144, pages 49–56. IET, 1997.

[51] S. Del Favero. *Analysis and development of consensus-based estimation schemes.* PhD thesis, degli Studi di Padova, 2010.

[52] P. Del Moral and L. Miclo. *Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering.* Springer, 2000.

[53] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339, April 1967.

[54] A. P. Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(2):205–247, January 1968.

[55] T. Denœux. The cautious rule of combination for belief functions and some extensions. In *9th International Conference on Information Fusion*, pages 1–8. IEEE, 2006.

[56] T. Denœux. Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence. *Artificial Intelligence*, 172(2):234–264, 2008.

[57] T. Denœux, N. El Zoghby, V. Cherfaoui, and A. Jouglet. Optimal object association in the Dempster-Shafer framework. *IEEE Transactions on Cybernetics*, 44:2521–2531, 2014.

[58] T. Denœux. Likelihood-based belief function: Justification and some extensions to low-quality data. *International Journal of Approximate Reasoning*, 55(7):1535–1547, October 2014.

[59] S. Destercke, D. Dubois, and E. Chojnacki. Cautious conjunctive merging of belief functions. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 332–343. Springer, 2007.

[60] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[61] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[62] A. Doucet, N. J Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, 2001.

[63] C. Drouet. En 40 ans, une dizaine d'avions civils ont été abattus en vol, Le monde : 18 Juillet 2014.

[64] D. Dubois and H. Prade. Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4(3):244–264, 1988.

[65] M.E. El Najjar and P. Bonnifait. A road-matching method for precise vehicle localization using belief theory and Kalman filtering. *Autonomous Robots*, 19(2):173–191, 2005.

[66] N. El-Sheimy, E-H. Shin, and X. Niu. Kalman filter face-off: Extended vs. unscented kalman filters for integrated gps and mems inertial. *Inside GNSS*, 1(2):48–54, 2006.

[67] N. El Zoghby, V. Cherfaoui, and T. Denoeux. Optimal object association from pairwise evidential mass functions. In *Proceedings of the 16th International Conference on Information Fusion*. IEEE, 2013.

[68] A. Farina, F. Gini, M.V. Greco, and L. Verrazzani. High resolution sea clutter data: statistical analysis of recorded live data. In *Radar, Sonar and Navigation*, volume 144, pages 121–130. IET, 1997.

[69] F. Fayad and K. Hamadeh. Object-to-track association in a multisensor fusion system under the tbm framework. In *In 11th International Conference on Information Sciences, Signal Processing and their Applications (ISSPA'2012)*, pages 1001–1006, Montreal (Quebec), Canada, 2012.

[70] A. Fiche, A. Martin, and autres. Approche bayesienne et fonctions de croyance continues pour la classification. *Rencontre francophone sur la Logique Floue et ses Applications, Annecy, France*, 2009.

[71] J.L. Fisher and D.P. Casasent. Fast JPDA multitarget tracking algorithm. *Applied Optics*, 28(2):371–376, 1989.

[72] B. Fortin, R. Lherbier, and J-C. Noyer. A track-before-detect approach for extended target tracking in multi-lidar systems using a low-level centralized fusion framework. In *17th International Conference on Information Fusion*, pages 1–8. IEEE, 2014.

[73] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *Proceedings of the 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes.*, volume 19, pages 807–812. IEEE, 1980.

[74] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983.

[75] C-W. Gardiner. *Stochastic methods*. Springer-Verlag, 1985.

[76] F. Garin and L. Schenato. A survey on distributed estimation and control applications using linear consensus algorithms. In *Networked Control Systems*, pages 75–107. Springer, 2010.

[77] R. Georgescu, P. Willett, and S. Schoenecker. Gm-cphd and ml-pda applied to the metron multi-static sonar dataset. In *13th Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2010.

[78] N. Gordon. A hybrid bootstrap filter for target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):353–358, 1997.

[79] N.J. Gordon, S. Maskell, and T. Kirubarajan. Efficient particle filters for joint tracking and classification. In *AeroSense*, pages 439–449. International Society for Optics and Photonics, 2002.

[80] N.J. Gordon, D.J. Salmond, and A.F-M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

[81] M.S. Grewal and A.P. Andrews. *Kalman filtering: theory and practice using Matlab.* John Wiley & Sons, 2011.

[82] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.

[83] G. Grisetti, G.D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi. Fast and accurate SLAM with rao-blackwellized particle filters. *Robotics and Autonomous Systems*, 55(1):30–38, 2007.

[84] D. Gruyer and V. Berge-Cherfaoui. Multi-objects association in perception of dynamical situation. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 255–262. Morgan Kaufmann Publishers Inc., 1999.

[85] D. Gruyer and E. Pollard. Credibilistic IMM likelihood updating applied to outdoor vehicle robust ego-localization. In *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8. IEEE, 2011.

[86] M.B. Guldogan, D. Lindgren, F. Gustafsson, H. Habberstad, and U. Orguner. Multi-target tracking with PHD filter using doppler-only measurements. *Digital Signal Processing*, 27:1–11, 2014.

[87] S. Hachour, F. Delmotte, and D. Mercier. Comparison of credal assignment algorithms in kinematic data tracking context. In *15th International conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, pages 200–211, 2014.

[88] S. Hachour, F. Delmotte, and D. Mercier. A new parameterless credal method to track-to-track assignment problem. In *3rd International Conference on Belief Functions*, pages 403–411, Oxford, UK, September 2014.

[89] S. Hachour, F. Delmotte, and D. Mercier. A distributed solution for multi-object tracking and classification. In *17th International Conference on Information Fusion.* IEEE, Salamanca, spain, 2014.

[90] S. Hachour, F. Delmotte, D. Mercier, and E. Lefèvre. Object tracking and credal classification with kinematic data in a multi-target context. *Information Fusion*, 20:174–188, November 2014.

[91] M. Hadzagic, H. Michalska, and A. Jouan. IMM-JVC and IMM-JPDA for closely maneuvering targets. In *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1278–1282. IEEE, 2001.

[92] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[93] A.H. Jazwinski. *Stochastic processes and filtering theory*. Courier Dover Publications, 2007.

[94] S.J. Julier and J.K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, volume 3, pages –. Orlando, FL, 1997.

[95] A. Kallel and S. Hégarat-Mascle. Combination of partially non-distinct beliefs: The cautious-adaptive rule. *International Journal of Approximate Reasoning*, 50(7):1000–1021, 2009.

[96] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[97] R.E. Kalman and R.S. Bucy. New results in linear filtering and prediction theory. *Journal of Fluids Engineering*, 83(1):95–108, 1961.

[98] A-Y. Kibangou. Finite-time average consensus based protocol for distributed estimation over awgn channels. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 5595–5600. IEEE, 2011.

[99] A-Y. Kibangou. Graph Laplacian based matrix design for finite-time distributed average consensus. In *American Control Conference (ACC)*, pages 1901–1906. IEEE, 2012.

[100] 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.

[101] P. Konstantinova, A. Udvarev, and T. Semerdjiev. A study of a target tracking algorithm using global nearest neighbor approach. In *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech-03)*, 2003.

[102] G. Kouemou, C. Neumann, and F. Opitz. Exploitation of track accuracy information in fusion technologies for radar target classification using Dempster-Shafer rules. In *12th International Conference on Information Fusion*, pages 217–223. IEEE, 2009.

[103] H.W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[104] J. Lancaster and S. Blackman. Joint IMM/MHT tracking and identification for multi-sensor ground target tracking. In *9th International Conference on Information Fusion*, pages 1–7. IEEE, 2006.

[105] J-P. Lauffenberger, J. Daniel, and O. Saif. Object-to-track association in a multisensor fusion system under the tbm framework. In *In IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications (ACATTA 2013)*, pages —, Istanbul, Turkey, 2013.

[106] E. Lefèvre, O. Colot, and P. Vannoorenberghe. Belief function combination and conflict management. *Information fusion*, 3(2):149–162, 2002.

[107] H. Li, F. Xiao, J. Zhou, and X-R. Li. Nonlinear distributed estimation fusion that reduces mean square error. In *Proceedings of the 16th International Conference on Information Fusion*, 2013.

[108] W. Li, Y. Jia, J. Du, and J. Zhang. PHD filter for multi-target tracking with glint noise. *Signal Processing*, 94:48–56, January 2014.

[109] Y. Li, H. Xiao, Z. Song, R. Hu, and H. Fan. A new multiple extended target tracking algorithm using PHD filter. *Signal Processing*, 93(12):3578–3588, December 2013.

[110] M. Liggins, D. Hall, and J. Llinas. *Handbook of multisensor data fusion: theory and practice*. CRC press, 2008.

[111] J.S. Liu and R. Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

[112] T.E. Luginbuhl, E.H. Giannopoulos, and P.L. Ainsleigh. A modified jpda. In *9th International Conference on Information Fusion*, pages 1–8. IEEE, 2006.

[113] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71, 2000.

[114] R. Maher. A survey of PHD filter and CPHD filter implementations. In *Proceedings of SPIE*, volume 6567,65670O. SPIE, 2007.

[115] R.P. Mahler. *Statistical multisource-multitarget information fusion*, volume 685. Artech House, 2007.

[116] R.P.S. Mahler, B-T. Vo, and B-N. Vo. CPHD filtering with unknown clutter rate and detection profile. *IEEE Transactions on Signal Processing*, 59(8):3497–3513, 2011.

[117] M. Mallick, J. Krant, and Y. Bar-Shalom. Multi-sensor multi-target tracking using out-of-sequence measurements. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 1, pages 135–142. IEEE, 2002.

[118] M. Marron, J.C. Garcia, M.A. Sotelo, M. Cabello, D. Pizarro, F. Huerta, and J. Cerro. Comparing a Kalman filter and a particle filter in a multiple objects tracking application. In *IEEE International Symposium on Intelligent Signal Processing*, pages 1–6. IEEE, 2007.

[119] A. Martin. La fusion d'informations. *Polycopié de cours ENSIETA-Réf*, 1484:117, 2005.

[120] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: a survey. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(1):103–123, 1998.

[121] S. McGinnity and G.W. Irwin. Multiple model bootstrap filter for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic systems*, 36:1006–1012, 2000.

[122] G-J. McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.

[123] N. Megherbi, S. Ambellouis, O. Colot, and F. Cabestaing. Multimodal data association based on the use of belief functions for multiple target tracking. In *Proceedings of the 8th International Conference on Information Fusion*, volume 2, pages 7–14. IEEE, 2005.

[124] D. Mercier, T. Denœux, and M.H. Masson. Belief function correction mechanisms. *Foundations of Reasoning under Uncertainty*, 249:203–222, 2010.

[125] D. Mercier, É. Lefèvre, and D. Jolly. Object association with belief functions, an application with vehicles. *Information Sciences*, 181(24):5485–5500, 2011.

[126] D. Mercier, B. Quost, and T. Denœux. Refined modeling of sensor reliability in the belief function framework using contextual discounting. *Information Fusion*, 9(2):246–258, 2008.

[127] H-B. Mitchell. *Data Fusion: Concepts and Ideas*. Springer, 2012.

[128] A. Moriya. *Tracking an air target in multistatic radar networks*. PhD thesis, UCL (University College London), 2012.

[129] B. Mourllion, Dominique Gruyer, C. Royere, and S. Théroude. Multi-hypotheses tracking algorithm based on the belief theory. In *8th International Conference on Information Fusion*, volume 2, pages 8–16. IEEE, 2005.

[130] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics*, 5(1):32–38, 1957.

[131] D. Musicki and R. Evans. Linear joint integrated probabilistic data association-LJIPDA. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 3, pages 2415–2420. IEEE, 2002.

[132] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, 39(6):1237–1241, 1994.

[133] D. Musicki and R.obin Evans. Joint integrated probabilistic data association: JIPDA. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1093–1099, 2004.

[134] G. Nassreddine. *Estimation d'état par la théorie des fonctions de croyance: application à la localisation routière*. PhD thesis, Compiègne, 2009.

[135] G. Nassreddine, F. Abdallah, and T. Denœux. A new method for state estimation of dynamic system based on Dempster-Shafer theory. In *International Conference on Advances in Computational Tools for Engineering Applications*, pages 101–106. IEEE, 2009.

[136] G. Nassreddine, F. Abdallah, and T. Denoeux. A state estimation method for multiple model systems using belief function theory. In *12th International Conference on Information Fusion*, pages 506–513. IEEE, 2009.

[137] D.H. Nguyen, J.H. Kay, B.J. Orchard, and R.H. Whiting. Classification and tracking of moving ground vehicles. *Lincoln Laboratory Journal*, 13(2):275–308, 2002.

[138] A. Oka and L. Lampe. Distributed target tracking using signal strength measurements by a wireless sensor network. *IEEE Journal on Selected Areas in Communications*, 28(7):1006–1015, 2010.

[139] R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *44th IEEE Conference on Decision and Control*, pages 8179–8184. IEEE, 2005.

[140] R. Olfati-Saber. Distributed kalman filtering for sensor networks. In *46th IEEE Conference on Decision and Control*, pages 5492–5498. IEEE, 2007.

[141] R. Olfati-Saber, E. Franco, E. Frazzoli, and J-S. Shamma. Belief consensus and distributed hypothesis testing in sensor networks. In *Networked Embedded Sensing and Control*, pages 169–182. Springer, 2006.

[142] A. Olshevsky and J-N. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM review*, 53(4):747–772, 2011.

[143] F. Opitz, T. Kausch, et al. Ukf controlled variable-structure imm algorithms using coordinated turn models. In *Proc. Int. Conf. on Information Fusion*. Citeseer, 2004.

[144] K. Panta, D. E Clark, and B-N. Vo. Data association and track management for the Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):1003–1016, 2009.

[145] M.K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[146] E. Pollard, B. Pannetier, and M. Rombaut. Hybrid algorithms for multitarget tracking using MHT and GM-CPHD. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2):832–847, 2011.

[147] Graham W Pulford and Rob J Evans. A multipath data association tracker for over-the-horizon radar. *IEEE Transactions on Aerospace and Electronic Systems*, 34(4):1165–1183, 1998.

[148] E. Ramasso. *Reconnaissance de séquences d'états par le Modèle des Croyances Transférables. Application à l'analyse de vidéos d'athlétisme.* PhD thesis, Université Joseph-Fourier-Grenoble I, 2007.

[149] E. Ramasso, C. Panagiotakis, M. Rombaut, and D. Pellerin. Belief scheduler based on model failure detection in the TBM framework. application to human activity recognition. *International journal of approximate reasoning*, 51(7):846–865, 2010.

[150] B-S. Rao and H-F. Durrant-Whyte. Fully decentralised algorithm for multisensor kalman filtering. In *Control Theory and Applications*, volume 138, pages 413–420. IET, 1991.

[151] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[152] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*, volume 685. Artech house Boston, 2004.

[153] B. Ristic, D. Clark, and B-N. Vo. Improved SMC implementation of the PHD filter. In *13th Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2010.

[154] B. Ristic, D. Clark, B.-N. Vo, and B.T. Vo. Adaptive target birth intensity for phd and cphd filters. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2):1656–1668, 2012.

[155] B. Ristic, N. Gordon, and A. Bessell. On target classification using kinematic data. *Information Fusion*, 5(1):15–21, 2004.

[156] J.A. Roecker. Multiple scan joint probabilistic data association. *IEEE Transactions on Aerospace and Electronic Systems*, 31(3):1204–1210, 1995.

[157] J.A. Roecker and G.L. Phillis. Suboptimal joint probabilistic data association. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):510–517, 1993.

[158] C. Royère, D. Gruyer, and V. Cherfaoui. Data association with belief theory. In *Proceedings of the 3rd International Conference on Information Fusion*, volume 1, pages TUD2–3. IEEE, 2000.

[159] J.H. Ryu, Kyun K.L. Du H.H., and T.L. Song. Prediction-based interacting multiple model estimation algorithm for target tracking with large sampling periods. *update*, 1(1):1, 2008.

[160] D.J. Salmond. Mixture reduction algorithms for target tracking in clutter. In *OE/LASE'90, 14-19 Jan., Los Angeles, CA*, pages 434–445. International Society for Optics and Photonics, 1990.

[161] N-F. Sandell and R. Olfati-Saber. Distributed data association for multi-target tracking in sensor networks. In *47th IEEE Conference on Decision and Control.*, pages 1085–1090. IEEE, 2008.

[162] G. Shafer. *A mathematical theory of evidence*. Princeton university press, 1976.

[163] D. Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[164] R.A. Singer. Estimating optimal tracking filter performance for manned maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, (4):473–483, 1970.

[165] A. Sinha, Z. Ding, T. Kirubarajan, and M. Farooq. Track quality based multitarget tracking approach for Global Nearest-Neighbor association. *IEEE Transactions on Aerospace and Electronic Systems*, 48(2):1179–1191, 2012.

[166] F. Smarandache and J. Dezert. An introduction to DSm theory of plausible, paradoxist, uncertain, and imprecise reasoning for information fusion. *Octogon Mathematical Magazine*, 15(2):681–722, 2007.

[167] F. Smarandache and J. Dezert. An introduction to DSmt. In *Proc. of Conference on Advances and Applications of DSmT for Information Fusion. Rehoboth, USA: American Research Press*, pages 3–74, 2009.

[168] P. Smets. Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning*, 9(1):1–35, 1993.

[169] P. Smets. What is dempster-shafer's model. *Advances in the Dempster-Shafer theory of evidence*, pages 5–34, 1994.

[170] P. Smets. Decision making in a context where uncertainty is represented by belief functions. In *Belief functions in business decisions*, pages 17–61. Springer, 2002.

[171] P. Smets. Belief functions on real numbers. *International Journal of Approximate Reasoning*, 40(3):181–223, November 2005.

[172] P. Smets. Decision making in the TBM: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning*, 38(2):133–147, 2005.

[173] P. Smets. Analyzing the combination of conflicting belief functions. *Information Fusion*, 8(4):387–412, 2007.

[174] P. Smets and R. Kennes. The transferable belief model. *Artificial intelligence*, 66(2):191–234, 1994.

[175] P. Smets and B. Ristic. Kalman filter and joint tracking and classification based on belief functions in the TBM framework. *Information Fusion*, 8(1):16–27, 2007.

[176] D. Smith and S. Singh. Approaches to multisensor data fusion in target tracking: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 18(12):1696–1710, 2006.

[177] R.L. Streit and T.E. Luginbuhl. Probabilistic multi-hypothesis tracking. Technical report, DTIC Document, 1995.

[178] D. Svensson, J. Wintenby, and L. Svensson. Performance evaluation of MHT and GM-CPHD in a ground target tracking scenario. In *12th International Conference on Information Fusion*, pages 300–307. IEEE, 2009.

[179] A. Tchamova, J. Dezert, T. Semerdjiev, and P. Konstantinova. Target tracking with generalized data association based on the general DSm rule of combination. *Fusion04*, 2004.

[180] G. Trehard, E. Pollard, B. Bradai, and F. Nashashibi. Tracking both pose and status of a traffic light via an interacting multiple model filter. In *17th International Conference on Information Fusion*, pages 1–7. IEEE, 2014.

[181] B-N. Vo and W-K Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, 2006.

[182] B-N. Vo, S. Singh, and A. Doucet. Sequential monte carlo implementation of the phd filter for multi-target tracking. In *Proceeding of the 6th International Conference on Information Fusion*, pages 792–799. IEEE, 2003.

[183] B-N. Vo, S. Singh, and A. Doucet. Sequential monte carlo methods for multitarget filtering with random finite sets. *Aerospace and Electronic Systems, IEEE Transactions on*, 41(4):1224–1245, 2005.

[184] B-T. Vo. *Random finite sets in multi-object filtering*. Citeseer, 2008.

[185] B-T. Vo, B-N. Vo, and A. Cantoni. Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 55(7):3553–3567, 2007.

[186] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[187] R.R. Yager. On the Dempster-Shafer framework and new combination rules. *Information sciences*, 41(2):93–137, 1987.

[188] T. Zajic and R.P.S. Mahler. Particle-systems implementation of the phd multitarget-tracking filter. In *AeroSense 2003*, pages 291–299. International Society for Optics and Photonics, 2003.

[189] H. Zhang, Z. Jing, and S. Hu. Gaussian mixture CPHD filter with gating technique. *Signal Processing*, 89(8):1521–1530, 2009.

[190] K.S. Zhang, X.R. Li, H. Chen, and M. Mallick. Multi-sensor multi-target tracking with out-of-sequence measurements. In *Proc. of International Conf. on Information Fusion*, pages 672–679, 2003.

[191] Y. Zhang and X. Rong Li. Detection and diagnosis of sensor and actuator failures using IMM estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 34(4):1293–1313, 1998.

[192] B. Zhou and N.K. Bose. Multitarget tracking in clutter: fast algorithms for data association. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):352–363, 1993.

Résumé

Suivi et Classification d'Objets Multiples : Contributions avec la Théorie des Fonctions de Croyance

Cette thèse aborde le problème du suivi et de la classification de plusieurs objets simultanément. Il est montré dans la thèse que les fonctions de croyance permettent d'améliorer les résultats fournis par des méthodes classiques à base d'approches Bayésiennes. En particulier, une précédente approche développée dans le cas d'un seul objet est étendue au cas de plusieurs objets. Il est montré que dans toutes les approches multi-objets, la phase d'association entre observations et objets connus est fondamentale. Cette thèse propose également de nouvelles méthodes d'association crédales qui apparaissent plus robustes que celles trouvées dans la littérature. Enfin, est abordée la question de la classification multi-capteurs qui nécessite une seconde phase d'association. Dans ce dernier cas, deux architectures de fusion des données capteurs sont proposées, une dite centralisée et une autre dite distribuée. De nombreuses comparaisons illustrent l'intérêt de ces travaux, que les classes des objets soient constantes ou variantes dans le temps.

Mots clés : suivi et classification d'objets, association de données, fonctions de croyance, fusion multi-capteurs.

Abstract

Multi-Object Tracking and Classification: Contributions with Belief Functions Theory

This thesis deals with multi-objet tracking and classification problem. It was shown that belief functions allow the results of classical Bayesian methods to be improved. In particular, a recent approach dedicated to a single object classification which is extended to multi-object framework. It was shown that detected observations to known objects assignment is a fundamental issue in multi-object tracking and classification solutions. New assignment solutions based on belief functions are proposed in this thesis, they are shown to be more robust than the other credal solutions from recent literature. Finally, the issue of multi-sensor classification that requires a second phase of assignment is addressed. In the latter case, two different multi-sensor architectures are proposed, a so-called centralized one and another said distributed. Many comparisons illustrate the importance of this work, in both situations of constant and changing objects classes.

Key words: multi-object tracking and classification, data assignment, belief functions, multi-sensor data fusion.