

Auto-localisation et suivi de cibles dans les réseaux de capteurs mobiles

THÈSE

pour l'obtention du grade de

Docteur de l'Université de Technologie de Troyes
(spécialité Optimisation et Sécurité des Systèmes)

présentée et soutenue par

Farah Mourad

le 7 décembre 2010

Composition du jury

<i>Rapporteurs :</i>	Luc Jaulin	PU à l'ENSIETA
	Nacim Ramdani	PU à l'université d'Orléans
<i>Examineurs :</i>	Michel Kieffer	MdC (HDR) à l'université Paris-Sud
	Fahed Abdallah	MdC à l'université de technologie de Compiègne
<i>Directeurs de thèse :</i>	Cédric Richard	PU à l'université de Nice Sophia-Antipolis
	Hichem Snoussi	PU à l'université de technologie de Troyes

A LA MÉMOIRE DE MON GRAND-PÈRE,

A MES PARENTS ET MES SŒURS,

A MON FIANCÉ

Remerciements

Je tiens tout d'abord à remercier mes directeurs de thèse, Hichem Snoussi et Cédric Richard, pour la confiance et l'aide qu'ils m'ont accordées. Tout au long de ces trois années, ils ont su orienter mes recherches aux bons moments tout en me laissant une bonne marge d'autonomie. Pour leur soutien, leur disponibilité et surtout leur amitié, je les remercie vivement.

Un grand merci aux membres du jury pour leur participation à l'examen de cette thèse, en particulier Luc Jaulin et Nacim Ramdani en leur qualité de rapporteurs et Michel Kieffer et Fahed Abdallah en leur qualité d'examineurs.

Je remercie également Fahed Abdallah pour sa participation dans l'orientation de cette thèse et pour ses conseils. Je remercie en outre Farouk Yalaoui et Lionel Amodeo. Les collaborations que nous avons menées ensemble au cours de cette thèse m'ont été très enrichissantes que ce soit au niveau professionnel ou personnel.

Mes remerciements à Christophe Bérenguer, le directeur du laboratoire LM2S, Antoine Grall, le directeur du pôle ROSAS, et les secrétaires Marie-José Rousselet et Véronique Banse pour m'avoir accueillie au sein de l'UTT.

Une pensée chaleureuse à tous les amis de l'UTT avec lesquels j'ai partagé une salle, un café ou un repas, notamment Amélie, Paul, Frédéric, Farah, Naim, Jing, Ahmad, Elias, Zineb, Slim et plein d'autres. Un grand merci ému à mes meilleures amies, de plus ou moins longue date, en particulier Nisrine, Sahar, Mayassa, Zeina, Judy, Diala et Randa.

Je ne peux pas manquer l'occasion de remercier ma famille : mon père Souheil, ma mère Maha, mes sœurs Nour et Maya, ma tante Faten, ma grand-mère Hadia, mes beaux-parents Mohamad et Fatmé et mes beaux-frères Wissam, Hilal et Samer. Un immense merci pour leur amour, leur soutien et leurs encouragements.

Je remercie mon cher fiancé Hicham qui a su faire preuve de beaucoup de patience et de compréhension surtout pour la période de rédaction. Merci pour ton soutien et ton amour.

Enfin merci à toutes les personnes que je n'ai pas citées ici et qui se reconnaîtront dans ces quelques lignes...

Résumé

Cette thèse porte sur les problèmes d'auto-localisation et de suivi de cibles dans les réseaux de capteurs mobiles. L'étude de ces problèmes se situe dans le cadre de la théorie des intervalles, permettant ainsi de représenter les incertitudes par des intervalles. Nous traitons d'abord dans ce mémoire le problème d'auto-localisation dans des environnements parfaits. Nous présentons ainsi des méthodes basées sur les mesures de connectivités, relatives aux ancrés ou aux non-ancrés. Deux extensions de ces techniques, profitant des ancrés plus éloignées ou faisant de la rétropropagation, sont également présentées. Des méthodes sans modèle, plus robustes que celles basées sur le modèle de connectivité, sont ensuite proposées. Ces dernières effectuent une comparaison des puissances des signaux échangés afin de définir les mesures d'observation. Nous considérons par la suite le problème d'auto-localisation dans les environnements imparfaits. Nous supposons alors que les fiabilités des mesures sont connues. La solution du problème se situe à la fois dans le cadre de la théorie des intervalles et de la théorie des croyances. La deuxième problématique traitée est le suivi de cibles en cas de capteurs à mobilité contrôlée. Une stratégie de gestion de mobilité est alors présentée afin d'optimiser l'estimation de la position de la cible. Cette stratégie réalise un repositionnement des capteurs à l'aide d'une technique de triangulation suivie d'une optimisation par colonies de fourmis.

Mots-clé :

- Réseaux de capteurs (technologie)
- Calcul sur des intervalles
- Estimation, Théorie de l'
- Traitement du signal

Table des matières

Table des figures	xiii
Liste des tableaux	xvii
I Introduction	1
I.1 Réseaux de capteurs sans fil	2
I.1.1 Description des réseaux de capteurs sans fil	2
I.1.2 Capteurs intelligents	2
I.1.3 Topologies des réseaux de capteurs	4
I.1.4 Applications	6
I.2 Problème de localisation	8
I.2.1 Description du problème	8
I.2.2 Méthodes de localisation	9
I.3 Problème de suivi de cibles	10
I.3.1 Description du problème	11
I.3.2 Méthodes de suivi de cibles	11
I.4 Réseaux de capteurs mobiles	14
I.4.1 Mobilité des capteurs	14
I.4.2 Quelques applications des réseaux de capteurs mobiles	15
I.4.3 Modèles de mobilité	17
I.4.4 Localisation des capteurs mobiles	21
I.4.5 Suivi de cibles dans un réseau mobile	22
I.5 Contributions et organisation du document	23
I.5.1 Plan du manuscrit	23
I.5.2 Publications	24
II Théorie des intervalles	27
II.1 Introduction	28
II.2 Définition des intervalles	28
II.3 Outils de base de la théorie des intervalles	29
II.3.1 Opérations ensemblistes	29
II.3.2 Opérations arithmétiques	31
II.3.3 Fonctions d'inclusion	32

TABLE DES MATIÈRES

II.3.4	Tests d'inclusion	35
II.4	Sous-pavages	36
II.4.1	Sous-pavage régulier	36
II.4.2	Représentation par un arbre binaire	37
II.4.3	Algorithme d'inversion ensembliste : SIVIA	38
II.5	Contracteurs	41
II.5.1	Définition des contracteurs	42
II.5.2	Contracteur basé sur l'élimination de Gauss	42
II.5.3	Contracteur basé sur l'algorithme de Gauss-Seidel	45
II.5.4	Contracteur basé sur la méthode de Krawczyk	46
II.5.5	Contracteur basé sur la propagation <i>forward-backward</i>	47
II.6	Conclusion	48
III	Auto-localisation par connectivités	51
III.1	Introduction	52
III.2	Problématique	53
III.2.1	Modèle de mobilité	53
III.2.2	Modèle d'observation généralisé	53
III.3	Théorie des intervalles	54
III.4	Auto-localisation par connectivités vis-à-vis des ancrés	55
III.4.1	Mesures de connectivités	55
III.4.2	Définition du problème	56
III.4.3	Algorithme	56
III.4.4	Versions simplifiées de l'algorithme	63
III.5	Auto-localisation <i>back-propagated</i>	65
III.6	Auto-localisation <i>multi-hop</i>	66
III.6.1	Modèle d'observation	68
III.6.2	Algorithme	69
III.7	Auto-localisation basée sur les ancrés et les non-ancrés	73
III.8	Simulations	75
III.8.1	Evaluation de la méthode par connectivités basées sur les ancrés	78
III.8.2	Evaluation de la méthode <i>back-propagated</i>	82
III.8.3	Evaluation de la méthode <i>multi-hop</i>	83
III.8.4	Evaluation de la méthode utilisant les non-ancrés	84
III.9	Conclusion	85
IV	Auto-localisation sans modèle	87
IV.1	Introduction	88
IV.2	Problématique	88
IV.2.1	Mesures d'observation	89
IV.2.2	Modèle de mobilité	91
IV.2.3	Problème d'auto-localisation	91
IV.3	Auto-localisation sans modèle basée sur les ancrés	92
IV.3.1	Algorithme	92

IV.3.2 Versions simplifiées de l’algorithme	94
IV.4 Auto-localisation sans modèle utilisant les non-ancres	99
IV.4.1 Mesures d’observation	99
IV.4.2 Algorithme	100
IV.5 Simulations	103
IV.5.1 Evaluation de la méthode ALAA	105
IV.5.2 Efficacité de l’algorithme de Waltz	107
IV.5.3 Comparaison de la méthode ALAA à la méthode ALCA	107
IV.5.4 Comparaison de ALAA à des méthodes de Monte-Carlo	108
IV.5.5 Comparaison de la méthode ALAAN à la méthode ALAA	109
IV.5.6 Sensibilité de la méthode ALAAN à la densité d’ancres	112
IV.6 Conclusion	112
V Auto-localisation en présence d’imperfections de mesures	115
V.1 Introduction	116
V.2 Théorie des fonctions de croyance	116
V.2.1 Introduction à la TFC	117
V.2.2 Représentation des informations dans le TBM	117
V.2.3 Règles de combinaison	120
V.2.4 Prise de décision	121
V.3 Problématique	122
V.3.1 Modèle de mobilité	122
V.3.2 Modèle d’observation	123
V.4 Auto-localisation avec des mesures aberrantes	125
V.4.1 Description de la méthode basée sur la TFC	125
V.4.2 Variante de la méthode	130
V.5 Simulations	134
V.5.1 Evaluation de la méthode	135
V.5.2 Sensibilité aux valeurs de fiabilité	137
V.5.3 Comparaison à la méthode ALCA	139
V.5.4 Comparaison à la méthode MCB	140
V.6 Conclusion	141
VI Suivi de cibles dans les réseaux mobiles	145
VI.1 Introduction	146
VI.2 Estimation de la position de la cible	147
VI.2.1 Problématique	147
VI.2.2 Algorithme	148
VI.3 Prédiction de la position de la cible	150
VI.4 Définition des positions des capteurs	150
VI.4.1 Capteurs fixes	151
VI.4.2 Capteurs mobiles	152
VI.5 Optimisation du repositionnement des capteurs	157
VI.5.1 Algorithme d’optimisation par colonies de fourmis	157

TABLE DES MATIÈRES

VI.5.2	Repositionnement des capteurs	160
VI.6	Simulations	160
VI.6.1	Comparaison à une méthode statique basée sur les intervalles . . .	162
VI.6.2	Influence de la distance inter-capteurs sur les performances . . .	164
VI.6.3	Evaluation du modèle de prédiction	166
VI.6.4	Efficacité de la méthode ACO	167
VI.7	Conclusion	169
VII	Conclusion et perspectives	171
VII.1	Résumé et contributions	172
VII.2	Travaux futurs et suggestions	173
VII.2.1	Auto-localisation avec un maximum de q mesures aberrantes . . .	173
VII.2.2	Autres perspectives	175
	Références	177
	177

Table des figures

I.1	Schéma fonctionnel d'un capteur intelligent.	3
I.2	Des capteurs intelligents de plus en plus intégrés.	4
I.3	Topologies de réseau en étoile en (a), en grille en (b) et hybride en (c).	5
I.4	Applications des réseaux de capteurs.	7
I.5	Illustrations de la mobilité contrôlée en (a) et de la mobilité non-contrôlée en (b).	15
I.6	Projet ZebraNet.	16
I.7	Exemple de capteurs développés pour la Great Duck Island à gauche et des pétrels cul-blanc à droite.	16
I.8	Fonctionnement d'un réseau de capteurs installé pour le suivi des patients.	17
I.9	Illustration du Random Walk Mobility Model (à période constante).	18
I.10	Illustration du Random Waypoint Mobility Model.	19
I.11	Illustration du Nomadic Community Mobility Model.	20
I.12	Illustration du Reference Point Group Mobility Model.	20
II.1	Exemples sur les opérations \cap et \setminus pour les pavés.	31
II.2	Illustration des fonctions d'inclusion.	33
II.3	Illustration de l'exemple 4.	34
II.4	Quatre fonctions d'inclusion naturelles de la même fonction.	35
II.5	Pavage et sous-pavage (en gris) réguliers du pavé $[0, 2] \times [0, 2]$	37
II.6	Arbre binaire associé au sous-pavage en gris de FIG. II.5.	38
II.7	Quatre situations rencontrées par l'algorithme SIVIA.	40
II.8	Illustration de l'exemple 9.	41
III.1	Etape de propagation.	57
III.2	Auto-localisation par connectivités vis-à-vis des ancrés.	62
III.3	Effet de la simplification partielle sur l'algorithme d'auto-localisation.	63
III.4	Effet de la simplification totale sur l'algorithme d'auto-localisation.	64
III.5	Auto-localisation <i>back-propagated</i> : Propagation et correction en (a) et rétropropagation en (b).	66
III.6	Contraintes d'observation de la méthode <i>multi-hop</i>	70

TABLE DES FIGURES

III.7	Auto-localisation <i>multi-hop</i>	72
III.8	Influence des erreurs de seuillage sur les mesures d'observation en (a) et l'effet de l'utilisation d'intervalles sur les seuils en (b).	73
III.9	Perte de la garantie en (a) et correction de l'erreur en (b).	74
III.10	Auto-localisation basée sur les ancrés et les non-ancrés.	75
III.11	Mouvement de groupe de 50 capteurs.	77
III.12	Pavés estimés à l'aide des méthodes ALCA et ALCA _t	79
III.13	Rapports des surfaces des pavés obtenus avec ALCA, ALCA _p et ALCA _t	79
III.14	Variation de l'erreur d'estimation et de la surface des pavés en fonction du paramètre v_{max}	80
III.15	Pavés estimés avec différentes valeurs du paramètre v_{max}	80
III.16	Courbe de l'erreur d'estimation en fonction du paramètre r	81
III.17	Pavés estimés avec différentes valeurs du paramètre r	81
III.18	Estimation à l'aide de la méthode ALCA et la méthode MCB avec $N = 50$	82
III.19	Estimation à l'aide de la méthode ALCA et la méthode MCB avec $N = 500$	82
III.20	Variations de la surface moyenne des pavés et de l'erreur moyenne en fonction de K_{BP} avec la méthode ALBP.	83
III.21	Pavés obtenus à l'aide de la méthode ALCA et la méthode ALBP avec $K_{BP} = 5$	83
III.22	Variations de la surface moyenne et de l'erreur moyenne en fonction de K_{MH} avec la méthode ALMH.	84
III.23	Pavés obtenus à l'aide de la méthode ALCA et la méthode ALMH avec $K_{MH} = 7$	84
III.24	Comparaison de la méthode ALCAN à la méthode ALCA.	85
III.25	Comparaison de la méthode ALCAN à la méthode ALMH.	85
IV.1	Génération d'un anneau.	89
IV.2	Définition des rayons interne et externe d'un anneau.	90
IV.3	Etape de propagation.	93
IV.4	Anneau d'observation dans le contexte ensembliste.	94
IV.5	Auto-localisation sans modèle basée sur les ancrés.	96
IV.6	Version à simplification partielle de l'auto-localisation sans modèle basée sur les ancrés.	97
IV.7	Simplification d'un anneau d'observation.	98
IV.8	Version à simplification totale de l'auto-localisation sans modèle basée sur les ancrés.	98
IV.9	Génération des anneaux à l'aide des ancrés et des non-ancrés.	101
IV.10	Auto-localisation sans modèle utilisant les ancrés et les non-ancrés.	105
IV.11	Mouvement de groupe de 50 capteurs.	106
IV.12	Pavés estimés à l'aide des méthodes ALAA et ALAA _t	107

IV.13	Rapports des surfaces des pavés obtenus avec ALAA, ALAAp et ALAAt.	107
IV.14	Variation de l'erreur moyenne et de la surface moyenne en fonction du nombre maximal d'itérations permises dans l'algorithme de Waltz.	108
IV.15	Pavés obtenus avec la méthode ALAA où les contraintes ne sont pas itérées ($K_W = 1$) et celle avec l'algorithme de Waltz avec $K_W = 5$.	108
IV.16	Pavés estimés à l'aide des méthodes ALAA et ALCA.	109
IV.17	Rapports des erreurs d'estimation et des surfaces des pavés obtenus avec ALAA et ALCA.	109
IV.18	Illustration des pavés estimés à l'aide de la méthode ALAA et des particules issues de la méthode MCB.	110
IV.19	Illustration des pavés estimés à l'aide de la méthode ALAA et des particules issues de la méthode MMCB.	110
IV.20	Pavés estimés à l'aide des méthodes ALAAN et ALAA.	111
IV.21	Variations des rapports des erreurs et des surfaces des pavés obtenues avec ALAAN sur celles obtenues avec ALAA.	111
IV.22	Rapports moyens des erreurs et des surfaces des pavés en fonction du nombre d'ancres.	113
IV.23	Rapports moyens des erreurs et des surfaces des pavés en fonction du nombre de nœuds.	113
V.1	Propagation des ensembles focaux précédents.	126
V.2	Illustration d'une FM catégorique issue d'une source fiable.	127
V.3	Exemple d'un problème d'auto-localisation.	129
V.4	Combinaison des observations.	133
V.5	Combinaison des pavés d'observations aux pavés de mobilité.	134
V.6	Mouvement de groupe de 30 nœuds et 15 ancres.	135
V.7	Nombres de mesures correctes et erronées obtenues à chaque instant.	136
V.8	Valeurs de fiabilité accordées aux mesures correctes (flèches vers le haut) et erronées (flèches vers le bas), illustrées aux instants impairs.	136
V.9	Pavés obtenus à l'aide de notre méthode.	137
V.10	Pavés estimés obtenus à l'aide des méthodes ALR et ALRa.	138
V.11	Pavés obtenus à l'aide de notre méthode avec des fiabilités supérieures à 0.5.	139
V.12	Nombre des mesures erronées et indicateur de garantie de notre méthode avec des fiabilités supérieures à 0.5.	139
V.13	Illustrations des contraintes aux instants $t = 13s$ à gauche et $t = 87s$ à droite.	140
V.14	Pavés obtenus à l'aide des méthodes ALR et ALCA.	141
V.15	Pavés et particules obtenus à l'aide des méthodes ALR et MCB.	142
V.16	Pavés et particules obtenus à l'aide des méthodes ALR et MCB avec des fiabilités des mesures erronées allant de 0 à 1.	142
VI.1	Estimation de la position de la cible.	149

TABLE DES FIGURES

VI.2	Illustration d'un arrangement de capteurs fixes.	152
VI.3	Dépendance de la précision de l'estimation à la longueur des côtés du triangle.	152
VI.4	Un illustration d'un triangle de capteurs.	153
VI.5	Un exemple d'une structure de dix capteurs.	155
VI.6	Un exemple d'une structure garantie en (a) et d'une autre plus précise mais moins garantie en (b) permettant de couvrir un même pavé prédit.	157
VI.7	Une illustration de la trajectoire de la cible avec des capteurs fixes uniformément répartis.	162
VI.8	Pavés estimés à l'aide des méthodes SMC et SSt.	163
VI.9	Rapports des surfaces des pavés en haut et des erreurs d'estimation en bas SMC/SSt.	164
VI.10	Variations du temps de calcul moyen par pas de temps en haut, du rapport moyen des surfaces des pavés au milieu et du rapport moyen des erreurs d'estimation SMC/SSt en bas en fonction du nombre total des capteurs considérés dans le réseau.	165
VI.11	Rapports des surfaces des pavés en haut et des erreurs d'estimation en bas SMCp/SMCg.	165
VI.12	Capteurs mobiles, position réelle et pavé estimé avec SMCp à l'instant $t = 28$	166
VI.13	Capteurs mobiles, position réelle et pavé estimé avec SMCg à l'instant $t = 28$	166
VI.14	Capteurs mobiles, position réelle et pavé estimé avec SMCp à l'instant $t = 14$	167
VI.15	Capteurs mobiles, position réelle et pavé estimé avec SMCg à l'instant $t = 14$	167
VI.16	Trajectoire non-prédictible de la cible et pavés estimés.	168
VI.17	Capteurs mobiles, positions réelles et pavés prédits et estimés aux instants $t = 18$ (en noir) et $t = 19$ (en rouge).	168
VI.18	Trajectoire de la cible et pavés estimés obtenus pareillement à l'aide de la méthode ACO ou la méthode exacte.	169
VI.19	Rapports des temps de calcul et des distances parcourues avec la méthode ACO sur ceux obtenus avec la méthode exacte.	170
VII.1	Illustration de la méthode basée sur SIVIA.	174
VII.2	Illustration de la méthode combinatoire.	174

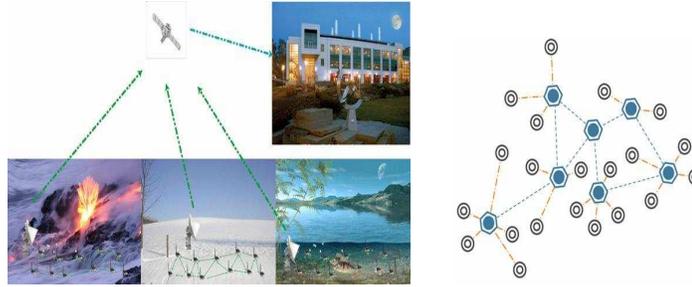
Liste des tableaux

I.1	Quelques compagnies développant des capteurs intelligents.	4
I.2	Caractéristiques des protocoles Zigbee, Bluetooth et Wi-Fi.	5
II.1	Quelques contracteurs de base.	42
III.1	Comparaison des différentes versions de la méthode ALCA.	79
III.2	Comparaison de la méthode ALCA à la méthode MCB basée sur Monte-Carlo.	82
III.3	Comparaison de la méthode ALCA à la méthode ALBP avec $K_{BP} = 5$	83
III.4	Comparaison de la méthode ALCA à la méthode ALMH avec $K_{MH} = 7$	84
IV.1	Comparaison des différentes versions de la méthode ALAA.	106
IV.2	Comparaison des méthodes ALAA et ALCA.	108
IV.3	Comparaison de la méthode ALAA aux méthodes MCB et MMCB.	109
V.1	Quatre exemples de FM's définies sur $\Omega = \{\omega_1, \omega_2, \omega_3\}$	118
V.2	Valeurs de croyance, plausibilité, implicabilité et communalité associées à la FM normale du TAB. V.1	120
V.3	Combinaisons, ensembles focaux et masses correspondant au problème de FIG. V.3	129
V.4	Combinaisons des observations du problème de FIG. V.3	133
V.5	Comparaison des deux versions de la méthode ALR.	137
V.6	Comparaison des deux méthodes ALR et ALCA.	140
V.7	Comparaison des deux méthodes ALR et MCB.	143

LISTE DES TABLEAUX

I

Introduction



Sommaire

- I.1 Réseaux de capteurs sans fil**
- I.2 Problème de localisation**
- I.3 Problème de suivi de cibles**
- I.4 Réseaux de capteurs mobiles**
- I.5 Contributions et organisation du document**

Les réseaux de capteurs sans fil sont des réseaux composés d'un grand nombre de micro-capteurs, dits aussi capteurs intelligents. En plus des unités de communication, ils intègrent des capteurs physiques et des unités de traitement et de stockage de données. La principale contrainte des micro-capteurs est leurs sources d'énergie limitées puisqu'ils sont munis de batteries non renouvelables. Ainsi, afin de maximiser la durée de vie du réseau, tous les algorithmes de traitement de données, les protocoles de communication, le matériel intégré, etc. doivent être réduits en consommation d'énergie. L'absence d'infrastructure fixe permettra aux capteurs sans fil de changer de positions. Il s'agit alors des réseaux de capteurs mobiles, dont la mobilité peut être contrôlée ou non contrôlée. Ce chapitre décrit en premier lieu les réseaux de capteurs sans fils, ainsi que deux des problèmes les plus traités dans ce domaine : l'auto-localisation et le suivi de cibles. Il introduit par la suite les réseaux de capteurs mobiles ainsi que les problèmes d'auto-localisation et de suivi de cibles en cas de mobilité. Il expose enfin les principales contributions de ce travail et l'organisation du manuscrit.

I.1 Réseaux de capteurs sans fil

Grâce aux progrès récents de la microélectronique, la microtechnique et les communications sans fil, une nouvelle technologie a été élaborée : les Réseaux de Capteurs Sans Fil (RCSF). Ils viennent remplacer les réseaux de capteurs filaires où l'acheminement des données observées se faisait par le moyen du câblage coûteux et encombrant. Selon le magazine Technology Review du MIT, le réseau de capteurs sans fil est l'une des dix nouvelles technologies qui bouleverseront le monde et notre manière de vivre et de travailler. Dans cette section, nous commençons par décrire les RCSF ainsi que leurs éléments de base : les capteurs intelligents. Nous définissons par la suite les différentes topologies de réseaux. Nous introduisons enfin diverses applications concernant les RCSF.

I.1.1 Description des réseaux de capteurs sans fil

Un réseau de capteurs sans fil est un réseau composé d'un grand nombre de micro-capteurs de quelques millimètres cubes de volume. Ces derniers sont munis d'un capteur physique (capteurs de chaleur, d'humidité, de vibrations,...), d'unités de traitement et de stockage de données et d'un module de communication sans fil [Akyildiz *et al.*, 2002a; Culler *et al.*, 2004; Sohraby *et al.*, 2007]. Ils sont autonomes à durée de vie limitée, due principalement à leurs batteries non renouvelables. Il conviendra ainsi de veiller à réduire au maximum la consommation d'énergie afin de maximiser la durée de vie du réseau.

Les réseaux de capteurs présentent des intérêts considérables pour le secteur industriel, mais aussi pour les organisations civiles où la surveillance et la reconnaissance de phénomènes physiques est une priorité [Karl et Willig, 2003; Czubak et Wojtanowski, 2009]. Ainsi, un réseau de capteurs peut être mis en place dans le but de surveiller une zone géographique plus ou moins étendue pour détecter l'apparition de phénomènes particuliers ou mesurer une grandeur physique (température, pression, humidité,...). En plus de ces applications civiles, il existe évidemment des applications militaires des réseaux de capteurs notamment dans la détection d'intrusion, la localisation des combattants sur un champ de bataille, etc.

Dans le domaine de la recherche, les réseaux de capteurs posent un certain nombre de défis scientifiques. Les chercheurs électroniciens sont confrontés à de nombreux défis au niveau de la taille des capteurs (miniaturisation maximale), mais aussi au niveau des performances. En informatique, les enjeux se situent au niveau du routage, des protocoles de communication et des architectures logicielles. Toutes ces recherches vont dans le même sens : réduire au maximum la consommation d'énergie du réseau tout en optimisant les performances.

I.1.2 Capteurs intelligents

Un réseau de capteurs sans fil est composé d'un ensemble de capteurs miniaturisés et peu coûteux, dits *micro-capteurs* ou *capteurs intelligents* [Akyildiz *et al.*, 2002a;

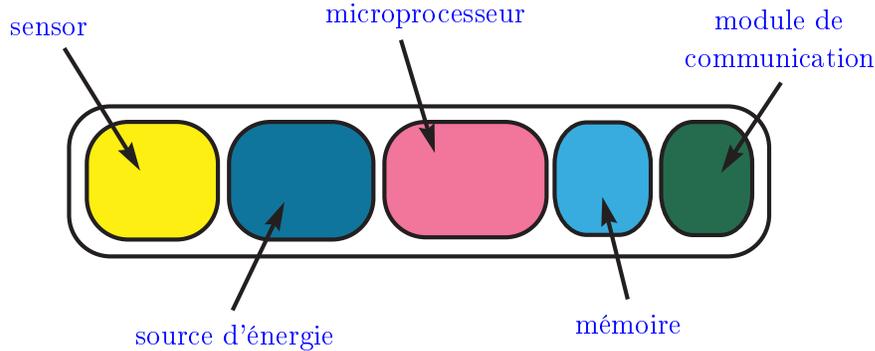


FIG. I.1: Schéma fonctionnel d'un capteur intelligent.

[Bharathidasan et Ponduru, 2002; Vieira *et al.*, 2003]. Ces composants sont formés principalement de cinq unités, comme suit :

- un **capteur physique**, dit aussi **sensor**, ou unité de captage qui permet de capter des grandeurs physiques (chaleur, humidité, vibrations, rayonnement, etc.) et de les transformer en données numériques,
- une **source d'énergie**, notamment des batteries non renouvelables,
- un **microprocesseur** ou unité de traitement qui a pour rôle de traiter les données numériques,
- une **mémoire** ou unité de stockage qui permet de sauvegarder des informations,
- et un **module de communication** qui permet de transmettre et de recevoir des signaux du voisinage.

Le schéma fonctionnel du capteur intelligent est illustré dans FIG. I.1.

Des recherches intenses ont été menées dans le but de miniaturiser les capteurs tout en maximisant leurs performances et leurs durées de vie. FIG. I.2 montre des exemples de capteurs de plus en plus intégrés. Ainsi, plusieurs technologies ont été élaborées afin de proposer des prototypes de capteurs intelligents. Le tableau TAB. I.1 reporte quelques entreprises présentes sur le marché. La compagnie Crossbow Technology, par exemple, a sorti en 2008 la série ëKo Pro de capteurs pour la surveillance de l'environnement, leur première cible étant l'agriculture de précision. Ces capteurs sans fil ont une portée de communication allant de 152 *m* à 3.2 *Km* selon les modèles, un poids de 544 *g*, un volume de $95 \times 89 \times 267 \text{ mm}^3$ et une espérance de vie de plus de cinq ans (au détriment de leur taille relativement grande).

Le fonctionnement des capteurs nécessite un protocole de communication, un système d'exploitation ainsi qu'un langage de programmation permettant d'implémenter des applications sur ces capteurs [Ahmed *et al.*, 2003; Ahmed et Eskicioglu, 2004; Al-Karaki et Kamal, 2004; Demirkol *et al.*, 2006]. Le système d'exploitation le plus utilisé actuellement est le **TinyOS** qui fût développé par des chercheurs à l'Université de Berkeley [Levis, 2006]. Le langage de programmation exclusif capable d'interagir avec le TinyOS est le **NesC**. Ce langage, dérivé du C, est conçu pour minimiser la consommation de

I. INTRODUCTION



FIG. I.2: Des capteurs intelligents de plus en plus intégrés.

Compagnie	Technologie	Applications
Crossbow Technology	Capteurs modulaires avec des sensors interchangeable	Surveillance de l'environnement et sécurité
Digital Sun	Réseaux de surveillance des sols pour l'arrosage intelligent	Surveillance de l'environnement et sécurité
Dust	Capteurs de quatre millimètres carrés	Gestion de stocks et surveillance
Ember	Capteurs qui s'auto-organisent et logiciels	Automatisation des bâtiments et des usines, défense
Millennial Net	Capteurs de petite taille et de faible puissance	Automatisation des bâtiments, lecteur de compteurs et gestion de chaînes d'approvisionnement
Xsily	Capteurs et logiciels	Surveillance de l'industrie et d'équipements, chauffage et ventilation

TAB. I.1: Quelques compagnies développant des capteurs intelligents.

mémoire et de puissance de calcul par les capteurs. Le TinyOS a alors pour particularité une taille très réduite en termes de mémoire (quelques kilo-octets). Concernant la communication, deux piles protocolaires ont été proposées : la Bluetooth et le **ZigBee** [Brunelli *et al.*, 2008]. La Bluetooth, standardisée sous la norme IEEE 802.15.1, a un grand défaut technique : une trop grande consommation d'énergie. Ainsi, elle ne peut pas être appliquée aux capteurs sans fil. Le protocole répondant le mieux aux besoins des réseaux de capteurs est le ZigBee. Combiné à la norme IEEE 802.15.4, il offre un faible débit de données limitant ainsi la consommation d'énergie dans le réseau. Le tableau TAB. I.2 illustre les différentes caractéristiques des technologies Zigbee, Bluetooth et Wi-Fi.

I.1.3 Topologies des réseaux de capteurs

Le choix d'une architecture du réseau constitue un point essentiel dans le fonctionnement des réseaux de capteurs. Ceci correspond à l'organisation du réseau définissant

Protocole	Zigbee	Bluetooth	Wi-Fi
Norme IEEE	802.15.4	802.15.1	802.11a/b/g/n/n-draft
Besoins mémoire	4-32 Ko	≥ 250 Ko	≥ 1 Mo
Autonomie avec pile	Années	Jours	Heures
Nombre de capteurs	≥ 65000	7	32
Vitesse de transfert	250 Ko.s^{-1}	1 Mo.s^{-1}	$11-54-108-320 \text{ Mo.s}^{-1}$

TAB. I.2: Caractéristiques des protocoles Zigbee, Bluetooth et Wi-Fi.

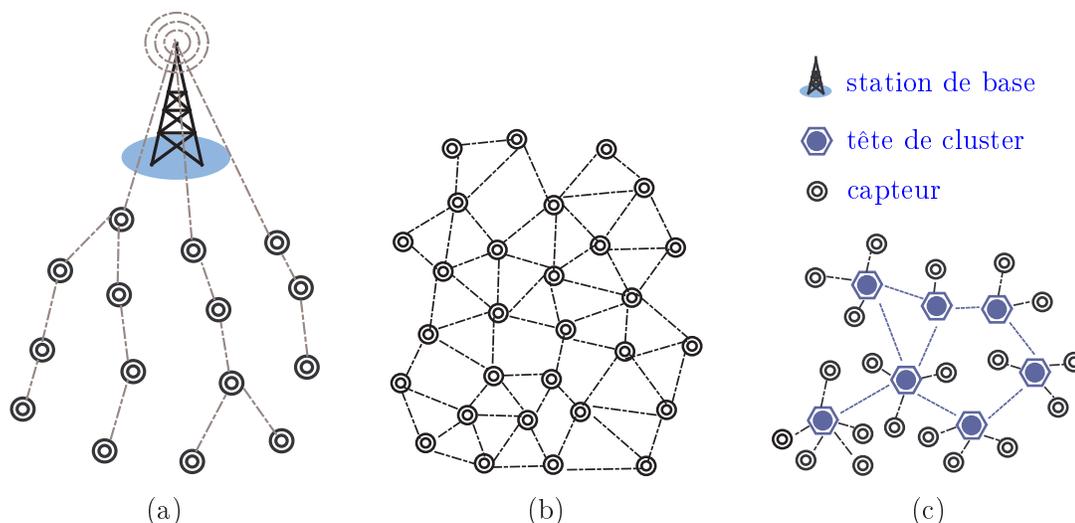


FIG. I.3: Topologies de réseau en étoile en (a), en grille en (b) et hybride en (c).

ainsi la manière dont les ondes électromagnétiques seront acheminées dans le réseau. Selon l'application, le choix de topologie influe fortement sur les performances du réseau. Nous distinguons, ci-dessous, trois topologies essentielles applicables aux réseaux de capteurs sans fil (cf. FIG. I.3) [Akyildiz *et al.*, 2002a; Bandyopadhyay et Coyle, 2003; Pan *et al.*, 2003; Jardosh et Ranjan, 2008; Fleury et Simplot-Ryl, 2009].

1. Topologie en étoile ou centralisée

Dans cette topologie, les capteurs acheminent leurs données à une station de base, appelée aussi centre de fusion. Les capteurs peuvent alors échanger des données uniquement avec la station de base qui collecte toutes les informations plus ou moins brutes du réseau. Dans ce type de structures, les messages sont véhiculés de proche en proche sans modification de contenu jusqu'à atteindre le centre de fusion. Cette topologie est surtout privilégiée par les applications de collecte de données. Parmi les avantages de cette architecture, on peut mentionner la simplicité du traitement où le problème se réduit à l'exécution de l'algorithme de décision en un même endroit. Cependant, le réseau à traitement centralisé est vulnérable puisqu'un dysfonctionnement de la station de base cause l'arrêt immédiat du réseau. De plus, cette topologie consomme énormément d'énergie vu

I. INTRODUCTION

que le coût de communication est supérieur au coût de traitement de données.

2. *Topologie en grille ou distribuée*

Dans cette topologie, tout capteur est capable d'échanger des signaux avec n'importe quel autre capteur situé dans sa portée de communication. La transmission de données entre deux capteurs hors portée l'un de l'autre pourra se faire à travers de capteurs intermédiaires. Les réseaux à traitement distribué sont alors plus robustes aux défaillances, un capteur tombant en panne ne pénalisant pas tout le réseau. Le traitement local des informations permettra d'optimiser les paquets à transmettre et donc de réduire la consommation d'énergie essentiellement due au coût de communication.

3. *Topologie hybride ou clustering*

Il s'agit d'une topologie hybride entre une topologie en étoile et une topologie en grille. Elle consiste à partitionner le réseau en des groupes de capteurs, appelés *clusters*, chacun ayant un chef, ou *tête de cluster*, chargé de réaliser les calculs et le routage des données. Le partitionnement peut être choisi a priori avec des chefs ayant des capacités supérieures en termes d'énergie et de calcul. Il est possible en outre de faire le découpage d'une manière dynamique selon la réserve énergétique et la proximité géographique.

I.1.4 Applications

L'évolution rapide des capteurs intelligents en termes de miniaturisation et de communications sans fil ainsi que la diversité de types de sensors disponibles ont permis aux réseaux de capteurs d'être applicables dans différents domaines. Ils se révèlent alors très utiles dans de nombreuses applications où il s'agit de collecter et de traiter des données [Gharavi et Kumar, 2003; Conner *et al.*, 2004; Arampatzis *et al.*, 2005; Ali *et al.*, 2008]. Nous citons en particulier les domaines militaire, environnemental, médical, domestique, de sécurité,... (cf. FIG. I.4).

1. *Applications militaires*

Le premier domaine d'application des réseaux de capteurs sans fil est le domaine militaire comme pour la plupart des technologies. La facilité de déploiement et le coût réduit des capteurs ont rendu les réseaux de capteurs un outil très appréciable dans ce domaine. Ils peuvent alors être utilisés pour surveiller un secteur stratégique afin de détecter et de poursuivre des cibles ennemies ou d'analyser des terrains afin de détecter la présence d'agents chimiques, biologiques ou radioactifs avant l'intervention humaine. L'armée américaine aurait déjà effectué des tests concluants sur les réseaux de capteurs sans fil dans le désert de Californie.

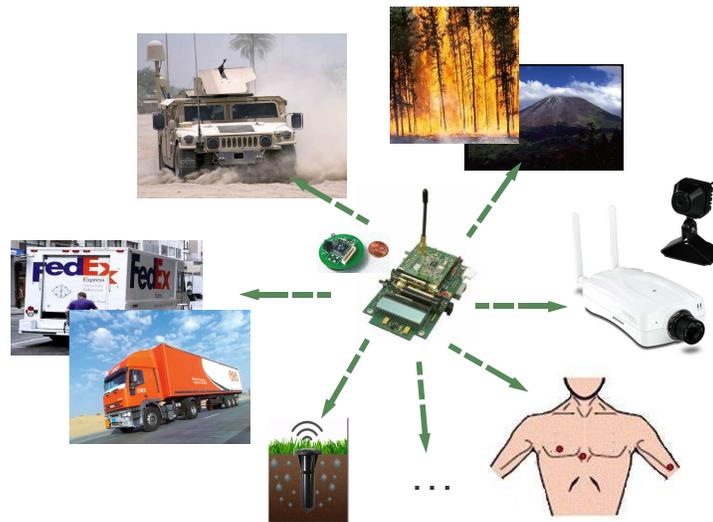


FIG. I.4: Applications des réseaux de capteurs.

2. *Applications environnementales*

Outre les applications militaires, les réseaux de capteurs peuvent être utilisés pour surveiller l'environnement ou recueillir des informations concernant la faune et la flore. Des capteurs peuvent être semés dans un champ d'agriculture afin de surveiller le niveau d'hydratation ou le taux de nutriment. Des capteurs peuvent être lancés par des avions ou des navires pour signaler des problèmes environnementaux tels que les incendies, la pollution, les épidémies, etc. Ils peuvent en outre servir à détecter des fuites de produits toxiques et alerter ainsi les secours plus rapidement dans les centrales nucléaires, les pétroliers, les sites industriels, les entrepôts chimiques, etc. Des capteurs avalés par les animaux ou placés sous leurs peaux peuvent servir à recueillir des informations sur leurs états et leurs comportements afin de proposer des solutions efficaces pour la conservation de la faune.

3. *Applications médicales*

Les réseaux de capteurs ont aussi beaucoup d'applications dans le domaine de la santé. Ils peuvent, par exemple, être utilisés pour surveiller des patients à distance. Ils permettent ainsi non seulement d'améliorer la qualité de vie des malades pouvant rester chez eux, mais aussi d'intervenir le plus rapidement possible si les informations relevées par les capteurs sont anormales. A titre d'exemple, Intel travaille sur un projet de recherche dans le but d'assister les personnes âgées. Dans ce projet, chaque objet de la maison pourrait être équipé d'un micro-capteur collectant des informations sur les activités quotidiennes de l'occupant. Les réseaux de capteurs pourraient en outre servir à la surveillance des fonctions vitales de l'être humain dans le futur. Les micro-capteurs seront alors avalés ou implantés sous

la peau. Actuellement, il existe des micro-caméras qui peuvent être avalées. Elles sont capables de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures. D'autres applications biomédicales ambitieuses existent aussi, telles que la détection de cancers. L'utilisation des réseaux de capteurs dans le domaine de la médecine pourrait ainsi apporter une possibilité de collecter des informations physiologiques en permanence, facilitant le diagnostic des maladies.

4. *Applications à la sécurité*

Les réseaux de capteurs pourraient être utilisés pour surveiller les structures des avions, des navires, des automobiles, des métros,... ainsi que les réseaux de circulation ou de distribution d'énergie. Ils pourraient servir à détecter l'altération des structures d'un bâtiment, d'une route, d'un quai, d'une voie ferrée, etc. Outre la surveillance de dégradation, les réseaux de capteurs peuvent constituer un système d'alarme qui servira à détecter les intrusions sur un large secteur. D'autres applications pourraient être envisagées telles que la surveillance de routes ou voies ferrées pour prévenir des accidents avec des animaux ou des êtres humains ou entre plusieurs véhicules.

5. *Applications commerciales*

Les réseaux de capteurs pourraient servir à des applications domestiques telles que l'optimisation des systèmes de chauffage et climatisation, d'éclairage ou de distribution d'eau. Des entreprises pourraient en outre suivre le procédé de fabrication de produits afin d'offrir une meilleure qualité de service. Les réseaux de capteurs pourraient par ailleurs améliorer le processus de stockage et de livraison, permettant par exemple au client de connaître en temps réel la position, l'état et la direction d'une cargaison.

I.2 Problème de localisation

L'une des problématiques principales des réseaux de capteurs sans fil est la localisation géographique des capteurs. Cette dernière s'avère primordiale à la fois pour la plupart des applications que pour les protocoles de routage. Dans cette section, nous commençons par décrire le problème de localisation. Nous exposons par la suite des méthodes existantes pour résoudre ce problème.

I.2.1 Description du problème

Dans de nombreuses applications telles que la surveillance des champs ou des forêts, le suivi de véhicule,..., les données mesurées doivent être liées à une position géographique pour pouvoir être utilisées. De la même manière, dans les protocoles de contrôle de topologie, où chaque capteur doit ajuster sa puissance de transmission pour minimiser sa consommation énergétique, les algorithmes ont le plus souvent besoin d'informations sur les positions des voisins [Rajaraman, 2002; Pan *et al.*, 2003]. De plus, les protocoles de routage géographique ou orienté position vont pouvoir fonctionner sans le coûteux

mécanisme de découverte de route proactive et ainsi économiser de l'énergie et améliorer le taux d'acheminement [Cheng *et al.*, 2004; Wu, 2006].

La première solution au problème de localisation fût d'équiper tous les capteurs par des systèmes de positionnement ou *Global Positioning Systems* (GPS) [Estrin *et al.*, 2001; Hofmann-Wellenhof *et al.*, 2004; El-Rabbany, 2006]. Le principe de la géolocalisation par GPS repose sur le calcul des distances qui séparent le récepteur GPS des différents satellites. Pour cette raison, ces derniers envoient des ondes électromagnétiques qui se propagent à la vitesse de la lumière. En utilisant le temps mis par l'onde pour parcourir le trajet, il est possible de calculer la distance séparant le satellite du récepteur. Il faut ainsi au minimum trois satellites pour avoir une localisation en deux dimensions à la surface de la Terre et quatre satellites pour accéder à l'altitude, un des satellites étant indispensable pour la synchronisation des horloges. Le GPS résout d'une manière précise le problème de la localisation dans les environnements extérieurs. Cependant, pour les réseaux de capteurs où les unités doivent être de très petites tailles et peu consommatrices à petits prix, équiper chaque capteur d'un récepteur GPS s'avère trop coûteux et donc peu pratique.

Les solutions alternatives consistent à équiper des capteurs peu nombreux par des récepteurs GPS. Ces capteurs, connaissant leurs positions, sont appelés **ancres**. Les autres capteurs non munis de GPS, ayant alors des positions inconnues, sont appelés **nœuds** ou **non-ancres**. Le problème de localisation se résume ainsi à l'estimation des positions des nœuds en se basant sur les mesures échangées avec les ancres. Supposons que tous les capteurs sont déployés dans le même plan. Les coordonnées des ancres et celles des nœuds sont alors données par les vecteurs à deux dimensions $\mathbf{a}_i = (a_{i,1}, a_{i,2})^T$, $1 \leq i \leq N_a$, et $\mathbf{x}_j = (x_{j,1}, x_{j,2})^T$, $1 \leq j \leq N_x$, respectivement où N_a et N_x sont les nombres respectifs des ancres et des nœuds dans le réseau. La localisation consiste alors à estimer les vecteurs \mathbf{x}_j , $1 \leq j \leq N_x$, en utilisant des informations concernant les \mathbf{a}_i , $1 \leq i \leq N_a$. Nous présenterons dans la suite quelques algorithmes de localisation pour les réseaux de capteurs sans fil, une des contraintes fondamentales de ces algorithmes étant la minimisation de la consommation d'énergie. Il faut noter que les capteurs sont supposés statiques, ayant ainsi des positions fixes au cours du temps.

I.2.2 Méthodes de localisation

Des algorithmes ont été proposés afin de contourner le problème de localisation par GPS. Les capteurs nœuds se basent sur les ancres pour connaître leurs positions. Parmi les solutions existantes, nous citons les méthodes basées sur la mesure de la puissance du signal reçue ou *Received Signal Strength Indicator*, le temps d'arrivée ou *Time of Arrival*, la différence de temps d'arrivée ou *Time Difference of Arrival*, l'angle d'arrivée ou *Angle of Arrival*, etc.

1. *Received Signal Strength Indicator (RSSI)*

Les algorithmes basés sur cette technique mesurent la puissance des signaux échangés afin d'estimer la distance séparant les capteurs [Wang et Xiao, 2007; Zanca *et al.*, 2008]. Sachant que la puissance du signal décroît d'une manière monotone avec l'augmentation de la distance parcourue, un nœud, ayant reçu un

signal, pourra en déduire la distance le séparant de l'émetteur. Chaque nœud aura ainsi la possibilité de calculer les distances le séparant de plusieurs ancrs pour en estimer sa position. L'estimation de la position pourra se faire de différentes manières selon la méthode proposée. Nous citons à titre d'exemple la méthode de multilatération où la position estimée correspond à l'intersection des différents cercles centrés sur les ancrs détectées et ayant les distances calculées comme rayons [Langendoen et Reijers, 2003; Nguyen et Rattenbury, 2003]. Cette approche ne nécessite pas de matériel supplémentaire. Cependant, le signal radio pourrait être très perturbé par l'environnement au cours de la propagation, ce qui pourrait conduire à une fausse estimation de la distance et donc de la position du nœud.

2. *Time of Arrival (ToA)*

Dans cette technique, la distance séparant un nœud d'une ancre est directement déduite du temps de propagation du signal entre les deux capteurs [Elson *et al.*, 2002; Elson et Romer, 2002; Moses *et al.*, 2003]. Connaissant la vitesse du signal, la distance sera égale au produit de la vitesse et de la durée de propagation. Cette technique nécessite une horloge de synchronisation entre les capteurs. Les distances calculées sont ainsi utilisées pour estimer les position des nœuds.

3. *Time Difference of Arrival (TDoA)*

Dans cette technique, chaque nœud mesure les temps de réception de signaux de types différents émis d'une même ancre [Rappaport *et al.*, 1996; Ward *et al.*, 1997; Nawaz et Jha, 2007]. Le décalage entre les temps de réception permettra de calculer la distance séparant le nœud de l'ancre. Les distances estimées serviront à la détermination des positions des nœuds.

4. *Angle of Arrival (AoA)*

Les méthodes basées sur cette technique estiment l'angle auquel les signaux sont reçus [Klukas et Fattouche, 1998; Pages-Zamora *et al.*, 2002; Peng et Sichertiu, 2006]. Une simple relation géométrique permettra ainsi de calculer les positions des nœuds.

Se basant sur le calcul des distances, les méthodes présentées ci-dessus permettent d'obtenir des estimées précises des positions des nœuds. Cependant, alors que la première approche est peu robuste suite aux perturbations de l'environnement, les autres induisent une grande consommation énergétique et un surcoût des capteurs nécessitant du matériel supplémentaire.

I.3 Problème de suivi de cibles

Le suivi de cibles ou *target tracking* est l'une des applications les plus prometteuses des réseaux de capteurs sans fil [Yang et Sikdar, 2003; Tran et Yang, 2006; Mostafaei *et al.*, 2009]. Cette application est d'une grande importance aussi bien pour les domaines militaires que civils (repérage de véhicules ennemis, détection de franchissement illégal des frontières, suivi de mouvements d'animaux sauvages dans les réserves fauniques,...).

Nous décrivons dans la suite le problème de suivi de cibles. Nous exposons ensuite des méthodes existantes pour résoudre ce problème.

I.3.1 Description du problème

L'objectif du suivi de cibles est de détecter une cible (ou éventuellement plusieurs) et d'identifier sa trajectoire. Le problème consiste alors à estimer les coordonnées de la cible en se basant sur les données mesurées par les capteurs. Il faut noter que le problème peut être abordé de différentes manières selon le scénario envisagé. On parle ainsi du *tracking passif* où la cible est supposée fixe et du *tracking actif* où la cible est plutôt mobile. Dans la suite, nous exposons quelques méthodes existantes pour le suivi de cibles mobiles.

I.3.2 Méthodes de suivi de cibles

Le suivi de cibles peut être assimilé à un problème de filtrage [Anderson et Moore, 1979; Ramachandra, 2000; Doucet *et al.*, 2002; Ristic *et al.*, 2004]. Le but consiste alors à mettre à jour la distribution a posteriori de l'état de la cible en utilisant à la fois les données mesurées et les états précédemment estimés. Afin d'être applicables aux réseaux de capteurs, les algorithmes classiques de filtrage ont été adaptés pour répondre aux limitations des capteurs en termes de calcul, énergie et communication. Parmi les filtres appliqués au suivi de cibles, nous citons le filtre de Kalman, le filtre particulaire et le filtre variationnel.

1. Suivi de cibles avec le filtre de Kalman

Le filtre de Kalman est un filtre récursif permettant d'estimer l'état d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées [Kalman, 1960]. Tenant compte de l'état précédent et des mesures actuelles, le filtre de Kalman estime l'état courant ainsi qu'une matrice mesurant la précision de l'état estimé. Ceci est réalisé en deux phases : prédiction et mise à jour. La phase de prédiction utilise l'état précédent pour prédire l'état courant alors que la phase de mise à jour tient compte des mesures courantes pour corriger l'état prédit dans le but d'augmenter la précision de l'estimation. Il faut noter que le filtre de Kalman est limité aux systèmes linéaires. Le filtre de Kalman étendu fût alors proposé pour remédier à cette limitation, la non-linéarité pouvant être associée au modèle d'évolution du processus, au modèle d'observation ou à tous les deux [Chen, 1993; Ribeiro, 2004]. Le problème est alors formulé comme suit,

$$\begin{cases} \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k), \\ \mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k), \end{cases} \quad (\text{I.1})$$

où f et h définissent les modèles d'évolution et d'observation, \mathbf{x}_k est le vecteur d'état à l'instant k , \mathbf{y}_k est le vecteur d'observation, \mathbf{u}_k est un vecteur de commande et \mathbf{w}_k et \mathbf{v}_k sont des bruits gaussiens à moyenne nulle.

L'application du filtrage de Kalman distribué aux réseaux de capteurs sans fil a été jugée possible par Hasu et Koivo en 2005 [Hasu et Koivo, 2005]. Un algorithme

I. INTRODUCTION

basé sur une approximation du filtre de Kalman distribué a été proposé par Spanos et al. [Spanos *et al.*, 2005]. Ceux-ci ont étudié la performance de l'algorithme proposé en présence d'imperfections, de retard, de perte de communication,... dans les réseaux sans fil. En 2006, Ribeiro et al. ont proposé l'algorithme *Sign of Innovation Kalman Filtering* (SOI-Kf) où l'observation se réduit à un seul bit [Ribeiro *et al.*, 2006]. Cette innovation améliore énormément la qualité du tracking tout en minimisant les coûts de communication. Cependant, le fait de considérer le bruit gaussien, alors qu'il ne l'est pas, peut avoir des conséquences dramatiques sur les performances de l'algorithme.

2. *Suivi de cibles avec le filtre particulaire*

Le filtre particulaire, connu aussi sous le nom de méthode séquentielle de Monte-Carlo, est un estimateur bayésien, basé sur les simulations [Doucet, 1998; Doucet *et al.*, 2000]. Il constitue souvent une alternative au filtre de Kalman étendu. Il présente l'avantage, qu'avec suffisamment d'échantillons, il se rapproche de l'estimé bayésien optimal. Il peut être ainsi plus précis que le filtre de Kalman. L'idée de base du filtrage particulaire consiste à générer un nombre fixe d'échantillons, appelés *particules*, qui approximent la distribution a posteriori de l'état. Les particules sont alors accordées des poids d'importance reflétant une approximation de la distribution a posteriori. Le problème est défini par un modèle Markovien d'évolution ayant $p(\mathbf{x}_0)$ comme distribution initiale et un modèle d'observation donnés comme suit,

$$\begin{cases} \mathbf{x}_k \sim p_{\mathbf{x}}(\mathbf{x}|\mathbf{x}_{k-1}), \\ \mathbf{y}_k \sim p_{\mathbf{y}}(\mathbf{y}|\mathbf{x}_k). \end{cases} \quad (\text{I.2})$$

Ceci peut être aussi défini par le système d'équations d'évolution et d'observation (I.1). Résoudre le problème revient à générer N_p particules selon une fonction d'importance $\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{0:k})$ généralement définie par la distribution a priori de l'état $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. A chaque instant k , ayant N_p particules provenant de l'instant précédent $\mathbf{x}_{k-1}^{(l)}$ avec leurs poids normalisés $\omega_{k-1}^{(l)}$, $1 \leq l \leq N_p$, l'algorithme est décrit comme suit :

- (a) **Génération des particules** : $\mathbf{x}_k^{(l)} \sim \pi(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{y}_{0:k})$,
- (b) **Attribution des poids** : $\hat{\omega}_k^{(l)} = \omega_{k-1}^{(l)} \cdot \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(l)}) \cdot p(\mathbf{x}_k^{(l)}|\mathbf{x}_{k-1}^{(l)})}{\pi(\mathbf{x}_k^{(l)}|\mathbf{x}_{0:k-1}, \mathbf{y}_{0:k})}$,
- (c) **Normalisation des poids** : $\omega_k^{(l)} = \frac{\hat{\omega}_k^{(l)}}{\sum_{j=1}^{N_p} \hat{\omega}_k^{(j)}}$,

avec l compris entre 1 et N_p . La distribution obtenue par les couples $\{\mathbf{x}_k^{(l)}, \omega_k^{(l)}\}$ se rapproche de la distribution a posteriori optimale quand N_p est de plus en plus grand. Un rééchantillonnage peut être réalisé pour améliorer les résultats. L'état estimé est ainsi donné par $\sum_{l=1}^{N_p} \omega_k^{(l)} \cdot \mathbf{x}_k^{(l)}$.

Plusieurs méthodes ont été proposées afin d'approximer le processus de génération de particules [Kotecha et Djurić, 2003a,b; Sheng *et al.*, 2005]. Dans [Kotecha

et Djurić, 2003a] et [Kotecha et Djurić, 2003b], Kotecha *et al.* ont décrit respectivement le filtrage particulaire gaussien ou *Gaussian Particle Filtering* et le filtrage particulaire basé sur les mélanges gaussiens, dit *Gaussian Sum Particle Filtering*. Ces deux algorithmes permettent d'approximer la distribution a posteriori par une seule distribution gaussienne ou par une somme pondérée de plusieurs gaussiennes respectivement. Dans [Sheng *et al.*, 2005], les auteurs ont proposé une stratégie collaborative pour le suivi de plusieurs cibles basée sur le modèle de mélanges gaussiens. Le principal inconvénient de ces méthodes se situe au niveau de la propagation de l'erreur, due à l'approximation de la distribution des particules par un nombre fini de gaussiennes.

3. *Suivi de cibles avec le filtre variationnel*

L'approche variationnelle est une méthode d'approximation déterministe permettant de résoudre un problème d'intérêt, tout en le définissant comme un problème d'optimisation [Jaakkola, 2000; MacKay, 2003]. Ceci est réalisé en introduisant au problème des degrés de liberté, connus sous le nom de paramètres variationnels [Jaakkola et Jordan, 1999]. En considérant le problème (I.2), le filtrage variationnel définit le modèle d'évolution comme étant un modèle de mélange, où l'état \mathbf{x}_k est supposé avoir une distribution gaussienne ayant une moyenne $\boldsymbol{\mu}_k$ et une matrice de précision $\boldsymbol{\lambda}_k$ aléatoires comme suit,

$$\begin{cases} \mathbf{x}_k \sim \mathcal{N}(\mathbf{x}_k | \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k), \\ \boldsymbol{\lambda}_k \sim \mathcal{W}_{\bar{n}}(\boldsymbol{\lambda}_k | \bar{\mathbf{V}}), \\ \boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_k | \boldsymbol{\mu}_{k-1}, \bar{\boldsymbol{\lambda}}), \end{cases} \quad (\text{I.3})$$

où $\bar{\boldsymbol{\lambda}}$, \bar{n} et $\bar{\mathbf{V}}$ sont respectivement la matrice de précision de la marche aléatoire, le degré de liberté et la matrice de précision de la distribution Wishart. La moyenne et la matrice de précision de \mathbf{x}_k suivent alors une loi normale et une loi de Wishart respectivement. L'état caché augmenté devient ainsi $\boldsymbol{\alpha}_k = (\mathbf{x}_k, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k)$. Le principe de l'approche variationnelle en ligne consiste à approcher la distribution $p(\boldsymbol{\alpha}_k | \mathbf{y}_{1:k})$ par une autre fonctionnelle plus simple $q(\boldsymbol{\alpha}_k)$ tout en minimisant la divergence de Kullback-Leibler par rapport à la vraie distribution de filtrage, donnée comme suit,

$$D_{KL}(q||p) = \int q(\boldsymbol{\alpha}_k) \cdot \log \frac{q(\boldsymbol{\alpha}_k)}{p(\boldsymbol{\alpha}_k | \mathbf{y}_{1:k})} d\boldsymbol{\alpha}_k. \quad (\text{I.4})$$

En imposant une forme séparable $q(\boldsymbol{\alpha}_k) = q(\mathbf{x}_k) \cdot q(\boldsymbol{\mu}_k) \cdot q(\boldsymbol{\lambda}_k)$ et en minimisant la divergence de Kullback-Leibler (I.4) avec les outils du calcul variationnel, on obtient la procédure itérative suivante [Vermaak *et al.*, 2003; Snoussi et Richard, 2006; Tzikas *et al.*, 2008],

$$q(\boldsymbol{\alpha}_k^{(\ell)}) \propto \exp\langle \log p(\mathbf{y}_{1:k}, \boldsymbol{\alpha}_k) \rangle_{\prod_{i \neq \ell} q(\boldsymbol{\alpha}_k^{(i)})}, \quad (\text{I.5})$$

où $\ell = 1, 2, 3$, $\boldsymbol{\alpha}_k^{(\ell)}$ est la ℓ -ème composante du vecteur d'état augmenté (vecteur d'état, sa moyenne et sa précision) et $\langle x \rangle_P$ est l'espérance de la variable aléatoire

I. INTRODUCTION

x selon la distribution P . La mise à jour de $q(\boldsymbol{\alpha}_k)$ est implémentée d'une manière séquentielle en prenant compte uniquement de la distribution précédente de la moyenne $q(\boldsymbol{\mu}_{k-1})$. La dépendance temporelle est ainsi limitée à la fonctionnelle d'une seule composante. De plus, un simple calcul permet de montrer que cette fonctionnelle est une gaussienne dont la communication et la sauvegarde se résument à une moyenne et une covariance.

Une approche variationnelle a été proposée par Snoussi et Richard [Snoussi et Richard, 2006]. Cette méthode respecte les contraintes de communication des réseaux de capteurs sans fil. Teng *et al.* ont proposé des méthodes basées sur le filtrage variationnel pour le suivi de cibles [Teng *et al.*, 2007a,b]. Ils proposent en outre des méthodes traitant le problème d'auto-localisation des capteurs et de suivi de cibles simultanés [Teng *et al.*, 2008, 2009] dans les réseaux de capteurs sans fil.

I.4 Réseaux de capteurs mobiles

La miniaturisation des technologies de communications sans fil a permis la réalisation de capteurs communicants sans fil. Ces capteurs sans fil se voient alors pourvus de nouvelles capacités, notamment la mobilité, et leurs champs d'utilisation s'en trouvent fortement augmentés. Dans cette section, nous introduisons les réseaux de capteurs mobiles. Nous exposons par la suite quelques applications spéciales aux réseaux mobiles. Nous définissons ensuite différents modèles de mobilité des capteurs. Nous présentons enfin les problèmes d'auto-localisation et de suivi de cibles pour les réseaux de capteurs mobiles.

I.4.1 Mobilité des capteurs

Grâce à l'absence d'infrastructure fixe, les capteurs ont la capacité d'être mobiles, leur mobilité pouvant être contrôlée ou non-contrôlée, dite aussi passive [Labioud, 2006; Shorey, 2006; Zeinalipour-Yazti et Chrysanthis, 2009]. Dans le premier cas, les capteurs se déplacent suite à une commande interne ou externe. Il est alors possible de profiter de la mobilité des capteurs pour améliorer les performances du réseau vis-à-vis des applications. Il s'agit par exemple des robots pouvant bouger dans différentes directions avec différentes vitesses (cf. FIG. I.5.(a)). Dans le cas de la mobilité passive, les capteurs se déplacent d'une manière incontrôlée suite à une force externe. Il est alors indispensable de relocaliser les capteurs puisque toutes les mesures réalisées sont fortement dépendantes des positions géographiques des capteurs. Supposons, par exemple, le cas de micro-capteurs déployés pour la surveillance d'un terrain. Un vent soufflant serait capable de déplacer les capteurs, ceux-ci étant passifs au déplacement (cf. FIG. I.5.(b)). Nos travaux de recherche se situent au niveau de l'auto-localisation des capteurs dans le cas de la mobilité incontrôlée et du déplacement des capteurs afin d'optimiser le suivi de cibles dans le cas où les capteurs ont une mobilité contrôlée.

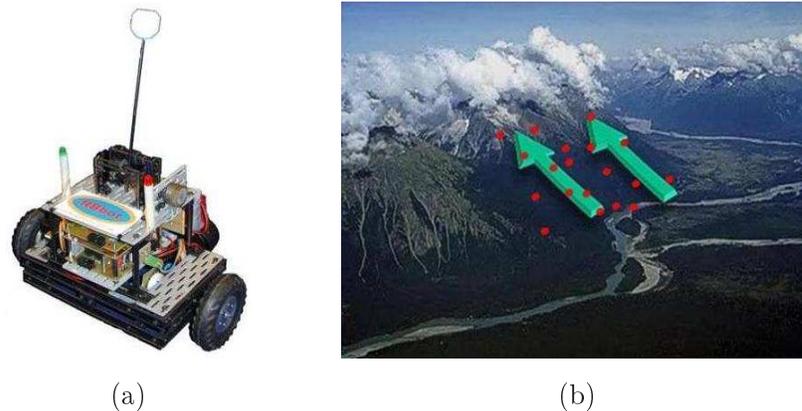


FIG. I.5: Illustrations de la mobilité contrôlée en (a) et de la mobilité non-contrôlée en (b).

I.4.2 Quelques applications des réseaux de capteurs mobiles

Il existe plusieurs classes de réseaux de capteurs mobiles : 1) les réseaux **hyper-mobiles**, où les composants bougent à des vitesses très élevées, citons par exemple les voitures, les personnes munies de téléphones portables, les avions, etc. ; 2) les réseaux **quasi-statiques**, où les capteurs se déplacent avec des vitesses faibles comme dans le cas de la surveillance d'un magasin à l'aide de robots mobiles ; et 3) les réseaux **mobiles hybrides**, regroupant les deux classes, tel l'exemple d'un avion équipé de capteurs mobiles à l'intérieur et à l'extérieur. Parmi les applications des réseaux de capteurs mobiles, nous citons ci-dessous quelques exemples issus de la recherche scientifique [Akyildiz *et al.*, 2002b] :

◇ ZebraNet

ZebraNet est un projet interdisciplinaire né des efforts communs des mondes de la biologie et des systèmes informatiques [Juang *et al.*, 2002]. Il s'agit d'une application de surveillance de l'environnement, dans laquelle des micro-capteurs sont attachés à des zèbres vivant en complète liberté (cf. FIG. I.6). Son but est de développer, d'évaluer et de tester des réseaux de capteurs mobiles, intégrant la communication sans fil, le stockage de données, etc. Cette technologie permet alors aux biologistes d'étudier les mouvements migratoires d'une espèce ainsi que ses différentes interactions avec d'autres espèces.

◇ Great Duck Island

La *Great Duck Island* est une île située dans le golfe du Maine. Selon l'organisation américaine de protection de l'environnement *The Nature Conservancy*, cette île abrite l'une des cinq espèces d'oiseaux de mer du Maine. Great Duck est un site actif pour la recherche environnementale menée par *The College of the Atlantic*, une grande partie de cette recherche s'intéressant en particulier au pétrel cul-blanc (*Oceanodroma leucorhoa*). Cette colonie de mouettes souffre malheureusement d'une hausse de mortalité des œufs de 20% à cause de la visite de l'homme rien que pour une quinzaine

I. INTRODUCTION



FIG. I.6: Projet ZebraNet.



FIG. I.7: Exemple de capteurs développés pour la Great Duck Island à gauche et des pétrels cul-blanc à droite.

de minutes [Anderson, 1995]. Pour remédier à ce problème, Mainwaring *et al.* ont eu l'idée de déployer des capteurs fixes et mobiles afin de pouvoir étudier cette colonie (cf. FIG. I.7) [Mainwaring *et al.*, 2002]. Cet exemple illustre les avantages des réseaux de capteurs mobiles dans la surveillance de l'environnement. L'utilisation des capteurs mobiles permet en effet de réduire considérablement la présence de l'homme dans des écosystèmes sensibles protégeant ainsi le milieu d'intérêt.

◇ Suivi de patients

Les réseaux de capteurs mobiles connaissent aussi un réel succès dans le monde de la santé. Citons en particulier l'exemple du suivi de patients dans un hôpital. La surveillance des signes vitaux des patients dans un hôpital nécessite en effet du matériel câblé, obligeant les patients à rester au lit. Pour cette raison, Hande *et al.* ont développé un réseau de capteurs sans fil mobiles capables de mesurer une variété de signes vitaux des patients et de les communiquer à une station de base en temps réel [Hande *et al.*, 2006]. Ces capteurs peuvent être munis de sensors fournissant des électrocardiogrammes (signaux ECGs), des mesures de la pression artérielle ou autre des patients. Ils intègrent en outre des cellules photovoltaïques permettant la recharge des piles grâce à la lumière fluorescente. Le personnel médical pourra ainsi surveiller l'état de santé des patients à n'importe quel moment sans devoir se déplacer dans leurs chambres. FIG. I.8, empruntée à l'article de Hange *et al.* [Hande *et al.*, 2006], illustre le fonctionnement d'un tel réseau.

◇ Détection de bombes radiologiques

L'une des applications militaires des réseaux de capteurs mobiles se situe dans le

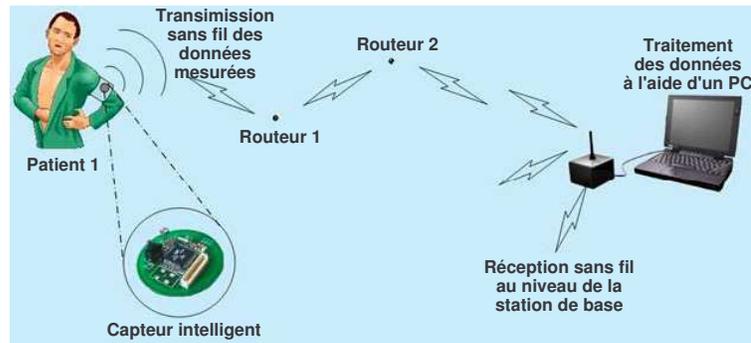


FIG. I.8: Fonctionnement d'un réseau de capteurs installé pour le suivi des patients.

cadre de la prévention. En effet, selon la chaîne de télévision canadienne publique CBC, l'aéroport international d'Ottawa développe un projet de détection de bombes radiologiques, dites aussi bombes sales [CBC, 2004]. Le système est composé de capteurs fixes placés dans des corridors et de capteurs mobiles placés sur des véhicules de sécurité. Le rôle de ces capteurs est de recueillir des données sur le niveau de radiation dans les différentes parties de l'aéroport. Toutes les données mesurées sont ensuite collectées dans une salle de contrôle où elles seront affichées sur des écrans en couleur.

I.4.3 Modèles de mobilité

Les chercheurs ont souvent recours aux techniques de simulation pour évaluer les performances des algorithmes développés pour les réseaux de capteurs mobiles. En conséquence, il est impératif de concevoir des modèles de mobilité qui reproduisent avec exactitude les mouvements des capteurs, pour lesquels les algorithmes sont éventuellement développés. Ce ne serait que dans ce type de scénarios que l'on pourrait juger l'algorithme proposé. Deux grandes classes de modèles existent : les traces et les modèles synthétiques. Les traces sont les schémas reproduisant avec précision des mouvements observés dans la vraie vie. Cependant, dans les cas où les traces ne sont pas accessibles, l'utilisation de modèles synthétiques devient nécessaire. Ceux-ci sont conçus de manière à imiter les mouvements réels des capteurs mobiles. Ils doivent, à titre d'exemple, proposer des changements de direction et de vitesse au cours du mouvement. Ils se regroupent en deux grandes catégories, les modèles d'entités et les modèles de groupes [Hong *et al.*, 1999; Camp *et al.*, 2002; Boudec et Vojnovic, 2005].

◇ Modèles d'entités

Les modèles d'entités sont des modèles qui définissent des mouvements indépendants. Ils décrivent ainsi les mobilités des capteurs en tant qu'entités. Plusieurs modèles d'entités existent, nous citons en particulier le *random walk mobility model* et le *random waypoint mobility model*.

1. *Random walk mobility model*

Le modèle de marche aléatoire ou *Random Walk Mobility Model* (RWMM)

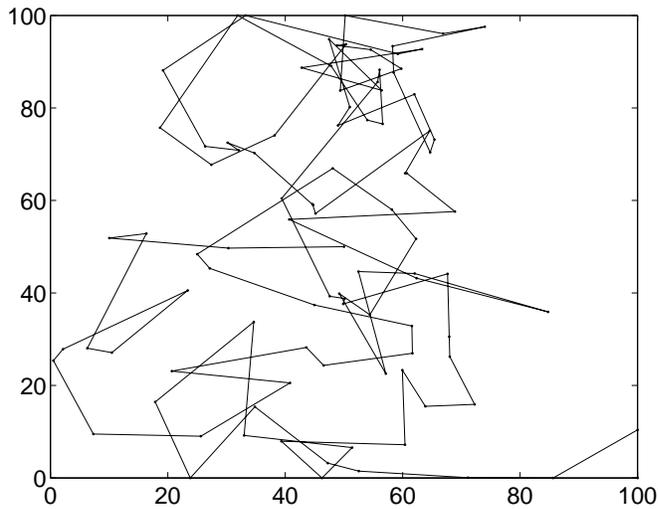


FIG. I.9: Illustration du Random Walk Mobility Model (à période constante).

a été décrit mathématiquement par Einstein en 1926. Le RWMM est développé pour représenter les mouvements irréguliers, partant du fait qu’il existe dans la nature des entités se déplaçant d’une manière imprévisible. Dans ce modèle, le capteur mobile part de sa position actuelle vers une nouvelle position avec une vitesse et une direction aléatoires. Ces nouvelles grandeurs sont choisies parmi des intervalles prédéfinis $[vitesse_{min}, vitesse_{max}]$ et $[0, 2\pi]$, respectivement. Les mouvements se produisent soit pour une durée constante Δt , soit pour une distance parcourue constante d . Au bout de chaque intervalle, un nouveau couple (vitesse, direction) est calculé. Un capteur arrivant à l’une des frontières de la zone de simulation rebondit avec un angle déterminé par la nouvelle direction. FIG. I.9 illustre un mouvement de 100s généré selon le modèle de marche aléatoire. La zone de simulation est supposée carrée de dimensions $100m \times 100m$. L’intervalle de vitesse est choisi à $[0, 30m.s^{-1}]$ et la période est supposée constante égale à 1s.

2. *Random waypoint mobility model*

Le modèle *Random Waypoint Mobility Model* (RWayMM) inclut des temps de pause entre les changements de direction et/ou de vitesse. Le capteur mobile commence par rester dans la même position pour une certaine durée de temps, dite temps de pause. Une fois ce temps écoulé, le capteur choisit aléatoirement une destination de la zone de simulation et une vitesse pour y parvenir. La vitesse est uniformément distribuée sur un intervalle prédéfini $[vitesse_{min}, vitesse_{max}]$. Le capteur se déplace ainsi vers sa nouvelle position avec la vitesse sélectionnée. Dès qu’il atteint sa destination, il choisit un temps de pause aléatoire pour y rester et ainsi de suite. FIG. I.10 illustre un mouvement de 200s généré selon le modèle RWayMM. La zone de simulation est supposée carrée de dimensions $100m \times 100m$. L’intervalle de vitesse est choisi à $[0, 30m.s^{-1}]$ et celui du temps

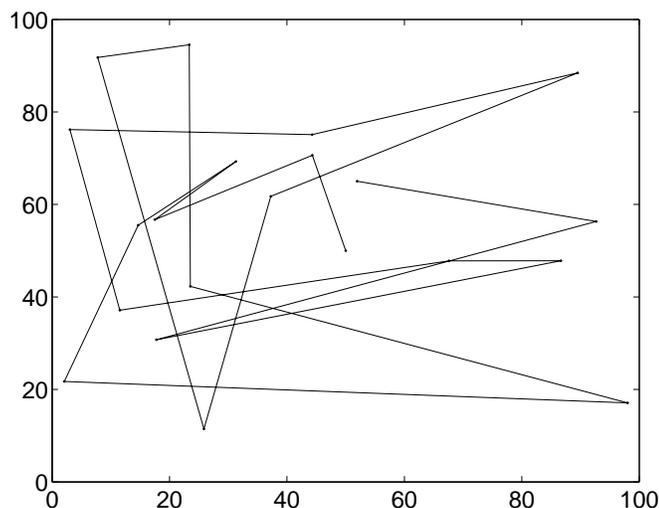


FIG. I.10: Illustration du Random Waypoint Mobility Model.

de pause est fixé à $[0, 2s]$.

◇ Modèles de groupes

Les modèles de groupes décrivent les mouvements de groupes de capteurs se déplaçant d'une manière coopérative. Ceci est utile dans des scénarios tels que le mouvement d'un groupe de soldats attribués une même tâche. Plusieurs modèles de groupes existent, nous citons en particulier le *nomadic community mobility model* et le *reference point group mobility model*.

1. *Nomadic community mobility model*

Tout comme les sociétés nomades se déplaçaient d'un endroit à l'autre, le modèle de mobilité de la communauté nomade ou *Nomadic Community Mobility Model* (NCMM) représente des groupes de capteurs allant collectivement d'un point à l'autre. Au sein de chaque groupe, les individus maintiennent leurs espaces propres où ils sont capables de bouger d'une manière aléatoire. Un tel scénario correspond à une classe d'étudiants faisant un tour dans un musée. Dans ce modèle, dès qu'un point de référence est choisi, les capteurs se déplacent vers le nouveau point. Chacun d'eux a par la suite la possibilité de bouger autour du point de référence à l'aide d'un modèle d'entités, celui de la marche aléatoire à titre d'exemple. FIG. I.11 illustre un mouvement de groupe de cinq capteurs selon le modèle NCMM. La figure montre en trait plein la trajectoire du point de référence et en trait interrompu celles des capteurs.

2. *Reference point group mobility model*

Le modèle *Reference Point Group Mobility Model* (RPGMM) décrit le mouvement aléatoire d'un groupe de capteurs, eux-mêmes pouvant bouger d'une manière aléatoire au sein du groupe. Dans ce modèle, chaque capteur possède un point de référence. Ces points de références se déplacent suivant la même trajectoire aléatoire (avec une simple translation). Les capteurs ont en outre la

I. INTRODUCTION

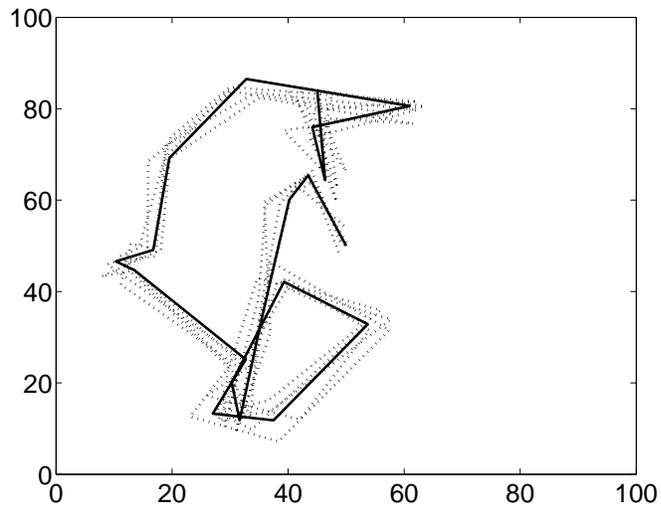


FIG. I.11: Illustration du Nomadic Community Mobility Model.

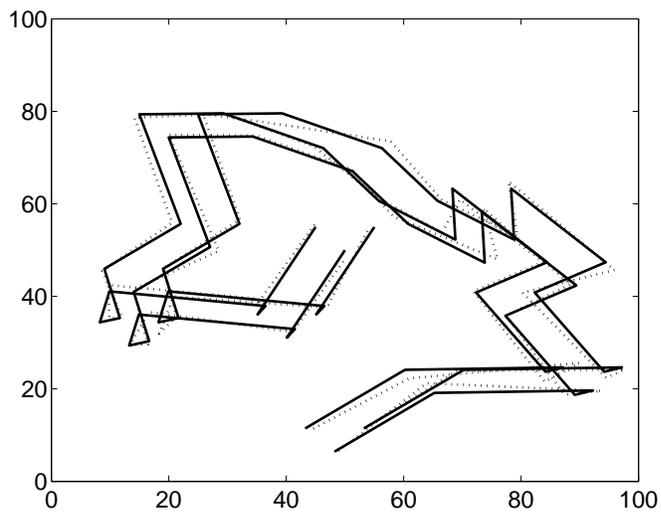


FIG. I.12: Illustration du Reference Point Group Mobility Model.

possibilité de bouger d'une manière aléatoire, chacun autour de son point de référence. FIG. I.12 illustre un mouvement de groupe de trois capteurs selon le modèle RPGMM. La figure montre en trait plein les trajectoires des points de référence et en trait interrompu celles des capteurs.

I.4.4 Localisation des capteurs mobiles

Suite à l'absence d'infrastructure dans les réseaux de capteurs sans fil, les capteurs ont la possibilité de changer de positions. Dans le cas où cette mobilité est due à une force extérieure, les capteurs sont passifs face au déplacement. En d'autres termes, ils n'ont pas conscience du changement de positions. Dans une telle situation, une localisation périodique devient inévitable vu que les données mesurées sont très liées aux positions des capteurs dans la plupart des applications. Plusieurs algorithmes ont été proposés afin de remédier au problème d'auto-localisation dans les réseaux de capteurs mobiles. Ces algorithmes distinguent deux types de capteurs : les **ancres**, équipées de GPS [Hofmann-Wellenhof *et al.*, 2004] et donc connaissant leurs positions précises, et les **non-ancres** ou **nœuds**, nécessitant des algorithmes d'auto-localisation.

Des méthodes de localisation statique répétitive ont été proposées dans [Zhang *et al.*, 1998; Bulusu *et al.*, 2000; Capkun *et al.*, 2001; Savvides *et al.*, 2001; Moore *et al.*, 2004; Vivekanandan et Wong, 2007]. Dans ces méthodes, les nœuds n'utilisent que les mesures effectuées instantanément vis-à-vis des ancres afin de définir leurs positions. Dans [Zhang *et al.*, 1998] et [Capkun *et al.*, 2001] par exemple, les auteurs proposent des méthodes répétitives basées sur les mesures des temps d'arrivée ou *Time of Arrival*. Dans [Bulusu *et al.*, 2000], Bulusu *et al.* définissent à chaque instant la position estimée par le barycentre des ancres se trouvant à proximité du nœud. Différemment, dans [Savvides *et al.*, 2001], les nœuds estiment les distances les séparant des ancres de son voisinage. La définition des positions se fait par la suite par une technique de multilatération. Moore *et al.* présentent dans [Moore *et al.*, 2004] une méthode de localisation itérative basée sur la théorie des graphes. Plus récemment, Vivekanandan et Wong proposent une méthode de localisation statique répétitive aussi [Vivekanandan et Wong, 2007]. Dans cette méthode, chaque ancre transmet plusieurs signaux avec différents niveaux de puissance. Ces signaux incluent la position de l'ancre, le niveau de puissance à l'émission et la distance maximale que peut parcourir le signal. Chaque nœud recevant un certain nombre de signaux d'une même ancre pourra définir une couronne dans laquelle il se situe. Sa position estimée est ainsi définie par le centre approximatif de la zone d'intersection de toutes les couronnes générées. D'autres chercheurs tentent à résoudre le problème de localisation en utilisant une certaine forme d'optimisation globale. Citons à titre d'exemple Doherty *et al.* qui décrivent la localisation en tant que problème d'optimisation convexe [Doherty *et al.*, 2001].

D'autres méthodes tenant compte de la mobilité des nœuds ont été proposées dans [Hu et Evans, 2004; Baggio et Langendoen, 2006; Yi *et al.*, 2007]. En plus des observations, ces méthodes utilisent des modèles de mobilité afin de raffiner l'estimation. La résolution du problème de localisation se fait dans ces méthodes à l'aide de l'approche séquentielle de Monte-Carlo ou encore le filtrage particulaire [Doucet, 1998; Doucet

I. INTRODUCTION

et al., 2000]. L'estimation consiste ainsi à générer des particules afin de couvrir toutes les solutions possibles du problème. Le barycentre de toutes les particules définit la position estimée. Dans [Hu et Evans, 2004], Hu et Evans utilisent le modèle de mobilité pour propager la position estimée à l'instant précédent. Un nombre fixe de positions, dites **particules**, sont alors générées à l'intérieur de la zone obtenue. Les observations sont ensuite utilisées pour corriger les particules prédites. Seules sont gardées les particules qui appartiennent à la zone d'intersection des mesures avec le modèle de mobilité. Les poids affectés aux particules valent ainsi 1 ou 0. Baggio et Langendoen proposent une amélioration à cette méthode dans [Baggio et Langendoen, 2006]. En effet, ils utilisent les mesures et le modèle de mobilité afin de définir une première zone approximative. Ceci permet de réduire la zone de prédiction et ainsi le temps de calcul de la méthode. Dans [Hu et Evans, 2004] et [Baggio et Langendoen, 2006], les mesures utilisées sont des mesures de connectivités. En d'autres termes, ces méthodes tiennent compte des mesures faites vis-à-vis des ancres situées dans les voisinages des nœuds. Dans [Yi *et al.*, 2007], Yi *et al.* proposent une méthode à sauts multiples ou *multi-hop*. Cette méthode ne se limite pas aux mesures des ancres à proximité du nœud. Elle tient compte désormais même des ancres les plus éloignées. En se basant sur les puissances des signaux reçus, chaque nœud définit des couronnes, centrées sur les ancres, auxquelles il appartient. Les rayons des couronnes sont des multiples de la portée de communication des capteurs. Grâce à cette technique, des réseaux à faibles densités d'ancres peuvent être considérés.

I.4.5 Suivi de cibles dans un réseau mobile

Grâce à la mobilité contrôlée des capteurs [Dantu *et al.*, 2005; Song *et al.*, 2008], il est possible de les repositionner afin d'optimiser les performances du réseau vis-à-vis des applications. Dans les travaux de la thèse, nous considérons en particulier l'application de suivi de cibles. Des méthodes sont proposées dans la littérature pour la gestion de mobilité. Leur but est d'améliorer la topologie du réseau, d'assurer la couverture, de prolonger la durée de vie du réseau, etc. [Kansal *et al.*, 2004; Tirta *et al.*, 2004; Basagni *et al.*, 2008; Lambrou *et al.*, 2009]. Peu d'attention a été consacrée au problème du contrôle de la mobilité pour le suivi de cibles [Zou et Chakrabarty, 2007; chit Chin *et al.*, 2010]. Dans [Zou et Chakrabarty, 2007], Zou et Chakrabarty proposent une stratégie de déplacement des capteurs basée sur l'estimation bayésienne. Chaque capteur a la possibilité de bouger vers une nouvelle position parmi une série de positions candidates. Cette série est composée des positions situées à un pas de la position courante, la largeur du pas dépendant de plusieurs paramètres. Le choix de la position est fait selon que le déplacement améliorera la qualité du suivi de cibles ou non. En effet, la méthode propose de générer toutes les combinaisons possibles des nouvelles positions des capteurs. Les combinaisons seront par la suite évaluées, celle correspondant à la meilleure estimation de la position de la cible sera choisie. Chin *et al.* envisagent le problème où la cible, nommée souris, essaie de s'évader de la détection des capteurs, nommés chats [chit Chin *et al.*, 2010]. La méthode étudie les différentes situations possibles du problème. Elle propose en outre un protocole de coordination permettant aux chats de capturer collaborativement la souris.

I.5 Contributions et organisation du document

Le travail présenté dans ce manuscrit se situe en premier lieu dans le cadre de l'auto-localisation des capteurs ayant une mobilité non-contrôlée. Il s'agit d'un problème d'estimation des vecteurs positions des capteurs non-équipés de GPS [Hofmann-Wellenhof *et al.*, 2004]. Le problème est défini par un modèle mathématique comprenant un modèle d'évolution, dû à la mobilité, et un modèle d'observation. En second lieu, le manuscrit traite le problème de suivi de cibles dans le cadre de la mobilité contrôlée. Une stratégie de gestion de mobilité est ainsi conçue afin d'optimiser l'estimation de la position de la cible. L'outil principal de l'estimation d'état est la théorie des intervalles [Moore, 1966; Jaulin *et al.*, 2001].

I.5.1 Plan du manuscrit

La suite du manuscrit s'articule autour de six chapitres comme suit :

Dans le deuxième chapitre, nous introduisons la théorie des intervalles. Cette approche à erreurs bornées permet une représentation ensembliste de différentes quantités. Nous commençons par définir l'élément de base de cette théorie : l'intervalle. On appelle intervalle réel, noté $[x]$, un ensemble de nombres réels délimité par une borne inférieure \underline{x} et une borne supérieure \bar{x} :

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}.$$

Nous décrivons ensuite les différentes notions de la théorie des intervalles, notamment les opérateurs, les fonctions d'inclusion, les contracteurs, les solveurs, etc.

Dans le troisième chapitre, nous exposons le problème d'auto-localisation basé sur les mesures de connectivité. Nous proposons en premier lieu une méthode basée uniquement sur les ancrs situées à proximité du nœud. Nous effectuons par la suite une localisation *multi-hop*, tenant compte des ancrs plus éloignées, ainsi qu'une approche dite *back-propagated*, permettant de mettre à jour les estimés précédents. Nous proposons enfin une méthode d'auto-localisation basée aussi bien sur les ancrs que sur les nœuds voisins. Celle-ci est privilégiée dans les réseaux de capteurs à faibles densités d'ancres.

Dans le quatrième chapitre, nous considérons le problème d'auto-localisation où les mesures sont conçues sans modèle. En effet, une comparaison des puissances reçues des différents signaux échangés permet de localiser les nœuds dans des couronnes centrées sur les ancrs. Les rayons de ces couronnes correspondent à des distances inter-ancres. Cette méthode permet de faire face aux problèmes de paramétrage des modèles d'observation. Nous résolvons ensuite ce problème en se basant à la fois sur les ancrs et les non-ancres.

Dans le cinquième chapitre, nous proposons une méthode d'auto-localisation dans les environnements imparfaits. Nous supposons alors avoir les mesures de fiabilité des différentes mesures. La résolution du problème consiste à combiner toutes les informations acquises en utilisant la théorie des fonctions de croyance. Le résultat se résume à un ensemble de pavés, ayant chacun un poids spécifique.

I. INTRODUCTION

Dans le sixième chapitre, nous traitons le problème de suivi de cibles dans les réseaux de capteurs mobiles. Nous proposons ainsi une stratégie de repositionnement des capteurs afin d'optimiser l'application. La méthode est composée de plusieurs étapes : 1) l'estimation de la position actuelle de la cible en utilisant la théorie des intervalles, 2) la prédiction de la position future de la cible également à l'aide des intervalles, 3) la définition d'un ensemble de positions que doivent prendre les capteurs, et 4) l'attribution d'une position parmi l'ensemble à chacun des capteurs tout en minimisant le déplacement. Cette dernière étape est effectuée à l'aide de l'optimisation par colonies de fourmis.

Le dernier chapitre ouvre quelques perspectives comme la localisation robuste en combinant les mesures de fiabilité à l'approche supposant un nombre maximal d'erreurs, le suivi de cibles multiples ou autre. Nous envisageons, en outre, l'utilisation d'autres méthodes d'optimisation, notamment l'algorithme génétique ou l'optimisation par essaims particulaires, en vue de comparaison. La définition des positions des capteurs peut être aussi améliorée en y introduisant des degrés de liberté, que l'on pourra optimiser.

I.5.2 Publications

Cette thèse a fait l'objet de plusieurs publications citées ci-dessous :

◇ **Revue internationale avec comité de lecture (1+3)**

- **Mourad, F.**, Snoussi, H., Abdallah, F. & Richard, C. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *IEEE Transactions on Signal Processing*, vol. 57 (8), p. 3226-3239, 2009.
- **Mourad, F.**, Snoussi, H. & Richard, C. Interval-based localization using RSSI comparison in MANETs. *IEEE Transactions on Aerospace and Electronic Systems*, 2009 (AQ, en 2^{ème} révision).
- **Mourad, F.**, Chehade, H., Snoussi, H., Yalaoui, F., Amodeo, L. & Richard, C. Controlled mobility sensor networks for target tracking using ant colony optimization. *IEEE Transactions on Mobile Computing*, soumise en avril 2010.
- **Mourad, F.**, Snoussi, H., Abdallah, F. & Richard, C. A robust localization algorithm for mobile sensors using belief functions and interval theory. *IEEE Transactions on Vehicular Technology*, soumise en juillet 2010.

◇ **Conférences internationales avec actes (2)**

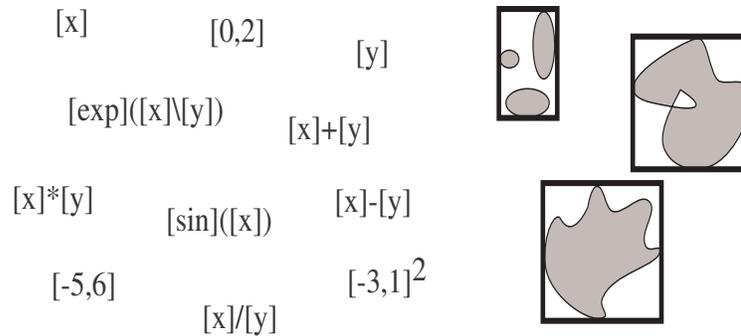
- **Mourad, F.**, Snoussi, H., Abdallah, F. & Richard, C. Guaranteed boxed localization in MANETs by interval analysis and constraints propagation techniques. *IEEE Globecom*, Nouvelle-Orléans, Etats-Unis, 30 nov.-4 déc. 2008.
- **Mourad, F.**, Snoussi, H., Abdallah, F. & Richard, C. Model-free interval-based localization in MANETs. *IEEE DSP/SPE*, Marco Island, Etats-Unis, 4-7 janvier 2009.

◇ **Conférence nationale avec acte (1)**

- **Mourad, F.**, Snoussi, H., Abdallah, F. & Richard, C. Localisation par intervalles basée sur les ancrés et les non-ancrés dans les réseaux de capteurs mobiles. *GRETSI*, Dijon, France, 8-11 septembre 2009.

I. INTRODUCTION

II Théorie des intervalles



Sommaire

- II.1 Introduction
- II.2 Définition des intervalles
- II.3 Outils de base de la théorie des intervalles
- II.4 Sous-pavages
- II.5 Contracteurs
- II.6 Conclusion


 ans ce chapitre, nous introduisons la théorie des intervalles. Cette approche à erreurs bornées permet une modélisation simplifiée de l'incertitude. Les nombres réels sont remplacés par les intervalles auxquels ils appartiennent. L'avantage d'une telle représentation, par rapport aux approches probabilistes, est la simplicité de manipuler les quantités incertaines. En particulier, il n'est pas nécessaire de représenter la manière dont l'incertitude est répartie entre les deux bornes. Les intervalles possèdent une double nature, celle du nombre réel et celle de l'ensemble. Les algorithmes proposés dans le cadre de la théorie des intervalles profitent de cette dualité pour combiner les opérations ensemblistes aux opérations arithmétiques. Dans ce chapitre, nous décrivons les différentes notions et outils de la théorie des intervalles, notamment les opérateurs, les fonctions d'inclusion, les contracteurs, etc.

II.1 Introduction

La théorie des intervalles a été proposée pour la première fois par Moore en 1966 [Moore, 1966]. L'idée de base de cette théorie consiste à remplacer les nombres réels par les intervalles auxquels ils appartiennent. Selon Moore [Moore, 1966, 1979; Moore *et al.*, 2009], il s'agit d'une extension du concept du nombre réel. En d'autres termes, l'intervalle réel est considéré comme une *nouvelle classe de nombres*, représentée par une paire de nombres réels : les deux bornes de l'intervalle. Appliquée à l'estimation d'état, la théorie des intervalles est une approche à erreurs bornées. Elle remédie aux problèmes d'erreurs d'arrondi et de propagation d'erreurs, ceux-ci étant dus principalement à l'incertitude au niveau des paramètres ou des valeurs initiales des modèles mathématiques. La résolution d'un problème à l'aide des intervalles engendre un intervalle contenant toutes les solutions possibles du problème. D'après Richtmyer (R. D. Ritchmyer, Math. Comput., 22 (1968), p. 221), bien que l'analyse par intervalles soit juste un nouveau langage d'inégalités, elle reste un langage très puissant ayant une application directe dans la résolution des problèmes de calculs numériques.

II.2 Définition des intervalles

Un **intervalle réel** $[x]$ [Moore, 1966; Jaulin *et al.*, 2001] est un sous-ensemble connexe de \mathbb{R} , défini comme suit,

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}, \quad (\text{II.1})$$

tel que $\underline{x} = \inf\{[x]\}$ et $\bar{x} = \sup\{[x]\}$ sont respectivement les bornes inférieure et supérieure de l'intervalle $[x]$. Les bornes des intervalles réels peuvent être aussi bien finies qu'infinies [Hickey *et al.*, 2007]. En d'autres termes, $] - \infty, +\infty[$ par exemple est un intervalle réel. L'ensemble vide \emptyset est pareillement un intervalle réel. L'ensemble de tous les intervalles réels est noté \mathbb{IR} .

L'intervalle réel est caractérisé par son centre $c([x])$,

$$c([x]) = \frac{\underline{x} + \bar{x}}{2}, \quad (\text{II.2})$$

et par sa largeur $w([x])$, dite aussi taille,

$$w([x]) = \bar{x} - \underline{x}. \quad (\text{II.3})$$

La taille de l'intervalle $w([x])$ dépend de l'incertitude présente sur la quantité x . Un intervalle est dit dégénéré si sa taille est nulle, i.e. $[x, x]$. Il est identifié au nombre réel x et est aussi noté x .

Un **pavé** $[\mathbf{x}]$ [Moore, 1966; Jaulin *et al.*, 2001], dit aussi **boîte**, est un intervalle multidimensionnel défini par le produit cartésien de plusieurs intervalles réels comme suit,

$$[\mathbf{x}] = [x_1] \times \cdots \times [x_n] = [\underline{x}_1, \bar{x}_1] \times \cdots \times [\underline{x}_n, \bar{x}_n]. \quad (\text{II.4})$$

L'ensemble des pavés est noté $\mathbb{I}\mathbb{R}^n$. Le centre d'un pavé est donné par le vecteur comprenant les centres de chacun des intervalles réels composants le pavé. Par ailleurs, la taille du pavé est donnée par la taille maximale des différents éléments,

$$w([\mathbf{x}]) = \max_{j \leq n} \{w([x_j])\}. \quad (\text{II.5})$$

II.3 Outils de base de la théorie des intervalles

Un intervalle réel possède une double nature, celle du nombre réel et celle des ensembles [Moore, 1979]. La théorie des intervalles profite de cette dualité pour définir des algorithmes adaptés aux intervalles. Dans cette section, nous présentons les outils de base de la théorie des intervalles, notamment les opérateurs ensemblistes et arithmétiques, les fonctions d'inclusion et les tests d'inclusion.

II.3.1 Opérations ensemblistes

Les opérateurs ensemblistes constituent l'un des premiers outils de base du calcul ensembliste [Moore, 1979; Kearfott et Kreinovich, 1996; Jaulin *et al.*, 2001]. Nous commençons tout d'abord par les définitions concernant les intervalles réels. Nous étendons par la suite ces définitions aux intervalles multidimensionnels.

1. Cas des intervalles réels

Les intervalles héritent naturellement de l'**inclusion** entre les ensembles de nombres réels,

$$[x] \subseteq [y] \text{ si } \forall x \in [x], x \in [y]. \quad (\text{II.6})$$

Si l'on considère les bornes des intervalles, ceci est équivalent au fait que $\underline{x} \geq \underline{y}$ et $\bar{x} \leq \bar{y}$.

De la même manière, l'**intersection** de deux intervalles réels donne un intervalle défini comme suit,

$$[x] \cap [y] = \begin{cases} \emptyset, & \text{si } \bar{x} < \underline{y} \text{ ou } \bar{y} < \underline{x}, \\ [\max\{\underline{x}, \underline{y}\}, \min\{\bar{x}, \bar{y}\}], & \text{sinon.} \end{cases} \quad (\text{II.7})$$

La réunion ensembliste, par contre, n'est pas une opération interne à $\mathbb{I}\mathbb{R}$. En effet, la réunion de deux intervalles disjoints donne un ensemble non connecté. En conséquence, une nouvelle opération de **réunion d'intervalles** \sqcup est définie,

$$[x] \sqcup [y] = [\min\{\underline{x}, \underline{y}\}, \max\{\bar{x}, \bar{y}\}]. \quad (\text{II.8})$$

L'intervalle résultant englobe les deux intervalles réels $[x]$ et $[y]$.

Un intervalle $[x]$ **privé** de l'intervalle $[y]$ engendre un intervalle vide si $[x] \subseteq [y]$. Dans le cas contraire, ceci correspond à l'intervalle englobant le complémentaire de $[y]$ dans $[x]$ comme suit,

$$[x] \setminus [y] = [\{x \in [x] \mid x \notin [y]\}]. \quad (\text{II.9})$$

II. THÉORIE DES INTERVALLES

En d'autres termes,

$$[x] \setminus [y] = \begin{cases} \emptyset, & \text{si } [x] \subseteq [y], \\ [\underline{x}, \underline{y}], & \text{si } \underline{x} < \underline{y} \leq \bar{x} \leq \bar{y}, \\ [\bar{y}, \bar{x}], & \text{si } \underline{y} \leq \underline{x} \leq \bar{y} < \bar{x}, \\ [x], & \text{sinon.} \end{cases} \quad (\text{II.10})$$

Exemple 1.

$$\begin{aligned} [1, 4] \cap [3, 5] &= [3, 4], \\ [-2, -1] \sqcup [1, 3] &= [-2, 3], \\ [0, 4] \setminus [2, 5] &= [0, 2]. \end{aligned}$$

2. Cas des pavés

Tous les opérateurs ensemblistes définis pour les intervalles réels peuvent s'étendre aux pavés. Par exemple, un pavé est inclus dans un autre si toutes les composantes réelles du premier sont incluses dans celles du deuxième,

$$[\mathbf{x}] \subseteq [\mathbf{y}] \text{ si } \forall i, 1 \leq i \leq n, [x_i] \subseteq [y_i]. \quad (\text{II.11})$$

De la même manière, la réunion de deux pavés $[x]$ et $[y]$ de \mathbb{R}^n engendre un pavé dont les composantes sont issues de la réunion d'intervalles des composantes de $[x]$ et $[y]$,

$$[\mathbf{x}] \sqcup [\mathbf{y}] = ([x_1] \sqcup [y_1]) \times \cdots \times ([x_n] \sqcup [y_n]). \quad (\text{II.12})$$

L'intersection de deux pavés $[\mathbf{x}]$ et $[\mathbf{y}]$ de \mathbb{R}^n est un pavé de \mathbb{R}^n donné comme suit,

$$[\mathbf{x}] \cap [\mathbf{y}] = \begin{cases} \emptyset^n, & \text{si } \exists i, 1 \leq i \leq n, [x_i] \cap [y_i] = \emptyset, \\ ([x_1] \cap [y_1]) \times \cdots \times ([x_n] \cap [y_n]), & \text{sinon,} \end{cases} \quad (\text{II.13})$$

tel que \emptyset^n est le produit cartésien de n fois \emptyset . Dans la suite, nous utiliserons la notation \emptyset tout court pour les intervalles et les pavés.

Un pavé $[\mathbf{x}]$ de \mathbb{R}^n , privé d'un pavé $[\mathbf{y}]$, est un pavé $[\mathbf{z}] = [\mathbf{x}] \setminus [\mathbf{y}]$, dont les composantes réelles $[z_i]$, $1 \leq i \leq n$, sont données par

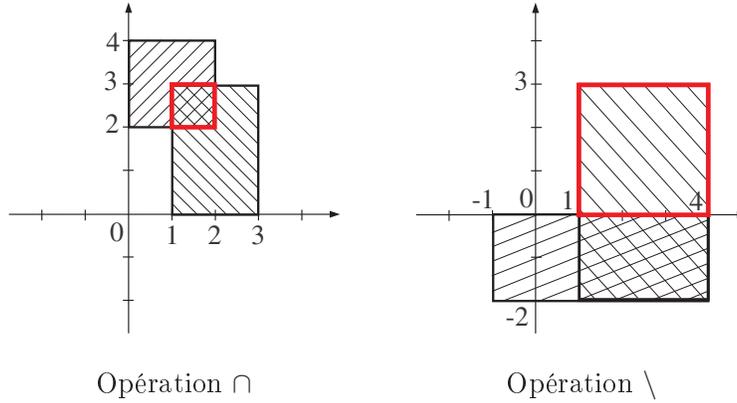
$$[z_i] = \begin{cases} [x_i] \setminus [y_i], & \text{si } \forall j \neq i, 1 \leq j \leq n, [x_j] \subseteq [y_j], \\ [x_i], & \text{sinon.} \end{cases} \quad (\text{II.14})$$

Le pavé $[\mathbf{z}]$ est vide, $[\mathbf{z}] = \emptyset$, si $[\mathbf{x}] \subseteq [\mathbf{y}]$.

Exemple 2.

$$\begin{aligned} [1, 3] \times [0, 3] \cap [0, 2] \times [2, 4] &= [1, 2] \times [2, 3], \\ [1, 4] \times [-2, 3] \setminus [-1, 4] \times [-2, 0] &= [1, 4] \times [0, 3]. \end{aligned}$$

FIG. II.1 montre une illustration des exemples ci-dessus.


 FIG. II.1: Exemples sur les opérations \cap et \setminus pour les pavés.

II.3.2 Opérations arithmétiques

L'arithmétique classique des réels est fondée principalement sur quatre opérateurs, notamment l'**addition** $+$, la **soustraction** $-$, la **multiplication** $*$ et la **division** $/$. L'arithmétique des intervalles profite de la double nature des intervalles pour leur appliquer ces opérateurs [Moore, 1979; Kearfott et Kreinovich, 1996; Jaulin *et al.*, 2001]. Soit \circ un opérateur de la liste suivante $\{+, -, *, /\}$. L'application de \circ à deux intervalles réels donne un intervalle défini comme suit,

$$[x] \circ [y] = [\{x \circ y \mid x \in [x], y \in [y]\}]. \quad (\text{II.15})$$

En utilisant les bornes inférieures et supérieures des intervalles, cette définition généralisée sera donnée par

$$[x] \circ [y] = [\min\{\underline{x} \circ \underline{y}, \underline{x} \circ \bar{y}, \bar{x} \circ \underline{y}, \bar{x} \circ \bar{y}\}, \max\{\underline{x} \circ \underline{y}, \underline{x} \circ \bar{y}, \bar{x} \circ \underline{y}, \bar{x} \circ \bar{y}\}]. \quad (\text{II.16})$$

La multiplication de deux intervalles sera notée indifféremment dans la suite $[x] * [y]$ ou $[x].[y]$.

Remarque 1. Cette définition n'est vraie pour la division que si les bornes des deux arguments sont soit strictement positives, soit strictement négatives. En effet, la division d'un intervalle réel par un autre intervalle contenant le zéro donne lieu à un intervalle réel ayant des bornes infinies.

L'équation (II.16) peut être simplifiée pour l'addition et la soustraction de la manière suivante,

$$\begin{cases} [x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}], \\ [x] - [y] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]. \end{cases} \quad (\text{II.17})$$

Outre les opérateurs classiques, d'autres opérateurs sont considérés tels que la **négation** et la **puissance carrée**,

$$\begin{cases} -[x] = [\{-x \mid x \in [x]\}] = [-\bar{x}, -\underline{x}], \\ [x]^2 = [\{x^2 \mid x \in [x]\}]. \end{cases} \quad (\text{II.18})$$

II. THÉORIE DES INTERVALLES

Si les bornes des arguments sont strictement positives, les intervalles réels engendrés par la multiplication, la division et la puissance carrée sont donnés par

$$\begin{cases} [x] * [y] = [\underline{x} * \underline{y}, \bar{x} * \bar{y}], \\ [x]/[y] = [\underline{x}/\bar{y}, \bar{x}/\underline{y}], \\ [x]^2 = [\underline{x}^2, \bar{x}^2]. \end{cases} \quad (\text{II.19})$$

Remarque 2. *Il est à noter ici que $[x]^2 \neq [x] * [x]$. Par exemple, si $[x] = [-1, 1]$, alors $[x]^2 = [0, 1]$, tandis que $[x] * [x] = [-1, 1]$.*

Exemple 3.

$$\begin{aligned} [1, 4] + [0, 2] &= [1, 6], \\ [0, 3] - [1, 4] &= [-4, 2], \\ -[-2, 3] &= [-3, 2], \\ [-1, 5] * [0, 3] &= [-3, 15], \\ [1, 2]/[-1, 1] &=] - \infty, +\infty[, \\ [0, 2]^2 &= [0, 4]. \end{aligned}$$

Les opérations arithmétiques définies pour les intervalles peuvent être étendues aux pavés. Pour tout opérateur \circ considéré dans la liste élargie $\{+, -, *, /, \wedge\}$, l'application de \circ conduit à un pavé défini comme suit,

$$[\mathbf{x}] \circ [\mathbf{y}] = ([x_1] \circ [y_1]) \times \cdots \times ([x_n] \circ [y_n]). \quad (\text{II.20})$$

Certaines propriétés des opérations sur les intervalles se déduisent directement des opérations correspondantes sur les réels. Il est facile de montrer que l'addition et la multiplication sont associatives et commutatives. En revanche, la multiplication est sous-distributive par rapport à l'addition,

$$[x] * ([y] + [z]) \subseteq [x] * [y] + [x] * [z]. \quad (\text{II.21})$$

Tous les opérateurs sur les intervalles vérifient en outre la propriété d'*inclusion minimale* définie par

$$\text{Si } [x_1] \subseteq [x_2] \text{ et } [y_1] \subseteq [y_2], \text{ alors } [x_1] \circ [y_1] \subseteq [x_2] \circ [y_2]. \quad (\text{II.22})$$

II.3.3 Fonctions d'inclusion

Soit \mathbf{f} une fonction définie de \mathbb{R}^n dans \mathbb{R}^m ,

$$\mathbf{f} : \mathbf{x} \in \mathbb{R}^n \longmapsto \mathbf{f}(\mathbf{x}) \in \mathbb{R}^m \quad (\text{II.23})$$

Si l'on suppose que l'argument de \mathbf{f} est un vecteur appartenant à un pavé $[\mathbf{x}]$ de \mathbb{R}^n , la fonction d'intervalles associée à $[\mathbf{x}]$ sera donnée par l'ensemble suivant,

$$\mathbf{f}([\mathbf{x}]) = \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\}. \quad (\text{II.24})$$

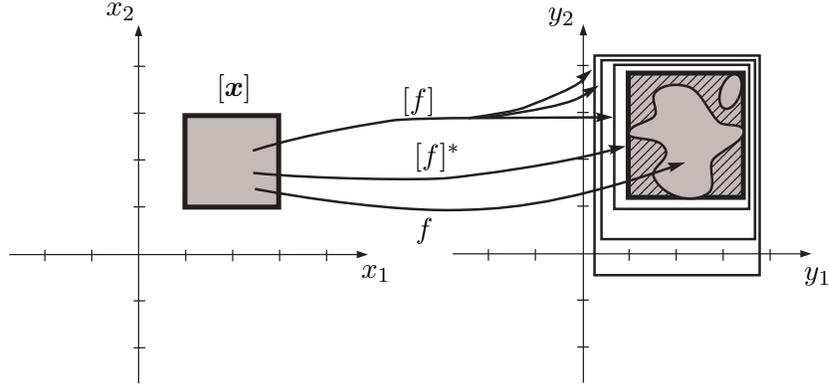


FIG. II.2: Illustration des fonctions d'inclusion.

L'ensemble résultant n'est pas nécessairement un pavé. Pour cette raison, nous introduisons les fonctions d'inclusion [Jaulin *et al.*, 2001], notées $[f]$ pour une fonction donnée f . Celles-ci, associées à un pavé $[x]$, engendrent des pavés englobant $f([x])$,

$$[f]([x]) = \{f(x) \mid x \in [x]\} \supseteq f([x]). \quad (\text{II.25})$$

Il est évident qu'une fonction donnée f peut avoir une infinité de fonctions d'inclusion. La fonction d'inclusion minimale de f , notée $[f]^*$, correspond au plus petit pavé englobant le résultat (cf. FIG. II.2).

Exemple 4. Soit x un vecteur bidimensionnel comprenant deux composantes : x_1 , comprise entre 0 et 4 et x_2 , comprise entre 0 et 2π . Dans le cadre des intervalles, nous pouvons remplacer les variables x_1 et x_2 par les intervalles auxquels elles appartiennent : $[x_1] = [0, 4]$ et $[x_2] = [0, 2\pi]$ respectivement. Considérons la fonction $f(x) = (f_1(x), f_2(x))$ définie par

$$\begin{cases} f_1(x) = x_1 \cdot \cos(x_2), \\ f_2(x) = x_1 \cdot \sin(x_2). \end{cases}$$

La fonction d'intervalles correspondant à f aura les composantes suivantes :

$$\begin{cases} f_1([x]) = [x_1] \cdot \cos([x_2]), \\ f_2([x]) = [x_1] \cdot \sin([x_2]). \end{cases}$$

Les équations ci-dessus correspondent à celles d'un disque centré sur le point $(0, 0)$ et ayant 4 comme rayon (cf. FIG. II.3). Le résultat n'est donc pas un pavé. La figure illustre en outre une série de fonctions d'inclusion de f , la minimale étant notée $[f]^*$.

Si l'on considère en particulier le cas d'une fonction réelle de \mathbb{R} dans \mathbb{R} , la fonction d'inclusion minimale de f associée à tout $[x] \in \mathbb{R}$ le plus petit intervalle réel incluant $f([x])$. Toute fonction élémentaire f continue et monotone a comme fonction d'inclusion minimale

$$[f]^*([x]) = [\min\{f(\underline{x}), f(\overline{x})\}, \max\{f(\underline{x}), f(\overline{x})\}]. \quad (\text{II.26})$$

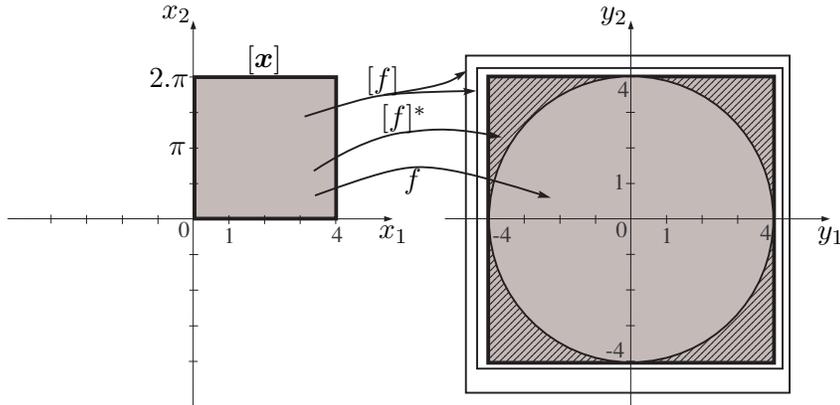


FIG. II.3: Illustration de l'exemple 4.

Exemple 5. Les fonctions d'inclusion minimales des fonctions \exp et \ln appliquées à $[x] = [\underline{x}, \bar{x}]$ sont données par

$$[\exp]([x]) = [\exp(\underline{x}), \exp(\bar{x})],$$

$$[\ln]([x]) = [\ln(\underline{x}), \ln(\bar{x})].$$

Il existe plusieurs méthodes pour calculer les fonctions d'inclusion d'une fonction donnée [Jaulin *et al.*, 2001]. Dans ce travail, nous utilisons les fonctions d'inclusion naturelles. Celles-ci sont obtenues en remplaçant chaque variable scalaire par l'intervalle correspondant et chaque fonction élémentaire par son extension aux intervalles. Vu la nature enveloppante des intervalles, les fonctions d'inclusion naturelles ne sont pas minimales en général. La précision de l'intervalle résultant est très liée à l'expression de la fonction. Une illustration de cette dépendance est donnée dans l'exemple ci-dessous, emprunté au livre de Jaulin *et al.* [Jaulin *et al.*, 2001].

Exemple 6. Considérons les expressions suivantes de la même fonction $f(x)$,

$$f_1(x) = x.(x + 1),$$

$$f_2(x) = x.x + x,$$

$$f_3(x) = x^2 + x,$$

$$f_4(x) = \left(x + \frac{1}{2}\right)^2 - \frac{1}{4}.$$

Si l'on évalue les fonctions d'inclusion naturelles de ces fonctions pour $x = [-1, 1]$, on obtient

$$[f_1]([x]) = [x].[x] + 1 = [-1, 1].([-1, 1] + 1) = [-2, 2],$$

$$[f_2]([x]) = [x].[x] + [x] = [-1, 1].[-1, 1] + [-1, 1] = [-2, 2],$$

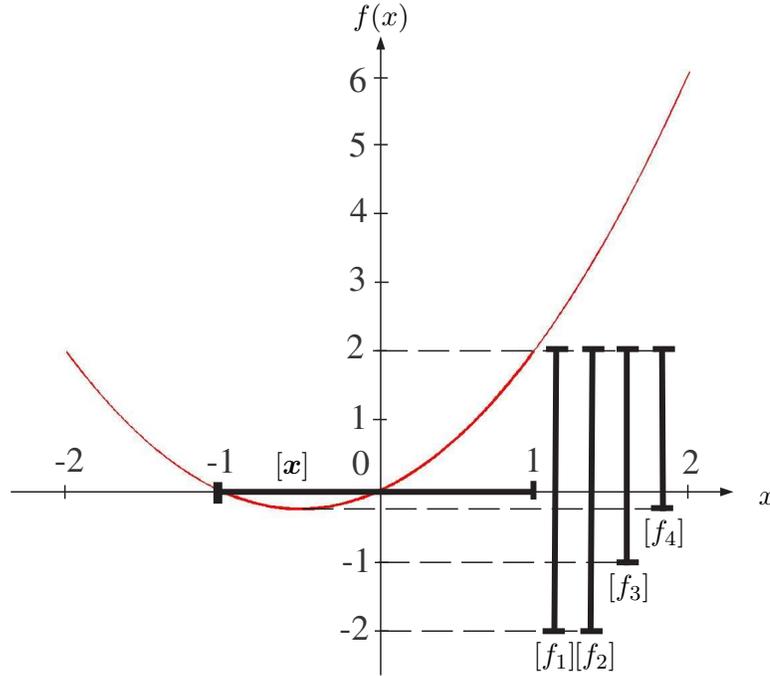


FIG. II.4: Quatre fonctions d'inclusion naturelles de la même fonction.

$$\begin{aligned}
 [f_3]([x]) &= [x]^2 + [x] = [-1, 1]^2 + [-1, 1] = [-1, 2], \\
 [f_4]([x]) &= ([x] + \frac{1}{2})^2 - \frac{1}{4} = ([-1, 1] + \frac{1}{2})^2 - \frac{1}{4} = [-\frac{1}{4}, 2].
 \end{aligned}$$

La précision de l'intervalle résultant dépend donc de l'expression de f . $[f_4]$ est minimale vu que x apparaît une fois dans f_4 et que f_4 est continue. FIG. II.4 illustre les résultats obtenus pour les différentes formes de f .

II.3.4 Tests d'inclusion

Soit λ un test défini de \mathbb{R}^n dans $\{1, 0\}$. Un test d'inclusion $[\lambda]$ pour λ est une fonction donnée par

$$[\lambda] : [x] \in \mathbb{R}^n \mapsto \lambda([x]) \in \{1, 0, \emptyset, [0, 1]\}, \quad (\text{II.27})$$

telle que, pour tout $[x]$ de \mathbb{R}^n ,

$$[\lambda]([x]) = \begin{cases} 1, & \text{si } \forall \mathbf{x} \in [x], \lambda(\mathbf{x}) = 1, \\ 0, & \text{si } \forall \mathbf{x} \in [x], \lambda(\mathbf{x}) = 0, \\ [0, 1], & \text{si } \exists \mathbf{a}, \mathbf{b} \in [x], \lambda(\mathbf{a}) = 0 \text{ et } \lambda(\mathbf{b}) = 1, \\ \emptyset, & \text{si le test est impossible.} \end{cases} \quad (\text{II.28})$$

Ainsi, le résultat du test est vrai pour $\lambda([x]) = 1$, faux pour $\lambda([x]) = 0$, impossible pour $\lambda([x]) = \emptyset$ ou indéterminé pour $\lambda([x]) = [0, 1]$ [Jaulin *et al.*, 2001]. Les tests d'inclusion

II. THÉORIE DES INTERVALLES

sont définis pour prouver que tous les points d'un pavé donné satisfont une certaine propriété.

Exemple 7. Soit le test λ donné par

$$\lambda : \mathbf{x} = (x_1, x_2) \in \mathbb{R}^2 \mapsto \lambda(\mathbf{x}) \in \{1, 0\},$$

tel que

$$\lambda(\mathbf{x}) = \begin{cases} 1, & \text{si } x_1 - x_2 \geq 0, \\ 0, & \text{sinon.} \end{cases}$$

Le test d'inclusion minimal associé à λ est donné par

$$[\lambda]([\mathbf{x}]) = \begin{cases} 1, & \text{si } \underline{x}_1 - \overline{x}_2 \geq 0, \\ 0, & \text{si } \overline{x}_1 - \underline{x}_2 < 0, \\ [0, 1], & \text{sinon.} \end{cases}$$

Par exemple, pour $[\mathbf{x}] = [2, 5] \times [1, 3]$, $[\lambda]([\mathbf{x}]) = [0, 1]$ puisque $-1 \leq x_1 - x_2 \leq 4$ pour tout $(x_1, x_2) \in [\mathbf{x}]$. Plus particulièrement, pour $(x_1, x_2) = (2, 3)$, $x_1 - x_2 = -1 < 0$. D'un autre côté, pour $(x_1, x_2) = (5, 1)$, $x_1 - x_2 = 4 \geq 0$.

II.4 Sous-pavages

Un **sous-pavage** \mathbb{X} d'un pavé $[\mathbf{x}] \subset \mathbb{R}^n$ est un ensemble (fini ou non) de sous-pavés de $[\mathbf{x}]$ de tailles non nulles [Jaulin, 1994; Kieffer, 1999; Jaulin *et al.*, 2001]. Ces pavés ne se recouvrent pas, c'est-à-dire que leur intersection deux à deux est vide sauf éventuellement aux frontières. Les sous-pavages ont pour but d'approximer d'une manière garantie des sous-ensembles quelconques de \mathbb{R}^n . Quand un sous-pavage de $[\mathbf{x}]$ couvre $[\mathbf{x}]$, il est alors appelé **pavage** de $[\mathbf{x}]$. Dans la suite, nous commençons par définir les sous-pavages réguliers. Nous introduisons ensuite les représentations en arbres binaires des sous-pavages. Nous présentons enfin l'algorithme d'inversion ensembliste SIVIA.

II.4.1 Sous-pavage régulier

Avant de définir plus précisément les sous-pavages, nous introduisons quelques définitions. Soit $[\mathbf{x}]$ un pavé de \mathbb{R}^n ,

$$[\mathbf{x}] = [x_1] \times \cdots \times [x_n] = [\underline{x}_1, \overline{x}_1] \times \cdots \times [\underline{x}_n, \overline{x}_n], \quad (\text{II.29})$$

et soit j l'indice de la première composante de $[\mathbf{x}]$ ayant la taille maximale,

$$j = \min\{i \mid w([x_i]) = w([\mathbf{x}])\}. \quad (\text{II.30})$$

Nous définissons alors les sous-pavés $G[\mathbf{x}]$ et $D[\mathbf{x}]$ de $[\mathbf{x}]$ comme suit,

$$\begin{aligned} G[\mathbf{x}] &= [\underline{x}_1, \overline{x}_1] \times \cdots \times \left[\underline{x}_j, \frac{\underline{x}_j + \overline{x}_j}{2}\right] \times \cdots \times [\underline{x}_n, \overline{x}_n], \\ D[\mathbf{x}] &= [\underline{x}_1, \overline{x}_1] \times \cdots \times \left[\frac{\underline{x}_j + \overline{x}_j}{2}, \overline{x}_j\right] \times \cdots \times [\underline{x}_n, \overline{x}_n]. \end{aligned} \quad (\text{II.31})$$

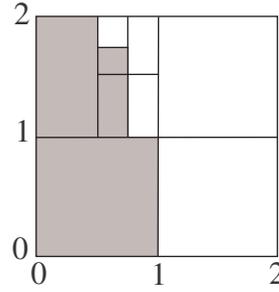


FIG. II.5: Pavage et sous-pavage (en gris) réguliers du pavé $[0, 2] \times [0, 2]$.

$G[\mathbf{x}]$ et $D[\mathbf{x}]$ sont respectivement les **enfants gauche** et **droit** de $[\mathbf{x}]$. La génération de ces deux enfants est appelée la **bissection** de $[\mathbf{x}]$. La **réunification** est l'opération de fusion des enfants $G[\mathbf{x}]$ et $D[\mathbf{x}]$ pour obtenir le pavé parent $[\mathbf{x}]$.

Exemple 8. Soit $[\mathbf{x}] = [0, 4] \times [1, 3] \times [2, 6]$. La taille de $[\mathbf{x}]$ est égale à 4. Les enfants gauche et droit de $[\mathbf{x}]$ sont donnés par

$$\begin{aligned} G[\mathbf{x}] &= [0, 2] \times [1, 3] \times [2, 6], \\ D[\mathbf{x}] &= [2, 4] \times [1, 3] \times [2, 6]. \end{aligned}$$

Un sous-pavage de $[\mathbf{x}]$ est dit **régulier** si chacun de ses pavés peut être obtenu à partir de $[\mathbf{x}]$ par une succession finie de bisections et de sélections. Les sous-pavages réguliers sont des classes de sous-ensembles de \mathbb{R}^n facilement manipulables avec un ordinateur [Jaulin, 1994; Sam-Haroud et Faltings, 1996]. FIG. II.5 illustre un pavage régulier du pavé $[\mathbf{x}] = [0, 2] \times [0, 2]$. Elle montre également, en gris, un sous-pavage régulier du pavé $[\mathbf{x}]$. Ce dernier est obtenu par l'union de quatre sous-pavés de $[\mathbf{x}]$ obtenus suite aux bisections successives suivantes,

$$\begin{aligned} GG[\mathbf{x}] &= [0, 1] \times [0, 1], \\ GDG[\mathbf{x}] &= [0, \frac{1}{2}] \times [1, 2], \\ GDDGG[\mathbf{x}] &= [\frac{1}{2}, \frac{3}{4}] \times [1, \frac{3}{2}], \\ GDDDGG[\mathbf{x}] &= [\frac{1}{2}, \frac{3}{4}] \times [\frac{3}{2}, \frac{7}{4}]. \end{aligned}$$

II.4.2 Représentation par un arbre binaire

Sur l'ordinateur, un sous-pavage régulier est représenté par un arbre binaire formé de nœuds. L'ensemble de nœuds peut être vide, formé d'un seul nœud étant la racine de l'arbre ou composé de deux sous-arbres binaires ayant une intersection vide, en particulier les sous-arbres gauche et droit. L'arbre binaire associé au sous-pavage de FIG. II.5 est illustré dans FIG. II.6, où 1 indique que le nœud correspondant appartient au sous-pavage. Dans la figure, A est la racine de l'arbre et B et C sont respectivement

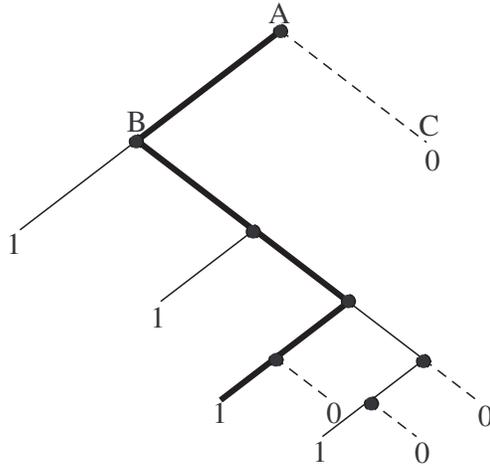


FIG. II.6: Arbre binaire associé au sous-pavage en gris de FIG. II.5.

les enfants gauche et droit de A. Le sous-arbre droit de A est vide. L'arbre est construit à partir de la liste des pavés contenus dans le sous-pavage. La croissance des branches dépend de la manière dont le pavé initial $[\mathbf{x}]$ est découpé. La liste des branches en gras de FIG. II.6 correspond par exemple au pavé $GDDGG[\mathbf{x}]$ du sous-pavage. Le nombre de branches dans une liste donnée correspond au nombre de bisections nécessaires pour obtenir le sous-pavé à partir de $[\mathbf{x}]$.

II.4.3 Algorithme d'inversion ensembliste : SIVIA

Soit \mathbf{f} une fonction, éventuellement non linéaire, définie de \mathbb{R}^n dans \mathbb{R}^m et soit \mathbb{Y} un sous-ensemble de \mathbb{R}^m . Le problème d'inversion ensembliste consiste à trouver le sous-ensemble \mathbb{X} défini comme suit,

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\}. \quad (\text{II.32})$$

Dans le cadre de l'analyse par intervalles, deux sous-pavages réguliers $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$ tels que

$$\underline{\mathbb{X}} \subseteq \mathbb{X} \subseteq \overline{\mathbb{X}}, \quad (\text{II.33})$$

peuvent être obtenus à l'aide de l'algorithme d'inversion ensembliste SIVIA (pour *Set Inverter Via Interval Analysis*) [Jaulin et Walter, 1993a,b].

SIVIA nécessite un pavé de recherche $[\mathbf{x}]_0$, dans lequel $\overline{\mathbb{X}}$ est sûrement inclus. Ce pavé initial est découpé successivement et chaque sous-pavé est testé afin de générer le sous-pavage solution. Soit $[\mathbf{f}]$ une fonction d'inclusion de \mathbf{f} . Quatre situations peuvent être rencontrées :

1. Si $[\mathbf{f}]([\mathbf{x}])$ est entièrement inclus dans \mathbb{Y} , alors $[\mathbf{x}]$ fait partie de la solution. Il est dit **acceptable**. Il sera alors ajouté aux sous-pavages $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$.

2. Si $[f]([x])$ a une intersection vide avec \mathbb{Y} , alors $[x]$ n'appartient pas à la solution. Dans ce cas, $[x]$ est dit **inacceptable** et il sera éliminé.
3. Si $[f]([x])$ a une intersection non-vide avec \mathbb{Y} , mais n'est pas inclus dans \mathbb{Y} , alors $[x]$ contient une partie de la solution. $[x]$ est dit **ambigu**. Si ce pavé a une largeur supérieure à une certaine précision ε , alors il sera découpé et le test sera récursivement appliqué aux nouveaux pavés, enfants de $[x]$.
4. Dans le dernier cas, le pavé considéré $[x]$ est ambigu, avec une largeur inférieure à la précision ε . Il est alors considéré trop petit et est stocké dans le sous-pavage externe $\overline{\mathbb{X}}$.

Une illustration de ces quatre situations est présentée dans FIG. II.7. La figure montre à droite les branches correspondant aux bisections du pavé initial $[x]_0$. Le sous-pavage $\Delta\mathbb{X} = \overline{\mathbb{X}} \setminus \underline{\mathbb{X}}$ correspond aux pavés contenus dans $\overline{\mathbb{X}}$ mais non inclus dans $\underline{\mathbb{X}}$. Ce sous-pavage, dit **couche d'incertitude**, est régulier avec des pavés ayant tous une largeur inférieure à ε . L'algorithme 1 illustre les démarches de l'approche SIVIA. Les sous-pavages $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}}$ sont vides à l'entrée initiale de la fonction SIVIA(\cdot).

Algorithme 1 : Fonction SIVIA($[x]_0, f, \mathbb{Y}, \varepsilon, \underline{\mathbb{X}}, \overline{\mathbb{X}}$).

Entrées : $[x]_0, f, \mathbb{Y}, \varepsilon, \underline{\mathbb{X}}, \overline{\mathbb{X}}$;
Sorties : $\underline{\mathbb{X}}, \overline{\mathbb{X}}$;
si $[f]([x]_0) \subseteq \mathbb{Y}$ **alors**
 $\underline{\mathbb{X}} \leftarrow [x]_0, \overline{\mathbb{X}} \leftarrow [x]_0$;
sinon si $[f]([x]_0) \cap \mathbb{Y} = \emptyset$ **alors**
 Éliminer $[x]_0$;
sinon
 si $w([x]_0) \geq \varepsilon$ **alors**
 SIVIA($G[x]_0, f, \mathbb{Y}, \varepsilon, \underline{\mathbb{X}}, \overline{\mathbb{X}}$), SIVIA($D[x]_0, f, \mathbb{Y}, \varepsilon, \underline{\mathbb{X}}, \overline{\mathbb{X}}$);
 sinon
 $\overline{\mathbb{X}} \leftarrow [x]_0$;
 fin
fin

Exemple 9. Soit \mathbb{X} l'ensemble des points (x_1, x_2) de \mathbb{R}^2 tels que

$$\mathbb{X} = \{(x_1, x_2) \mid x_1^2 + x_2^2 \in [1, 2]\}.$$

Le domaine initial de ces points est donné par $[x]_0 = [-4, 4] \times [-4, 4]$ et la précision est fixée à $\varepsilon = 0.1$. La solution correspond à la couronne limitée par les cercles centrés sur $(0, 0)$ et ayant 1 et $\sqrt{2}$ comme rayons respectifs. L'application de SIVIA donne lieu à deux sous-pavages de $[x]_0$ interne et externe à la couronne. FIG. II.8 illustre en bleu le sous-pavage $\underline{\mathbb{X}}$ interne à \mathbb{X} , en rouge la couche d'incertitude $\Delta\mathbb{X} = \overline{\mathbb{X}} \setminus \underline{\mathbb{X}}$ et en vert les cercles limitant la couronne solution. Selon la valeur de ε , la couche d'incertitude sera plus ou moins large.

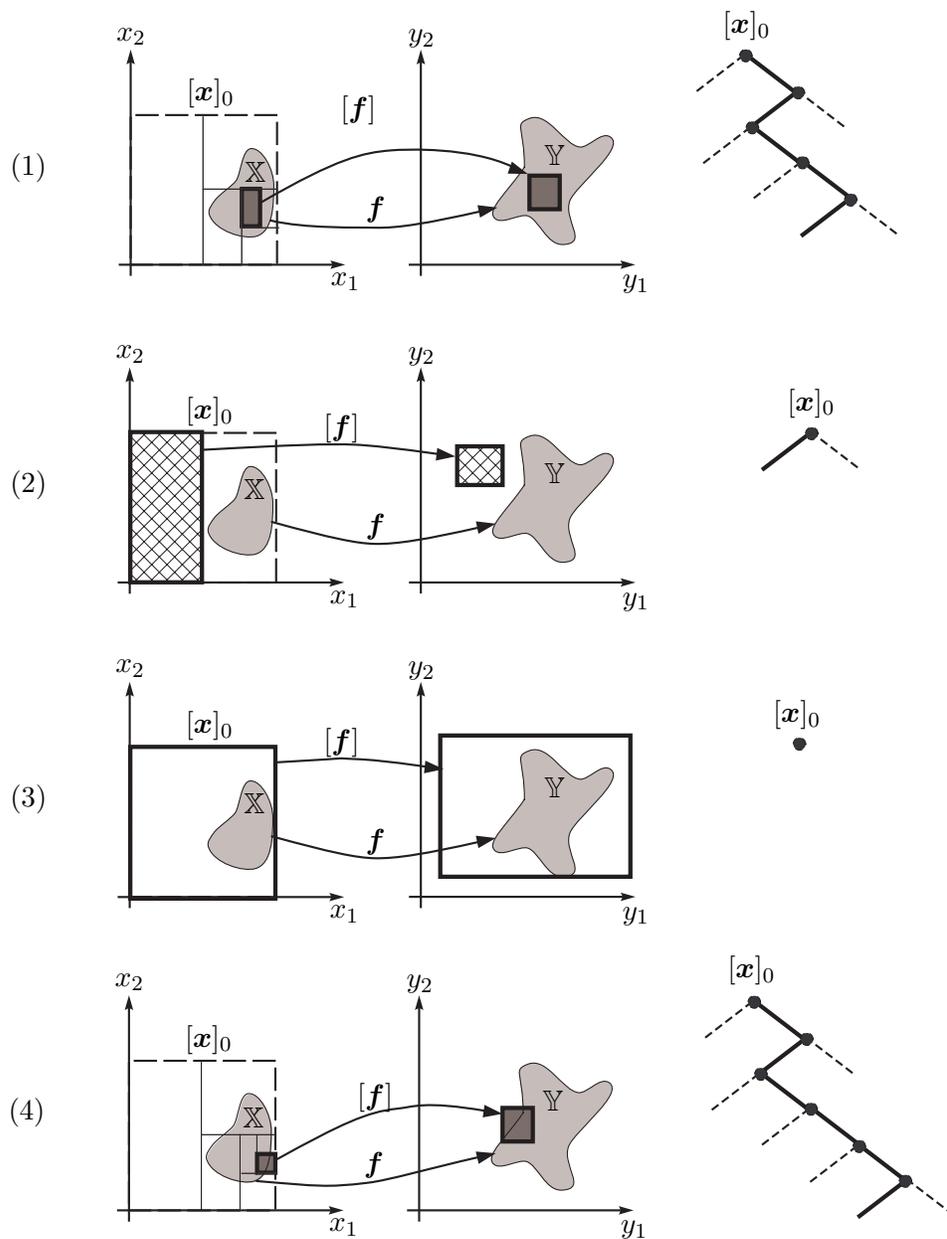


FIG. II.7: Quatre situations rencontrées par l'algorithme SIVIA.

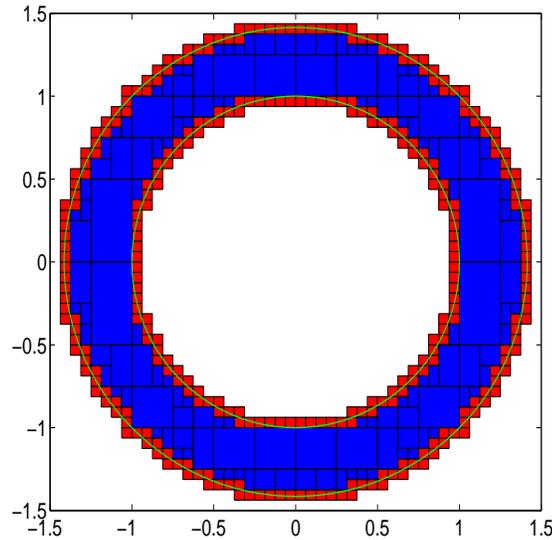


FIG. II.8: Illustration de l'exemple 9.

En considérant un problème défini par un système d'équations et/ou d'inéquations, SIVIA permet de résoudre le problème d'une manière garantie. SIVIA partage l'espace de recherche en des pavés acceptables, inacceptables ou ambigus selon les contraintes. Seuls les pavés acceptables et ambigus de tailles inférieures à la précision requise sont sélectionnés. L'algorithme SIVIA a une complexité exponentielle en temps et en espace dépendamment des dimensions du problème. En effet, SIVIA est efficace pour des problèmes dont le nombre de paramètres à estimer n'est pas élevé. Dans les applications où le nombre de paramètres est assez élevé, d'autres types d'algorithmes sont utilisés. Nous en citons les contracteurs que nous présentons dans la section suivante.

II.5 Contracteurs

Soient $x_i, i \in \{1, \dots, n\}$, n variables réelles et soient $f_j, j \in \{1, \dots, m\}$, m contraintes de la forme suivante,

$$f_j(x_1, \dots, x_n) = 0, \quad j \in \{1, \dots, m\}. \quad (\text{II.34})$$

Chaque variable x_i appartient à un domaine D_i . Dans la suite, nous supposons que les domaines D_i sont des intervalles réels, notés $[x_i]$. Le vecteur $\mathbf{x} = (x_1, \dots, x_n)$ appartient ainsi à un domaine a priori $[\mathbf{x}] = [x_1] \times \dots \times [x_n]$. Soit \mathbf{f} la fonction ayant pour coordonnées les contraintes f_j . L'équation (II.34) peut alors s'écrire sous forme vectorielle, $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. Tel problème est dit **problème de satisfaction de contraintes** ou CSP (pour *Constraint Satisfaction Problem*) [Cleary, 1987; Lhomme, 1993; Sam-Haroud, 1995; Haroud *et al.*, 1995; Sam-Haroud et Faltings, 1996], qui s'écrit comme suit,

$$\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}]). \quad (\text{II.35})$$

II. THÉORIE DES INTERVALLES

Les CSPs peuvent impliquer des contraintes d'égalités et d'inégalités. Dans cette section, nous commençons par une définition des contracteurs, permettant de résoudre un CSP. Nous présentons ensuite différents contracteurs proposés dans la littérature.

II.5.1 Définition des contracteurs

La solution du CSP \mathcal{H} défini dans (II.35) est un sous-ensemble de \mathbb{R}^n donné comme suit,

$$\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) = \mathbf{0}\}. \quad (\text{II.36})$$

Dans le contexte ensembliste, résoudre le problème consiste à définir une approximation externe de \mathbb{S} . L'avantage principal de cette approche est qu'elle produit, d'une manière garantie, un pavé englobant le sous-ensemble solution \mathbb{S} . Plusieurs algorithmes, nommés **contracteurs**, ont été proposés dans cet esprit.

La **contraction** d'un CSP \mathcal{H} consiste à remplacer le domaine initial $[\mathbf{x}]$ par un domaine plus petit $[\mathbf{x}']$ de telle sorte que la solution reste intacte, $\mathbb{S} \subseteq [\mathbf{x}'] \subseteq [\mathbf{x}]$. La contraction optimale de \mathcal{H} consiste à remplacer $[\mathbf{x}]$ par la plus petite boîte englobant \mathbb{S} . Tout opérateur réalisant une contraction de \mathcal{H} est appelé **contracteur**. Les contracteurs seront notés \mathcal{C} avec un indice indiquant le principe sur lequel ils sont basés. Afin de garder une complexité polynomiale, ces contracteurs n'impliquent pas des bisections. TAB. II.1 illustre la liste des contracteurs de base qui seront présentés dans ce travail. Notons qu'il est parfois commode de définir plusieurs types de domaines pour les quantités incertaines, tels que $[\mathbf{A}]$, $[\mathbf{b}]$ et $[\mathbf{p}]$ dans TAB. II.1.

Contracteur	Principe de base
$\mathcal{C}_{EG}(\mathbf{A}\cdot\mathbf{p} - \mathbf{b} = \mathbf{0}, [\mathbf{A}], [\mathbf{p}], [\mathbf{b}])$	Elimination de Gauss
$\mathcal{C}_{GS}(\mathbf{A}\cdot\mathbf{p} - \mathbf{b} = \mathbf{0}, [\mathbf{A}], [\mathbf{p}], [\mathbf{b}])$	Algorithme de Gauss-Seidel
$\mathcal{C}_K(\mathbf{f}(\mathbf{x}) = \mathbf{0}, [\mathbf{x}])$	Méthode de Krawczyk
$\mathcal{C}_{\uparrow\downarrow}(\mathbf{f}(\mathbf{x}) = \mathbf{0}, [\mathbf{x}])$	Propagation <i>forward-backward</i>

TAB. II.1: Quelques contracteurs de base.

II.5.2 Contracteur basé sur l'élimination de Gauss

Les contracteurs les plus souvent utilisés définissent des **sous-solveurs finis** afin de résoudre un CSP \mathcal{H} . Ces sous-solveurs sont des fonctions exprimant certaines variables en fonction d'autres. Soit \mathbf{u} un sous-vecteur de \mathbf{x} , $\mathbf{u} = (x_{i_1}, \dots, x_{i_{n_u}})$. $\mathcal{I} = \{i_1, \dots, i_{n_u}\}$ est l'ensemble des indices des variables de \mathbf{x} sélectionnées dans \mathbf{u} . \mathbf{u} est également noté $\mathbf{x}_{\mathcal{I}}$. Un sous-solveur fini ϕ de \mathcal{H} associe à \mathbf{u} un autre sous-vecteur \mathbf{v} , tel que $\mathbf{v} = \mathbf{x}_{\mathcal{J}}$, $\mathcal{J} \subset \{1, \dots, n\}$ et $\mathcal{I} \cap \mathcal{J} = \emptyset$,

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \text{ alors } \mathbf{v} \in \phi(\mathbf{u}). \quad (\text{II.37})$$

En général, $\phi(\mathbf{u})$ est un singleton. La relation (II.37) s'écrit alors

$$\mathbf{v} = \phi(\mathbf{u}). \quad (\text{II.38})$$

L'exemple ci-dessous illustre le concept des sous-solveurs.

Exemple 10. *Supposons le CSP suivant,*

$$\mathcal{H} : \left(\begin{array}{l} x_1.x_2 - x_3 = 0 \\ x_4 - \exp(x_2) = 0 \\ [x_1] = [x_2] = [x_3] = [x_4] = \mathbb{R} \end{array} \right).$$

Plusieurs sous-solveurs de \mathcal{H} peuvent être obtenus, nous en citons quatre,

$$\begin{aligned} x_3 &= \phi_a(x_1, x_2) = x_1.x_2, \\ x_2 &= \phi_b(x_1, x_3) = x_3/x_1 \text{ si } x_1 \neq 0, \mathbb{R} \text{ sinon,} \\ x_1 &= \phi_c(x_2, x_3) = x_3/x_2 \text{ si } x_2 \neq 0, \mathbb{R} \text{ sinon,} \\ x_4 &= \phi_d(x_2) = \exp(x_2). \end{aligned}$$

Mettre sous forme d'intervalles les sous-solveurs consiste à remplacer les variables par leurs domaines d'appartenance et les fonctions par des fonctions d'inclusion qui leur correspondent. Si l'on considère le sous-solveur ϕ_d de l'exemple 10, l'utilisation d'intervalles conduit à la forme suivante,

$$[x_4] = [\phi_d]([x_2]) = [\exp](\mathbb{R}).$$

Considérons un CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$ et l'un de ses sous-solveurs ϕ ayant comme entrée le vecteur $\mathbf{u} = \mathbf{x}_{\mathcal{I}}$ et comme sortie le vecteur $\mathbf{v} = \mathbf{x}_{\mathcal{J}}$. Si $[\phi]$ est une fonction d'inclusion de ϕ , alors une contraction de \mathcal{H} peut être réalisée en remplaçant chaque domaine $[x_j]$, $j \in \mathcal{J}$, par le domaine $[x_j] \cap [\phi_j](\mathbf{u})$,

$$[x_j] \leftarrow [x_j] \cap [\phi_j](\mathbf{u}). \quad (\text{II.39})$$

Exemple 11. *Considérons à nouveau le CSP de l'exemple 10 avec le sous-solveur ϕ_d . Le domaine $[x_4]$ sera réduit comme suit,*

$$[x_4] = [x_4] \cap [\phi_d]([x_2]) = \mathbb{R} \cap [\exp](\mathbb{R}) = \mathbb{R} \cap [0, +\infty[= [0, +\infty[.$$

Le contracteur basé sur l'élimination de Gauss, noté \mathcal{C}_{EG} , effectue ce type de contraction. Il a été proposé pour résoudre les CSP à systèmes d'équations linéaires à base d'intervalles. Le problème consiste à trouver le plus petit pavé englobant l'ensemble solution du CSP

$$\mathcal{H} : (\mathbf{A}.\mathbf{p} - \mathbf{b} = \mathbf{0}, \mathbf{A} \in [\mathbf{A}], \mathbf{p} \in [\mathbf{p}], \mathbf{b} \in [\mathbf{b}]), \quad (\text{II.40})$$

où

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n_p} \\ \vdots & & & \vdots \\ a_{n_p 1} & a_{n_p 2} & \dots & a_{n_p n_p} \end{pmatrix}, \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_{n_p} \end{pmatrix}. \quad (\text{II.41})$$

Les variables de \mathcal{H} forment alors le vecteur

$$\mathbf{x} = (a_{11}, \dots, a_{n_p n_p}, b_1, b_{n_p}, p_1, \dots, p_{n_p}). \quad (\text{II.42})$$

II. THÉORIE DES INTERVALLES

Le contracteur \mathcal{C}_{EG} tend à réduire le domaine de \mathbf{p} en supposant le vecteur $\mathbf{u} = (a_{11}, \dots, a_{n_p n_p}, b_1, b_{n_p})$ connu. L'algorithme est composé de deux étapes, la première consiste à réduire le système d'équations en une forme triangulaire ou échelonnée. Ceci est obtenu grâce à des opérations élémentaires sur les lignes du système. La seconde étape consiste ensuite à trouver la solution par substitution. Les domaines que l'on réduit dans cet algorithme sont ceux des variables p_1, \dots, p_{n_p} . Le pseudo-code du contracteur \mathcal{C}_{EG} est illustré dans l'algorithme 2.

Algorithme 2 : Pseudo-code du contracteur basé sur l'élimination de Gauss.

Entrées : $[a_{11}], \dots, [a_{n_p n_p}], [b_1], \dots, [b_{n_p}], [p_1], \dots, [p_{n_p}];$

Sorties : $[p_1], \dots, [p_{n_p}];$

pour $i = 1, \dots, n_p - 1$ **faire**

pour $j = i + 1, \dots, n_p$ **faire**

$[\alpha_j] = [a_{ji}] / [a_{ii}];$

$[b_j] = [b_j] - [\alpha_j] \cdot [b_i];$

pour $k = i + 1, \dots, n_p$ **faire**

$[a_{jk}] = [a_{jk}] - [\alpha_j] \cdot [a_{ik}];$

fin

fin

fin

pour $i = n_p, \dots, 1$ **faire**

$[p_i] = [p_i] \cap ([b_i] - \sum_{j=i+1}^{n_p} [a_{ij}] \cdot [p_j]) / [a_{ii}];$

fin

Exemple 12. *Considérons l'exemple du CSP linéaire tiré du livre de Jaulin et al. [Jaulin et al., 2001],*

$$\mathcal{H} : (\mathbf{A} \cdot \mathbf{p} - \mathbf{b} = \mathbf{0}),$$

avec

$$[\mathbf{A}] = \begin{pmatrix} [4, 5] & [-1, 1] & [\frac{3}{2}, \frac{5}{2}] \\ [-\frac{1}{2}, \frac{1}{2}] & [-7, -5] & [1, 2] \\ [-\frac{3}{2}, -\frac{1}{2}] & [-\frac{7}{10}, -\frac{1}{2}] & [2, 3] \end{pmatrix}, [\mathbf{b}] = \begin{pmatrix} [3, 4] \\ [0, 2] \\ [3, 4] \end{pmatrix} \text{ et } [\mathbf{p}] = \begin{pmatrix} \mathbb{R} \\ \mathbb{R} \\ \mathbb{R} \end{pmatrix}.$$

L'application du contracteur \mathcal{C}_{EG} réduit le domaine de \mathbf{p} comme suit,

$$[\mathbf{p}] = \begin{pmatrix} [-1.81928, 1.16873] \\ [-0.414071, 1.72523] \\ [0.700233, 3.42076] \end{pmatrix}.$$

II.5.3 Contracteur basé sur l'algorithme de Gauss-Seidel

Le contracteur basé sur l'algorithme de Gauss-Seidel, noté \mathcal{C}_{GS} , est également proposé pour les CSPs linéaires du type

$$\mathcal{H} : (\mathbf{A}.\mathbf{p} - \mathbf{b} = \mathbf{0}, \mathbf{A} \in [\mathbf{A}], \mathbf{p} \in [\mathbf{p}], \mathbf{b} \in [\mathbf{b}]), \quad (\text{II.43})$$

où la matrice \mathbf{A} est supposée carrée. \mathbf{A} peut s'écrire comme la somme d'une matrice diagonale et d'une matrice à diagonale nulle,

$$\mathbf{A} = \text{diag}(\mathbf{A}) + \text{extdiag}(\mathbf{A}). \quad (\text{II.44})$$

La relation $\mathbf{A}.\mathbf{p} - \mathbf{b} = \mathbf{0}$ est ainsi équivalente à

$$\text{diag}(\mathbf{A}).\mathbf{p} + \text{extdiag}(\mathbf{A}).\mathbf{p} = \mathbf{b}. \quad (\text{II.45})$$

En supposant que $\text{diag}(\mathbf{A})$ est inversible, l'équation (II.45) peut être reformulée comme suit,

$$\mathbf{p} = (\text{diag}(\mathbf{A}))^{-1}.(\mathbf{b} - \text{extdiag}(\mathbf{A}).\mathbf{p}). \quad (\text{II.46})$$

Un sous-solveur en virgule fixe de \mathcal{H} est ainsi donné par

$$\psi \left(\begin{array}{c} \mathbf{A} \\ \mathbf{b} \\ \mathbf{p} \end{array} \right) = \left(\begin{array}{c} \mathbf{A} \\ \mathbf{b} \\ (\text{diag}(\mathbf{A}))^{-1}.(\mathbf{b} - \text{extdiag}(\mathbf{A}).\mathbf{p}) \end{array} \right). \quad (\text{II.47})$$

Une fonction d'inclusion de ψ est donnée par

$$[\psi] \left(\begin{array}{c} [\mathbf{A}] \\ [\mathbf{b}] \\ [\mathbf{p}] \end{array} \right) = \left(\begin{array}{c} [\mathbf{A}] \\ [\mathbf{b}] \\ (\text{diag}([\mathbf{A}]))^{-1}.([\mathbf{b}] - \text{extdiag}([\mathbf{A}]).[\mathbf{p}]) \end{array} \right). \quad (\text{II.48})$$

Le contracteur \mathcal{C}_{GS} consiste à réduire le domaine $[\mathbf{p}]$ en appliquant le sous-solveur ψ ,

$$\left(\begin{array}{c} [\mathbf{A}] \\ [\mathbf{b}] \\ [\mathbf{p}] \end{array} \right) \leftarrow \left(\begin{array}{c} [\mathbf{A}] \\ [\mathbf{b}] \\ [\mathbf{p}] \cap (\text{diag}([\mathbf{A}]))^{-1}.([\mathbf{b}] - \text{extdiag}([\mathbf{A}]).[\mathbf{p}]) \end{array} \right). \quad (\text{II.49})$$

Exemple 13. *Considérons le même CSP de l'exemple 12 avec un domaine initial de \mathbf{p} plus restreint,*

$$[\mathbf{A}] = \left(\begin{array}{ccc} [4, 5] & [-1, 1] & [\frac{3}{2}, \frac{5}{2}] \\ [-\frac{1}{2}, \frac{1}{2}] & [-7, -5] & [1, 2] \\ [-\frac{3}{2}, -\frac{1}{2}] & [-\frac{7}{10}, -\frac{1}{2}] & [2, 3] \end{array} \right), [\mathbf{b}] = \left(\begin{array}{c} [3, 4] \\ [0, 2] \\ [3, 4] \end{array} \right) \text{ et } [\mathbf{p}] = \left(\begin{array}{c} [-10, 10] \\ [-10, 10] \\ [-10, 10] \end{array} \right).$$

L'inverse de la matrice diagonale ainsi que la matrice à diagonale nulle associée à $[\mathbf{A}]$ sont données par

$$(\text{diag}([\mathbf{A}]))^{-1} = \left(\begin{array}{ccc} [\frac{1}{5}, \frac{1}{4}] & [0, 0] & [0, 0] \\ [0, 0] & [-\frac{1}{5}, -\frac{1}{7}] & [0, 0] \\ [0, 0] & [0, 0] & [\frac{1}{3}, \frac{1}{2}] \end{array} \right)$$

II. THÉORIE DES INTERVALLES

et

$$\text{extdiag}([\mathbf{A}]) = \begin{pmatrix} [0, 0] & [-1, 1] & [\frac{3}{2}, \frac{5}{2}] \\ [-\frac{1}{2}, \frac{1}{2}] & [0, 0] & [1, 2] \\ [-\frac{3}{2}, -\frac{1}{2}] & [-\frac{7}{10}, -\frac{1}{2}] & [0, 0] \end{pmatrix}.$$

L'application du contracteur \mathcal{C}_{GS} produit le résultat suivant,

$$[\mathbf{p}] = \begin{pmatrix} [-8, 9.75] \\ [-5.4001, 5.0001] \\ [-9.5, 10] \end{pmatrix}.$$

Il est clair que le contracteur \mathcal{C}_{GS} fonctionne moins bien que le contracteur \mathcal{C}_{EG} pour cet exemple, ce qui n'est pas toujours le cas. L'itération de \mathcal{C}_{GS} permettra de réduire au fur et à mesure l'incertitude au niveau du résultat.

II.5.4 Contracteur basé sur la méthode de Krawczyk

Le contracteur \mathcal{C}_K est basé sur la méthode de Krawczyk. Il a été proposé pour contracter les CSPs généralement non linéaires, $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$. De la même façon que \mathcal{C}_{GS} , le contracteur \mathcal{C}_K utilise des sous-solveurs en virgule fixe. Pour toute matrice \mathbf{M} inversible, si $n_f = n_x$ et \mathbf{f} est différentiable,

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \Leftrightarrow \mathbf{M} \cdot \mathbf{f}(\mathbf{x}) = \mathbf{0} \Leftrightarrow \mathbf{x} - \mathbf{M} \cdot \mathbf{f}(\mathbf{x}) = \mathbf{x}. \quad (\text{II.50})$$

Un sous-solveur en virgule fixe de \mathcal{H} est ainsi donné par

$$\psi(\mathbf{x}) = \mathbf{x} - \mathbf{M} \cdot \mathbf{f}(\mathbf{x}). \quad (\text{II.51})$$

La fonction d'inclusion centrée de ψ est donnée par

$$[\psi]([\mathbf{x}]) = \psi(\mathbf{x}_0) + [\mathbf{J}_\psi]([\mathbf{x}]) \cdot ([\mathbf{x}] - [\mathbf{x}_0]), \quad (\text{II.52})$$

où $[\mathbf{J}_\psi]([\mathbf{x}])$ est une fonction d'inclusion du Jacobien de ψ et \mathbf{x}_0 est le centre du pavé $[\mathbf{x}]$. Le contracteur \mathcal{C}_K consiste alors à réduire le domaine $[\mathbf{x}]$ comme suit,

$$[\mathbf{x}] = [\mathbf{x}] \cap [\psi]([\mathbf{x}]) = [\mathbf{x}] \cap (\psi(\mathbf{x}_0) + [\mathbf{J}_\psi]([\mathbf{x}]) \cdot ([\mathbf{x}] - \mathbf{x}_0)). \quad (\text{II.53})$$

Si l'on remplace $\psi(\mathbf{x})$ par $\mathbf{x} - \mathbf{M} \cdot \mathbf{f}(\mathbf{x})$, on obtient

$$[\mathbf{x}] = [\mathbf{x}] \cap (\mathbf{x}_0 - \mathbf{M} \cdot \mathbf{f}(\mathbf{x}_0) + (\mathbf{I} - \mathbf{M} \cdot [\mathbf{J}_f]([\mathbf{x}])) \cdot ([\mathbf{x}] - \mathbf{x}_0)), \quad (\text{II.54})$$

avec \mathbf{I} la matrice identité et $[\mathbf{J}_f]$ une matrice d'inclusion du Jacobien de \mathbf{f} . En général, la matrice \mathbf{M} est prise égale à l'inverse du Jacobien de \mathbf{f} en \mathbf{x}_0 , $\mathbf{M} = ([\mathbf{J}_f](\mathbf{x}_0))^{-1}$. Le pseudo-code de \mathcal{C}_K est illustré dans l'algorithme 3.

Algorithme 3 : Pseudo-code du contracteur basé sur la méthode de Krawczyk.

Entrées : $[\mathbf{f}], [\mathbf{x}]$;

Sorties : $[\mathbf{x}]$;

$\mathbf{x}_0 = c([\mathbf{x}])$;

$\mathbf{M} = ([\mathbf{J}_f](\mathbf{x}_0))^{-1}$;

$[\mathbf{J}_\psi]([\mathbf{x}]) = \mathbf{I} - \mathbf{M} \cdot [\mathbf{J}_f]([\mathbf{x}])$;

$[\mathbf{x}] = [\mathbf{x}] \cap (\mathbf{x}_0 - \mathbf{M} \cdot \mathbf{f}(\mathbf{x}_0) + [\mathbf{J}_\psi]([\mathbf{x}]) \cdot ([\mathbf{x}] - \mathbf{x}_0))$;

II.5.5 Contracteur basé sur la propagation *forward-backward*

Le contracteur *forward-backward* ou propagation et rétro-propagation, noté $\mathcal{C}_{\uparrow\downarrow}$, est basé sur la propagation de contraintes [Waltz, 1975; Cleary, 1987; Davis, 1987; Benhamou *et al.*, 1999]. Ce contracteur réduit tous les domaines du CSP $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$ en itérant toutes les contraintes une par une sans aucun ordre a priori. La dimension de \mathbf{f} n'est pas nécessairement égale à la dimension de \mathbf{x} . Chaque contrainte élémentaire de \mathbf{f} donne lieu à différents sous-solveurs finis. La contraction de \mathcal{H} consiste alors à intersecter le domaine de chaque variable avec le résultat des sous-solveurs lui correspondant.

Exemple 14. Soient la contrainte

$$x_1 + x_2 - x_3 = 0,$$

et le pavé donné par

$$[\mathbf{x}] = [x_1] \times [x_2] \times [x_3] = [1, 5] \times [2, 6] \times [0, 4].$$

Cette contrainte induit trois sous-solveurs finis,

$$x_1 = \phi_1(x_2, x_3) = x_3 - x_2,$$

$$x_2 = \phi_2(x_1, x_3) = x_3 - x_1,$$

$$x_3 = \phi_3(x_1, x_2) = x_1 + x_2.$$

La contraction des domaines est effectuée comme suit,

$$[x_1] = [x_1] \cap [\phi_1]([x_2], [x_3]) = [1, 5] \cap [-6, 2] = [1, 2],$$

$$[x_2] = [x_2] \cap [\phi_2]([x_1], [x_3]) = [2, 6] \cap [-5, 3] = [2, 3],$$

$$[x_3] = [x_3] \cap [\phi_3]([x_1], [x_2]) = [0, 4] \cap [3, 11] = [3, 4].$$

Le nouveau domaine s'écrit alors $[\mathbf{x}] = [1, 2] \times [2, 3] \times [3, 4]$.

Soient $f_j(x_1, \dots, x_n)$, $j = 1, \dots, n_f$, les contraintes du CSP impliquant différents opérateurs et fonctions élémentaires tels que $+$, $-$, $*$, $/$, \sin , \cos , etc. Il est alors possible de les décomposer en des **contraintes primitives** [Lhomme, 1993]. Celles-ci sont des contraintes ne contenant qu'un seul opérateur ($+$, $-$, $*$, $/$) ou qu'une seule fonction

II. THÉORIE DES INTERVALLES

élémentaire (sin, cos, exp,...). Par exemple, la contrainte $x_1 \cdot \sin(x_2) + x_3 = 0$ peut être décomposée en plusieurs contraintes primitives,

$$\begin{aligned} a_1 &= \sin(x_2), \\ a_2 &= x_1 \cdot a_1, \\ a_2 + x_3 &= 0, \end{aligned}$$

Les domaines associés aux variables intermédiaires (ici a_1 et a_2) sont pris égaux à \mathbb{R} . Le contracteur $\mathcal{C}_{\downarrow\uparrow}$ itère toutes les contraintes primitives du CSP, dans un ordre quelconque, afin de contracter les domaines, jusqu'à ce qu'il n'y ait plus de contraction. Ce contracteur est connu aussi sous le nom de **contracteur de Waltz** [Waltz, 1972, 1975]. Il est particulièrement avantageux pour :

- son indépendance par rapport à la linéarité ou la non linéarité des contraintes,
- son indépendance par rapport à la taille des domaines initiaux,
- son temps d'exécution raisonnable,
- et sa facilité de mise en œuvre.

Exemple 15. *Considérons le CSP \mathcal{H} donné par*

$$\left(\begin{array}{l} y_1 = x_1 - 3 \cdot x_2 \in [-2, 1] \\ y_2 = x_1 + x_2 \in [-1, 1] \\ \mathbf{x} = [x_1] \times [x_2] = [-10, 10] \times [-10, 10] \end{array} \right)$$

L'application du contracteur de Waltz consiste à itérer l'ensemble des contraintes primitives associées à \mathcal{H} . Une itération engendre les contractions suivantes,

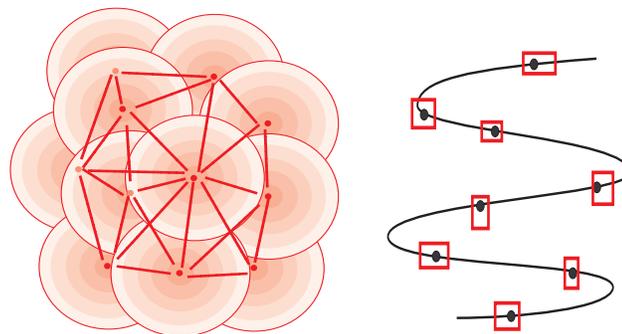
$$\begin{aligned} [y_1] &= [y_1] \cap ([x_1] - 3 \cdot [x_2]) = [-2, 1] \cap [-40, 40] = [-2, 1], \\ [y_2] &= [y_2] \cap ([x_1] + [x_2]) = [-1, 1] \cap [-20, 20] = [-1, 1], \\ [x_1] &= [x_1] \cap ([y_1] + 3 \cdot [x_2]) = [-10, 10] \cap [-31, 32] = [-10, 10], \\ [x_2] &= [x_2] \cap \frac{1}{3} \cdot ([x_1] - [y_1]) = [-10, 10] \cap [-\frac{11}{3}, 4] = [-\frac{11}{3}, 4], \\ [x_1] &= [x_1] \cap ([y_2] - [x_2]) = [-10, 10] \cap [-5, \frac{14}{3}] = [-5, \frac{14}{3}], \\ [x_2] &= [x_2] \cap ([y_2] - [x_1]) = [-\frac{11}{3}, 4] \cap [-\frac{17}{3}, 6] = [-\frac{11}{3}, 4]. \end{aligned}$$

II.6 Conclusion

Dans ce chapitre, nous avons présenté brièvement les outils principalement utilisés dans ce travail. En effet, nous avons décrit les intervalles réels et multidimensionnels, ainsi que les opérateurs arithmétiques et ensemblistes appliqués aux intervalles. Nous nous sommes intéressés par la suite aux méthodes de résolution d'un système d'équations. La première méthode définit la solution sous forme de sous-pavages. Elle consiste à découper un domaine initial de recherche à l'aide de l'algorithme SIVIA. Le résultat est un ensemble de pavés dont l'union couvre l'ensemble solution du problème. Cette technique est précise mais elle a une complexité exponentielle. La seconde méthode, moins précise, traite le problème comme un problème de satisfaction de contraintes. Ayant un domaine initial, cette approche consiste à le réduire en utilisant un contracteur. Le

résultat est alors un pavé unique contenant l'ensemble des solutions du problème. Cette méthode, à précision non contrôlable, est sensiblement moins complexe que SIVIA au niveau du temps de calcul et de la mémoire mise en jeu.

III Auto-localisation par connectivités



Sommaire

- III.1 Introduction
- III.2 Problématique
- III.3 Théorie des intervalles
- III.4 Auto-localisation par connectivités vis-à-vis des ancrés
- III.5 Auto-localisation *back-propagated*
- III.6 Auto-localisation *multi-hop*
- III.7 Auto-localisation basée sur les ancrés et les non-ancrés
- III.8 Simulations
- III.9 Conclusion

Dans ce chapitre, nous abordons le problème d'auto-localisation basée sur les mesures de connectivité. Les capteurs mobiles ont ainsi une mobilité passive ou non-contrôlée. Les méthodes proposées distinguent deux types de capteurs : les ancrés, munies de GPS et les non-ancrés, ne connaissant pas leurs positions. Ces derniers nécessiteront alors des algorithmes d'auto-localisation pour pouvoir estimer leurs positions. Dans ce travail, la résolution du problème se fait dans le cadre de la théorie des intervalles. Ainsi, les positions estimées sont des pavés bidimensionnels, les capteurs étant supposés bouger dans le même plan. Nous proposons en premier lieu une méthode basée uniquement sur les ancrés situées à proximité du nœud. Nous présentons par la suite une approche dite *back-propagated*, permettant de mettre à jour les estimés précédents. Nous effectuons également une localisation *multi-hop*, tenant compte des ancrés plus éloignés. Nous proposons enfin une méthode d'auto-localisation basée aussi bien sur les ancrés que sur les nœuds voisins. Ces deux dernières méthodes sont privilégiées dans les réseaux de capteurs à faibles densités d'ancrés.

III.1 Introduction

Les réseaux de capteurs mobiles forment de nos jours un domaine pluridisciplinaire très actif. Cette technologie a suscité un intérêt important au cours de la dernière décennie. L'attractivité de ce domaine réside à la fois dans ses aspects prometteurs théoriques et applicatifs [Akyildiz *et al.*, 2002b; Labiod, 2006; Shorey, 2006]. Dans les réseaux de capteurs mobiles, où la mobilité est non-contrôlée, l'auto-localisation des capteurs représente une question fondamentale. En effet, en l'absence d'information sur la position des capteurs, les données récoltées s'avèrent d'une utilité limitée dans la plupart des applications. La géolocalisation par GPS (pour *Global Positioning System*) permet de résoudre ce problème de localisation des capteurs [Hofmann-Wellenhof *et al.*, 2004; El-Rabbany, 2006]. Néanmoins, le fait d'équiper tous les capteurs par des récepteurs GPS se révèle irréalisable en pratique, à cause du coût excessif et de la grande consommation énergétique de tels équipements.

Une alternative respectant les contraintes énergétiques des réseaux de capteurs a été proposée. Elle consiste à équiper un faible nombre de capteurs par des récepteurs GPS. Ces capteurs, connaissant leurs positions, sont appelés **ancres**. Les autres capteurs, appelés **nœuds** ou **non-ancres**, communiquent avec les ancres afin d'estimer leurs propres positions. Les méthodes présentées dans ce chapitre sont des approches distribuées, le traitement des données et la prise de décision sont alors locaux sur chaque nœud. À l'aide de ces méthodes, les réseaux de capteurs sont robustes face aux pannes et aux attaques extérieures avec une consommation d'énergie réduite. Dans le cadre des méthodes distribuées, des algorithmes itérant des méthodes de localisation statique ont été proposés [Zhang *et al.*, 1998; Bulusu *et al.*, 2000; Capkun *et al.*, 2001; Savvides *et al.*, 2001; Moore *et al.*, 2004; Vivekanandan et Wong, 2007]. L'inconvénient de ces méthodes est qu'elles ne tiennent pas compte de la mobilité des capteurs. D'autres méthodes intégrant un modèle de mobilité existent [Hu et Evans, 2004; Baggio et Langendoen, 2006; Yi *et al.*, 2007]. Basées sur l'approche séquentielle de Monte-Carlo [Doucet *et al.*, 2001], ces méthodes génèrent des particules afin de définir la zone solution.

Dans ce chapitre, nous proposons des méthodes distribuées d'auto-localisation. La résolution des problèmes posés se situe dans le cadre de la théorie des intervalles. Les positions estimées sont des pavés bidimensionnels, les capteurs étant supposés mobiles dans le même plan. Dans la suite, nous définissons la problématique d'auto-localisation d'une manière générale. Nous proposons ensuite une méthode basée uniquement sur les observations des ancres situées à proximité du nœud. Nous présentons par la suite une approche dite *back-propagated*, permettant de mettre à jour les estimés précédents. Nous effectuons également une localisation *multi-hop*, tenant compte des ancres plus éloignées. Nous proposons enfin une méthode d'auto-localisation basée sur les observations de tous les capteurs voisins au nœud considéré, y compris les non-ancres.

III.2 Problématique

Nous distinguons deux types de capteurs : les ancrs et les non-ancres ou nœuds. Les ancrs ont des positions supposées connues, notées $\mathbf{a}_i(t) = (a_{i,1}(t), a_{i,2}(t))$, $i \in \{1, \dots, N_a\}$. Les positions des non-ancres sont notées $\mathbf{x}_j(t) = (x_{j,1}(t), x_{j,2}(t))$, $j \in \{1, \dots, N_x\}$, celles-ci étant des variables inconnues. Le problème revient alors à estimer les coordonnées $\mathbf{x}_j(t)$ en fonction de mesures effectuées vis-à-vis des ancrs. Outre les observations, les méthodes proposées profitent de la mobilité des capteurs pour raffiner le résultat. Nous utilisons ainsi un modèle de mobilité permettant de propager les positions antérieures. Dans cette section, nous présentons le modèle de mobilité commun à toutes les méthodes. Nous exposerons ensuite un modèle d'observation généralisé, que nous explicitons dans la suite selon la méthode proposée.

III.2.1 Modèle de mobilité

Dans ce travail, nous proposons un modèle général pouvant convenir à un grand nombre d'applications. Ce modèle a été inspiré du modèle de marche aléatoire ou Random Walk introduit dans le paragraphe I.4.3. Selon le modèle proposé, seule la vitesse maximale du j -ème nœud considéré est supposée connue. Elle est notée $v_{max,j}$. Le nœud est alors capable de bouger dans toutes les directions avec une vitesse quelconque limitée à $v_{max,j}$. En d'autres termes, si l'on suppose connaître la position antérieure $\mathbf{x}_j(t-1)$ du nœud j , la nouvelle position appartiendra au disque ayant comme centre $\mathbf{x}_j(t-1)$. Le modèle de mobilité s'écrit alors comme suit,

$$\begin{cases} x_{j,1}(t) = x_{j,1}(t-1) + \Delta t \cdot v_j \cdot \cos(\theta_j), \\ x_{j,2}(t) = x_{j,2}(t-1) + \Delta t \cdot v_j \cdot \sin(\theta_j), \end{cases} \quad (\text{III.1})$$

où θ_j et v_j sont des variables inconnues, $\theta_j \in [0, 2\pi]$, $v_j \in [0, v_{max,j}]$ et Δt est la période de localisation. Cette définition est équivalente à l'équation du disque donné par

$$(x_{j,1}(t) - x_{j,1}(t-1))^2 + (x_{j,2}(t) - x_{j,2}(t-1))^2 = (\Delta t \cdot v_j)^2, \quad (\text{III.2})$$

où v_j est une variable positive inconnue mais inférieure à $v_{max,j}$. Toute information supplémentaire concernant la mobilité des nœuds peut être utilisée pour raffiner le modèle proposé.

III.2.2 Modèle d'observation généralisé

En plus du modèle de mobilité, chaque nœud récolte des informations du milieu extérieur pour définir sa position. Soit \mathbf{y}_j le vecteur d'informations reçues. Alors \mathbf{y}_j peut s'écrire sous la forme d'une fonction des coordonnées des ancrs et des non-ancres du réseau. Le modèle d'observation général correspondant à un nœud j s'écrit comme suit,

$$\mathbf{y}_j(t) = \mathbf{f}(\mathbf{x}_k(t), k \in \{1, \dots, j, \dots, N_x\}, \mathbf{a}_i(t), i \in \{1, \dots, N_a\}, \mathbf{p}), \quad (\text{III.3})$$

où \mathbf{p} est un vecteur de paramètres dépendant du modèle considéré. Contrairement au modèle de mobilité, le modèle d'observation donné par le couple (\mathbf{f}, \mathbf{p}) sera explicité dans

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

la suite au fur et à mesure selon l'algorithme proposé. Le problème d'auto-localisation sera défini alors par le système d'équations correspondant aux deux modèles,

$$\begin{cases} (x_{j,1}(t) - x_{j,1}(t-1))^2 + (x_{j,2}(t) - x_{j,2}(t-1))^2 = (\Delta t \cdot v_j)^2, \\ \mathbf{y}_j(t) = \mathbf{f}(\mathbf{x}_k(t)), k \in \{1, \dots, j, \dots, N_x\}, \mathbf{a}_i(t), i \in \{1, \dots, N_a\}, \mathbf{p}), \\ v_j \in [0, v_{max,j}]. \end{cases} \quad (\text{III.4})$$

Résoudre le problème d'auto-localisation se résume ainsi à la résolution de ce système.

III.3 Théorie des intervalles

La résolution du problème d'auto-localisation se situe dans le cadre de l'analyse par intervalle. Pour cette raison, nous rappelons brièvement quelques notions de la théorie des intervalles. Cette approche représente un outil mathématique rigoureux, manipulant des intervalles au lieu des nombres réels. Un intervalle réel $[x]$ est un ensemble connexe de nombres réels limités entre deux bornes inférieure \underline{x} et supérieure \bar{x} ,

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}. \quad (\text{III.5})$$

Le centre d'un intervalle est donné par $c([x]) = \frac{\underline{x} + \bar{x}}{2}$ et sa taille par $w([x]) = \bar{x} - \underline{x}$. Un pavé, nommé aussi boîte, est un intervalle multidimensionnel résultant du produit cartésien d'intervalles réels,

$$[\mathbf{x}] = [x_1] \times \dots \times [x_n] = [\underline{x}_1, \bar{x}_1] \times \dots \times [\underline{x}_n, \bar{x}_n]. \quad (\text{III.6})$$

Le centre du pavé est donné par le vecteur des centres des intervalles réels composants le pavé. Sa taille, par contre, est donnée par la taille maximale des différentes composantes.

Les intervalles ont une double nature, celle des nombres réels et celles des ensembles. La théorie des intervalles profitent de cette dualité pour étendre les opérations ensemblistes et arithmétiques aux intervalles. Nous en citons l'intersection (\cap), la réunion d'intervalles (\sqcup), la différence ensembliste (\setminus), l'addition (+), la soustraction (-), la multiplication (*), la division (/), etc. Outre les opérations, toutes les fonctions peuvent être appliquées aux intervalles. Si \mathbf{f} est une fonction de \mathbb{R}^n dans \mathbb{R}^m , l'association de \mathbf{f} à un pavé $[\mathbf{x}]$ engendre un ensemble $\mathbf{f}([\mathbf{x}])$ pas nécessairement connexe. En conséquence, les fonctions d'inclusion ont été élaborées pour remédier à ce problème. Celles-ci, notées $[\mathbf{f}]([\mathbf{x}])$, produisent des pavés englobant l'ensemble $\mathbf{f}([\mathbf{x}])$. Une fonction possède alors une infinité de fonctions d'inclusion, la minimale $[\mathbf{f}]^*([\mathbf{x}])$ étant celle qui engendre le plus petit pavé couvrant $\mathbf{f}([\mathbf{x}])$.

Le contexte ensembliste fournit une alternative à l'approximation ponctuelle. Soient \mathbf{x} une variable de \mathbb{R}^n et \mathbf{f} une fonction définie de \mathbb{R}^n dans \mathbb{R}^m . Alors $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ définit un ensemble de contraintes. Si $[\mathbf{x}]$ est le domaine initial de \mathbf{x} alors $\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])$ est dit problème de satisfaction de contraintes (CSP). Dans le cadre des intervalles, la résolution du problème se fait soit par contraction à l'aide des contracteurs (voir la section II.5), soit par découpage à l'aide de l'algorithme d'inversion ensembliste SIVIA (voir la section II.4.3). Dans ce chapitre, nous utilisons les contracteurs, en particulier

l'algorithme de Waltz pour son temps de calcul raisonnable, sa simplicité et sa facilité de mise en œuvre. Le contracteur de Waltz, du type *forward-backward*, définit des sous-solveurs finis, éventuellement primitifs, qu'il itère dans un ordre quelconque jusqu'à ce qu'aucune contraction n'est plus possible. Nous rappelons qu'un sous-solveur fini est une fonction, issue d'une contrainte, exprimant l'une des variables en fonction des autres. Le résultat final est un pavé inclus dans le domaine initial $[\mathbf{x}]$.

III.4 Auto-localisation par connectivités vis-à-vis des ancrs

Dans cette section, nous proposons une méthode d'auto-localisation basée sur les relations de détection réciproque. Ces dernières sont souvent qualifiées de mesures de connectivités ou de proximités inter-capteurs. Cette méthode sera notée ALCA (pour Auto-Localisation par Connectivités vis-à-vis des Ancres). Le nœud mobile détecte alors les ancrs situées à proximité de lui afin de définir ses mesures d'observation. Dans la suite, nous commençons par introduire le modèle d'observation basé sur les connectivités. Nous exposons ensuite le problème d'auto-localisation obtenu dans ce cas. Nous présentons enfin l'algorithme de résolution du problème proposé.

III.4.1 Mesures de connectivités

Les mesures de connectivités sont généralement définies sur la base des connexions radio établies entre capteurs voisins. Elles sont généralement liées d'une manière non-linéaire à la distance séparant les capteurs. Selon le modèle d'Okumura-Hata [Medeisis et Kajackas, 2000; Nadir *et al.*, 2008], la puissance $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)$ du signal émis par une ancre $\mathbf{a}_i(t)$ et reçu par le nœud $\mathbf{x}_j(t)$ décroît d'une façon monotone avec l'augmentation de la distance inter-capteurs parcourue,

$$\rho_{\mathbf{a}_i, \mathbf{x}_j}(t) = \rho_0 - 10 \cdot n_P \cdot \log_{10} \frac{d(\mathbf{a}_i(t), \mathbf{x}_j(t))}{d_0} + \varepsilon_{i,j}(t), \quad (\text{III.7})$$

où $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)$ est en *dBm*, ρ_0 en *dBm* est la puissance mesurée à une certaine distance parcourue d_0 , n_P est le paramètre de perte correspondant au canal, $d(\mathbf{a}_i(t), \mathbf{x}_j(t))$ est la distance entre $\mathbf{a}_i(t)$ et $\mathbf{x}_j(t)$ et $\varepsilon_{i,j}(t)$ est le bruit de mesure.

En général, la puissance du signal ne suit pas exactement le modèle proposé à cause du bruit de mesure et des erreurs commises dans l'estimation des paramètres du modèle (n_P , ρ_0 et d_0). D'où l'avantage des mesures de connectivités où seules les ancrs situées dans la portée de communication du nœud sont prises en compte. Selon cette approche, la puissance reçue n'est pas utilisée pour estimer la vraie valeur de la distance inter-capteur, mais par contre elle est employée dans une sorte de comparaison pour dire si l'ancre est dans la portée du nœud ou non. En effet, un nœud j recevant un signal d'une ancre i , $i \in I_j^*(t) \subseteq \{1, \dots, N_a\}$, compare la puissance du signal $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)$ à un seuil ρ_r fonction de la portée de communication r ,

$$\begin{cases} \text{si } \rho_{\mathbf{a}_i, \mathbf{x}_j}(t) \geq \rho_r, & \text{alors } d(\mathbf{a}_i(t), \mathbf{x}_j(t)) \leq r \Leftrightarrow \text{l'ancre } i \text{ est détectée,} \\ \text{si } \rho_{\mathbf{a}_i, \mathbf{x}_j}(t) < \rho_r, & \text{alors } d(\mathbf{a}_i(t), \mathbf{x}_j(t)) > r \Leftrightarrow \text{l'ancre } i \text{ n'est pas détectée.} \end{cases} \quad (\text{III.8})$$

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

Soit $I_j(t) \subseteq I_j^*(t)$ l'ensemble des indices des ancrs détectées par le nœud j à l'instant t . Ces ancrs sont toutes à une distance du nœud j inférieure à la portée r . Chaque nœud j engendre ainsi un vecteur binaire $\mathbf{y}_j(t) = (y_{j,i}(t))_{i \in \{1, \dots, N_a\}}$ de mesures selon les ancrs qu'il détecte,

$$y_{j,i}(t) = \begin{cases} 1 & \text{si } i \in I_j(t), \\ 0 & \text{sinon.} \end{cases} \quad (\text{III.9})$$

Ces mesures seront utilisées dans l'estimation de sa propre position. Basée sur les connectivités, cette technique réduit énormément le risque d'erreur dans le processus d'estimation. On supposera dans la suite que les mesures générées sont correctes, un 1 dans le vecteur $\mathbf{y}_j(t)$ correspondant à une ancre réellement détectée.

III.4.2 Définition du problème

Le problème d'auto-localisation est défini à la fois par le modèle de mobilité donné dans le paragraphe III.2.1 et le modèle d'observation basé sur les connectivités. Le nœud j , ayant un vecteur binaire d'observation $\mathbf{y}_j(t)$, reconnaît les ancrs détectés et en définit un ensemble de contraintes comme suit,

$$\begin{aligned} y_{j,i}(t) = 1 &\Leftrightarrow d(\mathbf{a}_i(t), \mathbf{x}_j(t)) \leq r, \\ &\Leftrightarrow (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq r^2. \end{aligned} \quad (\text{III.10})$$

Si $I_j(t)$ est l'ensemble des indices des ancrs détectés ($y_{j,i}(t) = 1$), alors le problème d'auto-localisation est formulé comme suit,

$$\left\{ \begin{array}{l} (x_{j,1}(t) - x_{j,1}(t-1))^2 + (x_{j,2}(t) - x_{j,2}(t-1))^2 = (\Delta t \cdot v_j)^2, \\ \text{avec } \left| \begin{array}{l} (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq r^2, \quad i \in I_j(t), \\ v_j \in [0, v_{\max,j}]. \end{array} \right. \end{array} \right. \quad (\text{III.11})$$

La solution correspondra à la zone d'intersection des différentes régions de mobilité et d'observation.

III.4.3 Algorithme

La résolution du problème se fait à l'aide des intervalles. Les variables seront remplacées par les pavés auxquels elles appartiennent et les fonctions par leurs fonctions d'inclusion naturelles. Les coordonnées des nœuds $\mathbf{x}_j(t)$, en particulier, seront des pavés $[\mathbf{x}_j](t) = [x_{j,1}](t) \times [x_{j,2}](t)$ couvrant les positions réelles. Nous définissons alors le problème tel un problème de satisfaction de contraintes (CSP) que nous résolvons à l'aide du contracteur de Waltz. Pour cette raison, nous aurons besoin d'un domaine initial, que nous contracterons par la suite. La méthode consiste alors en deux étapes : la propagation, donnant lieu au domaine initial, et la correction, où le domaine initial sera contracté.

1. *Etape de propagation*

Dans cette étape, il s'agit de propager le pavé estimé à l'instant précédent, dit pavé antérieur. La propagation est réalisée à l'aide du modèle de mobilité. Dans

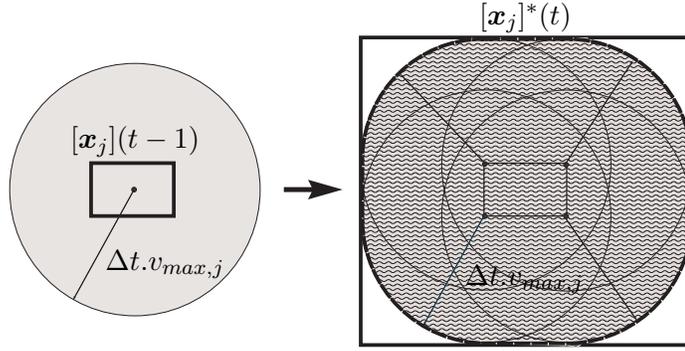


FIG. III.1: Etape de propagation.

le langage ensembliste, où chaque variable est remplacée par le domaine auquel elle appartient, l'équation (III.1) s'écrit comme suit,

$$\begin{cases} [x_{j,1}](t) &= [x_{j,1}](t-1) + [0, \Delta t.v_{max,j}].[\cos]([0, 2.\pi]) \\ &= [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}], \\ [x_{j,2}](t) &= [x_{j,2}](t-1) + [0, \Delta t.v_{max,j}].[\sin]([0, 2.\pi]) \\ &= [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]. \end{cases} \quad (\text{III.12})$$

Il faut noter que nous utilisons la formulation (III.1) du modèle de mobilité dans cette étape puisque nous ne connaissons pas encore les valeurs initiales de $[x_{j,1}](t)$ et de $[x_{j,2}](t)$ et que ces coordonnées ne sont pas des fonctions directes l'une de l'autre dans cette formulation. L'utilisation de la formulation (III.2) avec un domaine initial de $[\mathbf{x}_j(t)]$ égal à \mathbb{R}^2 aboutira au même résultat (voir la preuve III.1). La propagation du pavé antérieur est illustrée dans FIG. III.1. La figure montre que la propagation d'un pavé à l'aide d'une équation de disque engendre un rectangle aux sommets arrondis. En effet, il suffit d'appliquer l'équation du disque à tous les points du pavé. L'union des disques résultants donnera le rectangle concerné. La fonction d'inclusion du modèle de mobilité génère un pavé englobant le rectangle. Notons $[\mathbf{x}_j]^*(t) = [x_{j,1}]^*(t) \times [x_{j,2}]^*(t)$ le domaine initial de $\mathbf{x}_j(t)$. Ainsi,

$$\begin{cases} [x_{j,1}]^*(t) &= [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}], \\ [x_{j,2}]^*(t) &= [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]. \end{cases} \quad (\text{III.13})$$

Ce domaine sera contracté à l'aide de l'algorithme de Waltz dans l'étape suivante.

Preuve III.1. Si l'on considère la formulation de l'équation (III.2), celle-ci s'écrit à l'aide des intervalles sous la forme suivante,

$$([x_{j,1}](t) - [x_{j,1}](t-1))^2 + ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, (\Delta t.v_{max,j})^2],$$

avec $[x_{j,1}](t) \subseteq \mathbb{R}$ et $[x_{j,2}](t) \subseteq \mathbb{R}$.

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

Vu que $([x_{j,1}](t) - [x_{j,1}](t-1))^2$ et $([x_{j,2}](t) - [x_{j,2}](t-1))^2$ sont des carrés d'intervalles et donc positifs, alors chacun d'eux doit être inférieur à la somme maximale $(\Delta t.v_{max,j})^2$,

$$\begin{cases} ([x_{j,1}](t) - [x_{j,1}](t-1))^2 \subseteq [0, (\Delta t.v_{max,j})^2], \\ ([x_{j,2}](t) - [x_{j,2}](t-1))^2 \subseteq [0, (\Delta t.v_{max,j})^2], \end{cases}$$

$$\Downarrow$$

$$\begin{cases} [x_{j,1}](t) \subseteq [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}], \\ [x_{j,2}](t) \subseteq [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]. \end{cases}$$

Ainsi $([x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}])$ et $([x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}])$ peuvent former des domaines initiaux de $x_{j,1}(t)$ et $x_{j,2}(t)$ respectivement.

Afin de sortir un sous-solveur fini de $x_{j,1}(t)$ (resp. $x_{j,2}(t)$), il suffit de remplacer $[x_{j,2}](t)$ (resp. $[x_{j,1}](t)$) par son domaine initial dans l'équation de mobilité. Nous obtenons ainsi

$$\begin{cases} ([x_{j,1}](t) - [x_{j,1}](t-1))^2 = [0, +\infty[\cap([0, (\Delta t.v_{max,j})^2] - ([x_{j,2}](t) - [x_{j,2}](t-1))^2)], \\ ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, +\infty[\cap([0, (\Delta t.v_{max,j})^2] - ([x_{j,1}](t) - [x_{j,1}](t-1))^2)], \end{cases}$$

$$\Downarrow$$

$$\begin{cases} ([x_{j,1}](t) - [x_{j,1}](t-1))^2 = [0, +\infty[\cap([0, (\Delta t.v_{max,j})^2] - [0, (\Delta t.v_{max,j})^2]), \\ ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, +\infty[\cap([0, (\Delta t.v_{max,j})^2] - [0, (\Delta t.v_{max,j})^2]), \end{cases}$$

$$\Downarrow$$

$$\begin{cases} ([x_{j,1}](t) - [x_{j,1}](t-1))^2 = [0, +\infty[\cap([-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]^2)], \\ ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, +\infty[\cap([-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]^2)], \end{cases}$$

$$\Downarrow$$

$$\begin{cases} ([x_{j,1}](t) - [x_{j,1}](t-1))^2 = [0, (\Delta t.v_{max,j})^2], \\ ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, (\Delta t.v_{max,j})^2], \end{cases}$$

$$\Downarrow$$

$$\begin{cases} [x_{j,1}](t) = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}], \\ [x_{j,2}](t) = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]. \end{cases}$$

En d'autres termes, le domaine initial $[x_j](t)$ ne sera pas contracté. Ceci est équivalent au résultat obtenu en partant de la formulation (III.1) du modèle de mobilité. \square

2. Etape de correction

Une fois le domaine initial $[x_j]^*(t)$ est défini, les contraintes d'observation et de mobilité seront itérées dans l'algorithme de Waltz pour le contracter. Ces équations de mobilité et d'observation associées à un nœud j , se reformulent, dans le contexte ensembliste, respectivement par

$$([x_{j,1}](t) - [x_{j,1}](t-1))^2 + ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, (\Delta t.v_{max,j})^2], \quad (\text{III.14})$$

$$\text{et } ([x_{j,1}](t) - a_{i,1}(t))^2 + ([x_{j,2}](t) - a_{i,2}(t))^2 = [0, r^2], \quad i \in I_j(t). \quad (\text{III.15})$$

Les sous-solveurs finis que nous pouvons extraire de ces équations sont définis comme suit (voir la preuve III.2),

$$[\phi_{j,1}] = [x_{j,1}](t-1) + [-\overline{bm}_{j,1}(t), +\overline{bm}_{j,1}(t)], \quad (\text{III.16})$$

$$[\phi_{j,2}] = [x_{j,2}](t-1) + [-\overline{bm}_{j,2}(t), +\overline{bm}_{j,2}(t)], \quad (\text{III.17})$$

$$[\psi_{j,i,1}] = a_{i,1}(t) + [-\overline{bo}_{j,i,1}(t), +\overline{bo}_{j,i,1}(t)], \quad (\text{III.18})$$

$$[\psi_{j,i,2}] = a_{i,2}(t) + [-\overline{bo}_{j,i,2}(t), +\overline{bo}_{j,i,2}(t)], \quad (\text{III.19})$$

avec

$$\begin{cases} \overline{bm}_{j,1}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf(([x_{j,2}](t) - [x_{j,2}](t-1))^2)}, \\ \overline{bm}_{j,2}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf(([x_{j,1}](t) - [x_{j,1}](t-1))^2)}, \\ \overline{bo}_{j,i,1}(t) = \sqrt{r^2 - \inf(([x_{j,2}](t) - a_{i,2}(t))^2)}, \\ \overline{bo}_{j,i,2}(t) = \sqrt{r^2 - \inf(([x_{j,1}](t) - a_{i,1}(t))^2)}. \end{cases}$$

La contraction du problème s'effectue par intersection des sous-solveurs finis avec les différents domaines comme suit,

$$\begin{cases} [x_{j,1}](t) = [x_{j,1}](t) \cap [\phi_{j,1}], \\ [x_{j,2}](t) = [x_{j,2}](t) \cap [\phi_{j,2}], \\ [x_{j,1}](t) = [x_{j,1}](t) \cap [\psi_{j,i,1}], \\ [x_{j,2}](t) = [x_{j,2}](t) \cap [\psi_{j,i,2}]. \end{cases} \quad (\text{III.20})$$

Preuve III.2.

$$\begin{aligned} (\text{III.14}) \quad &\Leftrightarrow (\phi_{j,1} - [x_{j,1}](t-1))^2 + ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, (\Delta t.v_{max,j})^2], \\ &\Leftrightarrow ([\phi_{j,1}] - [x_{j,1}](t-1))^2 = [0, +\infty[\cap \dots \\ &\quad \dots([0, (\Delta t.v_{max,j})^2] - ([x_{j,2}](t) - [x_{j,2}](t-1))^2), \\ &\Leftrightarrow ([\phi_{j,1}] - [x_{j,1}](t-1))^2 = [0, (\Delta t.v_{max,j})^2 - \inf(([x_{j,2}](t) - [x_{j,2}](t-1))^2)], \\ &\Leftrightarrow [\phi_{j,1}] = [x_{j,1}](t-1) + [-\overline{bm}_{j,1}(t), +\overline{bm}_{j,1}(t)], \\ &\quad \text{avec } \overline{bm}_{j,1}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf(([x_{j,2}](t) - [x_{j,2}](t-1))^2)}, \\ &\quad \text{et } ([x_{j,2}](t) - [x_{j,2}](t-1))^2 \subseteq [0, (\Delta t.v_{max,j})^2]. \end{aligned}$$

$$\begin{aligned} (\text{III.14}) \quad &\Leftrightarrow ([x_{j,1}](t) - [x_{j,1}](t-1))^2 + (\phi_{j,2} - [x_{j,2}](t-1))^2 = [0, (\Delta t.v_{max,j})^2], \\ &\Leftrightarrow ([\phi_{j,2}] - [x_{j,2}](t-1))^2 = [0, +\infty[\cap \dots \\ &\quad \dots([0, (\Delta t.v_{max,j})^2] - ([x_{j,1}](t) - [x_{j,1}](t-1))^2), \\ &\Leftrightarrow ([\phi_{j,2}] - [x_{j,2}](t-1))^2 = [0, (\Delta t.v_{max,j})^2 - \inf(([x_{j,1}](t) - [x_{j,1}](t-1))^2)], \\ &\Leftrightarrow [\phi_{j,2}] = [x_{j,2}](t-1) + [-\overline{bm}_{j,2}(t), +\overline{bm}_{j,2}(t)], \\ &\quad \text{avec } \overline{bm}_{j,2}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf(([x_{j,1}](t) - [x_{j,1}](t-1))^2)}, \\ &\quad \text{et } ([x_{j,1}](t) - [x_{j,1}](t-1))^2 \subseteq [0, (\Delta t.v_{max,j})^2]. \end{aligned}$$

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

$$\begin{aligned}
\text{(III.15)} \quad &\Leftrightarrow (\psi_{j,i,1} - a_{i,1}(t))^2 + ([x_{j,2}](t) - a_{i,2}(t))^2 = [0, r^2], \\
&\Leftrightarrow ([\psi_{j,i,1}] - a_{i,1}(t))^2 = [0, +\infty[\cap([0, r^2] - ([x_{j,2}](t) - a_{i,2}(t))^2)], \\
&\Leftrightarrow ([\psi_{j,i,1}] - a_{i,1}(t))^2 = [0, r^2 - \inf(([x_{j,2}](t) - a_{i,2}(t))^2)], \\
&\Leftrightarrow [\psi_{j,i,1}] = a_{i,1}(t) + \frac{[-\overline{bo}_{j,i,1}(t), +\overline{bo}_{j,i,1}(t)]}{\sqrt{r^2 - \inf(([x_{j,2}](t) - a_{i,2}(t))^2)}}, \\
&\quad \text{avec } \overline{bo}_{j,i,1}(t) = \sqrt{r^2 - \inf(([x_{j,2}](t) - a_{i,2}(t))^2)}, \\
&\quad \text{et } ([x_{j,2}](t) - a_{i,2}(t))^2 \subseteq [0, r^2]. \\
\text{(III.15)} \quad &\Leftrightarrow ([x_{j,1}](t) - a_{i,1}(t))^2 + (\psi_{j,i,2} - a_{i,2}(t))^2 = [0, r^2], \\
&\Leftrightarrow ([\psi_{j,i,2}] - a_{i,2}(t))^2 = [0, +\infty[\cap([0, r^2] - ([x_{j,1}](t) - a_{i,1}(t))^2)], \\
&\Leftrightarrow ([\psi_{j,i,2}] - a_{i,2}(t))^2 = [0, r^2 - \inf(([x_{j,1}](t) - a_{i,1}(t))^2)], \\
&\Leftrightarrow [\psi_{j,i,2}] = a_{i,2}(t) + \frac{[-\overline{bo}_{j,i,2}(t), +\overline{bo}_{j,i,2}(t)]}{\sqrt{r^2 - \inf(([x_{j,1}](t) - a_{i,1}(t))^2)}}, \\
&\quad \text{avec } \overline{bo}_{j,i,2}(t) = \sqrt{r^2 - \inf(([x_{j,1}](t) - a_{i,1}(t))^2)}, \\
&\quad \text{et } ([x_{j,1}](t) - a_{i,1}(t))^2 \subseteq [0, r^2].
\end{aligned}$$

□

La résolution du problème d'auto-localisation pour un nœud $j \in \{1, \dots, N_x\}$ consiste alors à :

- i) propager le pavé antérieur pour générer un domaine initial $[x_j](t) = [x_j]^*(t)$,
- ii) détecter les ancres du voisinage du nœud définissant $I_j(t)$,
- iii) générer les sous-solveurs finis à partir du modèle de mobilité $\phi_{j,1}, \phi_{j,2}$,
- iv) générer les sous-solveurs finis à partir du modèle d'observation $\psi_{j,i,1}, \psi_{j,i,2}, i \in I_j(t)$,
- v) contracter le domaine initial à l'aide des sous-solveurs.

Les contractions à l'aide des sous-solveurs seront itérées jusqu'à ce que le pavé $[x_j](t)$ ne rétrécisse plus. En d'autres termes, le contracteur de Waltz continue à itérer les contraintes tant que la surface du pavé est réduite. D'autres critères d'arrêt peuvent être intégrés, nous citons par exemple un nombre d'itérations maximal qu'il ne faut pas dépasser ou une certaine mesure de précision sur le taux de réduction du pavé. Il faut noter que les sous-solveurs peuvent être décomposés en des sous-solveurs primitifs, contenant chacun une seule opération (+, -, *, /, ^) ou une seule fonction élémentaire. Ceci exige l'introduction de plusieurs variables auxiliaires dont les domaines seront contractés. Le pseudo-code de la méthode associée à un nœud $j \in \{1, \dots, N_x\}$ à un instant t donné est illustré dans l'algorithme 4. La position ponctuelle estimée sera égale au centre du pavé estimé. Ce code, implémenté sur chaque nœud, sera lancé à chaque pas de temps. Graphiquement, la contrainte de mobilité est représentée par un rectangle aux sommets arrondis et celles d'observation par des disques ayant les ancres détectées comme centres et r comme rayon. Une illustration est donnée dans FIG. III.2. Le pavé résultant sera, au mieux, équivalent à l'intersection du domaine initial, des disques d'observation et du rectangle de mobilité.

Remarque 3. *Le calcul ensembliste basé sur les intervalles permet la prise en compte de l'incertitude au niveau des positions des ancres et de la portée de communication. Dans les cas où les récepteurs GPS des ancres ne fournissent pas des positions précises ou que le champ de communication n'est pas homogène, des erreurs peuvent survenir au*

Algorithme 4 : Pseudo-code de la méthode d'auto-localisation par connectivités vis-à-vis des ancrés.

Entrées : $[x_j](t-1)$, $I_j(t)$, $\mathbf{a}_i(t)$, $i \in I_j(t)$, $v_{max,j}$, Δt , r ;
Sorties : $[x_j](t)$;

$$\begin{cases} [x_{j,1}]^*(t) = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]; \\ [x_{j,2}]^*(t) = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]; \end{cases}$$

$$\begin{aligned} [x_{j,1}](t) &= [x_{j,1}]^*(t), [x_{j,2}](t) = [x_{j,2}]^*(t); \\ A_j &= w([x_{j,1}](t)) * w([x_{j,2}](t)); \\ A_j^{old} &= A_j + 1; \end{aligned}$$

tant que $A_j < A_j^{old}$ **faire**
 $A_j^{old} = A_j$;

pour $i \in I_j(t)$ **faire**

$$\begin{aligned} [x_{j,2}](t) &= [x_{j,2}](t) \cap [a_{i,2}(t) - r, a_{i,2}(t) + r]; \\ \overline{bo}_{j,i,1}(t) &= \sqrt{r^2 - \inf((\overline{[x_{j,2}](t)} - a_{i,2}(t))^2)}; \\ [\psi_{j,i,1}] &= a_{i,1}(t) + [-\overline{bo}_{j,i,1}(t), +\overline{bo}_{j,i,1}(t)]; \\ [x_{j,1}](t) &= [x_{j,1}](t) \cap [\psi_{j,i,1}]; \\ [x_{j,1}](t) &= [x_{j,1}](t) \cap [a_{i,1}(t) - r, a_{i,1}(t) + r]; \\ \overline{bo}_{j,i,2}(t) &= \sqrt{r^2 - \inf((\overline{[x_{j,1}](t)} - a_{i,1}(t))^2)}; \\ [\psi_{j,i,2}] &= a_{i,2}(t) + [-\overline{bo}_{j,i,2}(t), +\overline{bo}_{j,i,2}(t)]; \\ [x_{j,2}](t) &= [x_{j,2}](t) \cap [\psi_{j,i,2}]; \end{aligned}$$

fin

$$\begin{aligned} \overline{bm}_{j,1}(t) &= \sqrt{(\Delta t.v_{max,j})^2 - \inf((\overline{[x_{j,2}](t)} - \overline{[x_{j,2}](t-1)})^2)}; \\ [\phi_{j,1}] &= [x_{j,1}](t-1) + [-\overline{bm}_{j,1}(t), +\overline{bm}_{j,1}(t)]; \\ [x_{j,1}](t) &= [x_{j,1}](t) \cap [\phi_{j,1}]; \\ \overline{bm}_{j,2}(t) &= \sqrt{(\Delta t.v_{max,j})^2 - \inf((\overline{[x_{j,1}](t)} - \overline{[x_{j,1}](t-1)})^2)}; \\ [\phi_{j,2}] &= [x_{j,2}](t-1) + [-\overline{bm}_{j,2}(t), +\overline{bm}_{j,2}(t)]; \\ [x_{j,2}](t) &= [x_{j,2}](t) \cap [\phi_{j,2}]; \\ A_j &= w([x_{j,1}](t)) * w([x_{j,2}](t)); \end{aligned}$$

fin

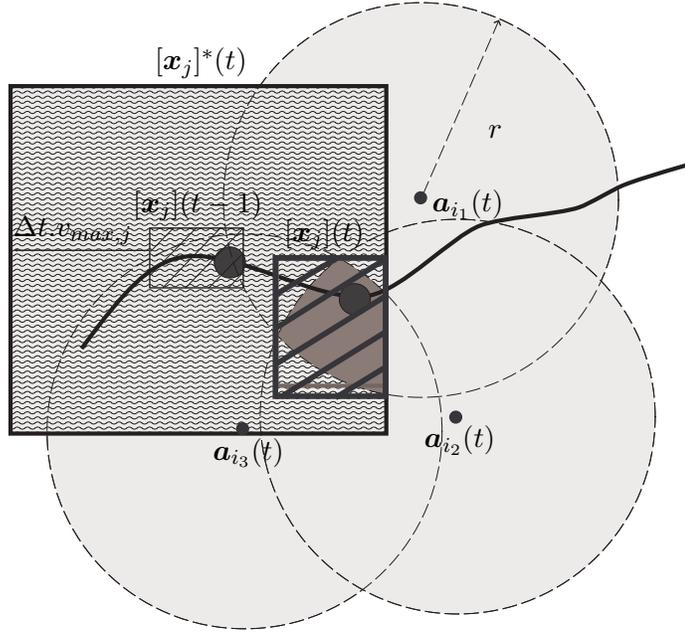


FIG. III.3: Effet de la simplification partielle sur l'algorithme d'auto-localisation.

III.4.4 Versions simplifiées de l'algorithme

Il est possible de simplifier le problème d'auto-localisation en relaxant les contraintes. Les équations de disques pourront être remplacées par des équations de carrés encadrant les disques. Une première simplification, dite simplification partielle, consiste à remplacer l'équation de mobilité par celle d'un carré de côté $2\Delta t.v_{max,j}$ comme suit,

$$\begin{cases} [x_{j,1}](t) = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}], \\ [x_{j,2}](t) = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]. \end{cases} \quad (\text{III.23})$$

Les sous-solveurs de mobilité $[\phi_{j,1}]$ et $[\phi_{j,2}]$ seront alors modifiés comme suit,

$$\begin{cases} [\phi_{j,1}] = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}], \\ [\phi_{j,2}] = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]. \end{cases} \quad (\text{III.24})$$

Dans ce cas, l'étape de propagation de l'algorithme 4 reste inchangée. Toutefois, il n'est plus nécessaire d'intégrer les sous-solveurs de mobilité dans l'étape de correction, vu qu'ils donneront le même pavé que le domaine initial. Le pavé initial sera alors moins contracté. Une illustration est donnée dans FIG. III.3. La figure montre en noir gris le pavé estimé et en gris le pavé issu de l'algorithme sans simplification. Il est clair qu'avec la simplification, le pavé devient légèrement plus grand, produisant ainsi une légère perte au niveau de la précision de l'estimation.

Une seconde simplification, dite simplification totale, consiste à relaxer à la fois les équations d'observation et de mobilité. Le modèle d'observation donnera des carrés

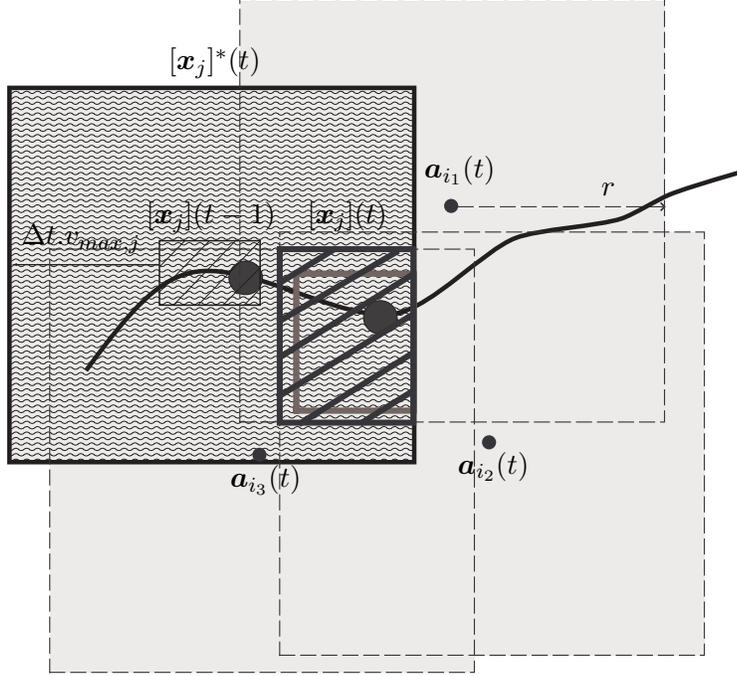


FIG. III.4: Effet de la simplification totale sur l'algorithme d'auto-localisation.

centrés sur les ancrs détectées et ayant $2.r$ comme côté comme suit,

$$\begin{cases} [x_{j,1}](t) = a_{i,1}(t) + [-r, r] \\ [x_{j,2}](t) = a_{i,2}(t) + [-r, r] \end{cases}, i \in I_j(t). \quad (\text{III.25})$$

Les sous-solveurs d'observation $[\psi_{j,i,1}]$ et $[\psi_{j,i,2}]$ seront alors reformulés par

$$\begin{cases} [\psi_{j,i,1}] = a_{i,1}(t) + [-r, r] \\ [\psi_{j,i,2}] = a_{i,2}(t) + [-r, r] \end{cases}, i \in I_j(t). \quad (\text{III.26})$$

Dans ce cas, l'étape de propagation demeure également inchangée. Dans l'étape de correction, par contre, les sous-solveurs d'observation seront remplacés par leurs nouvelles expressions. Ceux de mobilité peuvent être éliminés vu qu'ils n'apporteront pas de changement par rapport au domaine initial. Une illustration est donnée dans FIG. III.4. La figure montre en noir le nouveau pavé obtenu et en gris le pavé issu de la méthode sans simplification. La simplification totale engendre un pavé de plus en plus grand. Le temps de calcul sera réduit, au détriment de la précision de l'estimation.

Considérons le cas de la simplification totale. L'étape de propagation engendre le

domaine initial suivant,

$$\begin{cases} [x_{j,1}]^*(t) &= [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}] \\ &= [\underline{x}_{j,1}(t-1) - \Delta t.v_{max,j}, \bar{x}_{j,1}(t-1) + \Delta t.v_{max,j}], \\ [x_{j,2}]^*(t) &= [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}] \\ &= [\underline{x}_{j,2}(t-1) - \Delta t.v_{max,j}, \bar{x}_{j,2}(t-1) + \Delta t.v_{max,j}]. \end{cases} \quad (\text{III.27})$$

L'étape de correction consiste à itérer les contractions à l'aide des sous-solveurs d'observation donnés par

$$\begin{cases} [\psi_{j,i,1}] = a_{i,1}(t) + [-r, r] = [a_{i,1}(t) - r, a_{i,1}(t) + r] \\ [\psi_{j,i,2}] = a_{i,2}(t) + [-r, r] = [a_{i,2}(t) - r, a_{i,2}(t) + r] \end{cases}, i \in I_j(t). \quad (\text{III.28})$$

Le pavé estimé résulte alors de l'intersection des pavés issus de la propagation et des sous-solveurs. Une itération suffit vu que les deux coordonnées sont devenues indépendantes l'une de l'autre. Le résultat sera ainsi donné par

$$\begin{cases} [x_{j,1}](t) &= [x_{j,1}]^*(t) \cap (\bigcap_{i \in I_j(t)} [\psi_{j,i,1}]) \\ &= [\max\{\underline{x}_{j,1}(t-1) - \Delta t.v_{max,j}, \max_{i \in I_j(t)}(a_{i,1}(t) - r)\}, \\ &\quad \min\{\bar{x}_{j,1}(t-1) + \Delta t.v_{max,j}, \min_{i \in I_j(t)}(a_{i,1}(t) + r)\}], \\ [x_{j,2}](t) &= [x_{j,2}]^*(t) \cap (\bigcap_{i \in I_j(t)} [\psi_{j,i,2}]) \\ &= [\max\{\underline{x}_{j,2}(t-1) - \Delta t.v_{max,j}, \max_{i \in I_j(t)}(a_{i,2}(t) - r)\}, \\ &\quad \min\{\bar{x}_{j,2}(t-1) + \Delta t.v_{max,j}, \min_{i \in I_j(t)}(a_{i,2}(t) + r)\}]. \end{cases} \quad (\text{III.29})$$

Dans le cas de la simplification totale, l'estimation des positions des nœuds se résumera alors au calcul très rapide présenté dans (III.29).

III.5 Auto-localisation *back-propagated*

Dans cette section, nous proposons une première extension de la méthode ALCA exposée dans la section III.4. Cette méthode *back-propagated* effectue de la rétropropagation dans le temps pour corriger les pavés estimés précédemment. Nous l'appelons ALBP (pour Auto-Localisation *Back-Propagated*). Cette méthode est intéressante dans les applications où les positions estimées ne sont pas utilisables en temps réel. L'avantage de cette méthode est qu'elle permet d'obtenir plus de précision au niveau des positions estimées. Les pavés obtenus à l'instant t seront propagés, à l'aide du modèle de mobilité, pour corriger les pavés des instants $t-1$, $t-2$, $t-3$, etc. En d'autres termes, les observations courantes interviendront dans l'amélioration des positions précédentes. Soient $[\mathbf{x}_j](t)$ et $[\mathbf{x}_j](t-1)$ les pavés associés au nœud j respectivement aux instants t et $t-1$. Ces pavés sont estimés suivant la méthode ALCA, donc uniquement à l'aide des observations réalisées à leurs instants respectifs. Vu que $[\mathbf{x}_j](t)$ et $[\mathbf{x}_j](t-1)$ sont liés par le modèle de mobilité, ce dernier peut être utilisé pour exprimer $[\mathbf{x}_j](t-1)$ en fonction de $[\mathbf{x}_j](t)$, une fois $[\mathbf{x}_j](t)$ est contracté. Les deux sous-solveurs de rétropropagation associés à $[\mathbf{x}_j](t-1)$ sont donnés par

$$\begin{cases} [\phi_{j,1}^-] = [x_{j,1}](t) + [-\overline{bm}_{j,1}(t), +\overline{bm}_{j,1}(t)], \\ [\phi_{j,2}^-] = [x_{j,2}](t) + [-\overline{bm}_{j,2}(t), +\overline{bm}_{j,2}(t)], \end{cases} \quad (\text{III.30})$$

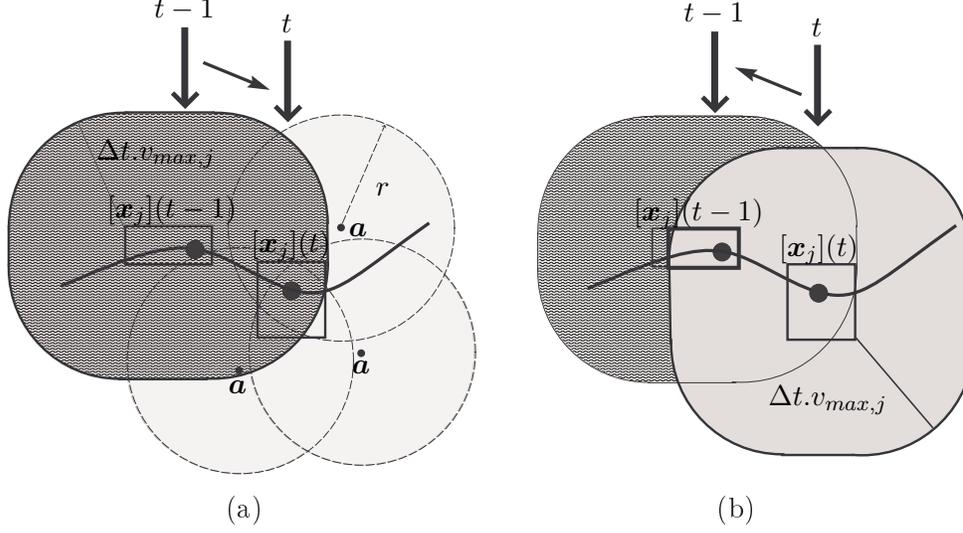


FIG. III.5: Auto-localisation *back-propagated* : Propagation et correction en (a) et rétro-propagation en (b).

avec

$$\begin{cases} \overline{bm}_{j,1}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf((x_{j,2}(t-1) - x_{j,2}(t))^2)}, \\ \overline{bm}_{j,2}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf((x_{j,1}(t-1) - x_{j,1}(t))^2)}. \end{cases}$$

Le nœud j construit une boucle de Waltz pour contracter le pavé obtenu à l'instant $t-1$ à l'aide des nouveaux sous-solveurs, comme suit,

$$\begin{aligned} [x_{j,1}](t-1) &= [x_{j,1}(t-1) \cap [\phi_{j,1}^-], \\ [x_{j,2}](t-1) &= [x_{j,2}(t-1) \cap [\phi_{j,2}^-]. \end{aligned} \quad (\text{III.31})$$

Une illustration est présentée dans FIG. III.5. Le pavé $[x_j](t-1)$, s'il est contracté, sera à son tour utilisé pour construire de nouveaux sous-solveurs pour contracter le pavé $[x_j](t-2)$, et ainsi de suite jusqu'à ce que le pavé d'un instant passé ne soit plus contracté ou que nous ayons atteint un certain nombre K_{BP} d'instant passés. Ainsi, la méthode ALBP est composée de trois étapes : la propagation, la correction et la rétropropagation. Les deux premières reprennent la méthode ALCA. Le pseudo-code correspondant à la rétropropagation est donné dans l'algorithme 5.

III.6 Auto-localisation *multi-hop*

La méthode *multi-hop*, appelée ALMH (pour Auto-Localisation *Multi-Hop*), représente une autre extension de la méthode basée sur les connectivités ALCA. Dans cette méthode, les nœuds ont la possibilité de communiquer avec des ancres plus loin que la portée pour définir leurs positions. Comparée à la méthode ALCA, cette méthode évoque plus d'observations, permettant ainsi de réduire de plus en plus les pavés estimés. La méthode ALMH est avantageuse dans les réseaux à faibles densités d'ancres.

Algorithme 5 : Pseudo-code de l'étape rétropropagation de la méthode ALBP.

Entrées : $[x_j](t)$, K_{BG} , $[x_j](t-k)$, $1 \leq k \leq K_{BG}$, $v_{max,j}$, Δt ;

Sorties : $[x_j](t-k)$, $1 \leq k \leq K_{BG}$;

$A_{BP,j,0}^- = w([x_{j,1}](t)) * w([x_{j,2}](t))$, $A_{BP,j,0} = A_{BP,0}^- - 1$;

$k = 1$;

tant que $A_{BP,j,k-1} < A_{BP,j,k-1}^-$ **et** $k \leq K_{BP}$ **faire**

$A_{BP,j,k}^- = w([x_{j,1}](t-k)) * w([x_{j,2}](t-k))$;

$A_j = A_{BP,j,k}^-$;

$A_j^{old} = A_j + 1$;

tant que $A_j < A_j^{old}$ **faire**

$A_j^{old} = A_j$;

$\overline{bm}_{j,1}^-(t-k+1) = \sqrt{(\Delta t \cdot v_{max,j})^2 - \inf((\overline{[x_{j,2}]}(t-k) - \overline{[x_{j,2}]}(t-k+1))^2)}$;

$[\phi_{j,1}^-] = [x_{j,1}](t-k+1) + [-\overline{bm}_{j,1}^-(t-k+1), +\overline{bm}_{j,1}^-(t-k+1)]$;

$[x_{j,1}](t-k) = [x_{j,1}](t-k) \cap [\phi_{j,1}^-]$;

$\overline{bm}_{j,2}^-(t-k+1) = \sqrt{(\Delta t \cdot v_{max,j})^2 - \inf((\overline{[x_{j,1}]}(t-k) - \overline{[x_{j,1}]}(t-k+1))^2)}$;

$[\phi_{j,2}^-] = [x_{j,2}](t-k+1) + [-\overline{bm}_{j,2}^-(t-k+1), +\overline{bm}_{j,2}^-(t-k+1)]$;

$[x_{j,2}](t-k) = [x_{j,2}](t-k) \cap [\phi_{j,2}^-]$;

$A_j = w([x_{j,1}](t-k)) * w([x_{j,2}](t-k))$;

fin

$A_{BP,j,k} = A_j = w([x_{j,1}](t-k)) * w([x_{j,2}](t-k))$;

$k = k + 1$;

fin

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

Dans la suite, nous présentons le modèle d'observation de la méthode ALMH. Nous exposons ensuite l'algorithme de résolution du problème.

III.6.1 Modèle d'observation

Selon le modèle d'Okumura-Hata présenté dans le paragraphe III.4.1, la puissance du signal décroît avec l'accroissement de la distance parcourue,

$$\rho_{\mathbf{a}_i, \mathbf{x}_j}(t) = \rho_0 - 10 \cdot n_P \cdot \log_{10} \frac{d(\mathbf{a}_i(t), \mathbf{x}_j(t))}{d_0} + \varepsilon_{i,j}(t), \quad (\text{III.32})$$

où $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)$ en dBm est la puissance à la réception du signal émis par une ancre $\mathbf{a}_i(t)$ et reçu par un nœud $\mathbf{x}_j(t)$, ρ_0 en dBm est la puissance mesurée à une certaine distance parcourue d_0 , n_P est le paramètre de perte correspondant au canal, $d(\mathbf{a}_i(t), \mathbf{x}_j(t))$ est la distance entre $\mathbf{a}_i(t)$ et $\mathbf{x}_j(t)$ et $\varepsilon_{i,j}(t)$ est le bruit de mesure.

Un nœud j , recevant un signal d'une ancre i , compare la puissance du signal à différents seuils $\rho_r, \rho_{2.r}, \rho_{3.r}, \dots$ contrairement à la méthode ALCA qui n'utilise qu'un seul seuil ρ_r . Les seuils sont les puissances associées aux multiples de la portée de communication $r, 2.r, 3.r, \dots$ selon le modèle (III.32). Si la puissance reçue $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)$ est comprise entre $\rho_{k_i.r}$ et $\rho_{(k_i-1).r}$, alors la distance entre $\mathbf{x}_j(t)$ et $\mathbf{a}_i(t)$ est comprise entre $(k_i - 1).r$ et $k_i.r$,

$$\text{si } \rho_{k_i.r} \leq \rho_{\mathbf{a}_i, \mathbf{x}_j}(t) < \rho_{(k_i-1).r}, \text{ alors } (k_i - 1).r < d(\mathbf{a}_i(t), \mathbf{x}_j(t)) \leq k_i.r. \quad (\text{III.33})$$

Dans ce cas, l'ancre $\mathbf{a}_i(t)$ est dite k_i -hop. Les mesures d'observations $\mathbf{y}_j(t)$ ne seront plus binaires, mais plutôt à valeurs entières positives définies comme suit,

$$y_{j,i}(t) = \begin{cases} 1 & \text{si } \rho_{\mathbf{a}_i, \mathbf{x}_j}(t) \geq \rho_r, \\ k_i & \text{si } \rho_{k_i.r} \leq \rho_{\mathbf{a}_i, \mathbf{x}_j}(t) < \rho_{(k_i-1).r}, k_i \geq 2, \\ 0 & \text{si pas de signal reçu de la part de l'ancre } i, \end{cases} \quad (\text{III.34})$$

avec $\mathbf{y}_j(t) = (y_{j,i}(t))_{i \in \{1, \dots, N_a\}}$.

Nous noterons dans cette méthode $I_j(t)$ l'ensemble des indices des composantes non nulles de $\mathbf{y}_j(t)$. En d'autres termes, $I_j(t)$ est l'ensemble des indices des ancres dont les signaux ont été reçus par le nœud j à l'instant t . dite ancres détectées. Le modèle d'observation de la méthode *multi-hop* est alors donné comme suit,

$$\begin{aligned} y_{j,i}(t) = k_i &\Leftrightarrow (k_i - 1).r < d(\mathbf{a}_i(t), \mathbf{x}_j(t)) \leq k_i.r \\ &\Leftrightarrow (k_i - 1)^2.r^2 < (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq k_i^2.r^2, \\ &\Leftrightarrow (y_{j,i}(t) - 1)^2.r^2 < (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq y_{j,i}(t)^2.r^2, \end{aligned} \quad (\text{III.35})$$

avec $k_i \geq 1$ et $i \in I_j(t)$. Pour toute ancre i communiquant avec le nœud j avec une mesure $k_i > 1$, un anneau sera défini, centré sur l'ancre et ayant respectivement $(k_i - 1).r$ et $k_i.r$ comme rayons interne et externe. Pour $k_i = 1$, le rayon interne devient nul et l'anneau se réduit à un disque de rayon r .

III.6.2 Algorithme

Le problème d'auto-localisation selon cette méthode est défini par l'ensemble des modèles de mobilité et d'observation suivants,

$$\begin{cases} (x_{j,1}(t) - x_{j,1}(t-1))^2 + (x_{j,2}(t) - x_{j,2}(t-1))^2 = (\Delta t \cdot v_j)^2, \\ (k_i - 1)^2 \cdot r^2 < (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq k_i^2 \cdot r^2, i \in I_j(t), \\ v_j \in [0, v_{max,j}]. \end{cases} \quad (\text{III.36})$$

Similairement à la méthode ALCA, le problème est considéré comme un CSP dont la résolution se fait à l'aide du contracteur de Waltz. La formulation du problème à l'aide des intervalles est donnée come suit,

$$\begin{cases} ([x_{j,1}](t) - [x_{j,1}](t-1))^2 + ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, (\Delta t \cdot v_{max,j})^2], \\ ([x_{j,1}](t) - a_{i,1}(t))^2 + ([x_{j,2}](t) - a_{i,2}(t))^2 = [(k_i - 1)^2 \cdot r^2, k_i^2 \cdot r^2], i \in I_j(t). \end{cases} \quad (\text{III.37})$$

La résolution du problème se fait selon deux étapes : étape de propagation, où un domaine initial est conçu, et étape de correction, où ce domaine est contracté. Seuls les sous-solveurs d'observation de cette méthode diffèrent de la méthode ALCA présentée dans la section III.4. L'étape de propagation, similairement à la méthode ALCA, consiste à définir un domaine initial $[\mathbf{x}_j]^*(t)$ à l'aide du modèle de mobilité comme suit,

$$\begin{cases} [x_{j,1}]^*(t) = [x_{j,1}](t-1) + [-\Delta t \cdot v_{max,j}, +\Delta t \cdot v_{max,j}], \\ [x_{j,2}]^*(t) = [x_{j,2}](t-1) + [-\Delta t \cdot v_{max,j}, +\Delta t \cdot v_{max,j}], \end{cases} \quad (\text{III.38})$$

où $[\mathbf{x}_j](t-1) = [x_{j,1}](t-1) \times [x_{j,2}](t-1)$ est le pavé estimé à l'instant $t-1$, $v_{max,j}$ est la vitesse maximale et Δt est la durée entre deux pas de temps consécutifs. Ce domaine est par la suite contracté à l'aide du contracteur de Waltz implémenté par des sous-solveurs. Pareillement à la méthode ALCA, les sous-solveurs de mobilité sont donnés par

$$\begin{cases} [\phi_{j,1}] = [x_{j,1}](t-1) + [-\overline{bm}_{j,1}(t), +\overline{bm}_{j,1}(t)], \\ [\phi_{j,2}] = [x_{j,2}](t-1) + [-\overline{bm}_{j,2}(t), +\overline{bm}_{j,2}(t)], \end{cases} \quad (\text{III.39})$$

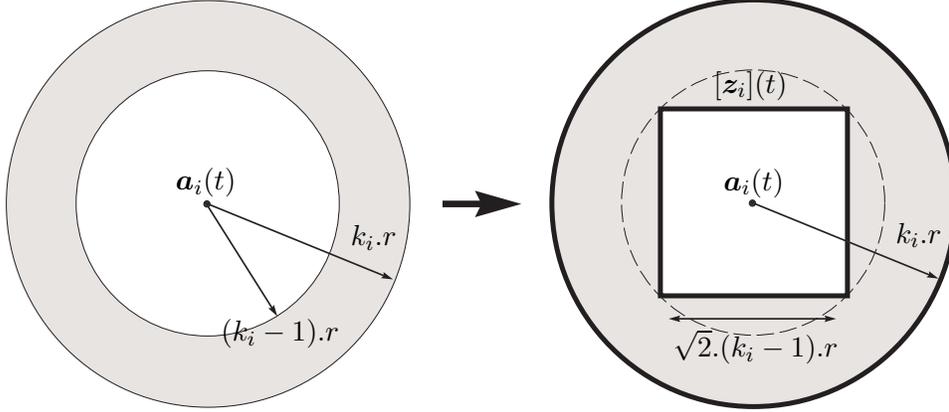
avec

$$\begin{cases} \overline{bm}_{j,1}(t) = \sqrt{(\Delta t \cdot v_{max,j})^2 - \inf(([x_{j,2}](t) - [x_{j,2}](t-1))^2)}, \\ \overline{bm}_{j,2}(t) = \sqrt{(\Delta t \cdot v_{max,j})^2 - \inf(([x_{j,1}](t) - [x_{j,1}](t-1))^2)}. \end{cases}$$

Les pavés à estimer seront alors contractés à l'aide du modèle de mobilité de la manière suivante,

$$\begin{cases} [x_{j,1}](t) = [x_{j,1}](t) \cap [\phi_{j,1}], \\ [x_{j,2}](t) = [x_{j,2}](t) \cap [\phi_{j,2}]. \end{cases} \quad (\text{III.40})$$

Le modèle d'observation définit des anneaux centrés sur les ancrs détectées. La position réelle du nœud tombera alors dans la zone d'intersection des anneaux. Pour un anneau donné i , si nous prenons séparément les cercles le limitant, le nœud considéré tombera à la fois à l'intérieur du disque de rayon $k_i \cdot r$ et à l'extérieur du disque de rayon $(k_i - 1) \cdot r$, l'ancree $\mathbf{a}_i(t)$ étant k_i -hop. Afin d'appliquer cette dernière contrainte dans


 FIG. III.6: Contraintes d'observation de la méthode *multi-hop*.

le cadre des intervalles, nous privons $[\mathbf{x}_j](t)$ du carré inscrit dans le disque interne de l'anneau comme suit,

$$[\mathbf{x}_j](t) = [\mathbf{x}_j](t) \setminus [\mathbf{z}_i](t), \quad (\text{III.41})$$

avec $[\mathbf{z}_i](t) = [z_{i,1}](t) \times [z_{i,2}](t)$ où

$$\begin{cases} [z_{i,1}](t) = [a_{i,1}(t) - \frac{\sqrt{2}}{2} \cdot (k_i - 1) \cdot r, a_{i,1}(t) + \frac{\sqrt{2}}{2} \cdot (k_i - 1) \cdot r], \\ [z_{i,2}](t) = [a_{i,2}(t) - \frac{\sqrt{2}}{2} \cdot (k_i - 1) \cdot r, a_{i,2}(t) + \frac{\sqrt{2}}{2} \cdot (k_i - 1) \cdot r]. \end{cases} \quad (\text{III.42})$$

La longueur du côté du carré inscrit est alors égale à $\sqrt{2} \cdot (k_i - 1) \cdot r$. Une illustration est donnée dans FIG. III.6. Pour $k_i = 1$, le carré se réduit à un point permettant ainsi l'élimination de la contrainte. Nous contractons également $[\mathbf{x}_j](t)$ à l'aide de l'équation du disque externe donné par

$$([x_{j,1}](t) - a_{i,1}(t))^2 + ([x_{j,2}](t) - a_{i,2}(t))^2 = [0, (k_i \cdot r)^2]. \quad (\text{III.43})$$

Pour cette raison, nous définissons des sous-solveurs d'observation de la manière suivante,

$$\begin{cases} [\psi_{j,i,1}] = [a_{i,1}(t) + [-\overline{bo}_{j,i,1}(t), +\overline{bo}_{j,i,1}(t)], \\ [\psi_{j,i,2}] = [a_{i,2}(t) + [-\overline{bo}_{j,i,2}(t), +\overline{bo}_{j,i,2}(t)], \end{cases} \quad (\text{III.44})$$

avec

$$\begin{cases} \overline{bo}_{j,i,1}(t) = \sqrt{(k_i \cdot r)^2 - \inf(([\mathbf{x}_{j,2}](t) - a_{i,2}(t))^2)}, \\ \overline{bo}_{j,i,2}(t) = \sqrt{(k_i \cdot r)^2 - \inf(([\mathbf{x}_{j,1}](t) - a_{i,1}(t))^2)}. \end{cases}$$

La contraction des pavés s'effectue alors comme suit,

$$\begin{cases} [x_{j,1}](t) = [x_{j,1}](t) \cap [\psi_{j,i,1}], \\ [x_{j,2}](t) = [x_{j,2}](t) \cap [\psi_{j,i,2}]. \end{cases} \quad (\text{III.45})$$

Le pseudo-code de la méthode ALMH pour un nœud j est donné dans l'algorithme 6. Nous illustrons également la méthode dans FIG. III.7.

Algorithme 6 : Pseudo-code de la méthode d'auto-localisation *multi-hop*.

Entrées : $[x_j](t-1)$, $I_j(t)$, $y_j(t)$, $\mathbf{a}_i(t)$, $i \in I_j(t)$, $v_{max,j}$, Δt , r ;

Sorties : $[x_j](t)$;

$$\begin{cases} [x_{j,1}]^*(t) = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]; \\ [x_{j,2}]^*(t) = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]; \end{cases}$$

$$[x_{j,1}](t) = [x_{j,1}]^*(t), [x_{j,2}](t) = [x_{j,2}]^*(t);$$

$$A_j = w([x_{j,1}](t)) * w([x_{j,2}](t));$$

$$A_j^{old} = A_j + 1;$$

tant que $A_j < A_j^{old}$ **faire**

$$A_j^{old} = A_j;$$

pour $i \in I_j(t)$ **faire**

$$k_i = y_{j,i}(t);$$

$$[x_{j,2}](t) = [x_{j,2}](t) \cap [a_{i,2}(t) - k_i.r, a_{i,2}(t) + k_i.r];$$

$$\overline{bo_{j,i,1}}(t) = \sqrt{(k_i.r)^2 - \inf(([x_{j,2}](t) - a_{i,2}(t))^2)};$$

$$[\psi_{j,i,1}] = a_{i,1}(t) + [-\overline{bo_{j,i,1}}(t), +\overline{bo_{j,i,1}}(t)];$$

$$[x_{j,1}](t) = [x_{j,1}](t) \cap [\psi_{j,i,1}];$$

$$[x_{j,1}](t) = [x_{j,1}](t) \cap [a_{i,1}(t) - k_i.r, a_{i,1}(t) + k_i.r];$$

$$\overline{bo_{j,i,2}}(t) = \sqrt{(k_i.r)^2 - \inf(([x_{j,1}](t) - a_{i,1}(t))^2)};$$

$$[\psi_{j,i,2}] = a_{i,2}(t) + [-\overline{bo_{j,i,2}}(t), +\overline{bo_{j,i,2}}(t)];$$

$$[x_{j,2}](t) = [x_{j,2}](t) \cap [\psi_{j,i,2}];$$

si $k_i > 1$ **alors**

$$[z_{i,1}](t) = [a_{i,1}(t) - \frac{\sqrt{2}}{2} \cdot (k_i - 1) \cdot r, a_{i,1}(t) + \frac{\sqrt{2}}{2} \cdot (k_i - 1) \cdot r];$$

$$[z_{i,2}](t) = [a_{i,2}(t) - \frac{\sqrt{2}}{2} \cdot (k_i - 1) \cdot r, a_{i,2}(t) + \frac{\sqrt{2}}{2} \cdot (k_i - 1) \cdot r];$$

$$[x_j](t) = [x_j](t) \setminus [z_i](t);$$

fin
fin

$$\overline{bm_{j,1}}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf(([x_{j,2}](t) - [x_{j,2}](t-1))^2)};$$

$$[\phi_{j,1}] = [x_{j,1}](t-1) + [-\overline{bm_{j,1}}(t), +\overline{bm_{j,1}}(t)];$$

$$[x_{j,1}](t) = [x_{j,1}](t) \cap [\phi_{j,1}];$$

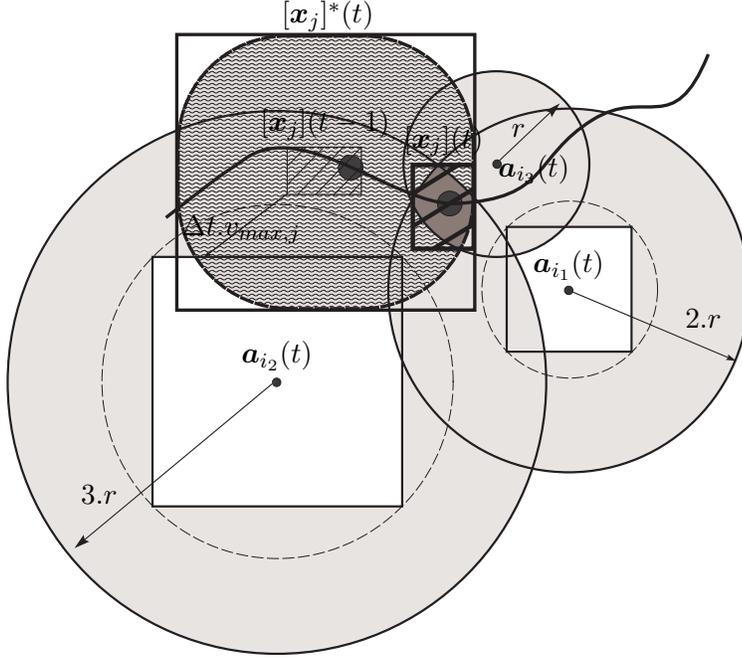
$$\overline{bm_{j,2}}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf(([x_{j,1}](t) - [x_{j,1}](t-1))^2)};$$

$$[\phi_{j,2}] = [x_{j,2}](t-1) + [-\overline{bm_{j,2}}(t), +\overline{bm_{j,2}}(t)];$$

$$[x_{j,2}](t) = [x_{j,2}](t) \cap [\phi_{j,2}];$$

$$A_j = w([x_{j,1}](t)) * w([x_{j,2}](t));$$

fin


 FIG. III.7: Auto-localisation *multi-hop*.

Remarque 4. Le bruit de mesure ou l'erreur du paramétrage du modèle d'Okumura-hata peut entraîner une erreur dans l'obtention des seuils de distances à partir des seuils de puissances. Cette erreur peut entraîner des mesures erronées et par la suite une perte de la garantie, où le pavé estimé ne contiendra plus la position réelle. Il est possible de tenir compte de l'incertitude sur ces seuils dans la méthode d'auto-localisation *multi-hop*. Soit $[k_i.r - \delta r, k_i.r + \delta r]$ l'intervalle couvrant l'incertitude sur le seuil $k_i.r$. Les équations d'observation s'écrivent dans ce cas de la manière suivante,

$$([x_{j,1}](t) - [a_{i,1}](t))^2 + ([x_{j,2}](t) - [a_{i,2}](t))^2 = [\max\{0, ((k_i - 1).r - \delta r)^2\}, (k_i.r + \delta r)^2], \quad (\text{III.46})$$

où les coordonnées des ancres sont supposées incertaines et donc représentées par les intervalles $[a_{i,1}](t)$ et $[a_{i,2}](t)$, $i \in I_j(t)$. Par exemple, pour $k_i = 2$, l'anneau, dit 2-hop, est limité par des cercles de rayons $r - \delta r$ et $2.r + \delta r$. Les anneaux seront ainsi plus larges couvrant la position réelle du nœud considéré. Une illustration est donnée dans FIG. III.8. La figure montre l'influence de l'erreur de seuillage sur les mesures d'observation à gauche et la correction de cette erreur à droite. En effet, les ancres a_{i_1} , a_{i_2} et a_{i_3} sont toutes supposées des ancres 2-hop ($k_{i_1} = k_{i_2} = k_{i_3} = 2$) alors que réellement, $k_{i_1} = 1$, $k_{i_2} = 2$ et $k_{i_3} = 3$. En d'autres termes, les distances entre les ancres et le nœud sont supposées comprises entre r et $2.r$ alors qu'elles ne le sont pas. Nous montrons dans la figure en petits disques pleins les positions des ancres correspondant aux coordonnées reçues et en cercles noirs les positions des ancres correspondant aux

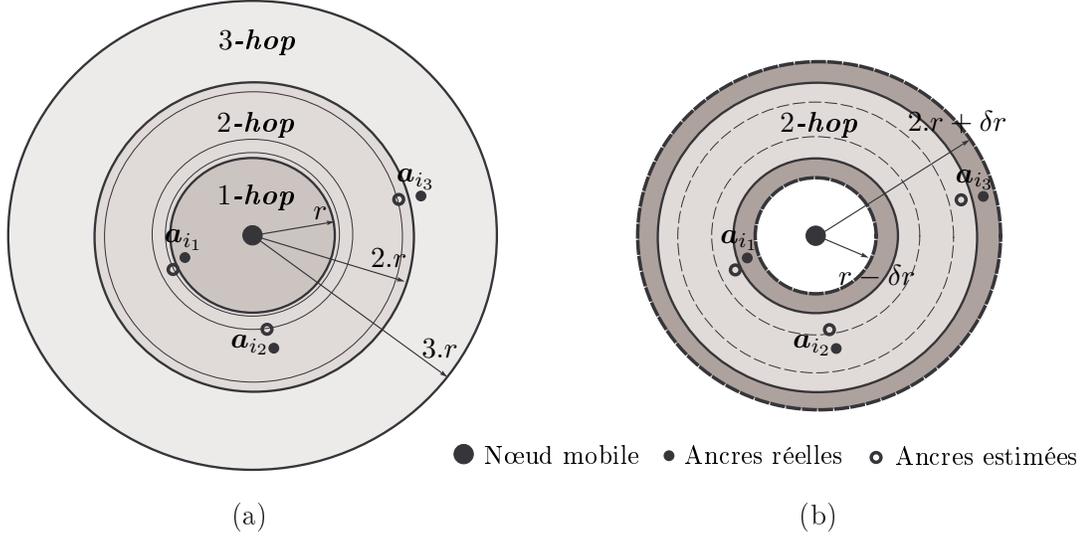


FIG. III.8: Influence des erreurs de seuillage sur les mesures d'observation en (a) et l'effet de l'utilisation d'intervalles sur les seuils en (b).

distances estimées incorrectes. En définissant le modèle d'observation tout en se basant sur les mesures erronées ($k = 2$), le pavé résultant ne contiendra pas le plus probablement la bonne position du nœud. L'ajout de l'incertitude aux seuillages permettra d'obtenir un anneau 2-hop plus large couvrant ainsi les bonnes positions des ancrés et donc corrigeant l'effet des mesures erronées. FIG. III.9 montre le pavé résultant de l'intersection des mesures erronées en (a) et celui obtenu suite à la correction en (b). La figure montre que la correction à l'aide des intervalles mis sur les seuils permet de regagner la garantie.

III.7 Auto-localisation basée sur les ancrés et les non-ancrés

Les méthodes présentées précédemment se basent uniquement sur les ancrés qu'elles soient à proximité du nœud ou plus loin. Dans cette section, nous utilisons les informations de proximité échangées à la fois avec les ancrés et les non-ancrés du voisinage du nœud considéré. Nous appelons la méthode ALCAN (pour Auto-Localisation par Connectivités vis-à-vis des Ancrés et des Non-ancrés). Cette méthode représente une extension directe de la méthode ALCA développée dans la section III.4. Outre les ancrés détectés, le nœud j détecte les non-ancrés situés à proximité de lui et en produit des contraintes d'observation supplémentaires.

Un nœud j , recevant des signaux des ancrés et des nœuds du réseau, compare les puissances reçues au seuil ρ_r , associé à la portée de communication r . Soient $\rho_{a_i, x_j}(t)$ et $\rho_{x_k, x_j}(t)$ les puissances à la réception des signaux émis respectivement par les ancrés et les nœuds, $i \in \{1, \dots, N_a\}$ et $k \in \{1, \dots, N_x\}, k \neq j$. Les ancrés et les nœuds détectés sont ceux pour lesquels $\rho_{a_i, x_j}(t) \geq \rho_r$ et $\rho_{x_k, x_j}(t) \geq \rho_r$. Soient $I_{a,j}(t)$ et $I_{x,j}(t)$ les ensembles

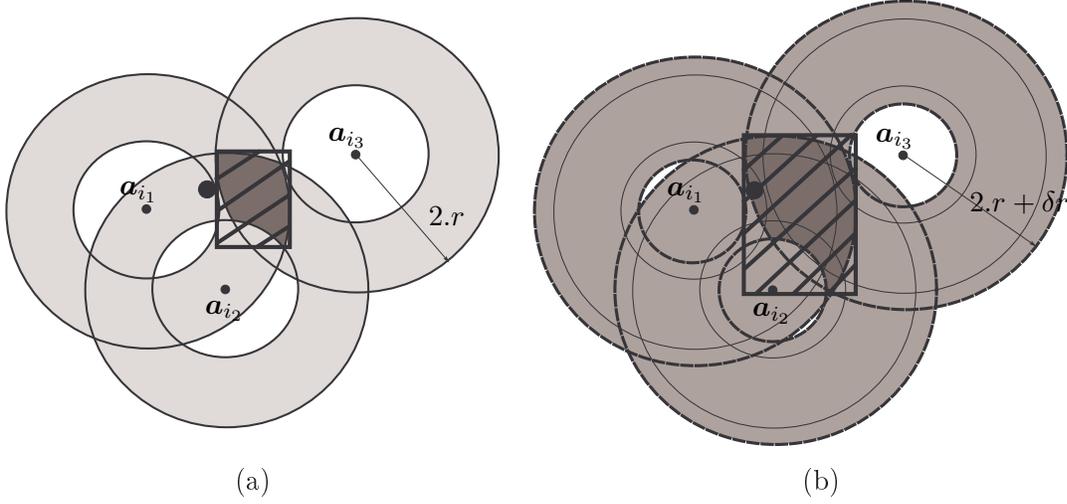


FIG. III.9: Perte de la garantie en (a) et correction de l'erreur en (b).

respectifs des indices des ancrs et des non-ancres détectés par le nœud j à l'instant t . Pour tout nœud k détecté, $k \in I_{x,j}(t)$, une contrainte d'observation est générée comme suit,

$$(x_{j,1}(t) - x_{k,1}(t))^2 + (x_{j,2}(t) - x_{k,2}(t))^2 \leq r^2. \quad (\text{III.47})$$

Cette contrainte s'écrit en fonction des intervalles de la manière suivante,

$$([x_{j,1}](t) - [x_{k,1}](t))^2 + ([x_{j,2}](t) - [x_{k,2}](t))^2 = [0, r^2]. \quad (\text{III.48})$$

Celle-ci induit deux sous-solveurs propres au nœud j comme suit,

$$\begin{cases} [\varphi_{j,k,1}] = [x_{k,1}](t) + [-\overline{bn}_{j,k,1}(t), +\overline{bn}_{j,k,1}(t)], \\ [\varphi_{j,k,2}] = [x_{k,2}](t) + [-\overline{bn}_{j,k,2}(t), +\overline{bn}_{j,k,2}(t)], \end{cases} \quad (\text{III.49})$$

avec

$$\begin{cases} \overline{bn}_{j,k,1}(t) = \sqrt{r^2 - \inf(([x_{j,2}](t) - [x_{k,2}](t))^2)}, \\ \overline{bn}_{j,k,2}(t) = \sqrt{r^2 - \inf(([x_{j,1}](t) - [x_{k,1}](t))^2)}. \end{cases}$$

La contraction grâce à ces deux sous-solveurs se fait alors comme suit, pour $k \in I_{x,j}(t)$,

$$\begin{aligned} [x_{j,1}](t) &= [x_{j,1}](t) \cap [\varphi_{j,k,1}], \\ [x_{j,2}](t) &= [x_{j,2}](t) \cap [\varphi_{j,k,2}]. \end{aligned} \quad (\text{III.50})$$

Chaque nœud j se localise en premier en se basant uniquement sur les ancrs. Il applique alors la méthode ALCA présentée dans l'algorithme 5 pour estimer sa position $[x_j](t)$. Les nœuds à proximité l'un de l'autre échangent ensuite leurs pavés estimés pour pouvoir définir leurs sous-solveurs supplémentaires et refaire une contraction totale. La méthode ALCAN présente alors trois étapes : i) la propagation, où le pavé initial est conçu grâce au modèle de mobilité, ii) la correction partielle, que nous effectuons grâce

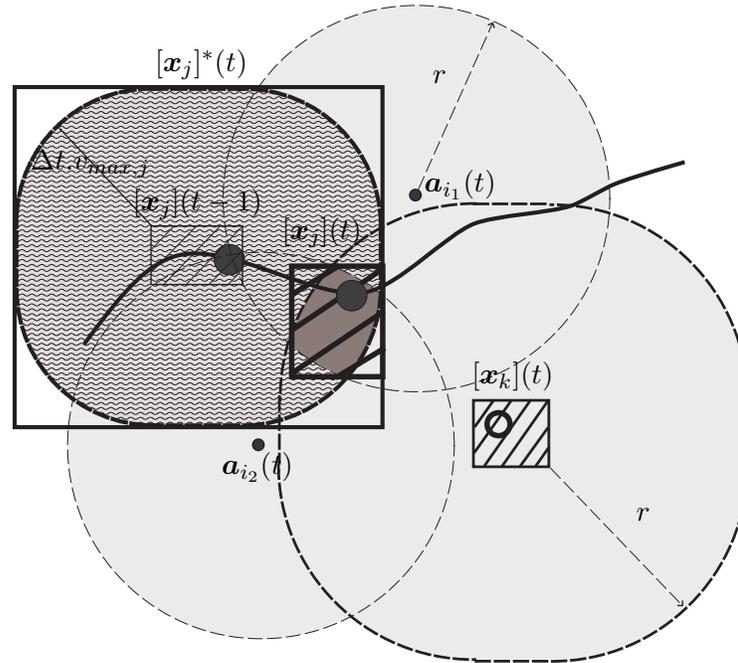


FIG. III.10: Auto-localisation basée sur les ancrés et les non-ancrés.

aux mesures des ancrés et iii) la correction totale où les nœuds échangent leurs pavés et effectuent une contraction grâce à toutes les mesures. La distinction de deux étapes de contraction permet de limiter l'échange des pavés entre les nœuds et de réduire ainsi le coût de communication. Le pseudo-code associé aux deux premières étapes reprend l'algorithme 4. Par ailleurs, le pseudo-code de la troisième étape est donné dans l'algorithme 7. Les nœuds voisins échangent ainsi leurs pavés à chaque itération de la dernière étape tant que leurs domaines se contractent. Cette méthode assure un gain de précision par rapport à la méthode basée sur les ancrés surtout dans les réseaux à faibles densités d'ancrés. Une illustration de cette méthode est donnée dans FIG. III.10.

III.8 Simulations

Dans cette section, nous évaluons les performances des différentes méthodes présentées dans ce chapitre. Les critères d'évaluation sont donnés par :

- ◇ la **précision de l'estimation**, exprimée par la surface du pavé estimé ;
- ◇ l'**erreur d'estimation**, définie par la distance entre la position réelle du nœud et le centre du pavé estimé ;
- ◇ et la **complexité de la méthode**, donnée par le temps de calcul.

Il faut noter que toutes les simulations sont réalisées sur Matlab 6.1, installé sur un CPU Intel(R) Core(TM)2 (2.40GHz, 1.00GB RAM).

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

Algorithme 7 : Pseudo-code de l'extension apportée par la méthode ALCAN par rapport à ALCA.

Entrées : $[\mathbf{x}_j](t), I_{x,j}(t), I_{a,j}(t), \mathbf{a}_i(t), i \in I_{a,j}(t), r$;

Sorties : $[\mathbf{x}_j](t)$;

$A_j = w([x_{j,1}](t)) * w([x_{j,2}](t))$;

$A_j^{old} = A_j + 1$;

tant que $A_j < A_j^{old}$ **faire**

$A_j^{old} = A_j$;

pour $k \in I_{x,j}(t)$ **faire**

$[\mathbf{x}_j](t) \leftrightarrow \mathbf{x}_k(t)$; $\mathbf{x}_j(t) \leftarrow [\mathbf{x}_k](t)$;

$[x_{j,2}](t) = [x_{j,2}](t) \cap [\underline{x}_{k,2}(t) - r, \bar{x}_{k,2}(t) + r]$;

$\overline{bn}_{j,k,1}(t) = \sqrt{r^2 - \inf((\underline{[x_{j,2}]}(t) - \underline{[x_{k,2}]}(t))^2)}$;

$[\varphi_{j,k,1}] = [x_{k,1}](t) + [-\overline{bn}_{j,k,1}(t), +\overline{bn}_{j,k,1}(t)]$;

$[x_{j,1}](t) = [x_{j,1}](t) \cap [\varphi_{j,k,1}]$;

$[x_{j,1}](t) = [x_{j,1}](t) \cap [\underline{x}_{k,1}(t) - r, \bar{x}_{k,1}(t) + r]$;

$\overline{bn}_{j,k,2}(t) = \sqrt{r^2 - \inf((\underline{[x_{j,1}]}(t) - \underline{[x_{k,1}]}(t))^2)}$;

$[\varphi_{j,k,2}] = [x_{k,2}](t) + [-\overline{bn}_{j,k,2}(t), +\overline{bn}_{j,k,2}(t)]$;

$[x_{j,2}](t) = [x_{j,2}](t) \cap [\varphi_{j,k,2}]$;

fin

pour $i \in I_{a,j}(t)$ **faire**

$[x_{j,2}](t) = [x_{j,2}](t) \cap [a_{i,2}(t) - r, a_{i,2}(t) + r]$;

$\overline{bo}_{j,i,1}(t) = \sqrt{r^2 - \inf((\underline{[x_{j,2}]}(t) - a_{i,2}(t))^2)}$;

$[\psi_{j,i,1}] = a_{i,1}(t) + [-\overline{bo}_{j,i,1}(t), +\overline{bo}_{j,i,1}(t)]$;

$[x_{j,1}](t) = [x_{j,1}](t) \cap [\psi_{j,i,1}]$;

$[x_{j,1}](t) = [x_{j,1}](t) \cap [a_{i,1}(t) - r, a_{i,1}(t) + r]$;

$\overline{bo}_{j,i,2}(t) = \sqrt{r^2 - \inf((\underline{[x_{j,1}]}(t) - a_{i,1}(t))^2)}$;

$[\psi_{j,i,2}] = [x_{i,2}](t) + [-\overline{bo}_{j,i,2}(t), +\overline{bo}_{j,i,2}(t)]$;

$[x_{j,2}](t) = [x_{j,2}](t) \cap [\psi_{j,i,2}]$;

fin

$A_j = w([x_{j,1}](t)) * w([x_{j,2}](t))$;

fin

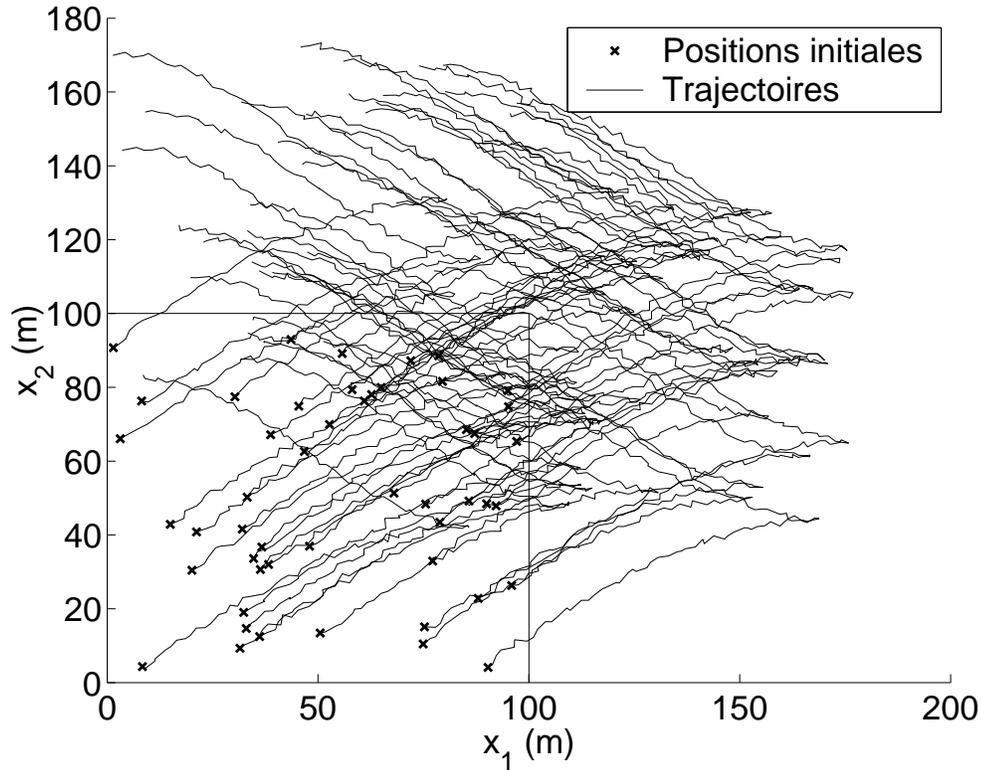


FIG. III.11: Mouvement de groupe de 50 capteurs.

Afin d'évaluer les différentes méthodes, nous déployons les capteurs dans une zone carrée de dimensions $100m \times 100m$, dite **zone de déploiement**. Nous supposons que tous les nœuds ont un mouvement de groupe. Les ancres par contre sont considérées fixes dans la zone de déploiement. La portée de communication des capteurs r est fixée à $10m$. Nous générons le mouvement de groupe selon le modèle *Reference Point Group Mobility Model* ou RPGMM (voir la section I.4.3). D'après ce modèle, les capteurs possèdent des points de référence propres se déplaçant tous selon la même trajectoire, dite trajectoire de base. Les capteurs ont également la possibilité de bouger aléatoirement autour de leurs points de référence. Dans ce chapitre, nous choisissons une trajectoire de base de $100s$ composée par l'association de deux sinusoides. Une illustration d'un mouvement de groupe de 50 capteurs suivant cette trajectoire est donnée dans FIG. III.11. Les positions initiales des capteurs sont choisies d'une manière aléatoire à l'intérieur de la zone carrée $100m \times 100m$. Initialisés sur ces positions, les points de référence suivent la trajectoire de base (avec une simple translation). Chacun des capteurs choisit par la suite une position aléatoire à moins de $1m$ de son point de référence. Selon la trajectoire de base, la vitesse maximale du capteur est égale à $v_{max} = 2.0343m.s^{-1}$, où la durée entre deux pas de temps consécutifs est égale à $\Delta t = 1s$.

III.8.1 Evaluation de la méthode par connectivités basées sur les ancrés

Dans cette partie, nous illustrons les performances de la méthode basée sur les connectivités vis-à-vis des ancrés, dite ALCA. Nous commençons alors par comparer les trois versions de cette méthode : la méthode ALCA exacte, ALCA avec simplification partielle, notée ALCAp, et ALCA avec simplification totale, notée ALCAt. Nous illustrons ensuite les effets des valeurs que nous donnons à la vitesse maximale et à la portée de communication. Nous comparons enfin la méthode proposée à une méthode d'auto-localisation basée sur l'approche de Monte-Carlo [Baggio et Langendoen, 2006]. Sans perte de généralité, nous considérons le problème d'auto-localisation d'un seul nœud mobile vu que le problème est défini pour chacun des nœuds indépendamment des autres. Sa mobilité est générée suivant la trajectoire de base. Nous supposons en outre les ancrés fixes. Nous déployons ainsi 120 ancrés aléatoirement dans la zone carrée de $100m \times 100m$. Des configurations sont alors générées aléatoirement jusqu'à ce que le nœud considéré ait au moins trois ancrés dans son voisinage à chaque instant. La vitesse maximale du nœud est prise en premier égale à $v_{max} = 2.0343m.s^{-1}$.

1. *Evaluation des trois versions de la méthode ALCA*

Dans ce paragraphe, nous comparons la méthode ALCA à ses deux versions simplifiées : ALCAp, où seul le modèle de mobilité est relaxé, et ALCAt, où à la fois les modèles de mobilité et d'observation sont approximés. FIG. III.12 montre les pavés obtenus à l'aide des méthodes ALCA et ALCAt. La figure montre en outre les positions des ancrés, la trajectoire du nœud considéré et les positions réelles du nœud aux instants illustrés. FIG. III.13 illustre les rapports des surfaces des pavés obtenus avec la méthode ALCA sur ceux obtenus avec la méthode ALCAp en trait interrompu et ceux obtenus avec ALCA sur ceux obtenus avec ALCAt en trait plein. La figure montre que la simplification partielle a un effet très négligeable sur la méthode exacte (moyenne des rapports ALCA/ALCAp = 0.9999). La simplification totale, impliquant une approximation du modèle d'observation, produit par contre des pavés visiblement plus larges (moyenne des rapports ALCA/ALCAt = 0.8176). Le nombre d'itérations effectuées dans l'algorithme de Waltz vaut 5 au maximum pour ALCA et ALCAp, alors qu'il est égal à 1 pour ALCAt. TAB. III.1 résume les résultats obtenus par les trois versions de la méthode. Le tableau illustre le temps de calcul moyen, l'erreur d'estimation moyenne et la surface moyenne des pavés obtenus par pas de temps. La version ALCAt avec simplification totale réduit énormément le temps de calcul au prix de la précision de l'estimation. Vu que la différence entre les méthodes ALCA et ALCAp est très négligeable au niveau de la précision et de l'erreur d'estimation, nous adoptons dans la suite la méthode avec simplification partielle ALCAp, que nous noterons ALCA tout court.

2. *Effet de la valeur accordée à la vitesse maximale*

Le modèle de mobilité que nous proposons repose sur la vitesse maximale du nœud. Les performances de la méthode dépendent alors de la valeur accordée au paramètre v_{max} à l'entrée de l'algorithme. Soit v^* la vraie vitesse maximale. Alors

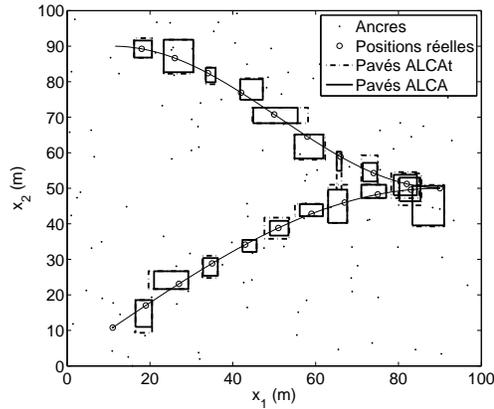


FIG. III.12: Pavés estimés à l'aide des méthodes ALCA et ALCAt.

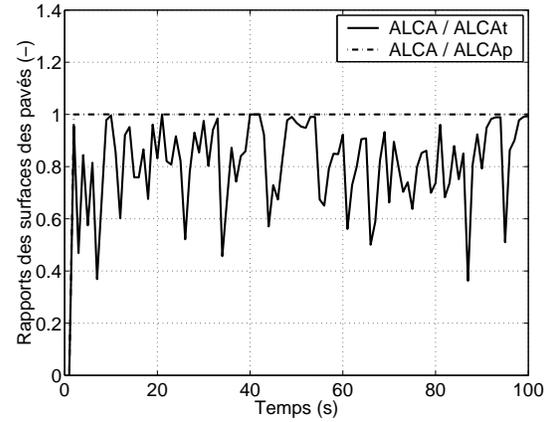


FIG. III.13: Rapports des surfaces des pavés obtenus avec ALCA, ALCAp et ALCAt.

Méthode	ALCA	ALCAp	ALCAt
Temps de calcul	0.00734s	0.00655s	0.00093s
Erreur moyenne	1.7775m	1.7776m	1.9830m
Surface moyenne	35.8408m ²	35.8411m ²	41.8440m ²

TAB. III.1: Comparaison des différentes versions de la méthode ALCA.

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

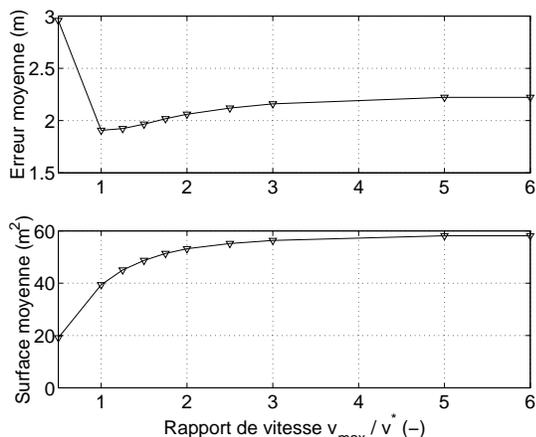


FIG. III.14: Variation de l'erreur d'estimation et de la surface des pavés en fonction du paramètre v_{max} .

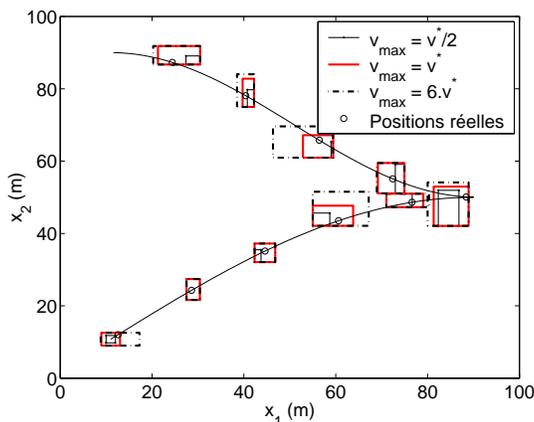


FIG. III.15: Pavés estimés avec différentes valeurs du paramètre v_{max} .

pour la trajectoire de base, $v^* = 2.0343m.s^{-1}$. Pour illustrer la dépendance de la méthode à la valeur de la vitesse, nous varions v_{max} en fonction de v^* tel que $v_{max} \in v^* * \{0.5, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 5, 6\}$. FIG. III.14 illustre les variations de l'erreur d'estimation moyenne et de la surface moyenne en fonction de la valeur de v_{max} . La figure montre que la moindre erreur est obtenue pour $v_{max} = v^*$. Pour $v_{max} < v^*$, nous obtenons des pavés plus petits mais ne contenant pas la vraie position pour plusieurs pas de temps. La garantie est alors perdue. Pour $v_{max} > v^*$, l'erreur et la précision de l'estimation augmente progressivement avec l'augmentation de v_{max} , jusqu'à la valeur $5.v^*$, à partir de laquelle les résultats de la localisation ne changent plus. Dans ces cas, le modèle de mobilité induit une région assez large pour contenir la zone d'intersection des disques d'observation sans la réduire. FIG. III.15 illustre les pavés obtenus avec la méthode ALCA pour $v_{max} = \frac{v^*}{2}$, $v_{max} = v^*$ et $v_{max} = 6.v^*$. Il est clair que pour $v_{max} = \frac{v^*}{2}$, la garantie est perdue dans la plupart des pas temps, alors que pour $v_{max} = 6.v^*$, les pavés sont larges impliquant une perte au niveau de la précision.

3. Impact de la valeur accordée à la portée de communication

Dans la méthode que nous proposons, le nœud estime sa position en utilisant des mesures de proximité vis-à-vis des ancrs. Une ancre est considérée à proximité du nœud si la puissance du signal émis par l'ancre et reçu par le nœud est supérieure à un certain seuil ρ_r . Ce seuil est fixé en fonction de la portée de communication r . Pour toute ancre détectée, une équation limitant la distance entre l'ancre et le nœud à r est obtenue. Une erreur au niveau de la valeur de r peut affecter les performances de la méthode. Soit r^* la vraie valeur de la portée correspondant au seuil ρ_r et soit r le paramètre que nous prenons à l'entrée de l'algorithme. Nous fixons r^* à $10m$ et nous varions r de telle sorte que r prenne les valeurs $\{6m, \dots, 10m, \dots, 14m\}$. Le paramètre vitesse maximale v_{max} prend sa

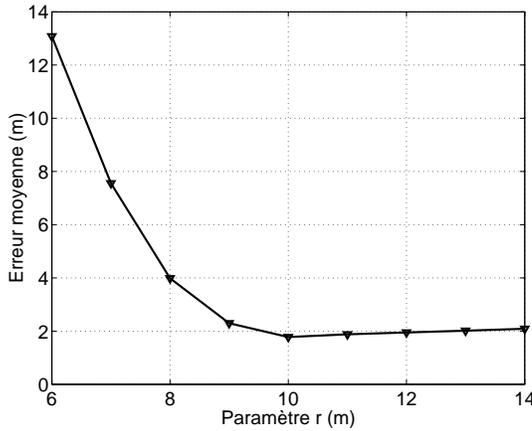


FIG. III.16: Courbe de l'erreur d'estimation en fonction du paramètre r .

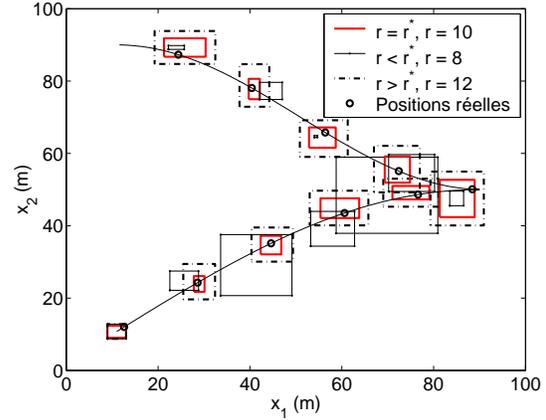


FIG. III.17: Pavés estimés avec différentes valeurs du paramètre r .

vraie valeur $2.0343m.s^{-1}$. FIG. III.16 montre la courbe de l'erreur d'estimation moyenne en fonction de la valeur de r . Le meilleur résultat est obtenu pour $r = r^*$. Pour $r < r^*$, les mesures peuvent induire des contraintes erronées, les disques leur correspondant ne couvrant pas la position réelle. Ceci conduit à la perte de garantie, d'où l'obtention d'une erreur d'estimation très élevée. Pour $r > r^*$, les mesures produisent des disques plus grands conduisant à une réduction de la précision. L'erreur d'estimation augmente progressivement mais moins vite qu'en cas de perte de garantie. FIG. III.17 illustre les pavés obtenus avec la méthode ALCA pour $r = 8m$, $r = 10m$ et $r = 12m$. Il est clair que pour $r = 8m$, la garantie est souvent perdue, alors que pour $r = 12m$, les pavés sont larges impliquant une réduction de la précision. Notons que pour $r < r^*$, le domaine initial résultant de la propagation du pavé précédent est adopté en cas d'intersection vide avec les mesures d'observation.

4. *Comparaison de la méthode ALCA à une approche de Monte-Carlo*

Dans ce paragraphe, nous comparons notre méthode à une technique existante basée sur l'approche séquentielle de Monte-Carlo [Baggio et Langendoen, 2006]. Cette dernière, nommée MCB pour *Monte-Carlo Boxed localization*, génère à chaque instant un nombre fixe de particules N satisfaisant toutes les contraintes. L'algorithme MCB comprend deux étapes : (i) la prédiction de particules, engendrées à l'intérieur de la zone d'intersection des pavés obtenus à l'aide des modèles simplifiés de mobilité et d'observation, et (ii) le filtrage des particules en ne gardant que celles qui satisfont les équations des disques. Ces deux étapes sont itérées jusqu'à ce que N particules soient gardées. La méthode MCB arrête le filtrage dès que le nombre d'itérations dépasse une certaine valeur, que nous fixons à 200. Les paramètres r et v_{max} sont attribués leurs vraies valeurs respectives $10m$ et $2.0343m.s^{-1}$. FIG. III.18 et FIG. III.19 illustrent les particules obtenues avec la méthode MCB respectivement pour $N = 50$ et $N = 500$, ainsi que les pavés

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

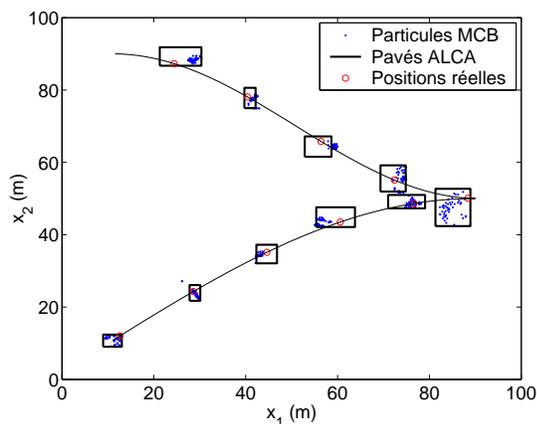


FIG. III.18: Estimation à l'aide de la méthode ALCA et la méthode MCB avec $N = 50$.

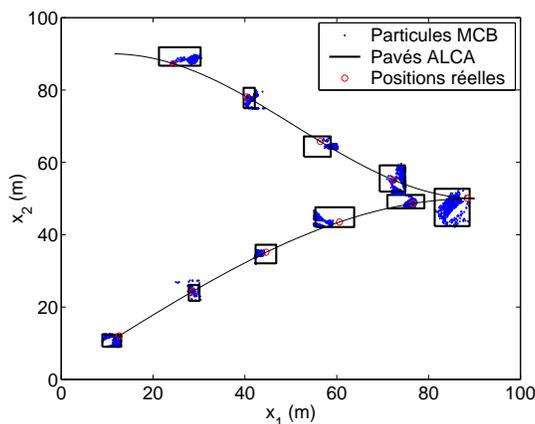


FIG. III.19: Estimation à l'aide de la méthode ALCA et la méthode MCB avec $N = 500$.

Méthode	ALCA	MCB $N = 50$	MCB $N = 500$
Temps de calcul	0.00655s	0.03354s	0.26333s
Erreur moyenne	1.7776m	2.9565m	2.8808m

TAB. III.2: Comparaison de la méthode ALCA à la méthode MCB basée sur Monte-Carlo.

obtenus avec la méthode ALCA. La figure montre qu'avec 50 particules, la position réelle n'est pas couverte pour la plupart des pas de temps ; alors qu'avec 500 particules, la zone solution est mieux définie au prix de la grande consommation de mémoire et du temps de calcul élevé. TAB. III.2 regroupe les temps de calcul moyens et les erreurs d'estimation moyennes par pas de temps correspondant à chacune des méthodes. Notons que l'erreur d'estimation pour MCB est égale à la distance entre la position réelle et le barycentre des particules. Les résultats obtenus illustrent bien les performances de la méthode que nous avons proposée comparée à la méthode MCB que ce soit au niveau du temps de calcul, de la mémoire consommée ou de l'erreur d'estimation.

III.8.2 Evaluation de la méthode *back-propagated*

La méthode *back-propagated* ALBP représente une extension de la méthode ALCA, où les pavés estimés à l'instant courant sont utilisés pour corriger les pavés précédents. Les simulations montrent que la rétropropagation permet de contracter les pavés antérieurs. Toutefois, le fait de repropager les pavés contractés vers l'instant présent n'a pas d'effet sur le pavé courant. Dans cette section, nous comparons la méthode ALBP à la méthode ALCA. Soit K_{BP} le nombre maximal de pas temps passés que l'algorithme peut atteindre à l'aide du processus de rétropropagation-correction. Nous varions en premier la valeur de K_{BP} de 1 jusqu'à 15 et nous illustrons les variations de la sur-

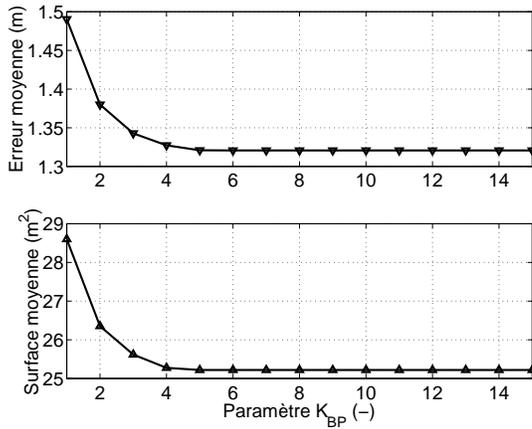


FIG. III.20: Variations de la surface moyenne des pavés et de l'erreur moyenne en fonction de K_{BP} avec la méthode ALBP.

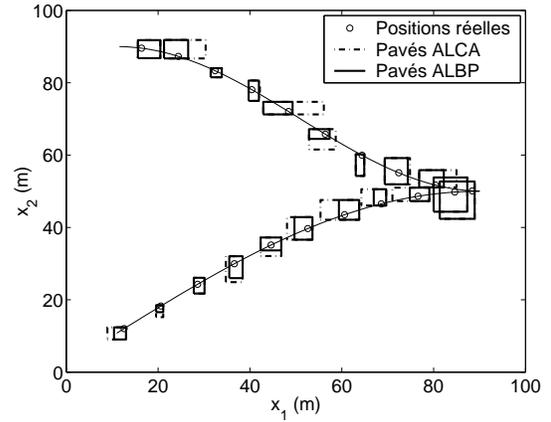


FIG. III.21: Pavés obtenus à l'aide de la méthode ALCA et la méthode ALBP avec $K_{BP} = 5$.

Méthode	ALCA	ALBP, $K_{BP} = 5$
Temps de calcul	0.00655s	0.00763s
Erreur moyenne	1.7776m	1.3504m
Surface moyenne	38.8411m ²	25.2197m ²

TAB. III.3: Comparaison de la méthode ALCA à la méthode ALBP avec $K_{BP} = 5$.

face moyenne des pavés et de l'erreur d'estimation moyenne en fonction de K_{BP} (cf. FIG. III.20). La figure montre qu'en moyenne, la rétropropagation n'a plus d'effet à partir du 5-ème instant. Il suffit alors de fixer K_{BP} à 5 pour cet exemple. FIG. III.21 montre les pavés que nous obtenons à l'aide des méthodes ALCA et ALBP pour $K_{BP} = 5$. Dans cette section, les ancres sont fixes, $v_{max} = 2.0343m.s^{-1}$ et $r = 10m$. Les temps de calcul moyens, les surfaces moyennes et les erreurs d'estimation moyennes par pas de temps sont regroupés dans TAB. III.3. Les résultats montrent que la méthode ALBP assure un gain important dans la précision de l'estimation pour une faible augmentation dans le temps de calcul. La méthode ALBP peut être alors privilégiée dans les applications ne présentant pas des contraintes temps réel.

III.8.3 Evaluation de la méthode *multi-hop*

Dans cette section, nous évaluons la méthode *multi-hop*, notée ALMH. Pour cette raison, nous la comparons à la méthode ALCA. Nous considérons alors la même configuration d'ancres que précédemment : 120 ancres fixes aléatoirement distribuées dans la zone de déploiement. Les paramètres v_{max} et r sont fixés respectivement à $2.0343m.s^{-1}$ et $10m$. Nous rappelons qu'une ancre est dite *k-hop* quand sa distance au nœud considéré est comprise entre $(k - 1).r$ et $k.r$. Soit K_{MH} le nombre maximal k sur toutes

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

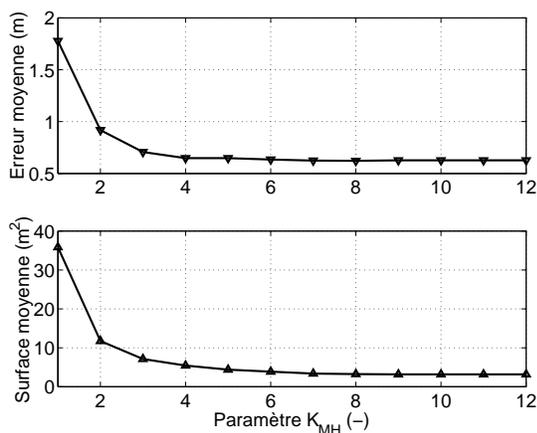


FIG. III.22: Variations de la surface moyenne et de l'erreur moyenne en fonction de K_{MH} avec la méthode ALMH.

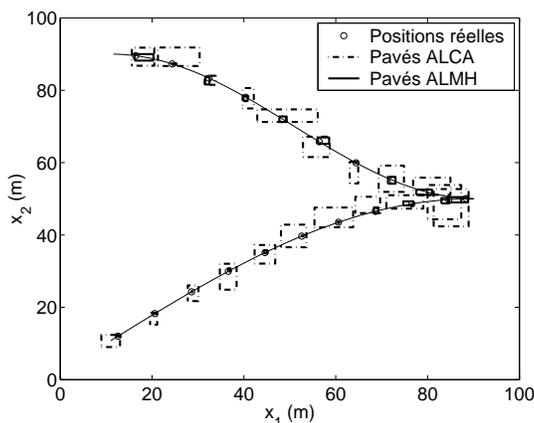


FIG. III.23: Pavés obtenus à l'aide de la méthode ALCA et la méthode ALMH avec $K_{MH} = 7$.

Méthode	ALCA	ALMH, $K_{MH} = 7$
Temps de calcul	0.00655s	0.17846s
Erreur moyenne	1.7776m	0.6241m
Surface moyenne	38.8411m ²	3.3820m ²

TAB. III.4: Comparaison de la méthode ALCA à la méthode ALMH avec $K_{MH} = 7$.

les ancres k -hop détectées. K_{MH} est fonction de la puissance des signaux à l'émission, de la dégradation dans le canal, de la distribution des ancres dans le réseau, etc. Nous limitons en premier le nombre maximal de hops K_{MH} considéré dans la méthode. Ceci limite le nombre d'ancres mises en jeu dans l'estimation de la position du nœud. FIG. III.22 illustre l'erreur d'estimation moyenne et la surface moyenne des pavés estimés en fonction de K_{MH} , $K_{MH} \in \{1, \dots, 12\}$. Il est clair que la qualité de l'estimation s'améliore avec l'augmentation du nombre d'ancres concernées. La figure montre également qu'à partir de $K_{MH} = 7$, les résultats ne changent plus. Ceci est dû au fait que pour $K_{MH} = 7$, toutes les ancres sont employées. FIG. III.23 illustre les pavés obtenus avec la méthode ALMH, $K_{MH} = 7$, et la méthode ALCA, équivalente à la méthode ALMH avec $K_{MH} = 1$. TAB. III.4 regroupe l'erreur moyenne, le temps de calcul et la surface moyenne des pavés correspondant aux deux méthodes. Les résultats démontrent bien l'efficacité de la méthode proposée, surtout au niveau de la précision de l'estimation, au prix du temps de calcul.

III.8.4 Evaluation de la méthode utilisant les non-ancres

La méthode d'auto-localisation évaluée dans cette section repose sur les mesures de connectivités effectuées vis-à-vis des ancres et des non-ancres. Elle sera notée ALCAN. En premier lieu, nous comparons cette méthode à la méthode ALCA, montrant ainsi

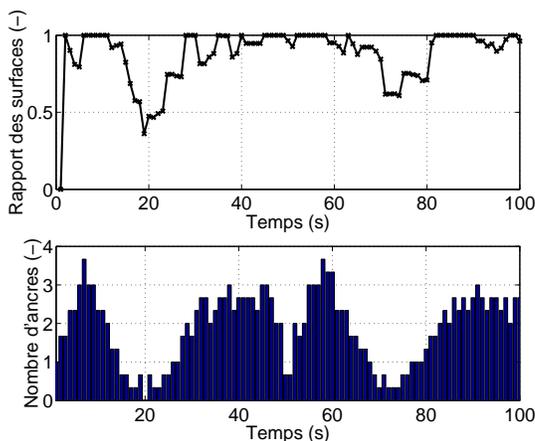


FIG. III.24: Comparaison de la méthode ALCAN à la méthode ALCA.

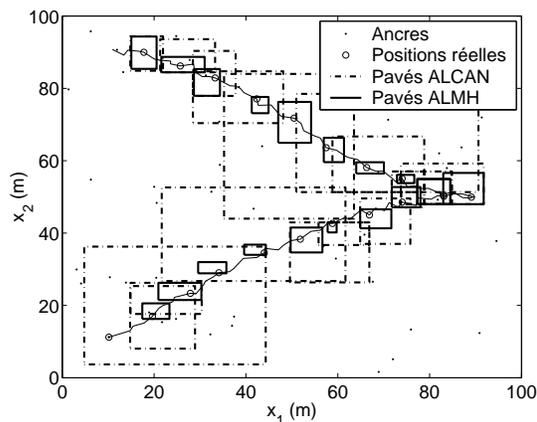


FIG. III.25: Comparaison de la méthode ALCAN à la méthode ALMH.

l'intérêt de l'utilisation des non-ancres voisins. Nous considérons alors 3 nœuds et 50 ancres fixes aléatoirement distribuées dans la zone de déploiement. Les nœuds, au voisinage l'un de l'autre, se déplacent selon un mouvement de groupe suivant la trajectoire de base. FIG. III.24 illustre le rapport moyen par nœud des surfaces des pavés obtenus avec les méthodes ALCA et ALCAN en fonction du temps. Elle illustre également le nombre moyen d'ancres situées au voisinage des nœuds en fonction du temps. La figure montre que la méthode ALCAN induit un gain au niveau de la précision de l'estimation surtout aux instants où le nombre des ancres voisins est faible. Notons qu'en moyenne, l'algorithme effectue 1.75 itérations pour réaliser la correction totale à chaque instant.

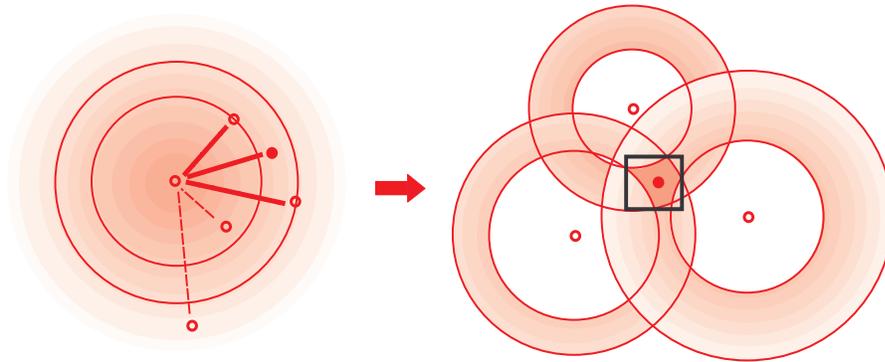
Nous comparons ensuite la méthode ALCAN à la méthode *multi-hop* ALMH, toutes les deux étant efficaces dans les réseaux à faibles densités en ancres. Nous choisissons $K_{MH} = 3$ pour la méthode ALMH. En d'autres termes, les ancres utilisées dans l'algorithme sont les ancres *1-hop*, *2-hop* et *3-hop*. FIG. III.25 illustrent les pavés obtenus à l'aide des méthodes ALCAN et ALMH pour l'un des nœuds mobiles. La figure montre que la méthode ALMH engendre beaucoup plus de précision avec un gain de 89% dans la surface des pavés par rapport à la méthode ALCAN. Toutefois, il faut noter que la méthode ALMH ($K_{MH} = 3$) nécessite un temps de calcul 4 fois plus que le temps mis par ALCAN (0.00448s par nœud par pas de temps pour ALCAN et 0.019263s par nœud par pas de temps pour ALMH, $K_{MH} = 3$). Notons également que cette différence dépend énormément de la configuration des capteurs dans le réseau.

III.9 Conclusion

Dans ce chapitre, nous avons proposé une méthode d'auto-localisation basée sur des mesures de connectivités effectuées vis-à-vis des ancres. Nous avons également présenté diverses extensions de cette méthode : la méthode *back-propagated*, effectuant une rétropropagation pour corriger les pavés antérieurs, la méthode *multi-hop*, utilisant des

III. AUTO-LOCALISATION PAR CONNECTIVITÉS

ancres plus ou moins éloignées, et la méthode basée sur les ancres et les non-ancres. Les techniques proposées utilisent un modèle de mobilité général afin de raffiner l'estimation. La résolution du problème se situe dans le cadre de la théorie des intervalles. Les positions estimées sont ainsi des pavés bidimensionnels englobant toutes les solutions possibles du problème. Avec cette approche à erreurs bornées, nous fournissons une méthode garantie, la position réelle se situant toujours à l'intérieur du pavé estimé. Les résultats de simulation illustrent bien les performances des méthodes proposées, comparées aux méthodes basées sur l'approche de Monte-Carlo. Toutefois, une erreur au niveau de la génération des mesures de connectivités peut entraîner une perte de la garantie. Dans le chapitre suivant, nous introduisons deux méthodes indépendantes du modèle de connectivités, permettant de remédier à ce problème.



Sommaire

IV.1 Introduction

IV.2 Problématique

IV.3 Auto-localisation sans modèle basée sur les ancrés

IV.4 Auto-localisation sans modèle utilisant les non-ancrés

IV.5 Simulations

IV.6 Conclusion

Dans ce chapitre, nous présentons deux méthodes d'auto-localisation sans modèle. Ces méthodes, également basées sur les intervalles, comparent les puissances des signaux reçus afin de définir des anneaux d'observation. Selon cette approche, les nœuds n'ont plus besoin d'estimer les paramètres du modèle de perte du canal, dit aussi modèle d'Okumura-Hata. Cette approche est ainsi plus robuste que celle présentée dans le chapitre précédent. Les positions estimées sont des pavés bidimensionnels, généralement plus précis que ceux obtenus à l'aide des mesures de connectivité. Outre les anneaux d'observation, les méthodes proposées tiennent compte de la mobilité des nœuds. La première méthode que nous présentons se base uniquement sur les ancrés. Nous proposons ensuite une extension de cette méthode pour aussi tenir compte des capteurs non-ancrés dans la génération des anneaux.

IV.1 Introduction

Une localisation fiable et précise des capteurs est un besoin essentiel pour toutes les applications des réseaux de capteurs mobiles. En effet, les données récoltées sont énormément liées aux emplacements où sont effectuées les mesures. Les méthodes d'auto-localisation proposées dans le chapitre III se basent sur le modèle de perte du canal, dit modèle d'Okumura-Hata [Medeisis et Kajackas, 2000; Nadir *et al.*, 2008]. Ce modèle à paramètres inconnus fournit une relation entre la puissance des signaux reçus et les distances parcourues par ces signaux. Une erreur dans l'estimation des paramètres pourra entraîner une erreur dans l'estimation de la position des nœuds. En d'autres termes, les performances des méthodes proposées dans le chapitre III dépendent du paramétrage du modèle.

Dans ce chapitre, nous proposons une approche permettant de remédier à ce problème. Celle-ci effectue une comparaison des puissances des signaux reçus afin de générer les observations [Liu *et al.*, 2004; Li, 2006; Nabaee et Olfat, 2008]. L'avantage principal de cette approche est qu'elle est indépendante de la valeur de la portée de communication. L'idée de base a été inspirée du travail de Liu *et al.* [Liu *et al.*, 2004], où ils effectuent une comparaison des puissances des signaux pour construire des anneaux. Selon ce travail, la position estimée est définie par le centre de gravité de l'intersection des anneaux. Cette méthode est alors statique ne tenant pas compte de la mobilité des capteurs. Contrairement à la méthode de Liu *et al.*, l'approche que nous proposons profite de la mobilité des capteurs pour définir un modèle de mobilité. Associé aux observations générées, ce modèle de mobilité permet de raffiner le problème d'auto-localisation. La résolution du problème se situe dans le cadre de l'analyse par intervalles, où chaque donnée est remplacée par l'intervalle auquel elle appartient [Moore, 1966, 1979; Jaulin *et al.*, 2001]. La solution proposée est alors un pavé bidimensionnel, contenant sûrement la position réelle.

La première méthode que nous présentons se base uniquement sur les ancrés pour générer les observations. La deuxième tient compte de toutes les informations possibles, y compris celles associées aux non-ancres. Celle-ci est intéressante dans les réseaux à faibles densités d'ancres. Dans la suite, nous exposons en premier la problématique générale où nous nous basons sur les ancrés. Nous proposons ensuite la première méthode où nous résolvons le problème posé. Nous présentons par la suite la deuxième méthode, celle-ci étant une extension de la première. Nous exposons enfin les résultats de simulation illustrant les performances de cette approche par rapport à celle proposée dans le chapitre III.

IV.2 Problématique

L'approche proposée distingue deux types de capteurs : les ancrés, munies de récepteurs GPS (pour *Global Positioning System*) [Hofmann-Wellenhof *et al.*, 2004], et les non-ancres, dits aussi nœuds, ayant des positions inconnues. Le problème consiste alors à estimer les positions des nœuds en se basant sur les positions des ancrés, celles-ci étant

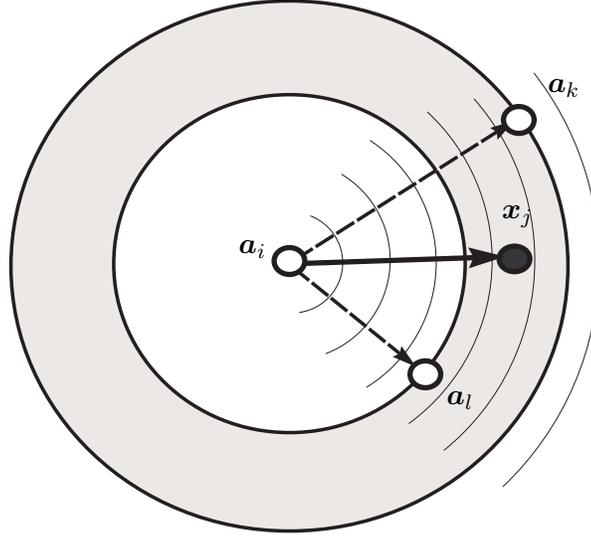


FIG. IV.1: Génération d'un anneau.

associées à des informations de mesures. Selon cette approche, les observations sont générées suite à une comparaison des puissances des signaux échangés entre les capteurs. Les nœuds profitent aussi de leur mobilité pour raffiner le problème d'estimation. Dans la suite, nous commençons par définir les mesures d'observation, suivies par le modèle de mobilité. Nous résumons ensuite le problème.

IV.2.1 Mesures d'observation

L'approche que nous proposons consiste à générer des anneaux centrés sur les ancres. Considérons le problème d'auto-localisation d'un nœud j , $j \in \{1, \dots, N_x\}$. Un anneau centré sur une ancre i est obtenu en comparant la puissance du signal émis par l'ancre i et reçu par le nœud j aux puissances des signaux émis par i et reçus par les autres ancres. Cette technique s'appuie sur le fait que la puissance du signal émis décroît d'une façon monotone avec l'accroissement de la distance parcourue. Soient \mathbf{a}_i , \mathbf{a}_k et \mathbf{a}_l trois ancres et soit \mathbf{x}_j le nœud considéré. Alors,

$$\text{si } \rho_{\mathbf{a}_i, \mathbf{a}_k} \leq \rho_{\mathbf{a}_i, \mathbf{x}_j} \leq \rho_{\mathbf{a}_i, \mathbf{a}_l}, \text{ alors } d(\mathbf{a}_i, \mathbf{a}_k) \geq d(\mathbf{a}_i, \mathbf{x}_j) \geq d(\mathbf{a}_i, \mathbf{a}_l), \quad (\text{IV.1})$$

où $\rho_{\mathbf{a}_i, \mathbf{a}_k}$, $\rho_{\mathbf{a}_i, \mathbf{x}_j}$ et $\rho_{\mathbf{a}_i, \mathbf{a}_l}$ sont respectivement les puissances des signaux émis par l'ancre \mathbf{a}_i et reçus par les capteurs \mathbf{a}_k , \mathbf{x}_j et \mathbf{a}_l et $d(\mathbf{a}_i, \mathbf{a}_k)$, $d(\mathbf{a}_i, \mathbf{x}_j)$ et $d(\mathbf{a}_i, \mathbf{a}_l)$ sont les distances respectives entre l'ancre \mathbf{a}_i et les capteurs \mathbf{a}_k , \mathbf{x}_j et \mathbf{a}_l . La relation (IV.1) permet de définir un anneau ayant comme centre l'ancre \mathbf{a}_i et comme rayons interne $d(\mathbf{a}_i, \mathbf{a}_l)$ et externe $d(\mathbf{a}_i, \mathbf{a}_k)$. Une illustration d'un tel anneau est donnée dans FIG. IV.1.

Soit $I_j(t)$ l'ensemble des indices des ancres pouvant communiquer avec le nœud j à l'instant t , $j \in \{1, \dots, N_x\}$, et soit $J_i(t)$ l'ensemble des indices des ancres communiquant

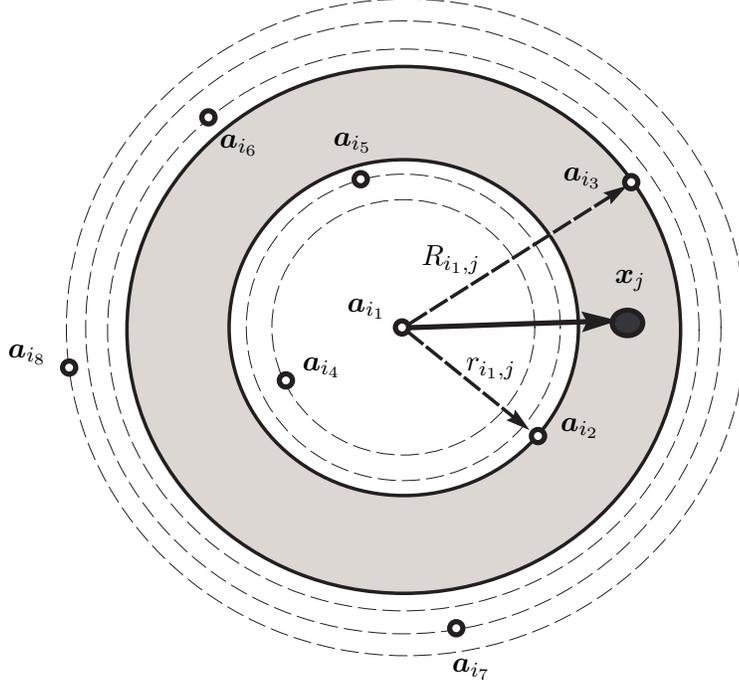


FIG. IV.2: Définition des rayons interne et externe d'un anneau.

avec l'ancre i à l'instant t , $i \in I_j(t)$. A chaque instant t , chacune des ancres i communiquant avec le nœud j émet des signaux dans le réseau. Ces signaux seront reçus par le nœud j et les ancres de $J_i(t)$ avec des puissances différentes. La puissance $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)$ au nœud j sera comparée aux autres puissances inter-ancres $\rho_{\mathbf{a}_i, \mathbf{a}_k}(t)$, $k \in J_i(t)$. Toute ancre $k \in J_i(t)$, telle que $\rho_{\mathbf{a}_i, \mathbf{a}_k} \leq \rho_{\mathbf{a}_i, \mathbf{x}_j}$, fournit une distance $d(\mathbf{a}_i, \mathbf{a}_k)$ supérieure à $d(\mathbf{a}_i, \mathbf{x}_j)$. De la même manière, toute ancre $k \in J_i(t)$, telle que $\rho_{\mathbf{a}_i, \mathbf{a}_k} \geq \rho_{\mathbf{a}_i, \mathbf{x}_j}$, fournit une distance $d(\mathbf{a}_i, \mathbf{a}_k)$ inférieure à $d(\mathbf{a}_i, \mathbf{x}_j)$. Ainsi, l'anneau le plus mince contenant le nœud j et centré sur l'ancre i a comme rayons

$$\begin{cases} r_{i,j} = d(\mathbf{a}_i, \mathbf{a}_{k_m}) \text{ tel que } k_m = \arg\{\min_{k \in J_i(t)} \{\rho_{\mathbf{a}_i, \mathbf{a}_k}(t) \mid \rho_{\mathbf{a}_i, \mathbf{a}_k}(t) \geq \rho_{\mathbf{a}_i, \mathbf{x}_j}(t)\}\}, \\ R_{i,j} = d(\mathbf{a}_i, \mathbf{a}_{k_M}) \text{ tel que } k_M = \arg\{\max_{k \in J_i(t)} \{\rho_{\mathbf{a}_i, \mathbf{a}_k}(t) \mid \rho_{\mathbf{a}_i, \mathbf{a}_k}(t) \leq \rho_{\mathbf{a}_i, \mathbf{x}_j}(t)\}\}, \end{cases} \quad (\text{IV.2})$$

où $r_{i,j}$ et $R_{i,j}$ sont respectivement les rayons interne et externe de l'anneau d'observation centré sur l'ancre i . Le message d'observation que l'ancre i transmettra au nœud j contiendra alors les deux rayons interne et externe ainsi que ses propres coordonnées. Une illustration est donnée dans FIG. IV.2.

Le nœud j , recevant des messages d'observation des ancres $i \in I_j(t)$, construit des contraintes sur sa position comme suit,

$$r_{i,j} \leq d(\mathbf{a}_i(t), \mathbf{x}_j(t)) \leq R_{i,j}, i \in I_j(t). \quad (\text{IV.3})$$

Ces contraintes d'observation seront explicitées de la manière suivante,

$$r_{i,j}^2 \leq (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq R_{i,j}^2, i \in I_j(t). \quad (\text{IV.4})$$

Le nœud j appartient ainsi à la région résultant de l'intersection de tous les anneaux. Notons que si l'ancre i ne trouve pas d'ancre plus éloignée que le nœud j , alors $R_{i,j}$ est pris égal à l'infini et la contrainte de supériorité sera éliminée. De la même manière, si l'ancre i ne trouve pas d'ancre plus proche que le nœud j , alors $r_{i,j}$ est pris égal à zéro et la contrainte d'infériorité sera éliminée. Dans ce cas, l'anneau se réduira à un simple disque.

IV.2.2 Modèle de mobilité

Bien que la mobilité semble rendre la localisation plus difficile, nous pouvons profiter de cette mobilité pour améliorer l'estimation des positions. Nous proposons ainsi un modèle de mobilité que nous associerons aux observations. L'objectif de nos méthodes exige un modèle assez général afin de pouvoir adapter nos méthodes à un plus grand nombre d'applications. Similairement au chapitre III, nous nous sommes inspirés du modèle de marche aléatoire [Camp *et al.*, 2002] pour proposer un modèle où seule la vitesse maximale du nœud est supposée connue. Selon ce modèle, les nœuds ont la possibilité de bouger dans toute direction, avec une vitesse quelconque mais inférieure à la vitesse maximale. Toutes les applications où les vitesses des nœuds sont limitées à une certaine valeur sont couvertes par ce modèle. Soit $\mathbf{x}_j(t-1) = (x_{j,1}(t-1), x_{j,2}(t-1))$ la position du nœud j à l'instant $t-1$. La position du nœud considéré à l'instant suivant est donnée comme suit,

$$\begin{cases} x_{j,1}(t) = x_{j,1}(t-1) + \Delta t \cdot v_j \cdot \cos(\theta_j), \\ x_{j,2}(t) = x_{j,2}(t-1) + \Delta t \cdot v_j \cdot \sin(\theta_j), \end{cases} \quad (\text{IV.5})$$

où θ_j et v_j sont des variables respectivement comprises entre 0 et 2π et 0 et $v_{max,j}$, $v_{max,j}$ étant la vitesse maximale du nœud j . Δt est la période de localisation. Ce modèle peut être reformulé de la manière suivante,

$$(x_{j,1}(t) - x_{j,1}(t-1))^2 + (x_{j,2}(t) - x_{j,2}(t-1))^2 = (\Delta t \cdot v_j)^2. \quad (\text{IV.6})$$

Si l'on suppose que la position $\mathbf{x}_j(t-1)$ estimée à l'instant $t-1$ est ponctuelle, la position courante appartiendra au disque ayant $v_{max,j}$ comme rayon et $\mathbf{x}_j(t-1)$ comme centre.

IV.2.3 Problème d'auto-localisation

Le problème d'auto-localisation est défini à la fois par le modèle de mobilité et les mesures d'observations. Ainsi, à chaque instant t , un nœud j reçoit des messages d'observation de toutes les ancres pouvant communiquer avec lui. Si $I_j(t)$ est l'ensemble des indices de ces ancres, les anneaux résultants sont construits comme suit,

$$r_{i,j}^2 \leq (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq R_{i,j}^2, i \in I_j(t), \quad (\text{IV.7})$$

IV. AUTO-LOCALISATION SANS MODÈLE

où $\mathbf{x}_j(t) = (x_{j,1}(t), x_{j,2}(t))$ représente les coordonnées inconnues du nœud j à l'instant t , $\mathbf{a}_i(t) = (a_{i,1}(t), a_{i,2}(t))$ est le vecteur coordonnées de l'ancre i et $r_{i,j}$ et $R_{i,j}$ sont respectivement les rayons interne et externe de l'anneau numéro i , centré sur l'ancre i et contenant le nœud j . Le problème d'auto-localisation est alors défini par le système suivant,

$$\begin{cases} (x_{j,1}(t) - x_{j,1}(t-1))^2 + (x_{j,2}(t) - x_{j,2}(t-1))^2 = (\Delta t \cdot v_j)^2, \\ \text{avec } r_{i,j}^2 \leq (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq R_{i,j}^2, i \in I_j(t), \\ \text{et } v_j \in [0, v_{max,j}], \end{cases} \quad (\text{IV.8})$$

où $v_{max,j}$ est la vitesse maximale du nœud j , v_j est une variable de vitesse inférieure à $v_{max,j}$ et Δt est la durée entre deux pas de temps consécutifs.

IV.3 Auto-localisation sans modèle basée sur les ancres

Dans cette section, nous proposons une méthode d'auto-localisation basée sur la comparaison des puissances des signaux échangés, notée ALAA (pour Auto-Localisation utilisant des Anneaux employant des Ancres). Basée uniquement sur les ancres, cette technique résout le problème posé dans la section IV.2. Dans la suite, nous exposons l'algorithme de résolution du problème dans le contexte ensembliste. Nous présentons ensuite des versions simplifiées de l'algorithme.

IV.3.1 Algorithme

La résolution du problème (IV.8) d'une manière garantie exige l'appel aux intervalles. Ainsi, les variables seront remplacées par les intervalles auxquels elles appartiennent et les fonctions par des fonctions d'inclusion leur correspondant. Les positions des nœuds $\mathbf{x}_j(t)$, en particulier, seront remplacées par des pavés bidimensionnels, notés $[\mathbf{x}_j](t) = [x_{j,1}](t) \times [x_{j,2}](t)$. Résoudre le problème consiste alors à le définir comme un problème de satisfaction de contraintes, qui sera contracté par la suite à l'aide de l'algorithme de Waltz.

Dans le contexte ensembliste, l'équation de mobilité sera reformulée de la manière suivante,

$$([x_{j,1}](t) - [x_{j,1}](t-1))^2 + ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, (\Delta t \cdot v_{max,j})^2]. \quad (\text{IV.9})$$

Similairement, les contraintes d'observation s'écrivent de la manière suivante,

$$([x_{j,1}](t) - a_{i,1}(t))^2 + ([x_{j,2}](t) - a_{i,2}(t))^2 = [r_{i,j}^2, R_{i,j}^2], i \in I_j(t). \quad (\text{IV.10})$$

La résolution du problème comprend deux étapes : la propagation, où un domaine initial est généré, et la correction, où ce domaine initial est contracté à l'aide de l'algorithme de Waltz. Dans la première étape, il s'agit de propager le pavé précédent à l'aide du modèle de mobilité. Le pavé résultant définit le domaine initial $[\mathbf{x}_j]^*(t)$ comme suit,

$$\begin{cases} [x_{j,1}]^*(t) = [x_{j,1}](t-1) + [-\Delta t \cdot v_{max,j}, +\Delta t \cdot v_{max,j}], \\ [x_{j,2}]^*(t) = [x_{j,2}](t-1) + [-\Delta t \cdot v_{max,j}, +\Delta t \cdot v_{max,j}]. \end{cases} \quad (\text{IV.11})$$

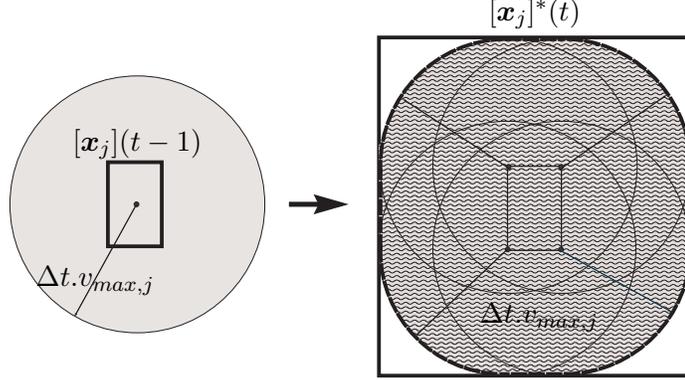


FIG. IV.3: Etape de propagation.

Voir le chapitre III, section III.4.3, pour plus de détails sur l'obtention de ce domaine. Une illustration de l'étape de propagation est donnée dans FIG. IV.3

Le domaine initial $[x_j]^*(t)$ sera contracté avec l'algorithme de Waltz, à l'aide des contraintes de mobilité et d'observation. Pour cette raison, nous définissons en premier les sous-solveurs finis issus du modèle de mobilité comme suit (voir la preuve III.2),

$$[\phi_{j,1}] = [x_{j,1}](t-1) + [-\overline{bm}_{j,1}(t), +\overline{bm}_{j,1}(t)], \quad (\text{IV.12})$$

$$[\phi_{j,2}] = [x_{j,2}](t-1) + [-\overline{bm}_{j,2}(t), +\overline{bm}_{j,2}(t)], \quad (\text{IV.13})$$

avec

$$\begin{cases} \overline{bm}_{j,1}(t) = \sqrt{(\Delta t \cdot v_{max,j})^2 - \inf((x_{j,2}(t) - x_{j,2}(t-1))^2)}, \\ \overline{bm}_{j,2}(t) = \sqrt{(\Delta t \cdot v_{max,j})^2 - \inf((x_{j,1}(t) - x_{j,1}(t-1))^2)}. \end{cases}$$

La contraction du problème à l'aide des sous-solveurs de mobilité s'effectue comme suit,

$$\begin{cases} [x_{j,1}](t) = [x_{j,1}](t) \cap [\phi_{j,1}], \\ [x_{j,2}](t) = [x_{j,2}](t) \cap [\phi_{j,2}]. \end{cases} \quad (\text{IV.14})$$

Si nous considérons les contraintes d'observation, celles-ci définissent des anneaux centrés sur les ancrés détectés. La position réelle du nœud tombera alors dans la zone d'intersection des anneaux. Pour un anneau donné i , si nous prenons séparément les cercles le limitant, le nœud considéré tombera à la fois à l'intérieur du disque de rayon $R_{i,j}$ et à l'extérieur du disque de rayon $r_{i,j}$. Chaque contrainte d'observation peut être décomposée en deux sous-contraintes, l'une correspondant à une équation de disque de rayon $R_{i,j}$ et l'autre effectuant une privation d'un disque de rayon $r_{i,j}$. Les deux sous-solveurs issus de la première sous-contrainte s'écrivent comme suit,

$$[\psi_{j,i,1}] = a_{i,1}(t) + [-\overline{bo}_{j,i,1}(t), +\overline{bo}_{j,i,1}(t)], \quad (\text{IV.15})$$

$$[\psi_{j,i,2}] = a_{i,2}(t) + [-\overline{bo}_{j,i,2}(t), +\overline{bo}_{j,i,2}(t)], \quad (\text{IV.16})$$

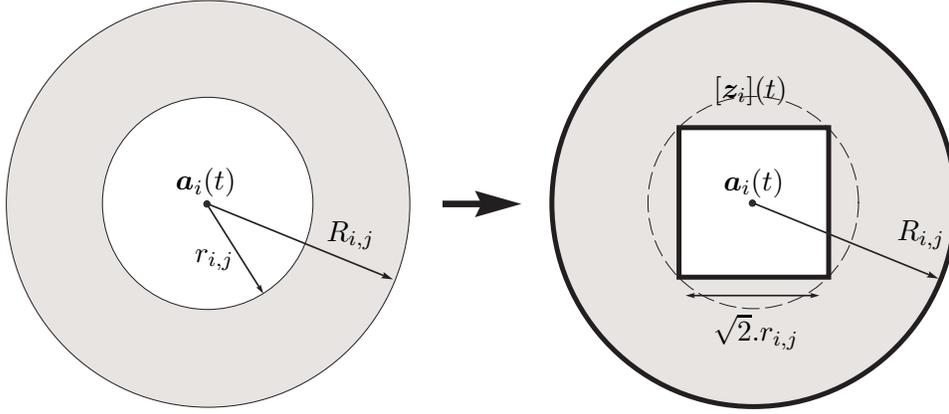


FIG. IV.4: Anneau d'observation dans le contexte ensembliste.

avec

$$\begin{cases} \overline{bo}_{j,i,1}(t) = \sqrt{R_{i,j}^2 - \inf((x_{j,2}(t) - a_{i,2}(t))^2)}, \\ \overline{bo}_{j,i,2}(t) = \sqrt{R_{i,j}^2 - \inf((x_{j,1}(t) - a_{i,1}(t))^2)}. \end{cases}$$

La contraction du problème à l'aide de ces deux sous-solveurs se fait de la manière suivante,

$$\begin{cases} [x_{j,1}](t) = [x_{j,1}(t) \cap [\psi_{j,i,1}], \\ [x_{j,2}](t) = [x_{j,2}(t) \cap [\psi_{j,i,2}]. \end{cases} \quad (\text{IV.17})$$

Afin d'appliquer la deuxième sous-contrainte dans le cadre des intervalles, il suffit de priver $[x_j](t)$ du carré inscrit dans le disque interne de l'anneau comme suit,

$$[x_j](t) = [x_j](t) \setminus [z_i](t), \quad (\text{IV.18})$$

avec $[z_i](t) = [z_{i,1}](t) \times [z_{i,2}](t)$ où

$$\begin{cases} [z_{i,1}](t) = [a_{i,1}(t) - \frac{\sqrt{2}}{2}.r_{i,j}, a_{i,1}(t) + \frac{\sqrt{2}}{2}.r_{i,j}], \\ [z_{i,2}](t) = [a_{i,2}(t) - \frac{\sqrt{2}}{2}.r_{i,j}, a_{i,2}(t) + \frac{\sqrt{2}}{2}.r_{i,j}]. \end{cases} \quad (\text{IV.19})$$

La longueur du côté du carré inscrit est alors égale à $\sqrt{2}.r_{i,j}$. Une illustration des contraintes d'observation est donnée dans FIG. IV.4. Le domaine obtenu contient le vrai anneau, permettant ainsi d'éviter la perte d'information probablement pertinente. Le pseudo-code de la méthode proposée est illustré dans l'algorithme 8. Une illustration de la méthode est également donnée dans FIG. IV.5.

IV.3.2 Versions simplifiées de l'algorithme

Deux versions simplifiées de la méthode ALAA peuvent être proposées. La première version représente une simplification partielle, où seul le modèle de mobilité est simplifié.

Algorithme 8 : Pseudo-code de la méthode d'auto-localisation sans modèle basée sur les ancres.

Entrées : $[\mathbf{x}_j](t-1)$, $I_j(t)$, $\mathbf{a}_i(t)$, $r_{i,j}$, $R_{i,j}$, $i \in I_j(t)$, $v_{max,j}$, Δt ;
Sorties : $[\mathbf{x}_j](t)$;

$$\begin{cases} [x_{j,1}]^*(t) = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]; \\ [x_{j,2}]^*(t) = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]; \end{cases}$$

$$\begin{aligned} [x_{j,1}](t) &= [x_{j,1}]^*(t), [x_{j,2}](t) = [x_{j,2}]^*(t); \\ A_j &= w([x_{j,1}](t)) * w([x_{j,2}](t)); \\ A_j^{old} &= A_j + 1; \end{aligned}$$

tant que $A_j < A_j^{old}$ **faire**
 $A_j^{old} = A_j$;

pour $i \in I_j(t)$ **faire**

$$[x_{j,2}](t) = [x_{j,2}](t) \cap [a_{i,2}(t) - R_{i,j}, a_{i,2}(t) + R_{i,j}];$$

$$\overline{bo}_{j,i,1}(t) = \sqrt{R_{i,j}^2 - \inf(([x_{j,2}](t) - a_{i,2}(t))^2)}$$

$$[\psi_{j,i,1}] = a_{i,1}(t) + [-\overline{bo}_{j,i,1}(t), +\overline{bo}_{j,i,1}(t)];$$

$$[x_{j,1}](t) = [x_{j,1}](t) \cap [\psi_{j,i,1}];$$

$$[x_{j,1}](t) = [x_{j,1}](t) \cap [a_{i,1}(t) - R_{i,j}, a_{i,1}(t) + R_{i,j}];$$

$$\overline{bo}_{j,i,2}(t) = \sqrt{R_{i,j}^2 - \inf(([x_{j,1}](t) - a_{i,1}(t))^2)}$$

$$[\psi_{j,i,2}] = a_{i,2}(t) + [-\overline{bo}_{j,i,2}(t), +\overline{bo}_{j,i,2}(t)];$$

$$[x_{j,2}](t) = [x_{j,2}](t) \cap [\psi_{j,i,2}];$$

$$[z_{i,1}](t) = [a_{i,1}(t) - \frac{\sqrt{2}}{2}.r_{i,j}, a_{i,1}(t) + \frac{\sqrt{2}}{2}.r_{i,j}];$$

$$[z_{i,2}](t) = [a_{i,2}(t) - \frac{\sqrt{2}}{2}.r_{i,j}, a_{i,2}(t) + \frac{\sqrt{2}}{2}.r_{i,j}];$$

$$[\mathbf{x}_j](t) = [\mathbf{x}_j](t) \setminus [\mathbf{z}_i](t);$$

fin

$$\overline{bm}_{j,1}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf(([x_{j,2}](t) - [x_{j,2}](t-1))^2)}$$

$$[\phi_{j,1}] = [x_{j,1}](t-1) + [-\overline{bm}_{j,1}(t), +\overline{bm}_{j,1}(t)];$$

$$[x_{j,1}](t) = [x_{j,1}](t) \cap [\phi_{j,1}];$$

$$\overline{bm}_{j,2}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf(([x_{j,1}](t) - [x_{j,1}](t-1))^2)}$$

$$[\phi_{j,2}] = [x_{j,2}](t-1) + [-\overline{bm}_{j,2}(t), +\overline{bm}_{j,2}(t)];$$

$$[x_{j,2}](t) = [x_{j,2}](t) \cap [\phi_{j,2}];$$

$$A_j = w([x_{j,1}](t)) * w([x_{j,2}](t));$$

fin

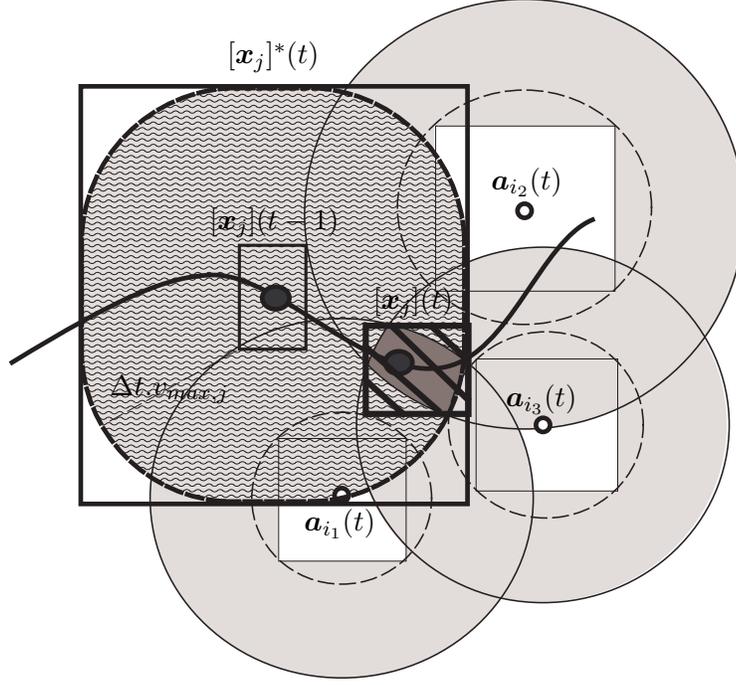


FIG. IV.5: Auto-localisation sans modèle basée sur les ancrés.

L'équation de disque de ce modèle sera remplacée par une équation de carré ayant $2.\Delta t.v_{max,j}$ comme côté. L'équation de mobilité est alors approximée comme suit,

$$\begin{cases} [x_{j,1}](t) = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}], \\ [x_{j,2}](t) = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]. \end{cases} \quad (\text{IV.20})$$

Les sous-solveurs de mobilité s'écrivent ainsi de la manière suivante,

$$\begin{cases} [\phi_{j,1}] = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}], \\ [\phi_{j,2}] = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]. \end{cases} \quad (\text{IV.21})$$

Dans cette version de la méthode, l'étape de propagation se trouve inaltérée. Cependant, vu que les sous-solveurs de mobilité engendrent le même domaine que le pavé initial, la contraction à l'aide de ces sous-solveurs sera omise du pseudo-code. L'algorithme de Waltz se limitera uniquement aux contraintes d'observation. Une illustration de cette version est donnée dans FIG. IV.6. La figure montre que le pavé obtenu est légèrement plus large que celui obtenu avec la méthode exacte.

La deuxième version de la méthode représente une simplification totale, où à la fois le modèle de mobilité et les mesures d'observation sont simplifiés. L'équation de mobilité sera remplacée par celle d'un carré de côté $2.\Delta t.v_{max,j}$. De plus, les équations des disques internes des anneaux seront remplacées par celles des carrés inscrits dans ces disques et celles des disques externes par les carrés les englobant. Les anneaux

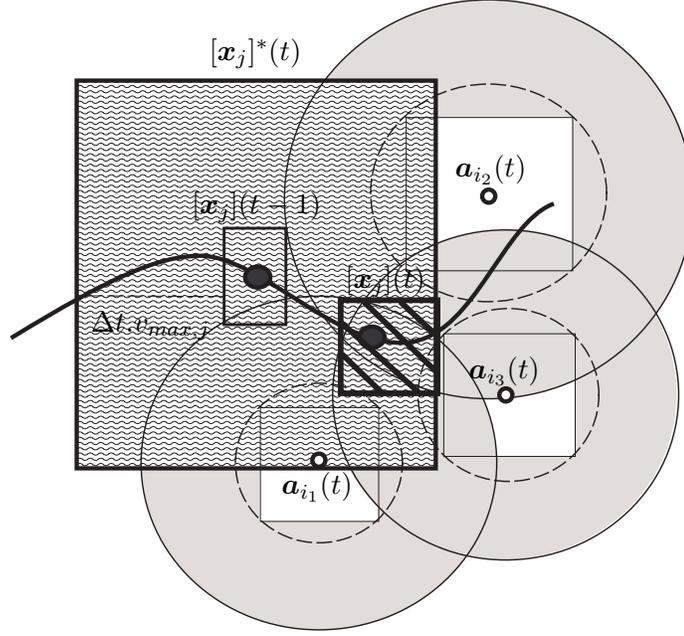


FIG. IV.6: Version à simplification partielle de l'auto-localisation sans modèle basée sur les ancrés.

circulaires donneront ainsi des anneaux rectangulaires (voir FIG. IV.7). La figure montre que l'anneau rectangulaire obtenu englobe le vrai anneau. La contrainte externe s'écrit alors comme suit,

$$\begin{cases} [x_{j,1}](t) = a_{i,1}(t) + [-R_{i,j}, +R_{i,j}], \\ [x_{j,2}](t) = a_{i,2}(t) + [-R_{i,j}, +R_{i,j}]. \end{cases} \quad (\text{IV.22})$$

Les sous-solveurs lui correspondant sont donnés de la manière suivante,

$$\begin{cases} [\psi_{j,i,1}] = a_{i,1}(t) + [-R_{i,j}, +R_{i,j}], \\ [\psi_{j,i,2}] = a_{i,2}(t) + [-R_{i,j}, +R_{i,j}]. \end{cases} \quad (\text{IV.23})$$

Le pavé interne, duquel le domaine de \mathbf{x}_j doit être déprivé, est donné par

$$\begin{cases} [z_{j,1}](t) = a_{i,1}(t) + [-\frac{\sqrt{2}}{2}.r_{i,j}, +\frac{\sqrt{2}}{2}.r_{i,j}], \\ [z_{j,2}](t) = a_{i,2}(t) + [-\frac{\sqrt{2}}{2}.r_{i,j}, +\frac{\sqrt{2}}{2}.r_{i,j}]. \end{cases} \quad (\text{IV.24})$$

Dans cette version, l'étape de propagation reste inchangée. En outre, la privation du domaine de \mathbf{x}_j du pavé $[z_i](t)$ se trouve inaltérée vu que la simplification aboutit au même pavé $[z_i](t)$. Toutefois, la contraction à l'aide des équations d'observation externes est modifiée, les sous-solveurs d'observation étant simplifiés. Une illustration est donnée dans FIG. IV.8. La figure montre que la simplification engendre un pavé encore plus large, entraînant une perte au niveau de la précision de l'estimation.

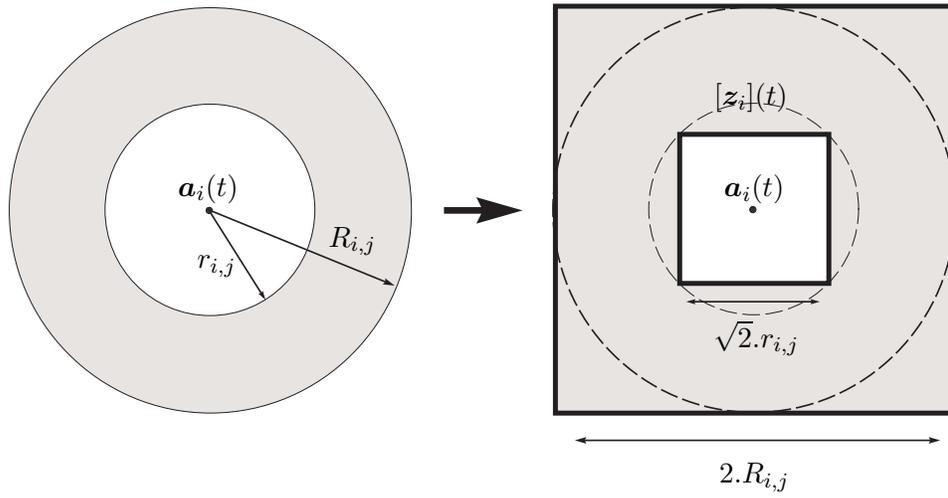


FIG. IV.7: Simplification d'un anneau d'observation.

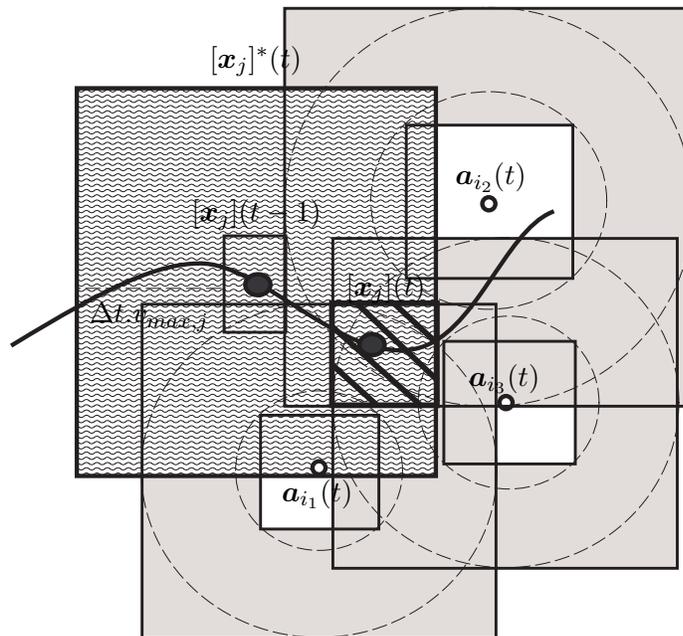


FIG. IV.8: Version à simplification totale de l'auto-localisation sans modèle basée sur les ancres.

IV.4 Auto-localisation sans modèle utilisant les non-ancres

Dans cette section, nous présentons une extension de la méthode ALAA, basée sur les ancres et les non-ancres. Celle-ci est notée ALAAN (pour Auto-Localisation utilisant des Anneaux basés sur les Ancres et les Non-ancres). Les contraintes d'observation sont alors des anneaux employant à la fois des ancres et des non-ancres. L'avantage principal de cette méthode est qu'elle engendre plus de précision, surtout dans les réseaux à faibles densités d'ancres. Dans la suite, nous commençons par définir les mesures d'observation tenant compte des ancres et des non-ancres. Nous exposons ensuite l'algorithme de résolution du problème basé sur l'analyse par intervalles.

IV.4.1 Mesures d'observation

La méthode ALAAN tient compte de tous les capteurs, y compris les non-ancres, pour estimer la position d'un nœud considéré. Les contraintes d'observation sont ainsi des anneaux centrés sur les ancres et les non-ancres pouvant communiquer avec ce nœud. Ces anneaux sont générés suite à une comparaison des puissances des signaux émis et reçus par tout type de capteurs. Les rayons des anneaux sont alors des distances inter-capteurs, ce qui constitue un désavantage pour les anneaux incorporant des nœuds. En effet, les nœuds ont des positions inconnues. De plus, l'incertitude présente sur ces positions produit des anneaux plus larges que ce qu'ils doivent être en réalité.

Afin de remédier à ce problème, nous proposons de générer deux anneaux pour chacun des capteurs communiquant avec le nœud considéré. L'un des anneaux a les rayons définis en n'utilisant que les puissances reçues par les ancres, l'autre n'utilisant que les nœuds. En effet, supposons un anneau centré sur une ancre et ayant le rayon externe fonction d'un nœud. Si nous ne distinguons pas les nœuds des ancres, il est possible que l'incertitude présente sur la position de ce nœud élargisse l'anneau de telle sorte qu'il devienne plus large que celui obtenu avec une ancre supposée plus loin. Dans ce cas, l'estimation devient moins précise, ce qui est dû à la perte d'information. La définition de deux types d'anneaux permet alors de remédier à ce problème.

Soient $I_{a,j}(t)$ et $I_{x,j}(t)$ les ensembles respectifs des indices des ancres et des non-ancres pouvant communiquer avec le nœud j à l'instant t . L'ensemble $I_j(t) = I_{a,j}(t) \cup I_{x,j}(t)$ correspond alors à tous les capteurs détectés par le nœud j à cet instant. Afin de générer les mesures d'observation, chaque capteur $\mathbf{s}_i(t)$, $i \in I_j(t)$ émet des signaux avec une même puissance initiale. En premier lieu, les puissances des signaux reçus par les ancres communiquant avec $\mathbf{s}_i(t)$ sont comparées pour générer le premier anneau. Soient $J_{a,i}(t)$ et $J_{x,i}(t)$ les ensembles des indices des ancres et des nœuds communiquant avec le capteur $\mathbf{s}_i(t)$. Les indices des ancres de $J_{a,i}(t)$ qui seront retenues pour la définition du premier anneau sont comme suit,

$$\begin{cases} l_{a,i} = \arg\{\min_{k \in J_{a,i}(t)} \{\rho_{\mathbf{s}_i, \mathbf{a}_k}(t) \mid \rho_{\mathbf{s}_i, \mathbf{a}_k}(t) \geq \rho_{\mathbf{s}_i, \mathbf{x}_j}(t)\}\}, \\ h_{a,i} = \arg\{\max_{k \in J_{a,i}(t)} \{\rho_{\mathbf{s}_i, \mathbf{a}_k}(t) \mid \rho_{\mathbf{s}_i, \mathbf{a}_k}(t) \leq \rho_{\mathbf{s}_i, \mathbf{x}_j}(t)\}\}, \end{cases} \quad (\text{IV.25})$$

où \mathbf{a}_k représente les coordonnées de l'ancre k , \mathbf{s}_i peut être une ancre ou un nœud, \mathbf{x}_j correspond au nœud j considéré, $j \in \{1, \dots, N_x\}$ et $\rho_{\mathbf{s}_i, \mathbf{a}_k}(t)$ est la puissance en dBm du

IV. AUTO-LOCALISATION SANS MODÈLE

signal émis par le capteur \mathbf{s}_i et reçu par l'ancre \mathbf{a}_k à l'instant t . $l_{a,i}$ et $h_{a,i}$ représentent les indices des ancres dont les distances au capteur \mathbf{s}_i définissent respectivement les rayons interne et externe de l'anneau. Ainsi, ces rayons, nommés respectivement $r_{a,i}$ et $R_{a,i}$, sont donnés par

$$\begin{cases} r_{a,i} = d(\mathbf{s}_i, \mathbf{a}_{l_{a,i}}), \\ R_{a,i} = d(\mathbf{s}_i, \mathbf{a}_{h_{a,i}}). \end{cases} \quad (\text{IV.26})$$

D'une manière similaire, la puissance du signal émis par \mathbf{s}_i et reçu par \mathbf{x}_j est comparée aux puissances des signaux reçus par les autres nœuds. Un deuxième anneau est établi, cette fois-ci basé plutôt sur les nœuds. Leurs indices sont définis comme suit,

$$\begin{cases} l_{x,i} = \arg\{\min_{k \in J_{x,i}(t) \cap I_{x,j}(t)} \{\rho_{\mathbf{s}_i, \mathbf{x}_k}(t) \mid \rho_{\mathbf{s}_i, \mathbf{x}_k}(t) \geq \rho_{\mathbf{s}_i, \mathbf{x}_j}(t)\}\}, \\ h_{x,i} = \arg\{\max_{k \in J_{x,i}(t) \cap I_{x,j}(t)} \{\rho_{\mathbf{s}_i, \mathbf{x}_k}(t) \mid \rho_{\mathbf{s}_i, \mathbf{x}_k}(t) \leq \rho_{\mathbf{s}_i, \mathbf{x}_j}(t)\}\}, \end{cases} \quad (\text{IV.27})$$

où $l_{x,i}$ et $h_{x,i}$ représentent les indices des nœuds dont les distances au capteur \mathbf{s}_i définissent respectivement les rayons interne et externe du deuxième anneau. Notons que le nœud j communique avec les nœuds $l_{x,i}$ et $h_{x,i}$ au cours de la localisation pour avoir leurs positions. Ces nœuds doivent alors appartenir obligatoirement à $I_{x,j}(t)$. Ainsi, les rayons interne et externe du deuxième anneau, nommés respectivement $r_{x,i}$ et $R_{x,i}$, sont définis par

$$\begin{cases} r_{x,i} = d(\mathbf{s}_i, \mathbf{x}_{l_{x,i}}), \\ R_{x,i} = d(\mathbf{s}_i, \mathbf{x}_{h_{x,i}}). \end{cases} \quad (\text{IV.28})$$

Les messages d'observation incluent les indices $l_{a,i}$, $h_{a,i}$, $l_{x,i}$ et $h_{x,i}$, ainsi que les coordonnées des ancres mises en jeu. Celles des nœuds, par contre, seront communiquées au cours du processus de localisation. Un anneau final résulte de l'intersection des deux anneaux. Ses rayons interne et externe, notés respectivement $r_{i,j}$ et $R_{i,j}$, sont obtenus comme suit,

$$\begin{cases} r_{i,j} = \max\{r_{a,i}, r_{x,i}\}, \\ R_{i,j} = \min\{R_{a,i}, R_{x,i}\}. \end{cases} \quad (\text{IV.29})$$

La contrainte issue de l'anneau final s'écrit alors de la manière suivante,

$$r_{i,j}^2 \leq (x_{j,1}(t) - s_{i,1}(t))^2 + (x_{j,2}(t) - s_{i,2}(t))^2 \leq R_{i,j}^2, \quad (\text{IV.30})$$

où $i \in I_j(t)$. FIG. IV.9 illustre les deux anneaux obtenus suivant cette méthode, où les nœuds sont supposés avoir des positions ponctuelles. La figure montre en trait interrompu l'anneau n'utilisant que les ancres et en trait plein l'anneau utilisant les nœuds, les deux étant centrés sur une ancre. L'anneau final est illustré en gras.

IV.4.2 Algorithme

La méthode ALAAN profite de la mobilité des capteurs pour améliorer la précision de l'estimation. Le problème d'auto-localisation est alors défini par le modèle de mobilité, associé aux mesures d'observation données par (IV.30),

$$\begin{cases} (x_{j,1}(t) - x_{j,1}(t-1))^2 + (x_{j,2}(t) - x_{j,2}(t-1))^2 = (\Delta t v_j)^2, \\ \text{avec } r_{i,j}^2 \leq (x_{j,1}(t) - s_{i,1}(t))^2 + (x_{j,2}(t) - s_{i,2}(t))^2 \leq R_{i,j}^2, i \in I_j(t), \\ \text{et } v_j \in [0, v_{max,j}], \end{cases} \quad (\text{IV.31})$$

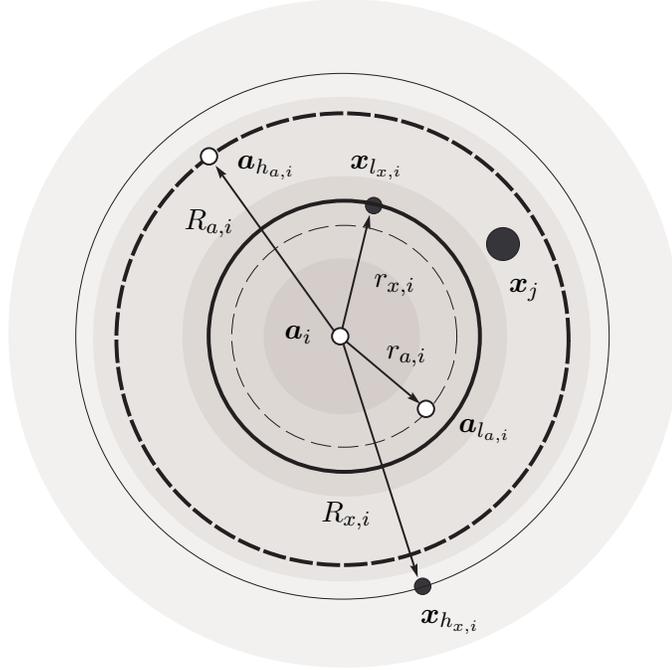


FIG. IV.9: Génération des anneaux à l'aide des ancres et des non-ancres.

où $v_{max,j}$ est la vitesse maximale du nœud j , v_j est une variable de vitesse inférieure à $v_{max,j}$ et Δt est la durée entre deux pas de temps consécutifs. Les valeurs des rayons $r_{i,j}$ et $R_{i,j}$ seront déterminées au cours du processus de localisation.

Généralement, le problème d'auto-localisation ne possède pas de solution unique, mais plutôt une zone couvrant un ensemble infini de solutions. La plupart des méthodes existantes réalisent une approximation interne de cette zone, générant une solution ponctuelle appartenant à la zone. Toutefois, la propagation de l'estimé ponctuel produit une erreur cumulative qui augmente d'un pas de temps à l'autre. D'où l'idée d'effectuer une approximation externe à l'aide des intervalles. La solution du problème sera alors un pavé couvrant toutes les solutions possibles. Ceci caractérise l'aspect garanti de la méthode. Ainsi, les variables seront remplacées par les intervalles auxquels elles appartiennent et les fonctions par des fonctions d'inclusion leur correspondant. Les positions des nœuds $\mathbf{x}_j(t)$, en particulier, seront remplacées par des pavés bidimensionnels, notés $[\mathbf{x}_j](t) = [x_{j,1}](t) \times [x_{j,2}](t)$. Résoudre le problème consiste alors à le définir tel un problème de satisfaction de contraintes que nous contracterons à l'aide de l'algorithme de Waltz.

L'équation de mobilité sera reformulée alors de la manière suivante,

$$([x_{j,1}](t) - [x_{j,1}](t-1))^2 + ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, (\Delta t \cdot v_{max,j})^2]. \quad (\text{IV.32})$$

IV. AUTO-LOCALISATION SANS MODÈLE

De la même façon, les contraintes d'observation s'écrivent comme suit,

$$([x_{j,1}](t) - [s_{i,1}](t))^2 + ([x_{j,2}](t) - [s_{i,2}](t))^2 = [r_{i,j}^2, R_{i,j}^2], i \in I_j(t), \quad (\text{IV.33})$$

où $[s_i](t) = [s_{i,1}](t) \times [s_{i,2}](t)$ est un intervalle à estimer si $s_i(t)$ est un nœud. Sinon, $[s_i](t) = s_i(t)$.

La résolution du problème est composée de deux étapes : la propagation où un domaine initial est défini et la correction où ce domaine sera contracté. La première étape consiste à propager le pavé précédent à l'aide du modèle de mobilité. Le pavé résultant, noté $[x_j]^*(t)$, s'écrit comme suit,

$$\begin{cases} [x_{j,1}]^*(t) = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}], \\ [x_{j,2}]^*(t) = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]. \end{cases} \quad (\text{IV.34})$$

Similairement à la méthode ALAA, le pavé $[x_j]^*(t)$ sera contracté à l'aide de l'algorithme de Waltz. Ce dernier sera implémenté par les contraintes de mobilité et d'observation, reformulées sous-forme de sous-solveurs. En fait, à chaque itération de l'algorithme, le nœud j communique avec les nœuds indiqués dans les messages d'observation afin d'obtenir leurs positions. Si nous considérons un message provenant d'un capteur $s_i(t)$, $i \in I_j(t)$, le nœud j reçoit des informations sur les positions des nœuds $l_{x,i}$, $h_{x,i}$ et i si s_i était un nœud. Notons $[s_i](t)$ la position du capteur s_i . Le pavé aura une taille nulle au cas où s_i est une ancre. Les rayons des anneaux seront alors calculés comme suit,

$$\left\{ \begin{array}{l} r_{a,i} = \inf(d([s_i](t), \mathbf{a}_{l_{a,i}}(t))) \\ \quad = \inf(\sqrt{(a_{l_{a,i},1}(t) - [s_{i,1}](t))^2 + (a_{l_{a,i},2}(t) - [s_{i,2}](t))^2}), \\ R_{a,i} = \sup(d([s_i](t), \mathbf{a}_{h_{a,i}}(t))) \\ \quad = \sup(\sqrt{(a_{h_{a,i},1}(t) - [s_{i,1}](t))^2 + (a_{h_{a,i},2}(t) - [s_{i,2}](t))^2}), \\ r_{x,i} = \inf(d([s_i](t), [\mathbf{x}_{l_{x,i}}](t))) \\ \quad = \inf(\sqrt{([x_{l_{x,i},1}](t) - [s_{i,1}](t))^2 + ([x_{l_{x,i},2}](t) - [s_{i,2}](t))^2}), \\ R_{x,i} = \sup(d([s_i](t), [\mathbf{x}_{h_{x,i}}](t))) \\ \quad = \sup(\sqrt{([x_{h_{x,i},1}](t) - [s_{i,1}](t))^2 + ([x_{h_{x,i},2}](t) - [s_{i,2}](t))^2}). \end{array} \right. \quad (\text{IV.35})$$

Notons que nous prenons la borne inférieure pour les rayons internes et celle supérieure pour les rayons externes en présence des données en intervalles. Avec ceci, nous obtenons des anneaux plus larges incluant toute l'incertitude existante. Les rayons interne et externe de l'anneau final centré sur s_i sont ainsi donné par $r_{i,j} = \max\{r_{a,i}, r_{x,i}\}$ et $R_{i,j} = \min\{R_{a,i}, R_{x,i}\}$, respectivement. Ces rayons seront recalculés à chaque itération de l'algorithme de Waltz. Similairement à la méthode ALAA, des sous-solveurs sont définis à partir du modèle de mobilité et du disque d'observation externe afin de contracter le pavé initial,

$$\begin{aligned} [\phi_{j,1}] &= [x_{j,1}](t-1) + [-\overline{bm}_{j,1}(t), +\overline{bm}_{j,1}(t)], \\ [\phi_{j,2}] &= [x_{j,2}](t-1) + [-\overline{bm}_{j,2}(t), +\overline{bm}_{j,2}(t)], \\ [\psi_{j,i,1}] &= [s_{i,1}](t) + [-\overline{bo}_{j,i,1}(t), +\overline{bo}_{j,i,1}(t)], \\ [\psi_{j,i,2}] &= [s_{i,2}](t) + [-\overline{bo}_{j,i,2}(t), +\overline{bo}_{j,i,2}(t)], \end{aligned} \quad (\text{IV.36})$$

avec

$$\begin{cases} \overline{bm}_{j,1}(t) = \sqrt{(\Delta t \cdot v_{max,j})^2 - \inf((x_{j,2}] (t) - [x_{j,2}] (t-1))^2)}, \\ \overline{bm}_{j,2}(t) = \sqrt{(\Delta t \cdot v_{max,j})^2 - \inf((x_{j,1}] (t) - [x_{j,1}] (t-1))^2)}, \\ \overline{bo}_{j,i,1}(t) = \sqrt{R_{i,j}^2 - \inf((x_{j,2}] (t) - [s_{i,2}] (t))^2)}, \\ \overline{bo}_{j,i,2}(t) = \sqrt{R_{i,j}^2 - \inf((x_{j,1}] (t) - [s_{i,1}] (t))^2)}. \end{cases}$$

Le domaine de \mathbf{x}_j sera également déprivé du pavé inscrit dans le disque interne comme suit,

$$[\mathbf{x}_j](t) = [\mathbf{x}_j](t) \setminus [\mathbf{z}_i](t), \quad (\text{IV.37})$$

avec $[\mathbf{z}_i](t) = [z_{i,1}](t) \times [z_{i,2}](t)$ où

$$\begin{cases} [z_{i,1}](t) = [\sup([s_{i,1}](t) - \frac{\sqrt{2}}{2} \cdot r_{i,j}), \inf([s_{i,1}](t) + \frac{\sqrt{2}}{2} \cdot r_{i,j})], \\ [z_{i,2}](t) = [\sup([s_{i,2}](t) - \frac{\sqrt{2}}{2} \cdot r_{i,j}), \inf([s_{i,2}](t) + \frac{\sqrt{2}}{2} \cdot r_{i,j})]. \end{cases} \quad (\text{IV.38})$$

Notons qu'en présence des intervalles, le plus petit pavé inscrit dans le disque interne est sélectionné dans le but de garder la garantie. Une illustration de cette méthode est donnée dans FIG. IV.10. La figure montre que l'utilisation des nœuds conduit à des anneaux tordus plus larges que ce qu'ils doivent être réellement. Toutefois, l'utilisation des nœuds réduit énormément la zone solution fournissant plus de précision. Le pseudo-code de la méthode est illustré dans l'algorithme 9.

IV.5 Simulations

Dans cette section, nous évaluons les performances des méthodes sans modèle. Similairement au chapitre III, ces méthodes seront évaluées selon différents critères, que nous citons ci-dessous,

- ◇ la **précision de l'estimation**, représentée par la surface du pavé estimé ;
- ◇ l'**erreur d'estimation**, définie par la distance entre la position réelle du nœud et le centre du pavé estimé ;
- ◇ et la **complexité de la méthode**, donnée par le temps de calcul.

Notons que toutes les simulations sont réalisées sur Matlab 6.1, installé sur un CPU Intel(R) Core(TM)2 (2.40GHz, 1.00GB RAM).

Afin d'évaluer les différentes méthodes, nous simulons un mouvement de groupe basé sur le modèle de mobilité dit *Reference Point Group Mobility Model* ou RPGMM, présenté dans la section I.4.3 du premier chapitre. Ce mouvement possède une trajectoire de base de 100s suivant laquelle tous les capteurs du groupe se déplacent. Celle-ci est composée de deux sinusoïdes présentant un changement brusque de direction. Selon cette trajectoire, la vitesse maximale du mouvement est égale à $3.1690m \cdot s^{-1}$. Nous supposons en premier lieu que seuls les nœuds sont mobiles en groupe, les ancres étant considérées fixes. Nous déplacerons ensuite tous les capteurs ancres et non-ancres suivant le même mouvement. Ainsi, nous déployons initialement les capteurs dans une zone carrée de dimensions $100m \times 100m$, dite zone de déploiement. Les capteurs mobiles se déplaceront alors, à partir de leurs positions initiales, suivant la trajectoire sinusoïdale,

IV. AUTO-LOCALISATION SANS MODÈLE

Algorithme 9 : Pseudo-code de la méthode d'auto-localisation sans modèle utilisant les non-ancres.

Entrées : $[\mathbf{x}_j](t-1)$, $I_{a,j}(t)$, $I_{x,j}(t)$, $\mathbf{a}_i(t)$, $\mathbf{a}_{l_{a,i}}(t)$, $\mathbf{a}_{h_{a,i}}(t)$, $l_{x,i}$, $h_{x,i}$, $i \in I_{a,j}(t)$, $v_{max,j}$, Δt ;

Sorties : $[\mathbf{x}_j](t)$;

$$\begin{cases} [x_{j,1}]^*(t) = [x_{j,1}](t-1) + [-\Delta t.v_{max,j}, \Delta t.v_{max,j}]; \\ [x_{j,2}]^*(t) = [x_{j,2}](t-1) + [-\Delta t.v_{max,j}, \Delta t.v_{max,j}]; \end{cases}$$

$$[x_{j,1}](t) = [x_{j,1}]^*(t), [x_{j,2}](t) = [x_{j,2}]^*(t);$$

$$A_j = w([x_{j,1}](t)) * w([x_{j,2}](t));$$

$$A_j^{old} = A_j + 1;$$

tant que $A_j < A_j^{old}$ **faire**

$$A_j^{old} = A_j;$$

pour $i \in I_j(t)$ **faire**

$$\mathbf{x}_j(t) \leftarrow [\mathbf{x}_{l_{x,i}}](t), \mathbf{x}_j(t) \leftarrow [\mathbf{x}_{h_{x,i}}](t);$$

si s_i **est un nœud alors**

$$\mathbf{x}_j(t) \leftarrow [s_i](t);$$

fin

$$R_{a,i} = \sup(\sqrt{(a_{h_{a,i},1}(t) - [s_{i,1}](t))^2 + (a_{h_{a,i},2}(t) - [s_{i,2}](t))^2});$$

$$R_{x,i} = \sup(\sqrt{([x_{h_{x,i},1}](t) - [s_{i,1}](t))^2 + ([x_{h_{x,i},2}](t) - [s_{i,2}](t))^2});$$

$$R_{i,j} = \min\{R_{a,i}, R_{x,i}\};$$

$$[x_{j,2}](t) = [x_{j,2}](t) \cap [\inf([s_{i,2}](t) - R_{i,j}), \sup([s_{i,2}](t) + R_{i,j})];$$

$$\overline{bo}_{j,i,1}(t) = \sqrt{R_{i,j}^2 - \inf((x_{j,2}](t) - [s_{i,2}](t))^2);$$

$$[\psi_{j,i,1}] = [s_{i,1}](t) + [-\overline{bo}_{j,i,1}(t), \overline{bo}_{j,i,1}(t)];$$

$$[x_{j,1}](t) = [x_{j,1}](t) \cap [\psi_{j,i,1}];$$

$$[x_{j,1}](t) = [x_{j,1}](t) \cap [\inf([s_{i,1}](t) - R_{i,j}), \sup([s_{i,1}](t) + R_{i,j})];$$

$$\overline{bo}_{j,i,2}(t) = \sqrt{R_{i,j}^2 - \inf((x_{j,1}](t) - [s_{i,1}](t))^2);$$

$$[\psi_{j,i,2}] = [s_{i,2}](t) + [-\overline{bo}_{j,i,2}(t), \overline{bo}_{j,i,2}(t)];$$

$$[x_{j,2}](t) = [x_{j,2}](t) \cap [\psi_{j,i,2}];$$

$$r_{a,i} = \inf(\sqrt{(a_{l_{a,i},1}(t) - [s_{i,1}](t))^2 + (a_{l_{a,i},2}(t) - [s_{i,2}](t))^2});$$

$$r_{x,i} = \inf(\sqrt{([x_{l_{x,i},1}](t) - [s_{i,1}](t))^2 + ([x_{l_{x,i},2}](t) - [s_{i,2}](t))^2});$$

$$r_{i,j} = \max\{r_{a,i}, r_{x,i}\};$$

$$[z_{i,1}](t) = [\sup([s_{i,1}](t) - \frac{\sqrt{2}}{2}.r_{i,j}), \inf([s_{i,1}](t) + \frac{\sqrt{2}}{2}.r_{i,j})];$$

$$[z_{i,2}](t) = [\sup([s_{i,2}](t) - \frac{\sqrt{2}}{2}.r_{i,j}), \inf([s_{i,2}](t) + \frac{\sqrt{2}}{2}.r_{i,j})];$$

$$[\mathbf{z}_i](t) = [z_i](t);$$

fin

$$\overline{bm}_{j,1}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf((x_{j,2}](t) - [x_{j,2}](t-1))^2);$$

$$[\phi_{j,1}] = [x_{j,1}](t-1) + [-\overline{bm}_{j,1}(t), \overline{bm}_{j,1}(t)];$$

$$[x_{j,1}](t) = [x_{j,1}](t) \cap [\phi_{j,1}];$$

$$\overline{bm}_{j,2}(t) = \sqrt{(\Delta t.v_{max,j})^2 - \inf((x_{j,1}](t) - [x_{j,1}](t-1))^2);$$

$$[\phi_{j,2}] = [x_{j,2}](t-1) + [-\overline{bm}_{j,2}(t), \overline{bm}_{j,2}(t)];$$

$$[x_{j,2}](t) = [x_{j,2}](t) \cap [\phi_{j,2}];$$

$$A_j = w([x_{j,1}](t)) * w([x_{j,2}](t));$$

fin

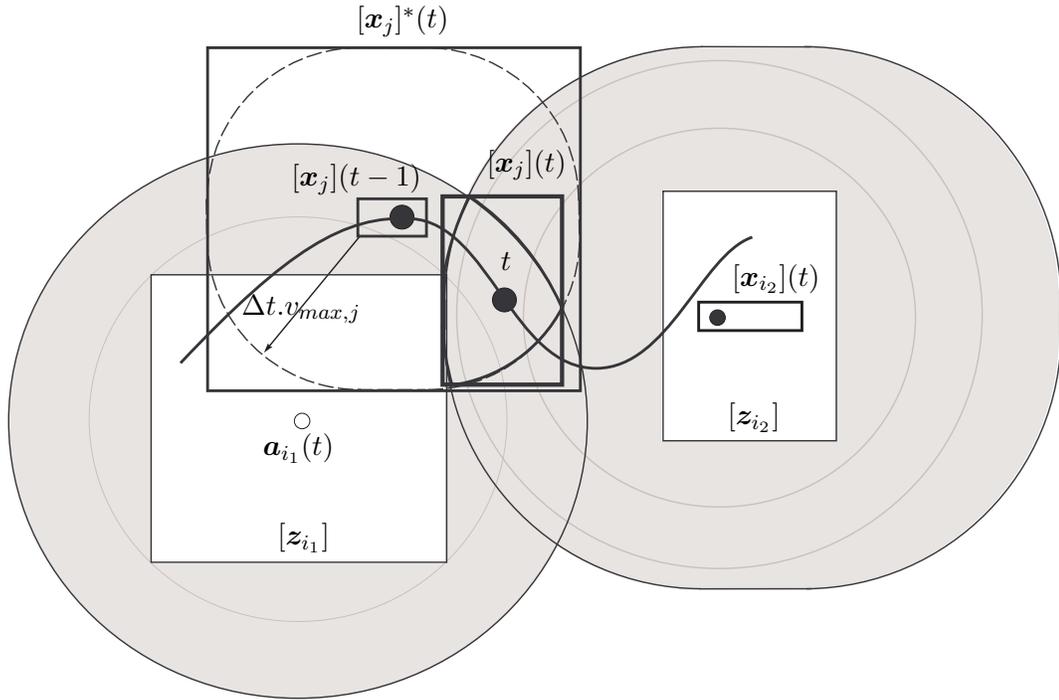


FIG. IV.10: Auto-localisation sans modèle utilisant les ancres et les non-ancres.

avec une simple translation. Une illustration d'un mouvement de groupe de 50 capteurs est donnée dans FIG. IV.11. Dans cette section, nous commençons tout d'abord par l'évaluation des différentes versions de la méthode ALAA basée sur les ancres. Nous montrons par la suite l'importance de l'algorithme de Waltz. Nous comparons ensuite la méthode ALAA à la méthode ALCA basée sur les connectivités vis-à-vis des ancres. Une comparaison à deux méthodes basées sur l'approche séquentielle de Monte-Carlo est également présentée. Nous passons ensuite à l'évaluation de la méthode ALAAN basée sur tous les capteurs. Nous la comparons ainsi à la méthode sans modèle basée sur les ancres. Nous illustrons ensuite la sensibilité de cette méthode à la densité d'ancres et des nœuds dans le réseau.

IV.5.1 Evaluation de la méthode ALAA

Dans cette section, nous évaluons les performances de la méthode d'auto-localisation sans modèle basée uniquement sur les ancres, dite ALAA. Nous illustrons ainsi les différentes versions de la méthode : la version exacte sans simplification, la version avec simplification partielle, notée ALAAp, où seule le modèle de mobilité est approximé, et la version avec simplification totale, notée ALAAt, où à la fois le modèle de mobilité et les mesures d'observation sont simplifiés. Sans perte de généralité, nous considérons dans cette section le problème d'auto-localisation d'un nœud unique, vu que chacun des nœuds estime sa position indépendamment des autres. De plus, nous supposons

IV. AUTO-LOCALISATION SANS MODÈLE

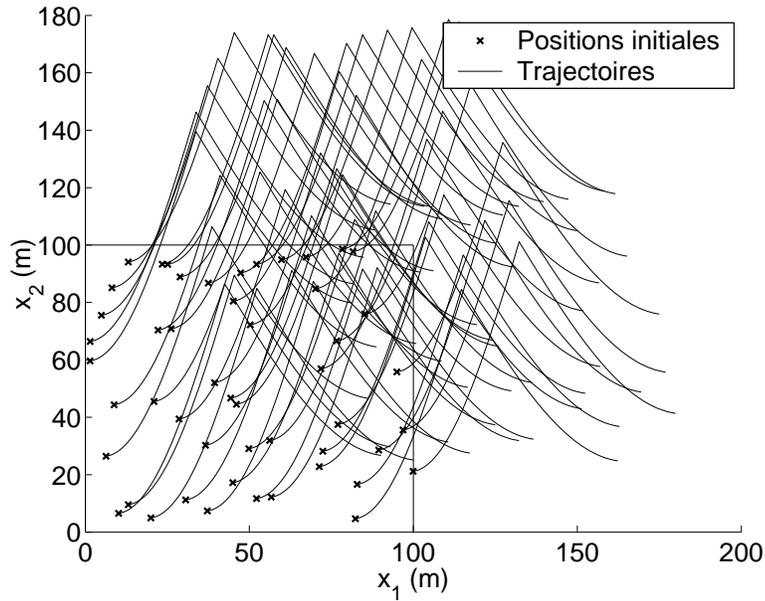


FIG. IV.11: Mouvement de groupe de 50 capteurs.

Méthode	ALAA	ALAAp	ALAAt
Temps de calcul	0.01689s	0.01201s	0.00421s
Erreur moyenne	0.7423m	0.7440m	1.7058m
Surface moyenne	6.6716m ²	6.6751m ²	18.6709m ²

TAB. IV.1: Comparaison des différentes versions de la méthode ALAA.

les ancrs fixes, 150 étant lancées aléatoirement dans la zone de déploiement. Etant donné que le nœud se déplace selon la trajectoire de base, sa vitesse maximale v_{max} est fixée à $3.1690m.s^{-1}$. Notons qu'à chaque instant, seules les ancrs situées dans la portée du nœud sont sollicitées. Celles-ci sont situées dans un rayon inférieur à $10m$ autour du nœud. FIG. IV.12 illustre les pavés obtenus à l'aide des méthodes ALAA et sa version simplifiée ALAAt. La figure montre qu'avec la simplification totale, nous perdons significativement au niveau de la précision. FIG. IV.13 illustre les rapports des surfaces des pavés obtenus avec ALAA sur celles obtenues avec ALAAt en trait plein et ALAA sur ALAAp en trait interrompu. Il est clair que la simplification partielle n'affecte pas beaucoup la qualité de l'estimation. TAB. IV.1 regroupe les valeurs du temps de calcul moyen, de l'erreur moyenne et de la surface moyenne par pas de temps obtenues avec les différentes versions de la méthode. La simplification totale apporte un gain important au niveau de calcul au prix de la qualité de l'estimation. Vu la différence négligeable entre la version avec simplification partielle et la version exacte, nous adopterons pour la suite la version avec simplification partielle, que nous noterons ALAA tout court.

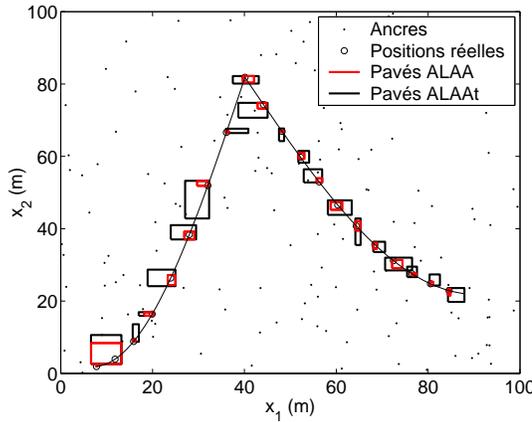


FIG. IV.12: Pavés estimés à l’aide des méthodes ALAA et ALAAat.

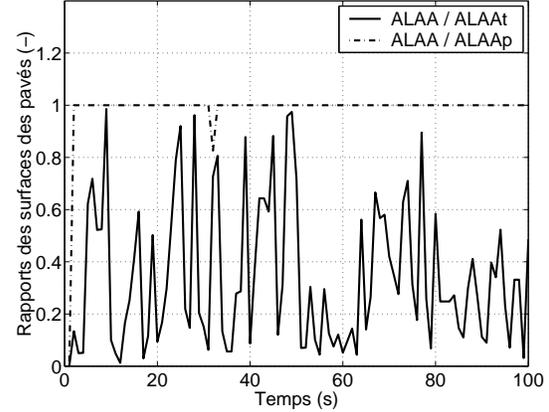


FIG. IV.13: Rapports des surfaces des pavés obtenus avec ALAA, ALAAp et ALAAat.

IV.5.2 Efficacité de l’algorithme de Waltz

Dans cette section, nous montrons l’importance de l’algorithme de Waltz. Nous rappelons que cet algorithme itère toutes les contraintes sans aucun ordre a priori jusqu’à ce que le pavé n’est plus contracté. Le contracteur de Waltz est un algorithme rapide et très simple à implémenter permettant de réduire la taille du pavé à estimer. Afin d’illustrer son importance, nous varions le nombre maximal d’itérations que l’algorithme est capable de réaliser. Celui-ci, noté K_W , est alors varié de 1 jusqu’à 10. FIG. IV.14 illustre les variations de l’erreur d’estimation moyenne et de la surface moyenne en fonction de K_W . La figure montre que le meilleur résultat est obtenu avec 5 itérations au maximum, nombre à partir duquel l’algorithme de Waltz ne contracte plus. FIG. IV.15 montre les pavés obtenus avec la méthode ALAA sans itération ($K_W = 1$) et celle-ci avec l’algorithme de Waltz capable de réaliser 5 itérations ($K_W = 5$). Notons que, pour cet exemple, la taille des pavés varie peu avec le nombre d’itérations permises, ce qui n’est pas toujours le cas.

IV.5.3 Comparaison de la méthode ALAA à la méthode ALCA

Dans cette section, nous comparons la méthode sans modèle ALAA à la méthode basée sur les mesures de connectivités ALCA, présentée dans le chapitre précédent. Pour cette raison, nous dispersons aléatoirement 150 ancres dans la zone de déploiement, celles-ci seront supposées fixes. Par ailleurs, nous considérons le problème d’auto-localisation d’un seul nœud se déplaçant selon la trajectoire de base. Son paramètre vitesse maximale est fixé à sa vraie valeur, $v_{max} = 3.1690 m.s^{-1}$. FIG. IV.16 illustre les pavés obtenus avec les deux méthodes. Il est clair que les pavés obtenus avec ALAA sont sensiblement plus petits que ceux engendrés par ALCA. FIG. IV.17 illustre les rapports des erreurs d’estimation et des surfaces des pavés obtenus avec la méthode

IV. AUTO-LOCALISATION SANS MODÈLE

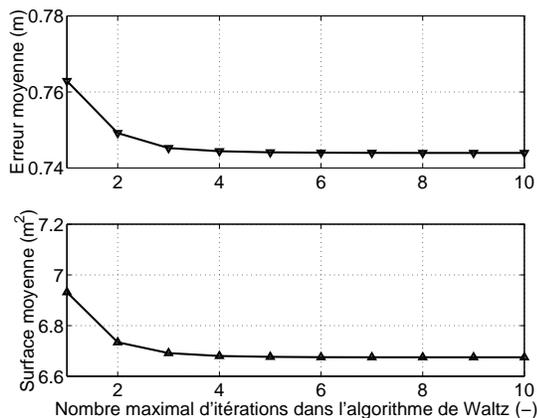


FIG. IV.14: Variation de l'erreur moyenne et de la surface moyenne en fonction du nombre maximal d'itérations permises dans l'algorithme de Waltz.

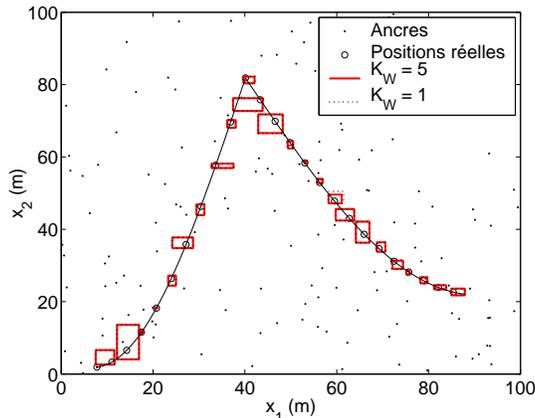


FIG. IV.15: Pavés obtenus avec la méthode ALAA où les contraintes ne sont pas itérées ($K_W = 1$) et celle avec l'algorithme de Waltz avec $K_W = 5$.

Méthode	ALAA	ALCA
Temps de calcul	0.01201s	0.00795s
Erreur moyenne	0.7440m	1.9151m
Surface moyenne	6.6751m ²	41.5928m ²

TAB. IV.2: Comparaison des méthodes ALAA et ALCA.

ALAA sur celles obtenues avec la méthode ALCA. TAB. IV.2 regroupe les valeurs du temps de calcul moyen, de l'erreur moyenne et de la surface moyenne obtenues avec les deux méthodes. Les gains en précision et en erreur valent respectivement 83.95% et 61.154% alors que l'accroissement du temps de calcul est de 51.07%, celui-ci restant tout de même très faible. Ainsi, l'utilisation des anneaux améliore significativement la qualité de l'estimation au prix d'une augmentation dans le temps de calcul.

IV.5.4 Comparaison de ALAA à des méthodes de Monte-Carlo

Pour illustrer l'efficacité de notre méthode, nous la comparons à deux méthodes basées sur l'approche séquentielle de Monte-Carlo [Doucet *et al.*, 2000]. La première méthode, présentée dans [Baggio et Langendoen, 2006], se base sur les mesures de connectivités. Celle-ci, notée MCB, génère des particules satisfaisant toutes les contraintes. Elle est composée de deux étapes : (i) la prédiction, où des positions sont engendrées à l'intérieur de la zone d'intersection des pavés approxinant les modèles de mobilité et d'observation, et (ii) le filtrage, où seules les particules satisfaisant les équations exactes des modèles sont gardées. Ces deux étapes sont itérées jusqu'à ce qu'un nombre N_p de particules soit obtenu ou que les itérations atteignent un nombre maximal, que nous fixons à 200. La deuxième méthode basée sur Monte-Carlo représente une version mo-

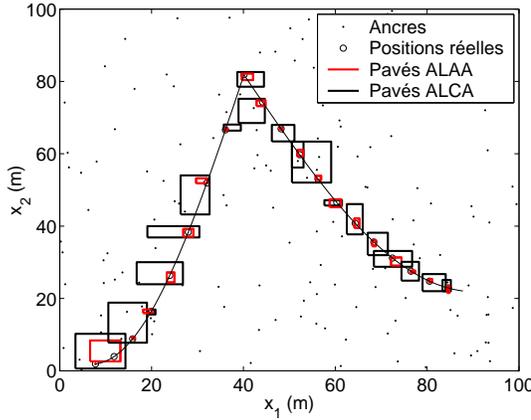


FIG. IV.16: Pavés estimés à l'aide des méthodes ALAA et ALCA.

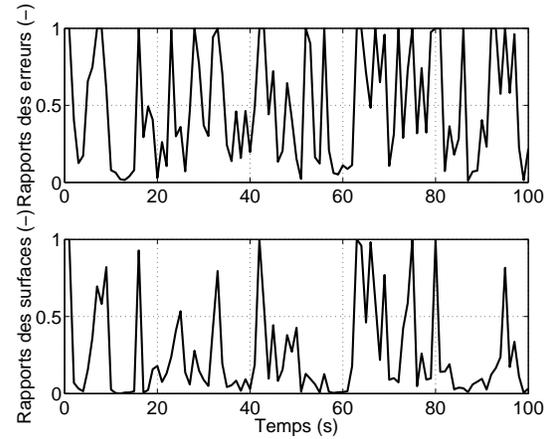


FIG. IV.17: Rapports des erreurs d'estimation et des surfaces des pavés obtenus avec ALAA et ALCA.

Méthode	ALAA	MCB	MMCB
Temps de calcul	0.01201s	0.03105s	0.4944s
Erreur moyenne	0.7440m	2.0672m	0.6986m

TAB. IV.3: Comparaison de la méthode ALAA aux méthodes MCB et MMCB.

difiée de MCB. Notée MMCB, cette méthode remplace les mesures de connectivités par les anneaux. Elle pose alors le même problème que la méthode ALAA, qu'elle résout à l'aide de l'approche de Monte-Carlo. Le nombre des particules pour les deux méthodes est fixé à 100. FIG. IV.18 et FIG. IV.19 illustrent les particules obtenues respectivement avec les méthodes MCB et MMCB. Elles montrent en outre les pavés engendrés avec la méthode ALAA. TAB. IV.3 regroupe les valeurs du temps de calcul moyen et de l'erreur d'estimation moyenne obtenues avec les trois méthodes. Les résultats illustrent les performances de notre méthode par rapport à la méthode MCB utilisant les connectivités. Les résultats montrent en outre que la méthode MMCB modifiée apporte un gain de 6.5% au niveau de l'erreur d'estimation par rapport à ALAA. Toutefois, ALAA entraîne un gain de 97.57% dans le temps de calcul. Le gain dans l'erreur est principalement dû au fait que les particules couvrent avec précision la zone solution. Notons que cette différence au niveau de l'erreur reste négligeable devant le temps de calcul relativement trop grand comparé à celui de ALAA.

IV.5.5 Comparaison de la méthode ALAAN à la méthode ALAA

Dans cette section, nous illustrons les performances de la méthode sans modèle ALAAN basée à la fois sur les ancrés et les non-ancrés. Pour cette raison, nous supposons un mouvement de groupe de 3 ancrés et 5 nœuds situés dans la portée l'un de l'autre tout au long du mouvement. FIG. IV.20 illustre les pavés obtenus avec les

IV. AUTO-LOCALISATION SANS MODÈLE

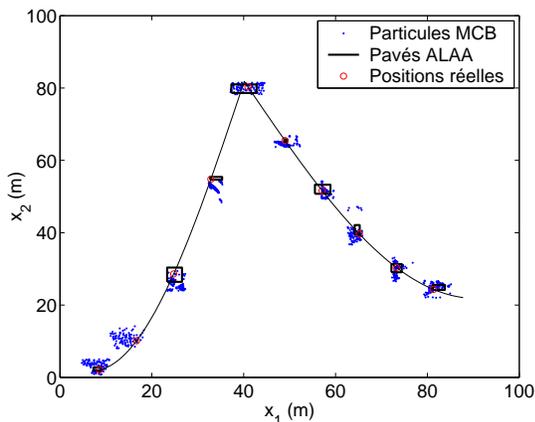


FIG. IV.18: Illustration des pavés estimés à l'aide de la méthode ALAA et des particules issues de la méthode MCB.

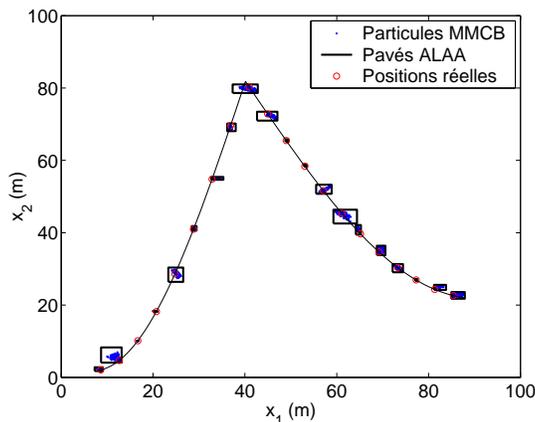


FIG. IV.19: Illustration des pavés estimés à l'aide de la méthode ALAA et des particules issues de la méthode MNCB.

méthodes ALAAN et ALAA pour l'un des nœuds considérés. Selon la figure, les pavés obtenus avec ALAAN sont plus petits que ceux obtenus avec ALAA. FIG. IV.21 illustre les courbes représentatives des rapports des erreurs et des surfaces obtenues avec la méthode ALAAN sur celles issues de la méthode ALAA. La figure montre les courbes des moyennes des rapports sur tous les nœuds ainsi que les courbes des plus petits rapports correspondant à l'un des cinq nœuds. Cette différence entre les courbes moyennes et les meilleures courbes est due à la dépendance des performances à la disposition des ancrés et des nœuds autour du nœud considéré. Les deux figures illustrent l'efficacité de l'utilisation des non-ancres dans l'estimation des positions des nœuds. Les temps de calcul moyen par nœud par pas de temps sont égaux à $0.011076s$ et $0.00568s$ respectivement pour les méthodes ALAAN et ALAA. Ainsi, la méthode ALAAN assure un gain dans la précision au prix du temps de calcul, qui reste relativement faible.

A présent, intéressons-nous à l'augmentation de l'énergie consommée causée par l'utilisation des nœuds. Cette consommation supplémentaire est principalement due à l'échange des coordonnées au cours de l'algorithme, à l'envoi et à la réception des messages d'observations et à la transmission des signaux afin de générer ces mesures d'observation. Soient m et n respectivement les nombres d'ancres et de nœuds présents dans le réseau. Supposons que tous ces capteurs peuvent communiquer ensemble. Nous étudierons dans ces conditions la consommation supplémentaire d'énergie associée à un nœud j au cours d'un seul pas temps. Notons que pour ceci, nous supposons que les nombres entiers ou réels sont codés sur 8 bits chacun.

- **Echange des coordonnées :** A chaque itération de l'algorithme de la méthode ALAAN, chaque nœud envoie ses coordonnées aux nœuds voisins et reçoit les leurs. Dans cet exemple, l'algorithme de Waltz parvient au résultat final dans seulement 2 itérations en moyenne. En d'autres termes, un nœud considéré envoie ses coordonnées à $n - 1$ nœuds et reçoit les leurs deux fois dans un pas de temps. Vu que la position

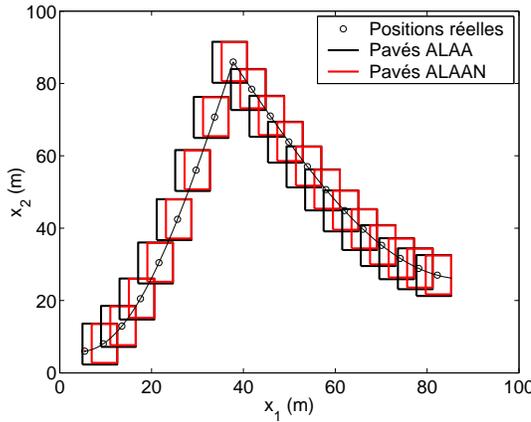


FIG. IV.20: Pavés estimés à l'aide des méthodes ALAAN et ALAA.

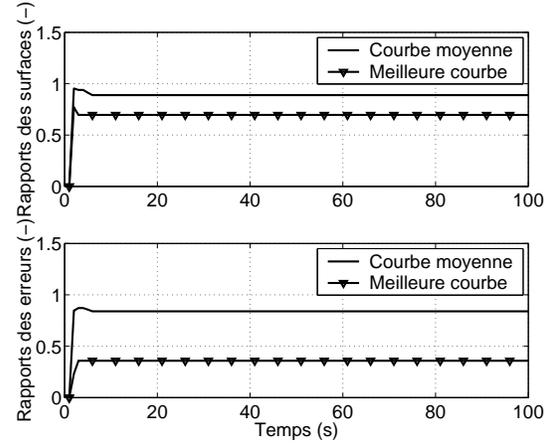


FIG. IV.21: Variations des rapports des erreurs et des surfaces des pavés obtenues avec ALAAN sur celles obtenues avec ALAA.

d'un nœud est un pavé bidimensionnel, ses coordonnées consisteront en 4 nombres réels. Si l'on considère que les messages échangés contiennent également l'identifiant de l'émetteur (1 entier), le nœud j échange alors $2 \cdot (4+1) \cdot (n-1) = 10 \cdot (n-1)$ nombres à chaque pas de temps. Ainsi, le nœud j reçoit $8 \cdot 10 \cdot (n-1) = 80 \cdot (n-1)$ bits et envoie $8 \cdot 10 \cdot (n-1) = 80 \cdot (n-1)$ bits. Soient P_e et P_r respectivement les énergies consommées pour l'envoi et la réception d'un bit. En conséquence, l'énergie consommée augmente de $80 \cdot (n-1) \cdot P_e + 80 \cdot (n-1) \cdot P_r$ par rapport à la méthode ALAA.

- **Echange des observations :** Un message d'observation émis par une ancre dans la méthode ALAAN contient supplément les indices des deux nœuds définissant le deuxième anneau. Chaque nœud reçoit ainsi 2 entiers en plus de chacune des ancrs, ce qui fait $2 \cdot m \cdot 8 = 16 \cdot m$ bits en plus. Considérons pour l'instant les messages d'observation émis par les nœuds. Un tel message contient l'identifiant de l'émetteur (1 entier), les coordonnées des ancrs définissant le premier anneau ($2 \cdot 2 = 4$ réels) et les indices des nœuds définissant le deuxième (2 entiers). Le message est alors codé sur $(1 + 4 + 2) \cdot 8 = 56$ bits. Chaque nœud échange de tels messages avec les $n-1$ nœuds voisins. Ceci contribue alors à l'envoi et à la réception de $56 \cdot (n-1)$ bits. D'où l'énergie consommée augmente de $56 \cdot (n-1) \cdot P_e + (56 \cdot (n-1) + 16 \cdot m) \cdot P_r$ par rapport à la méthode basée sur les ancrs.
- **Génération des observations :** Si l'on considère les messages émis par les ancrs, aucun signal supplémentaire n'est requis vu que les ancrs ont déjà les puissances mesurées aux nœuds. L'augmentation de l'énergie est alors due à la génération des mesures centrées sur les nœuds. Supposons en premier que le nœud j est le nœud qui construit les anneaux. Il envoie alors des signaux dans le réseau contenant son identifiant (1 entier). $m+n-1$ messages sont requis au minimum, ce qui conduit à

IV. AUTO-LOCALISATION SANS MODÈLE

$(m+n-1)*1*8 = 8.(m+n-1)$ bits. Les ancres et les autres nœuds reçoivent ces signaux et en mesurent les puissances qu'ils renvoient à l'émetteur. Celui-ci reçoit ainsi des messages de la part de $n-1$ nœuds contenant les valeurs des puissances reçues ainsi que leurs identifiants $((1+1)*(n-1)*8 = 16.(n-1)$ bits). Il reçoit également des messages de la part des ancres contenant leurs coordonnées, leurs identifiants and les valeurs des puissances mesurées, conduisant à la réception de $m*4*8 = 32.m$ bits. En tant que générateur d'observations, un nœud envoie alors $8.(m+n-1)$ bits au minimum et reçoit $32.m + 16.(n-1)$ bits. En tant que récepteur participant à la génération des mesures, un nœud reçoit les signaux des autres nœuds avec leurs identifiants (réception de $8.(n-1)$ bits). Il mesure les puissances des signaux et les renvoie aux émetteurs avec leurs identifiants (envoi de $2*(n-1)*8 = 16.(n-1)$ bits). Ainsi, la consommation d'énergie augmente au minimum de $(8.(m+n-1) + 16.(n-1)).P_e + (32.m + 16.(n-1) + 8.(n-1)).P_r = (8.m + 24.(n-1)).P_e + (32.m + 24.(n-1)).P_r$.

Enfin, si nous calculons l'augmentation totale de l'énergie consommée due à l'utilisation des nœuds, nous obtenons

$$(8.m + 160.(n-1)).P_e + (48.m + 160.(n-1)).P_r.$$

Nous nous retrouvons alors face à la recherche d'un compromis entre l'augmentation de la précision de l'estimation en utilisant les nœuds et la minimisation de la consommation d'énergie en négligeant quelques informations.

IV.5.6 Sensibilité de la méthode ALAAN à la densité d'ancres

Les performances de la méthode ALAAN, comparée à la méthode ALAA, dépendent de la densité des ancres et des nœuds dans le réseau. Afin d'illustrer cette dépendance, nous fixons en premier le nombre de nœuds à 5 et nous varions le nombre d'ancres situées dans leurs portées communes de 2 à 10. FIG. IV.22 illustre les variations des rapports de l'erreur moyenne et de la surface moyenne ALAAN / ALAA en fonction des nombre d'ancres. Comme prévu, la qualité de l'estimation est d'autant plus grande que le nombre d'ancres est plus petit. Nous fixons ensuite le nombre d'ancres à 3 et nous varions le nombre de nœuds de 2 à 10. FIG. IV.23 illustre les rapports de l'erreur et de la surface des pavés obtenues avec ALAAN sur celles obtenues avec ALAA en fonction du nombre de nœuds. La figure montre que les performances de ALAAN augmentent par rapport à ALAA avec l'augmentation du nombre de nœuds. Ainsi, la méthode utilisant les nœuds est de plus en plus privilégiée dans les réseaux à faibles densités d'ancres avec un nombre de nœuds de plus en plus grand.

IV.6 Conclusion

Dans ce chapitre, nous avons proposé une méthode d'auto-localisation sans modèle utilisant les ancres. Cette méthode est basée sur la comparaison des puissances des signaux échangés entre les capteurs afin de définir des anneaux d'observation. L'approche

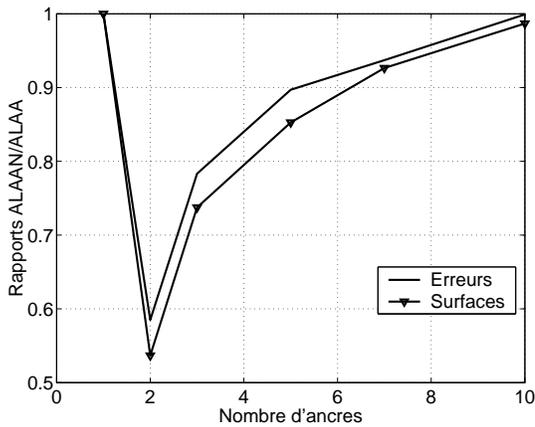


FIG. IV.22: Rapports moyens des erreurs et des surfaces des pavés en fonction du nombre d'ancres.

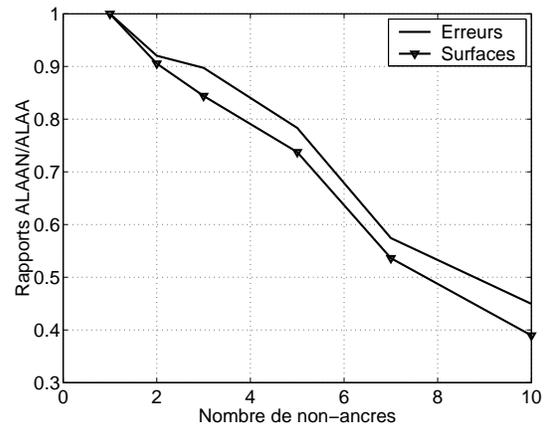
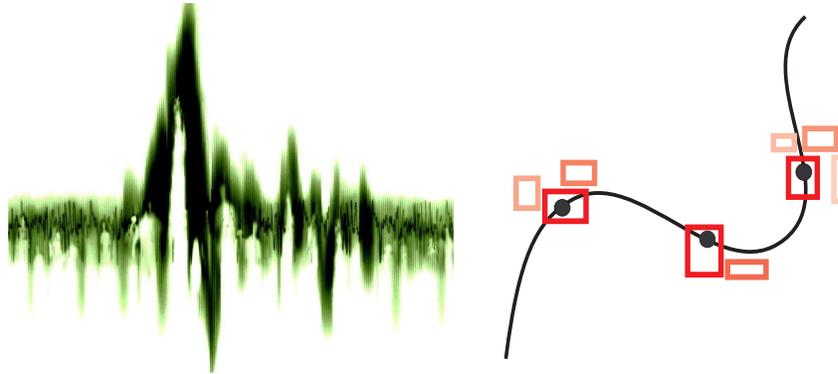


FIG. IV.23: Rapports moyens des erreurs et des surfaces des pavés en fonction du nombre de nœuds.

proposée permet de contourner les problèmes dus au calcul des paramètres du canal dans les méthodes basées sur les connectivités. Nous avons également présenté une extension de cette méthode, utilisant à la fois les ancres et les nœuds. Les deux méthodes proposées profitent de la mobilité des nœuds pour raffiner l'estimation des positions. La résolution des problèmes posés est effectuée dans le cadre de la théorie des intervalles. Les positions estimées sont alors des pavés bidimensionnels couvrant toutes les solutions possibles du problème. Les résultats de simulation illustrent les performances de ces méthodes comparées à l'approche exposée dans le chapitre précédent. Ainsi, les méthodes proposées sont efficaces dans les environnements parfaits. Toutefois, un bruit dans le canal ou une modification de la puissance d'envoi initiale peuvent générer des mesures fausses, conduisant à la perte de la garantie. Dans le chapitre suivant, nous traitons alors le problème d'auto-localisation dans les environnements imparfaits.

Auto-localisation en présence d'imperfections de mesures



Sommaire

- V.1 Introduction
 - V.2 Théorie des fonctions de croyance
 - V.3 Problématique
 - V.4 Auto-localisation avec des mesures aberrantes
 - V.5 Simulations
 - V.6 Conclusion
-

Un des objectifs essentiels des algorithmes d'auto-localisation est d'estimer correctement et avec précision les positions des capteurs. Ceci est généralement réalisé en fonction des mesures échangées entre les capteurs. Toutefois, il est possible d'avoir des mesures erronées capables de fausser les résultats de l'estimation. Dans ce chapitre, nous proposons une méthode d'auto-localisation dédiée aux environnements imparfaits. Dans cette méthode, nous supposons avoir les valeurs de fiabilité que nous pourrions associer aux différentes mesures. Le problème se composera alors d'un modèle de mobilité, des informations de connectivité et des mesures de fiabilité associées aux observations. La résolution de ce problème se situe dans le cadre de l'analyse par intervalles et de la théorie des fonctions de croyance. Les positions estimées seront alors des ensembles de pavés, combinant toutes les informations acquises et ayant chacun un poids spécifique. Ainsi, nous commencerons dans ce chapitre par l'introduction des fonctions de croyance. Nous proposerons ensuite un algorithme d'auto-localisation robuste aux imperfections de l'environnement. Nous illustrerons enfin les performances de cette méthode par quelques résultats de simulation sur des données synthétiques.

V.1 Introduction

Les réseaux de capteurs mobiles constituent un domaine de recherche en pleine effervescence. Cette technologie possède des applications potentielles dans différents domaines [Gharavi et Kumar, 2003; Arampatzis *et al.*, 2005; Czubak et Wojtanowski, 2009], notamment les domaines militaire, médical, civil, etc. L'un des points essentiels traités sous cette thématique est le problème d'auto-localisation des capteurs à mobilité incontrôlée. Plusieurs algorithmes ont été proposés afin de résoudre ce problème. Nous en citons les méthodes basées sur les connectivités et les méthodes sans modèle que nous avons présentées dans les chapitres précédents et les méthodes basées sur l'approche séquentielle de Monte-Carlo [Hu et Evans, 2004; Baggio et Langendoen, 2006; Yi *et al.*, 2007].

La plupart des algorithmes existants ont été conçus pour des environnements parfaits où les mesures effectuées sont supposées correctes. Toutefois, les réseaux de capteurs peuvent être déployés dans des environnements hostiles où les signaux échangés peuvent être altérés. Considérons en particulier les méthodes basées sur les mesures de connectivités. Celles-ci engendrent des mesures de proximité en se basant sur les puissances des signaux échangés. L'ajout de bruit, la réflexion des signaux, la panne des capteurs ou autres peuvent changer les valeurs des puissances reçues, ce qui aura pour conséquence de fausser probablement les mesures de connectivités. Dans telles conditions, la fusion des observations peut entraîner une fausse estimation des positions des capteurs.

Dans ce chapitre, nous proposons une méthode d'auto-localisation fiable dans les environnements imparfaits. Cette méthode se base sur les mesures de connectivités, dont les valeurs de fiabilité sont supposées connues. Outre le modèle d'observation, le problème posé profite de la mobilité des capteurs pour raffiner l'estimation. La résolution du problème emploie la théorie des fonctions de croyance, dite aussi théorie de Dempster-Shafer [Dempster, 1967; Shafer, 1976; Smets et Kennes, 1994]. Les informations collectées définissent alors des fonctions de croyance, dont la combinaison aboutit à la solution du problème. A chaque instant, un ensemble de pavés bidimensionnels, ayant chacun un poids spécifique, est défini. Le pavé associé au plus grand poids correspondra à la position recherchée. Toutefois, les autres pavés, comme ce dernier, seront sauvegardés afin d'être utilisés dans l'auto-localisation du capteur à l'instant suivant.

Dans la suite, nous commençons par introduire la théorie des fonctions de croyance. Nous présenterons par la suite la problématique que nous résoudrons à l'aide de cette théorie.

V.2 Théorie des fonctions de croyance

La Théorie des Fonctions de Croyance (TFC), dite également théorie de l'évidence ou encore théorie de Dempster-Shafer, est un outil mathématique permettant la fusion d'information. Dans cette section, nous commencerons par une brève introduction. Nous présenterons ensuite les outils de base de cette théorie.

V.2.1 Introduction à la TFC

La TFC constitue l'un des principaux cadres dédiés au raisonnement avec des informations incertaines et imprécises. Elle a été introduite par G. Shafer en 1976 [Shafer, 1976], sur la base des travaux d'A. Dempster en 1967 [Dempster, 1967]. P. Smets a ensuite décrit un cadre formel pour la TFC indépendant des notions de probabilités [Smets et Kennes, 1994]. Ce cadre est connu sous le nom de **modèle des croyances transférables** ou TBM (pour *Transferable Belief Model*). Le TBM est un modèle non probabiliste proposé d'une part, pour représenter l'imperfection des informations collectées et, d'autre part, pour combiner ces informations. Il comporte ainsi deux niveaux : un niveau **crédal**, pour représenter et combiner les informations, et un niveau **pignistique**, pour prendre une décision. Cette théorie est à l'heure actuelle employée dans divers domaines : fusion multi-capteur, reconnaissance des formes, fusion de classifieurs, etc [Dencœur, 1997; Ristic et Smets, 2005]. Dans ce chapitre, nous utilisons la TFC pour résoudre le problème d'auto-localisation des capteurs dans un environnement imparfait. Nous présenterons alors dans la suite les différentes notions proposées dans le TBM, celles-ci étant nécessaires à la mise en œuvre de la méthode développée dans ce chapitre.

V.2.2 Représentation des informations dans le TBM

Supposons un système quelconque dont on cherche à déterminer l'état x . Soit $\Omega = \{\omega_1, \dots, \omega_n\}$ le domaine de définition de x . Ω est connu sous le nom de **cadre de discernement**. Les valeurs $\{\omega_i\}_{i=1}^n$, susceptibles d'être prises par x , sont discrètes. Elles sont dites parfois **hypothèses**. Notons 2^Ω l'ensemble des parties de Ω :

$$2^\Omega = \{\emptyset, \{\omega_1\}, \dots, \Omega\}. \quad (\text{V.1})$$

L'ensemble vide \emptyset dépendra de la définition du système. Si Ω contient toutes les valeurs possibles de x , alors Ω représente l'événement certain. \emptyset désigne alors l'événement impossible et la croyance qui lui est affectée doit être nulle. Dans le cas contraire, si Ω ne contient pas toutes les valeurs possibles de x , \emptyset peut être défini par l'ensemble de toutes les valeurs de x non incluses dans Ω . Toute mesure fournissant des informations sur l'état x est considérée comme une source d'informations dans la TFC. Celle-ci offrira des croyances qui peuvent être représentées de différentes manières dans le cadre du TBM. Dans cette section, nous exposerons les représentations par les fonctions de masse, les fonctions de croyance, les fonctions de plausibilité, l'implicabilité et la communalité.

◇ Fonction de masse de croyance

Plusieurs formulations des croyances ont été proposées dans le TBM. La formulation la plus utilisée est celle de la **Fonction de Masse de croyance** (FM), notée généralement m_S^Ω et définie par

$$\begin{aligned} m_S^\Omega : 2^\Omega &\rightarrow [0, 1] \\ A &\mapsto m_S^\Omega(A), \end{aligned} \quad (\text{V.2})$$

V. AUTO-LOCALISATION EN PRÉSENCE D'IMPERFECTIONS DE MESURES

$A \in 2^\Omega$	catégorique	vide	bayésienne	normale
\emptyset	0	0	0	0
$\{\omega_1\}$	0	0	0.2	0.09
$\{\omega_2\}$	0	0	0.3	0.31
$\{\omega_3\}$	0	0	0.5	0.1
$\{\omega_1, \omega_2\}$	0	0	0	0.23
$\{\omega_1, \omega_3\}$	1	0	0	0.15
$\{\omega_2, \omega_3\}$	0	0	0	0.08
$\{\omega_1, \omega_2, \omega_3\}$	0	1	0	0.04

TAB. V.1: Quatre exemples de FM's définies sur $\Omega = \{\omega_1, \omega_2, \omega_3\}$.

où S représente la source d'information dont est issue m_S^Ω . Toute FM doit vérifier la condition suivante :

$$\sum_{A \in 2^\Omega} m_S^\Omega(A) = 1. \quad (\text{V.3})$$

La masse $m_S^\Omega(A)$ correspond à la part de croyance allouée par la source S à l'hypothèse selon laquelle l'état du système x est dans A . Dans la suite, la notation simplifiée m^Ω sera adoptée lorsqu'il n'est pas nécessaire de préciser la source d'information.

Tout sous-ensemble A de Ω tel que $m^\Omega(A) > 0$ est appelé **ensemble focal**. Soient A_1, \dots, A_r les ensembles focaux de m^Ω . Quelques cas particuliers de FM's peuvent être envisagés selon les ensembles focaux. Ainsi, une FM m^Ω est dite :

- **catégorique**, si $r = 1$. Elle sera alors assimilée à $A_{r=1}$.
- **vide**, si le cadre de discernement est le seul élément focal de cette FM. Elle représentera ainsi l'ignorance totale.
- **bayésienne**, si les ensembles focaux sont des singletons, $|A_i| = 1$ pour $i = 1, \dots, r$. Elle sera alors assimilée à une distribution de probabilité.
- **normale**, si la masse associée à l'ensemble vide est nulle, $m^\Omega(\emptyset) = 0$.
- **sous-normale**, si $m^\Omega(\emptyset) > 0$.

TAB. V.1 montre quatre exemples de FM's catégorique, vide, bayésienne et normale.

Généralement, le cadre de discernement est supposé fini, mais la TFC peut être étendue au cas infini [Petit-Renaud et Dencoux, 2004; Yager, 2004]. Si $\Omega = \mathbb{R}$, les éléments focaux seront des intervalles [Moore, 1979; Jaulin *et al.*, 2001]. Dans le cas multidimensionnel où $\Omega = \mathbb{R}^n$, les éléments focaux seront des pavés de dimensions n .

◇ Autres représentations

A partir de la FM, d'autres formulations équivalentes peuvent être déduites. Nous introduisons alors la **fonction de croyance**, la **fonction de plausibilité**, l'**implicabilité** et la **communalité**, toutes définies de 2^Ω dans $[0, 1]$.

- **Fonction de croyance** : La fonction de croyance, dite aussi **fonction de crédibilité**, est notée bel^Ω . Celle-ci, associée à la FM m^Ω , est définie par

$$bel^\Omega(A) = \sum_{\emptyset \neq B \subseteq A} m^\Omega(B), \forall A \subseteq \Omega. \quad (V.4)$$

La quantité $bel^\Omega(A)$ représente le degré de croyance en l'hypothèse « $x \in A$ ». Réciproquement, m^Ω peut être obtenue à partir de la fonction de croyance par la formule suivante,

$$m^\Omega(A) = \begin{cases} \sum_{\emptyset \neq B \subseteq A} (-1)^{|A|-|B|} bel^\Omega(B), & \forall A \subseteq \Omega, A \neq \emptyset, \\ 1 - bel^\Omega(\Omega), & \text{si } A = \emptyset. \end{cases} \quad (V.5)$$

- **Fonction de plausibilité** : La fonction de plausibilité, notée pl^Ω , associée à m^Ω , est définie par

$$pl^\Omega(A) = \sum_{B \cap A \neq \emptyset} m^\Omega(B), \forall A \subseteq \Omega. \quad (V.6)$$

La quantité $pl^\Omega(A)$ représente le degré maximal de croyance susceptible d'être alloué à l'hypothèse « $x \in A$ », après intégration de nouvelles informations. Réciproquement, m^Ω peut être obtenue à partir de la fonction de plausibilité par la formule suivante,

$$m^\Omega(A) = \begin{cases} \sum_{\emptyset \neq B \subseteq A} (-1)^{|A|-|B|+1} pl^\Omega(\overline{B}), & \forall A \subseteq \Omega, A \neq \emptyset, \\ 1 - pl^\Omega(\Omega), & \text{si } A = \emptyset. \end{cases} \quad (V.7)$$

où \overline{B} est le complémentaire de B dans Ω .

- **Implicabilité** : L'implicabilité de $A \subseteq \Omega$ est la somme des masses associées aux sous-ensembles de A . Elle est donnée par

$$b^\Omega(A) = \sum_{B \subseteq A} m^\Omega(B) = bel^\Omega(A) + m^\Omega(\emptyset). \quad (V.8)$$

- **Communalité** : La communalité d'un élément $A \subseteq \Omega$ est la somme des masses associées aux surensembles de A ,

$$q^\Omega(A) = \sum_{B \supseteq A} m^\Omega(B). \quad (V.9)$$

TAB. V.2 regroupe les valeurs de croyance, plausibilité, implicabilité et communalité associées à la FM normale du TAB. V.1. Notons que $bel^\Omega(A) \leq pl^\Omega(A)$ pour tout sous-ensemble A de Ω . Les valeurs $bel^\Omega(A)$ et $pl^\Omega(A)$ peuvent être considérées respectivement comme les bornes inférieure et supérieure de l'intervalle auquel appartient la probabilité associée à l'hypothèse « $x \in A$ », $P(A)$.

V. AUTO-LOCALISATION EN PRÉSENCE D'IMPERFECTIONS DE MESURES

$A \in 2^\Omega$	m^Ω	bel^Ω	pl^Ω	b^Ω	q^Ω
\emptyset	0	0	0	0	1
$\{\omega_1\}$	0.09	0.09	0.51	0.09	0.51
$\{\omega_2\}$	0.31	0.31	0.66	0.31	0.66
$\{\omega_3\}$	0.1	0.1	0.37	0.1	0.37
$\{\omega_1, \omega_2\}$	0.23	0.63	0.9	0.63	0.27
$\{\omega_1, \omega_3\}$	0.15	0.34	0.69	0.34	0.19
$\{\omega_2, \omega_3\}$	0.08	0.49	0.91	0.49	0.12
$\{\omega_1, \omega_2, \omega_3\}$	0.04	1	1	1	0.04

TAB. V.2: Valeurs de croyance, plausibilité, implicabilité et communalité associées à la FM normale du TAB. V.1.

◇ Affaiblissement d'une fonction de masse

La source d'information dont est extraite une FM peut ne pas être totalement fiable. Dans ce cas, il est possible d'introduire un coefficient d'affaiblissement $(1-\alpha) \in [0, 1]$ à la FM. Ce coefficient représente un élément de métaconnaissance concernant la fiabilité de la source dont est issue la FM [Smets et Kennes, 1994]. L'utilisation d'un tel coefficient permet de transférer une partie de la croyance vers l'ensemble Ω . Ainsi, nous obtenons une FM affaiblie, notée $m^{\alpha, \Omega}$, que nous déduisons de m^Ω de la manière suivante,

$$m^{\alpha, \Omega}(A) = \begin{cases} (1-\alpha).m^\Omega(A), & \text{si } A \subset \Omega, A \neq \Omega, \\ \alpha + (1-\alpha).m^\Omega(A), & \text{si } A = \Omega. \end{cases} \quad (\text{V.10})$$

V.2.3 Règles de combinaison

Afin de combiner les informations issues de différentes sources, Smets a développé plusieurs opérateurs de combinaison permettant d'obtenir une information plus fiable sur l'état recherché [Smets, 1993; Smets et Kennes, 1994]. Dans cette section, nous présentons différentes règles de combinaison proposées dans le cadre du TBM. Ces règles considèrent que les sources d'informations utilisées sont distinctes.

◇ Règle de combinaison conjonctive

Soient $m_{S_1}^\Omega$ et $m_{S_2}^\Omega$ deux FM issues de deux sources distinctes et fiables S_1 et S_2 , respectivement. Pour raisons de simplicité, nous noterons dans la suite m_1 et m_2 ces deux FM. La règle de combinaison **conjonctive**, notée CRC (pour *Conjunctive Rule of Combination*) et représentée par \odot est définie comme suit,

$$m_1 \odot_2(A) = (m_1 \odot m_2)(A) = \sum_{i,j | B_i \cap C_j = A} m_1(B_i).m_2(C_j), \forall A \subseteq \Omega. \quad (\text{V.11})$$

La CRC est commutative et associative. Cette règle transfère les masses des éléments focaux vers des ensembles de cardinalité plus petite, ayant ainsi un effet de spécialisation. La FM obtenue est ainsi plus informative que m_1 et m_2 .

Si les sources d'informations sont conflictuelles, la FM résultant d'une CRC est sous-normale. Afin d'obtenir une FM normale, il suffit d'appliquer la règle de Dempster, notée \oplus , qui consiste en une CRC suivie d'une étape de normalisation. Elle est définie par la formule suivante,

$$m_{1\oplus 2}(A) = (m_1 \oplus m_2)(A) = \begin{cases} \frac{m_1 \odot_2(A)}{1 - m_1 \odot_2(\emptyset)}, & \text{si } A \subseteq \Omega, A \neq \emptyset, \\ 0, & \text{si } A = \emptyset. \end{cases} \quad (\text{V.12})$$

◇ **Règle de combinaison disjonctive**

La règle de combinaison disjonctive, notée DRC (pour *Disjunctive Rule of Combination*) et représentée par \odot , est utilisée habituellement pour combiner deux FM issues de deux sources d'informations distinctes, dont l'une au moins est fiable sans que l'on sache laquelle. L'application de cette règle à deux FM m_1 et m_2 conduit au résultat suivant,

$$m_{1\odot 2}(A) = (m_1 \odot m_2)(A) = \sum_{i,j|B_i \cup C_j = A} m_1(B_i) \cdot m_2(C_j), \forall A \subseteq \Omega. \quad (\text{V.13})$$

La FM résultant de la DRC possède en général des éléments focaux de cardinalité plus élevée. Cette FM obtenue est alors moins spécialisée que les FM initiales. En conséquence, la DRC a un effet de généralisation induisant une FM moins informative.

V.2.4 Prise de décision

Dans la théorie des fonctions de croyance, plusieurs règles de décision sont utilisées. Nous en citons le maximum de masse de croyance, le maximum de plausibilité, le maximum de probabilité pignistique, etc [Dencœur, 1997]. Vu que les fonctions de masse et de plausibilité ont été explicitées précédemment, nous décrivons, dans cette section, la probabilité pignistique. Considérons une FM m^Ω associée à un cadre de discernement Ω . La probabilité pignistique, notée *BetP*, est une distribution de probabilité qui peut être calculée à partir de m^Ω en utilisant la transformation pignistique introduite par Smets *et al.* [Smets et Kennes, 1994]. Soient A_1, \dots, A_r les ensembles focaux de m^Ω . La transformation pignistique consiste à répartir uniformément la masse de chaque ensemble $A_i, i \in \{1, \dots, r\}$, entre les hypothèses élémentaires qui le composent. Ainsi,

$$BetP(\omega_k) = \sum_{i|A_i \subseteq \Omega, \omega_k \in A_i} \frac{m^\Omega(A_i)}{|A_i|}, \quad (\text{V.14})$$

où $|A_i|$ est le cardinal de A_i . Dans la définition ci-dessus, la FM m^Ω est supposée normale. Dans le cas contraire, où m^Ω est sous-normale, une étape de normalisation est requise avant la transformation pignistique.

Exemple 16. *La transformation pignistique de la FM normale du tableau V.1 fournit la distribution de probabilité suivante,*

$$\begin{cases} \text{Bet}P(\omega_1) = \frac{0.09}{1} + \frac{0.23}{2} + \frac{0.15}{2} + \frac{0.04}{3} = 0.293, \\ \text{Bet}P(\omega_2) = \frac{0.31}{1} + \frac{0.23}{2} + \frac{0.08}{2} + \frac{0.04}{3} = 0.520, \\ \text{Bet}P(\omega_3) = \frac{0.1}{1} + \frac{0.15}{2} + \frac{0.08}{2} + \frac{0.04}{3} = 0.228. \end{cases}$$

Si la décision doit porter sur les singletons du cadre de discernement, l'hypothèse ω_2 sera choisie selon le critère du maximum de la probabilité pignistique. Le choix aurait été similaire si l'on utilisait le critère de maximum de plausibilité.

La définition de la probabilité pignistique est présentée dans (V.15) pour un cadre de discernement fini. Dans le cas où $\Omega = \mathbb{R}$, et où les éléments focaux sont des intervalles $[x]_1, \dots, [x]_r$, la probabilité pignistique est caractérisée par la fonction de densité suivante,

$$\text{Bet}f(x) = \sum_{i=1}^r \mathbb{I}_{[x]_i}(x) \cdot \frac{m^\Omega([x]_i)}{|[x]_i|}, \quad (\text{V.15})$$

où $\mathbb{I}_{[x]_i}(x)$ est la fonction indicatrice explicitant l'appartenance ($\mathbb{I}_{[x]_i}(x) = 1$) ou non ($\mathbb{I}_{[x]_i}(x) = 0$) de x à $[x]_i$ et $|[x]_i|$ est la largeur de l'intervalle $[x]_i$.

V.3 Problématique

La méthode proposée introduit deux types de capteurs : les ancrs, munies de GPS [Hofmann-Wellenhof *et al.*, 2004], et les non-ancres, dits également nœuds, ne connaissant pas leurs positions. Les non-ancres échangent des informations de proximité avec les ancrs afin d'estimer leurs positions. Ainsi, la méthode que nous proposons dans ce chapitre est une méthode basée sur les mesures de connectivités vis-à-vis des ancrs. Toutefois, cette méthode est dédiée aux environnements imparfaits où les mesures réalisées sont accordées des valeurs de fiabilité. La méthode proposée profite également de la mobilité des capteurs pour raffiner l'estimation. Dans cette section, nous commençons tout d'abord par exposer le modèle de mobilité. Nous présenterons ensuite le modèle d'observation tenant compte des mesures aberrantes.

V.3.1 Modèle de mobilité

La méthode proposée tient compte de la mobilité des nœuds pour améliorer l'estimation des positions. Similairement aux méthodes présentées dans les chapitres III et IV, cette méthode propose un modèle de mobilité assez général pour couvrir un très grand nombre d'applications. Inspiré du modèle de mobilité de marche aléatoire [Camp *et al.*, 2002], ce modèle suppose que seule la vitesse maximale des nœuds est connue. Ainsi, les nœuds ont la liberté de bouger dans une direction quelconque avec une vitesse également quelconque mais limitée à une valeur maximale. Soient $\mathbf{x}_j(t) = (x_{j,1}(t), x_{j,2}(t))$ et $\mathbf{x}_j(t-1) = (x_{j,1}(t-1), x_{j,2}(t-1))$ les positions d'un nœud j considéré, respectivement

aux instants t et $t - 1$. Selon le modèle proposé, il est possible d'exprimer $\mathbf{x}_j(t)$ en fonction de $\mathbf{x}_j(t - 1)$ de la manière suivante,

$$\begin{cases} x_{j,1}(t) = x_{j,1}(t - 1) + \Delta t.v_j.\cos(\theta_j), \\ x_{j,2}(t) = x_{j,2}(t - 1) + \Delta t.v_j.\sin(\theta_j), \end{cases} \quad (\text{V.16})$$

où Δt est la durée entre deux pas de temps consécutifs, θ_j est la direction du nœud j entre $t - 1$ et t et v_j est la vitesse de ce dernier également entre ces deux instants. v_j et θ_j sont deux variables quelconques comprises respectivement entre 0 et $v_{max,j}$ et 0 et $2.\pi$, $v_{max,j}$ étant la vitesse maximale du nœud j . La relation (V.16) peut être reformulée comme suit,

$$(x_{j,1}(t) - x_{j,1}(t - 1))^2 + (x_{j,2}(t) - x_{j,2}(t - 1))^2 = (\Delta t.v_j)^2, \quad (\text{V.17})$$

avec $v_j \in [0, v_{max,j}]$. Toute information supplémentaire sur la mobilité des nœuds peut être utilisée afin de raffiner ce modèle. Si nous supposons connaître la position ponctuelle $\mathbf{x}_j(t - 1)$ du nœud j à l'instant $t - 1$, la position courante tombera alors dans le disque centré sur $\mathbf{x}_j(t - 1)$ et ayant $\Delta t.v_{max,j}$ comme rayon.

V.3.2 Modèle d'observation

Dans la méthode que nous proposons, les nœuds mobiles associent des mesures de connectivités au modèle de mobilité afin de déterminer leurs positions. Ces mesures, présentées dans la section III.4.1, se basent sur les valeurs des puissances des signaux échangés. Soit $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)$ la puissance, à la réception, du signal émis à l'instant t par une ancre i et reçu par le nœud j . Selon le modèle d'Okumura-Hata [Medeisis et Kajackas, 2000; Nadir et al., 2008], cette puissance décroît avec l'augmentation de la distance parcourue par le signal comme suit,

$$\rho_{\mathbf{a}_i, \mathbf{x}_j}(t) = \rho_0 - 10.n_P.\log_{10} \frac{d(\mathbf{a}_i(t), \mathbf{x}_j(t))}{d_0} + \varepsilon_{i,j}(t), \quad (\text{V.18})$$

où $\mathbf{a}_i(t)$ est la position de l'ancre i à l'instant t , $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)$ est en dBm , ρ_0 en dBm est la puissance mesurée à une certaine distance parcourue d_0 , n_P est le paramètre de perte correspondant au canal, $d(\mathbf{a}_i(t), \mathbf{x}_j(t))$ est la distance entre $\mathbf{a}_i(t)$ et $\mathbf{x}_j(t)$ et $\varepsilon_{i,j}(t)$ est le bruit de mesure.

Soit ρ_r la valeur de la puissance en dBm correspondant à la portée de communication r selon ce modèle, tout en négligeant le bruit du canal. Le nœud j considéré, recevant à l'instant t des signaux émis par des ancres, compare les puissances mesurées de ces signaux à ρ_r . Si $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t) \geq \rho_r$, alors $d(\mathbf{a}_i(t), \mathbf{x}_j(t)) \leq r$ et l'ancre i est supposée à proximité du nœud j . Dans le cas contraire, l'ancre ne sera pas prise en compte. Soit $I_j(t)$ l'ensemble des indices des ancres situées à proximité du nœud j à l'instant t , dites ancres détectées. Toutes ces ancres sont situées théoriquement à une distance du nœud j inférieure à la portée de communication r . D'où les contraintes d'observation définies comme suit,

$$(x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq r^2, \quad i \in I_j(t). \quad (\text{V.19})$$

V. AUTO-LOCALISATION EN PRÉSENCE D'IMPERFECTIONS DE MESURES

Le nœud j appartiendra alors à l'intersection des disques centrés sur les ancrs détectées et ayant r comme rayons.

En pratique, il est possible d'obtenir des mesures aberrantes ne suivant pas le modèle proposé. En effet, une ancre supposée à proximité du nœud j peut être située plus loin et vice-versa. Ceci est essentiellement dû au fait que les puissances des signaux ne suivent pas exactement le modèle d'Okumura-Hata. Soient $\rho_{\mathbf{a}_i, \mathbf{x}_j}^*(t)$ et $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)$ respectivement les valeurs théorique et mesurée de la puissance correspondant à la distance $d(\mathbf{a}_i(t), \mathbf{x}_j(t))$. Quatre cas peuvent se présenter en fonction des valeurs de ces puissances :

- **Cas 1** : $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t) \geq \rho_r$ et $\rho_{\mathbf{a}_i, \mathbf{x}_j}^*(t) \geq \rho_r$. L'ancre i est alors supposée à proximité du nœud j , ce qui est juste. Nous obtenons ainsi une mesure correcte.
- **Cas 2** : $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t) < \rho_r$ et $\rho_{\mathbf{a}_i, \mathbf{x}_j}^*(t) < \rho_r$. L'ancre i est alors supposée hors portée du nœud j , ce qui est toujours juste. Pas de mesure générée.
- **Cas 3** : $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t) < \rho_r \leq \rho_{\mathbf{a}_i, \mathbf{x}_j}^*(t)$. L'ancre i est supposée hors portée du nœud j , alors qu'elle est à proximité de lui. Dans ce cas, l'ancre i n'est pas considérée et une mesure est omise. Ceci conduit à une perte dans la précision de l'estimation. En revanche, pas de génération de fausses contraintes.
- **Cas 4** : $\rho_{\mathbf{a}_i, \mathbf{x}_j}(t) \geq \rho_r > \rho_{\mathbf{a}_i, \mathbf{x}_j}^*(t)$. L'ancre i est supposée à proximité du nœud j , alors qu'elle ne l'est pas. Dans ce cas, une mesure fautive est conçue, conduisant à une estimation incorrecte de la position du nœud. Dans la suite, nous appellerons ce type de mesures « mesures aberrantes » ou « mesures erronées ».

Dans la méthode que nous proposons, nous tenons compte des mesures aberrantes. Nous supposons alors que les contraintes d'observation sont affectées de mesures de fiabilité β_i , $i \in I_j(t)$, comprises entre 0 et 1. $\beta_i = 1$ signifie que la mesure associée à l'ancre i est complètement fiable. Ainsi, si l'on considère une ancre détectée i avec une fiabilité égale à β_i , le nœud j se situe à l'intérieur du disque centré sur l'ancre i et de rayon r avec une probabilité β_i . En revanche, il y a une probabilité de $1 - \beta_i$ que la mesure ne soit pas correcte et par conséquent, que la contrainte ne soit pas applicable. Le modèle d'observation revient alors aux mesures de connectivités, auxquelles sont associées des valeurs de fiabilité.

Remarque 5. *Dans ce travail, nous supposons que les mesures de fiabilité sont acquises. En d'autres termes, nous ne développons pas de méthodes pour le calcul de ces fiabilités. En revanche, nous présentons une suggestion de calcul basée sur les valeurs des puissances mesurées. Selon cette technique, nous accordons autant de confiance à la mesure que la puissance mesurée est supérieure au seuil ρ_r . Nous admettons que lorsque la puissance est beaucoup plus grande que le seuil, il y a plus de chance que la vraie puissance tombe au-dessus du seuil. Soient $\beta_{r,j}$ et $\bar{\beta}_j$ respectivement les valeurs minimale et maximale de fiabilité que nous pourrions accorder aux mesures. Celles-ci correspondent respectivement au seuil ρ_r et à une puissance maximale, notée \bar{p}_j . La puissance \bar{p}_j peut être choisie égale à la puissance initiale commune d'émission des signaux, à un multiple du seuil ρ_r ou autre. Dans tous les cas, \bar{p}_j doit être supérieure au maximum des*

puissances mesurées, $\bar{\rho}_j \geq \max_{i \in I_j(t)} \{\rho_{\mathbf{a}_i, \mathbf{x}_j}(t)\}$. Ainsi, selon la méthode proposée, les fiabilités sont proportionnelles aux puissances mesurées comme suit,

$$\beta_i = \beta_{r,j} + (\rho_{\mathbf{a}_i, \mathbf{x}_j}(t) - \rho_r) \cdot \frac{\bar{\beta}_j - \beta_{r,j}}{\bar{\rho}_j - \rho_r}. \quad (\text{V.20})$$

Notons que cette technique reste une suggestion. D'autres méthodes plus pertinentes peuvent être adaptées, celles-ci étant basées sur une analyse de la communication des informations dans le canal.

V.4 Auto-localisation avec des mesures aberrantes

Le problème d'auto-localisation est défini par l'association des modèles de mobilité et d'observation présentés dans la section précédente. Ainsi, pour tout nœud j du réseau, $j \in \{1, \dots, N_x\}$, le problème consiste à calculer les valeurs de $\mathbf{x}_j(t) = (x_{j,1}(t), x_{j,2}(t))$ qui vérifient le système suivant,

$$\begin{cases} (x_{j,1}(t) - x_{j,1}(t-1))^2 + (x_{j,2}(t) - x_{j,2}(t-1))^2 = (\Delta t \cdot v_j)^2, \\ \text{avec} \left| \begin{array}{l} (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq r^2, \text{ de fiabilité } \beta_i, i \in I_j(t), \\ v_j \in [0, v_{max,j}]. \end{array} \right. \end{cases} \quad (\text{V.21})$$

Dans ce chapitre, nous résolvons le problème à l'aide de la Théorie des Fonctions de Croyance (TFC), combinée à la théorie des intervalles [Moore, 1979; Jaulin *et al.*, 2001]. La méthode proposée consiste ainsi à compter chaque observation pour une source d'informations. Les sources obtenues sont distinctes vu que chaque mesure est réalisée vis-à-vis d'une ancre distincte des autres. L'idée de base revient alors à combiner ces sources d'informations afin de retrouver les positions des nœuds. Ces dernières seront représentées par des ensembles de pavés bidimensionnels, ayant chacun un poids spécifique. Dans la suite, nous décrivons en premier la méthode, basée sur la combinaison conjonctive des sources. Nous proposons ensuite une variante de cette méthode.

V.4.1 Description de la méthode basée sur la TFC

La technique proposée prend chaque mesure comme une source d'informations. Ainsi, pour toute contrainte d'observation, une Fonction de Masse (FM) est définie. Le modèle de mobilité est également utilisé pour définir une FM supplémentaire. En effet, les pavés précédents seront propagés à l'aide du modèle de mobilité, fournissant alors une source d'informations. Le nombre de FMs considérées, dites FMs élémentaires, est égal au nombre de contraintes d'observations incrémenté de 1, $|I_j(t)| + 1$. Ainsi, le problème reviendra à combiner ces FMs en une FM finale, possédant plusieurs ensembles focaux en forme de pavés bidimensionnels. Le cadre de discernement Ω de toutes les FMs correspondra à la plus grande zone où les capteurs sont susceptibles de bouger. Au cas où cette zone n'est pas limitée, Ω est pris égal à tout le plan \mathbb{R}^2 , vu que tous les capteurs sont supposés mobiles dans le même plan.

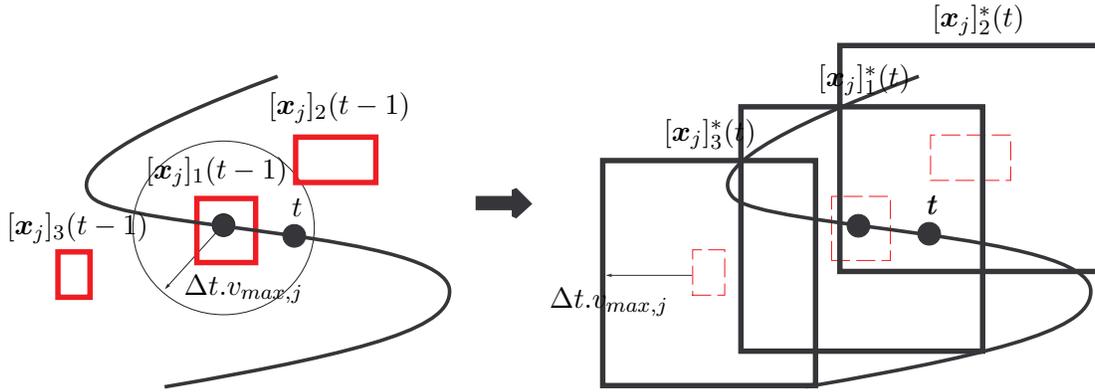


FIG. V.1: Propagation des ensembles focaux précédents.

Considérons le problème d'auto-localisation d'un nœud j , $j \in \{1, \dots, N_x\}$. Notons m_{t-1} la FM finale obtenue à l'instant $t-1$ pour le nœud j et $K(t-1)$ le nombre d'ensembles focaux de cette FM. Ces derniers seront notés $[x_j]_1(t-1), \dots, [x_j]_{K(t-1)}(t-1)$. La première étape de la méthode consiste à propager les ensembles focaux obtenus de l'instant $t-1$ à l'instant courant. Cette étape est réalisée à l'aide du modèle de mobilité, reformulé en fonction des intervalles comme suit,

$$([x_{j,1}](t) - [x_{j,1}](t-1))^2 + ([x_{j,2}](t) - [x_{j,2}](t-1))^2 = [0, (\Delta t \cdot v_{max,j})^2]. \quad (V.22)$$

Les pavés obtenus suite à la propagation sont alors définis de la manière suivante,

$$\begin{cases} [x_{j,1}]_k^*(t) = [x_{j,1}]_k(t-1) + [-\Delta t \cdot v_{max,j}, +\Delta t \cdot v_{max,j}], \\ [x_{j,2}]_k^*(t) = [x_{j,2}]_k(t-1) + [-\Delta t \cdot v_{max,j}, +\Delta t \cdot v_{max,j}]. \end{cases} \quad (V.23)$$

Les masses des ensembles focaux précédents seront transmis aux pavés obtenus, vu que la relation liant les deux est bijective. Une nouvelle FM, appelée FM de mobilité et notée m_t^* , est alors générée comme suit,

$$m_t^*([x_j]_k^*(t)) = m_{t-1}([x_j]_k(t-1)), \quad k \in \{1, \dots, K(t-1)\}. \quad (V.24)$$

Les ensembles focaux de cette FM seront donnés par les pavés $[x_j]_k^*(t)$, $k \in \{1, \dots, K(t-1)\}$. FIG. V.1 illustre la propagation de la FM m_{t-1} ayant trois ensembles focaux.

D'autres FMs sont obtenues à l'aide des contraintes d'observation. Une mesure fiable vis-à-vis de l'ancre i , $i \in I_j(t)$, produit une FM m_i ayant un ensemble focal unique. Ce dernier correspond au disque centré sur l'ancre i et ayant r comme rayon. Dans le contexte ensembliste, le disque sera remplacé par le pavé auquel il appartient. Ainsi, la FM m_i sera catégorique ayant ce carré comme l'unique ensemble focal. Notons par A_i , $i \in I_j(t)$, les pavés couvrant les disques d'observation. Ces derniers, de côtés $2 \cdot r$, seront centrés respectivement sur les ancres $\mathbf{a}_i(t)$,

$$A_i = [a_{i,1}(t) - r, a_{i,1}(t) + r] \times [a_{i,2}(t) - r, a_{i,2}(t) + r], \quad i \in I_j(t). \quad (V.25)$$

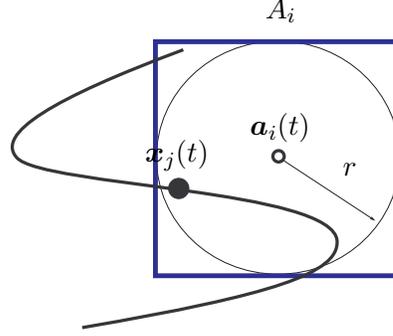


FIG. V.2: Illustration d'une FM catégorique issue d'une source fiable.

La FM correspondant à une mesure fiable i , $i \in I_j(t)$, sera donnée par

$$m_i(A) = \begin{cases} 1, & \text{si } A = A_i, \\ 0, & \text{sinon.} \end{cases} \quad (\text{V.26})$$

Une illustration d'une telle FM est donnée dans FIG. V.2. Or, puisque les sources ne sont pas fiables, les FMs générées seront affaiblies selon l'équation (V.10). Les coefficients d'affaiblissement α_i sont exprimés en fonction des fiabilités des mesures, $\alpha_i = 1 - \beta_i$. Les nouvelles FMs, dites FMs d'observation, seront définies comme suit,

$$m_i^\alpha(A) = \begin{cases} \beta_i, & \text{si } A = A_i, \\ 1 - \beta_i, & \text{si } A = \Omega, \\ 0, & \text{sinon.} \end{cases} \quad (\text{V.27})$$

Ainsi, une partie des croyances est transférée à tout le plan Ω . Les FMs obtenues possèdent chacune deux ensembles focaux, le pavé A_i et tout l'ensemble Ω .

Le problème d'auto-localisation se résume ainsi à la combinaison, selon la règle conjonctive, des différentes FMs obtenues m_t^* et m_i^α , $i \in I_j(t)$. La FM finale, notée m_t , sera plus informative que les FMs individuellement considérées. Afin de calculer m_t , il faut déterminer ses ensembles focaux et ensuite leur attribuer les masses adéquates. Soient \mathcal{F}^* et \mathcal{F}_i les ensembles regroupant les éléments focaux respectifs des FMs m_t^* et m_i^α , $i \in I_j(t)$. Alors $\mathcal{F}^* = \{[\mathbf{x}_j]_1^*, \dots, [\mathbf{x}_j]_{K(t-1)}^*\}$ et $\mathcal{F}_i = \{A_i, \Omega\}$. Puisque m_t résulte de la combinaison conjonctive des FMs élémentaires, ses ensembles focaux résultent de l'intersection des différentes combinaisons des éléments focaux de ces FMs. Chacune des combinaisons regroupera $|I_j(t)| + 1$ éléments focaux, chacun correspondant à l'une des FM élémentaires considérées. Soit $\mathcal{C} = (\mathcal{C}_\ell)_\ell$ l'ensemble de toutes les combinaisons possibles. \mathcal{C} est défini comme suit,

$$\mathcal{C} = \begin{pmatrix} \mathcal{C}_1 \\ \mathcal{C}_2 \\ \vdots \\ \mathcal{C}_{K_C} \end{pmatrix} = \begin{pmatrix} A_{i_1} & A_{i_2} & \cdots & A_{i_{|I_j(t)|}} & [\mathbf{x}_j]_1^*(t) \\ A_{i_1} & A_{i_2} & \cdots & A_{i_{|I_j(t)|}} & [\mathbf{x}_j]_2^*(t) \\ \vdots & \vdots & & \vdots & \vdots \\ \Omega & \Omega & \cdots & \Omega & [\mathbf{x}_j]_{K(t-1)}^*(t) \end{pmatrix}, \quad (\text{V.28})$$

V. AUTO-LOCALISATION EN PRÉSENCE D'IMPERFECTIONS DE MESURES

où i_s représente l'indice numéro s dans l'ensemble d'indices $I_j(t)$ des ancrs détectées. Le nombre de combinaisons possibles est alors égal à $K_C = K(t-1) \cdot 2^{|I_j(t)|}$. Le pavé résultant d'une combinaison \mathcal{C}_ℓ est donné par l'intersection de tous les ensembles focaux qui y sont considérés, $B_\ell = \bigcap_{s=1}^{|I_j(t)|+1} \mathcal{C}_{\ell,s}$. Selon la règle de combinaison conjonctive, la masse du pavé B_ℓ est donnée par

$$m_t(B_\ell) = m_t^*(\mathcal{C}_{\ell,|I_j(t)|+1}) \cdot \prod_{s=1}^{|I_j(t)|} m_{i_s}^\alpha(\mathcal{C}_{\ell,s}). \quad (\text{V.29})$$

Notons qu'en cas de conflit, des ensembles vides peuvent être obtenus. Dans tels cas, il est possible d'appliquer la règle de Dempster, où la règle de combinaison conjonctive est suivie d'une normalisation mettant à zéro la masse de l'ensemble vide.

Tout pavé non vide B_ℓ définit un ensemble focal de la FM normale finale. Toutefois, vu que généralement le nombre d'ensembles focaux tend à croître énormément, nous proposons dans notre méthode de limiter ce nombre à une certaine valeur $K(t)$. Cette dernière est soit fixée à l'avance pour tous les pas de temps, soit déterminée à chaque pas de temps tout en ne considérant que les ensembles focaux ayant une masse supérieure à une certaine limite inférieure. La FM finale ressortie aura les $K(t)$ pavés ayant les masses les plus élevées comme ensembles focaux. Une normalisation est requise par la suite afin d'avoir une somme des masses sélectionnées égale à l'unité. Nous noterons $[\mathbf{x}_j]_k(t)$, $k \in K(t)$, les ensembles focaux de la vraie FM finale m_t . La position ressortie à l'instant t correspond au pavé $[\mathbf{x}_j]_k(t)$ ayant la masse la plus élevée. Celui-ci sera sauvegardé avec les autres pour être utilisés dans l'estimation de la position du nœud à l'instant suivant.

Exemple 17. *Supposons le problème d'auto-localisation d'un nœud j comprenant trois mesures d'observation associées aux ancrs \mathbf{a}_1 , \mathbf{a}_2 et \mathbf{a}_3 avec $\beta_1 = 0.9$, $\beta_2 = 0.8$ et $\beta_3 = 0.2$. Nous supposons en outre avoir deux ensembles focaux précédents $[\mathbf{x}_j]_1(t-1)$ et $[\mathbf{x}_j]_2(t-1)$, ayant 0.7 et 0.3 comme masses respectives. FIG. V.3 illustre les différents ensembles focaux élémentaires correspondant à ce problème à l'instant courant A_1 , A_2 , A_3 , Ω , $[\mathbf{x}_j]_1^*$ et $[\mathbf{x}_j]_2^*$. Pour un tel problème, le nombre de combinaisons possibles est égal à $2^3 \cdot 2 = 16$. TAB. V.3 regroupe les combinaisons possibles, les ensembles focaux et les masses finales sous-normales m_t^\dagger et normales m_t leurs correspondants. Le tableau montre que la FM possède 7 ensembles focaux numérotés sur la figure V.3. L'ensemble focal ayant la meilleure masse est le pavé résultant de l'intersection des contraintes 1 et 2 avec le pavé de mobilité $[\mathbf{x}_j]_1^*$. Celui-ci est illustré en trait plein rouge sur la figure. Les autres ensembles focaux sont montrés en trait rouge interrompu et en trait plein noir (pavés de mobilité). Si l'on limite les ensembles focaux sélectionnés à ceux ayant une masse supérieure à 0.1, seuls deux ensembles emboîtés resteront dans la liste : $[\mathbf{x}_j]_1(t) = A_1 \cap A_2 \cap [\mathbf{x}_j]_1^*$ et $[\mathbf{x}_j]_2(t) = A_1 \cap [\mathbf{x}_j]_1^*$, avec les masses finales normalisées 0.8 et 0.2 respectivement. Ainsi, la position fournie à l'instant t correspondra au pavé $[\mathbf{x}_j]_1(t)$. Ce dernier ainsi que $[\mathbf{x}_j]_2(t)$ seront sauvegardés pour intervenir dans la localisation à l'instant suivant.*

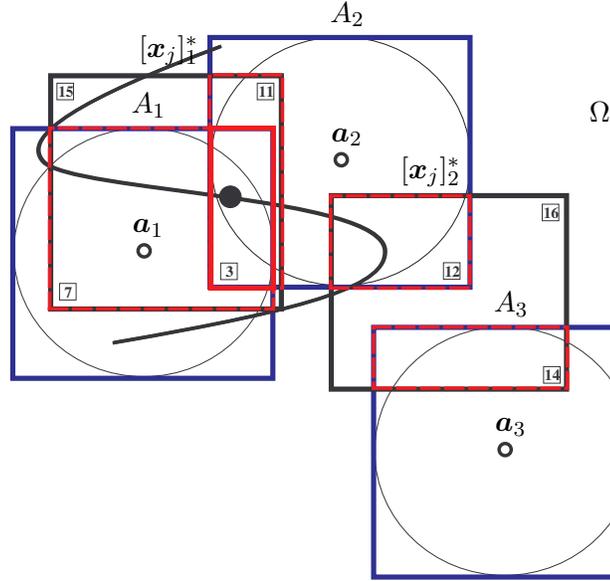


FIG. V.3: Exemple d'un problème d'auto-localisation.

ℓ	\mathcal{C}_ℓ	B_ℓ	$m_t^+(B_\ell)$	$m_t(B_\ell)$
1	$A_1, A_2, A_3, [\mathbf{x}_j]_1^*$	\emptyset	0.1008	0
2	$A_1, A_2, A_3, [\mathbf{x}_j]_2^*$	\emptyset	0.0432	0
3	$A_1, A_2, \Omega, [\mathbf{x}_j]_1^*$	$A_1 \cap A_2 \cap [\mathbf{x}_j]_1^*$	0.4032	0.689
4	$A_1, A_2, \Omega, [\mathbf{x}_j]_2^*$	\emptyset	0.1728	0
5	$A_1, \Omega, A_3, [\mathbf{x}_j]_1^*$	\emptyset	0.0252	0
6	$A_1, \Omega, A_3, [\mathbf{x}_j]_2^*$	\emptyset	0.0108	0
7	$A_1, \Omega, \Omega, [\mathbf{x}_j]_1^*$	$A_1 \cap [\mathbf{x}_j]_1^*$	0.1008	0.172
8	$A_1, \Omega, \Omega, [\mathbf{x}_j]_2^*$	\emptyset	0.0432	0
9	$\Omega, A_2, A_3, [\mathbf{x}_j]_1^*$	\emptyset	0.0112	0
10	$\Omega, A_2, A_3, [\mathbf{x}_j]_2^*$	\emptyset	0.0048	0
11	$\Omega, A_2, \Omega, [\mathbf{x}_j]_1^*$	$A_2 \cap [\mathbf{x}_j]_1^*$	0.0448	0.077
12	$\Omega, A_2, \Omega, [\mathbf{x}_j]_2^*$	$A_2 \cap [\mathbf{x}_j]_2^*$	0.0192	0.033
13	$\Omega, \Omega, A_3, [\mathbf{x}_j]_1^*$	\emptyset	0.0028	0
14	$\Omega, \Omega, A_3, [\mathbf{x}_j]_2^*$	$A_3 \cap [\mathbf{x}_j]_2^*$	0.0012	0.002
15	$\Omega, \Omega, \Omega, [\mathbf{x}_j]_1^*$	$[\mathbf{x}_j]_1^*$	0.0112	0.019
16	$\Omega, \Omega, \Omega, [\mathbf{x}_j]_2^*$	$[\mathbf{x}_j]_2^*$	0.0048	0.008

TAB. V.3: Combinaisons, ensembles focaux et masses correspondant au problème de FIG. V.3.

V.4.2 Variante de la méthode

La méthode proposée effectue une combinaison de toutes les informations collectées. Toutefois, celle-ci relâche les contraintes circulaires d'observation en les remplaçant par des équations de carrés. Dans cette section, nous proposons une variante de la méthode permettant de tenir compte des équations de disques. Celle-ci comprend trois étapes. La première consiste à propager les ensembles focaux issus du pas de temps précédent à l'aide de l'équation (V.23). Cette étape produit $K(t-1)$ pavés ayant chacun une masse spécifique. La FM de mobilité résultante est notée m_t^* . La deuxième étape consiste à combiner uniquement les mesures d'observation. Cette étape fournit une FM d'observation, notée m_o , qui sera combinée à m_t^* dans la troisième étape.

Dans le cadre de la TFC, une mesure d'observation de fiabilité β produit soit la contrainte de disque avec une masse β , soit tout le cadre de discernement avec une masse $1 - \beta$. Ceci est équivalent à l'utilisation ou non de la contrainte d'observation. Afin de combiner les observations, l'intersection des pavés présentée précédemment sera remplacée par une intersection de disques. Pratiquement, nous proposons de générer une matrice binaire de combinaisons où un 1 ou un 0 à la colonne s correspondent à la sélection ou non de la contrainte numéro i_s . Notons \mathcal{C}_o cette matrice. Alors,

$$\mathcal{C}_o = \begin{pmatrix} \mathcal{C}_{o,1} \\ \mathcal{C}_{o,2} \\ \vdots \\ \mathcal{C}_{o,K_o} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}. \quad (\text{V.30})$$

Le nombre de combinaison est égal à $K_o = 2^{|I_j(t)|}$. Une combinaison $\mathcal{C}_{o,\ell}$ produit un pavé $B_{o,\ell}$, issu de l'intersection des contraintes représentées par un 1 dans $\mathcal{C}_{o,\ell}$. La masse de ce pavé est donnée par

$$m_o(B_{o,\ell}) = \prod_{s=1}^{|I_j(t)|} q_{\ell,s}, \quad (\text{V.31})$$

avec

$$q_{\ell,s} = \begin{cases} \beta_{i_s}, & \text{si } \mathcal{C}_{o,\ell,s} = 1, \\ 1 - \beta_{i_s}, & \text{sinon,} \end{cases} \quad (\text{V.32})$$

où i_s représente l'indice numéro s dans l'ensemble d'indices $I_j(t)$ des ancrs détectées. En d'autres termes, $m_o(B_{o,\ell}) = \prod_{s=1}^{|I_j(t)|} (\beta_{i_s} \cdot \mathcal{C}_{o,\ell,s} + (1 - \beta_{i_s}) \cdot (1 - \mathcal{C}_{o,\ell,s}))$. Pour une combinaison $\ell \leq K_o$, le pavé $B_{o,\ell}$ résultera de l'intersection des disques des contraintes accordées un 1 dans $\mathcal{C}_{o,\ell}$. Ainsi, chaque contrainte d'observation i , reformulée en termes d'intervalles comme suit,

$$([x_{j,1}](t) - a_{i,1}(t))^2 + ([x_{j,2}](t) - a_{i,2}(t))^2 = [0, r^2], \quad (\text{V.33})$$

produira deux sous-solveurs $\psi_{j,i,1}$ et $\psi_{j,i,2}$ définis par

$$\begin{cases} [\psi_{j,i,1}] = a_{i,1}(t) + [-\overline{bo}_{j,i,1}(t), +\overline{bo}_{j,i,1}(t)], \\ [\psi_{j,i,2}] = a_{i,2}(t) + [-\overline{bo}_{j,i,2}(t), +\overline{bo}_{j,i,2}(t)], \end{cases} \quad (\text{V.34})$$

avec

$$\begin{cases} \overline{bo}_{j,i,1}(t) = \sqrt{r^2 - \inf((x_{j,2}]_1(t) - a_{i,2}(t))^2), \\ \overline{bo}_{j,i,2}(t) = \sqrt{r^2 - \inf((x_{j,1}]_2(t) - a_{i,1}(t))^2). \end{cases}$$

Pour chacune des combinaisons considérées, tous les sous-solveurs correspondants aux contraintes sélectionnées seront itérés dans l'algorithme de Waltz [Jaulin *et al.*, 2001] afin de contracter un domaine initial. Ce dernier est généralement pris égal à l'ensemble de discernement. Une relaxation de ces contraintes, attribuant à des équations de carrés, permettra de retrouver les résultats de la méthode proposée dans la section précédente.

Tous les pavés d'observation non vides seront par la suite intersectés avec les pavés de mobilité. Le nombre de combinaisons sera alors égal à $K(t-1) * K_o^+$, où $K_o^+ \leq K_o$ est le nombre de pavés d'observation non vides. Ainsi, à l'aide de cette technique, il est possible de réduire le nombre de combinaisons effectuées, surtout lorsque les observations ont des intersections vides. La masse de chacun des pavés sera égale au produit des masses des pavés de mobilité et d'observation concernés dans chaque combinaison. Notons que, pour obtenir une FM normale, il faut mettre à zéro les masses des ensembles vides obtenus. Un nombre $K(t)$ de pavés est ensuite fixé, ceux ayant les masses les plus élevées étant sélectionnés. La phase finale consiste alors à normaliser la distribution de masses finales. La position estimée à l'instant t correspondra au pavé ou l'union de pavés ayant la plus grande masse. Cependant, tous les pavés finaux seront sauvegardés en mémoire afin d'intervenir dans l'auto-localisation du nœud à l'instant suivant. Le pseudo-code de la méthode est donné dans l'algorithme 10, où $\max_{[K(t)]}\{m_t\}$ produit les $K(t)$ masses les plus élevées. Dans les simulations, nous adopterons cette méthode de résolution. Notons que l'utilisation des équations de disques permet de réduire les surfaces des pavés obtenus. Toutefois, l'itération de ces contraintes dans l'algorithme de Waltz peut être plus consommatrice en termes de temps de calcul. Nous permettrons ainsi la relaxation des contraintes d'observations selon les cas.

Exemple 18. *Supposons le même problème présenté dans l'exemple 16. La résolution de ce problème à l'aide de la variante de la méthode consiste à combiner les observations en premier. Le nombre de combinaisons requises est alors égal à $2^3 = 8$. TAB. V.4 regroupe les différentes combinaisons des mesures d'observation, les expressions des pavés résultants et les masses sous-normales m_o^+ et normales m_o qui leur sont attribuées. D_i représente le disque d'observation centré sur l'ancre i et $A_i = [D_i]$ le pavé englobant D_i . La FM d'observation normale possède 5 ensembles focaux. FIG. V.4 illustre ces ensembles avec des numérotations. Les pavés obtenus suite à ces combinaisons sont donnés par A_1, A_2, A_3, Ω et $[D_1 \cap D_2]$. Ces pavés seront par la suite combinés aux pavés de mobilité $[x_j]_1^*$ et $[x_j]_2^*$. Le nombre de combinaisons à faire est égal à $2 * 5 = 10$. Au final, 7 ensembles focaux seront obtenus, celui ayant la plus grande masse est illustré en trait plein rouge dans FIG. V.5. L'avantage principal de cette technique est que le pavé ayant la meilleure masse est plus petit que celui obtenu avec la méthode précédente. Ceci permet de réaliser un gain au niveau de la précision de l'estimation.*

V. AUTO-LOCALISATION EN PRÉSENCE D'IMPERFECTIONS DE MESURES

Algorithme 10 : Pseudo-code de la méthode d'auto-localisation en présence d'imperfections.

Entrées : $K(t)$, $[\mathbf{x}_j]_k(t-1)$, $1 \leq k \leq K(t-1)$, m_{t-1} , $I_j(t)$, $\mathbf{a}_i(t)$, β_i , $i \in I_j(t)$, $v_{max,j}$, Δt , r ;

Sorties : $[\mathbf{x}_j]_k(t)$, $k \in \{1, \dots, K(t)\}$;

pour $k \in \{1, \dots, K(t-1)\}$ **faire**

$[\mathbf{x}_{j,1}]_k^*(t) = [\mathbf{x}_{j,1}]_k(t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]$;

$[\mathbf{x}_{j,2}]_k^*(t) = [\mathbf{x}_{j,2}]_k(t-1) + [-\Delta t.v_{max,j}, +\Delta t.v_{max,j}]$;

$m_t^*([\mathbf{x}_j]_k^*(t)) = m_t([\mathbf{x}_j]_k(t-1))$;

fin

$$K_o = 2^{|I_j(t)|}, \mathcal{C}_o = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{pmatrix};$$

pour $\ell \in \{1, \dots, K_o\}$ **faire**

$J_\ell = \{s \mid \mathcal{C}_{o,\ell,s} = 1, s \in \{1, \dots, |I_j(t)|\}\}$;

$B_{o,\ell} = \Omega$;

$Ar_\ell = w(B_{o,\ell,1}) * w(B_{o,\ell,2})$, $Ar_\ell^{old} = Ar_\ell + 1$;

tant que $Ar_\ell < Ar_\ell^{old}$ **faire**

$Ar_\ell^{old} = Ar_\ell$;

pour $s \in J_\ell$ **faire**

$B_{o,\ell,2} = B_{o,\ell,2} \cap [a_{i_s,2}(t) - r, a_{i_s,2}(t) + r]$;

$\overline{b_{o,j,i_s,1}}(t) = \sqrt{r^2 - \inf((B_{o,\ell,2} - a_{i_s,2}(t))^2)}$;

$[\psi_{j,i_s,1}] = a_{i_s,1}(t) + [-\overline{b_{o,j,i_s,1}}(t), +\overline{b_{o,j,i_s,1}}(t)]$;

$B_{o,\ell,1} = B_{o,\ell,1} \cap [\psi_{j,i_s,1}]$;

$B_{o,\ell,1} = B_{o,\ell,1} \cap [a_{i_s,1}(t) - r, a_{i_s,1}(t) + r]$;

$\overline{b_{o,j,i_s,2}}(t) = \sqrt{r^2 - \inf((B_{o,\ell,1} - a_{i_s,1}(t))^2)}$;

$[\psi_{j,i_s,2}] = a_{i_s,2}(t) + [-\overline{b_{o,j,i_s,2}}(t), +\overline{b_{o,j,i_s,2}}(t)]$;

$B_{o,\ell,2} = B_{o,\ell,2} \cap [\psi_{j,i_s,2}]$;

fin

$Ar_\ell = w(B_{o,\ell,1}) * w(B_{o,\ell,2})$;

fin

$m_o(B_{o,\ell}) = \prod_{s=1}^{|I_j(t)|} (\beta_{i_s} \cdot \mathcal{C}_{o,\ell,s} + (1 - \beta_{i_s}) \cdot (1 - \mathcal{C}_{o,\ell,s}))$;

fin

$s = 0$;

pour $\ell \in \{1, \dots, K_o\}$ **faire**

si $B_{o,\ell} \neq \emptyset$ **alors**

pour $k \in \{1, \dots, K(t-1)\}$ **faire**

$s = s + 1$, $[\mathbf{x}_j]_s^+(t) = B_{o,\ell} \cap [\mathbf{x}_j]_k^*(t)$;

$m_t^+([\mathbf{x}_j]_s^+(t)) = m_o(B_{o,\ell}) * m_t^*([\mathbf{x}_j]_k^*(t))$;

fin

fin

fin

$S = \{s \mid m_t^+([\mathbf{x}_j]_s^+(t)) \in \max_{[K(t)]} \{m_t^+\}\}$;

pour $k \in \{1, \dots, K(t)\}$ **faire**

$[\mathbf{x}_j]_k(t) = [\mathbf{x}_j]_{S(k)}^+(t)$, $m_t([\mathbf{x}_j]_k(t)) = m_t^+([\mathbf{x}_j]_{S(k)}^+(t))$;

fin

ℓ	$C_{o,\ell}$	$B_{o,\ell}$	$m_o^+(B_{o,\ell})$	$m_o(B_{o,\ell})$
1	1, 1, 1	\emptyset	0.144	0
2	1, 1, 0	$[D_1 \cap D_2]$	0.576	0.716
3	1, 0, 1	\emptyset	0.036	0
4	1, 0, 0	A_1	0.144	0.179
5	0, 1, 1	\emptyset	0.016	0
6	0, 1, 0	A_2	0.064	0.080
7	0, 0, 1	A_3	0.004	0.005
8	0, 0, 0	Ω	0.016	0.020

TAB. V.4: Combinaisons des observations du problème de FIG. V.3.

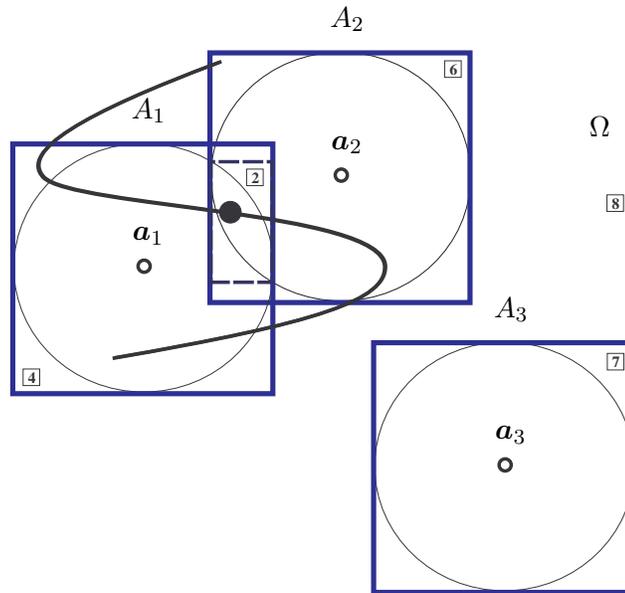


FIG. V.4: Combinaison des observations.

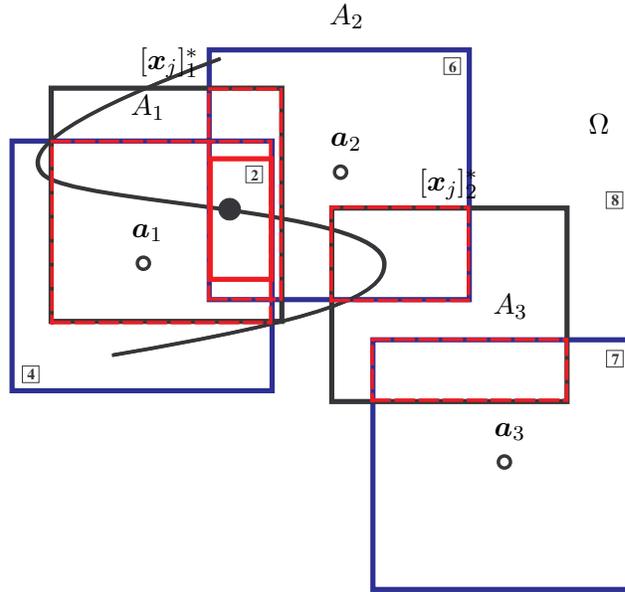


FIG. V.5: Combinaison des pavés d'observations aux pavés de mobilité.

V.5 Simulations

Dans cette section, nous évaluons les performances de la méthode proposée. L'étude est effectuée selon différents critères :

- ◇ l'**aspect garanti** de la méthode, exprimé par l'appartenance ou non de la position réelle au pavé estimé ;
- ◇ l'**erreur d'estimation**, définie par la distance entre la position réelle du nœud et le centre du pavé estimé ;
- ◇ la **complexité de la méthode**, donnée par le temps de calcul ;
- ◇ et la **précision de l'estimation**, exprimée par la surface du pavé estimé.

Il faut noter que toutes les simulations sont réalisées sur Matlab 6.1, installé sur un CPU Intel(R) Core(TM)2 (2.40GHz, 1.00GB RAM).

Afin de réaliser les simulations, nous déployons les capteurs dans une zone carrée de dimensions $100m \times 100m$, dite zone de déploiement. Nous supposons que tous les capteurs ont un mouvement de groupe où ils se déplacent dans la direction nord-est dans les 60 premières secondes puis dans la direction nord-ouest jusqu'à la 100-ème seconde. Un exemple de trajectoires de quinze ancres et trente nœuds selon ce mouvement est illustré dans FIG. V.6. Si nous fixons la période du mouvement à $\Delta t = 1s$, la vitesse maximale des capteurs sera égale à $v_{max} = 2.2719m.s^{-1}$. Nous fixons en outre la portée de communication à $r = 10m$. Sans perte de généralité, nous considérons un nœud unique, vu que le problème d'auto-localisation posé pour un nœud est indépendant des autres. Dans cette section, nous commençons par une illustration des performances de notre méthode dans un environnement imparfait. Nous montrons ensuite la sensibilité de

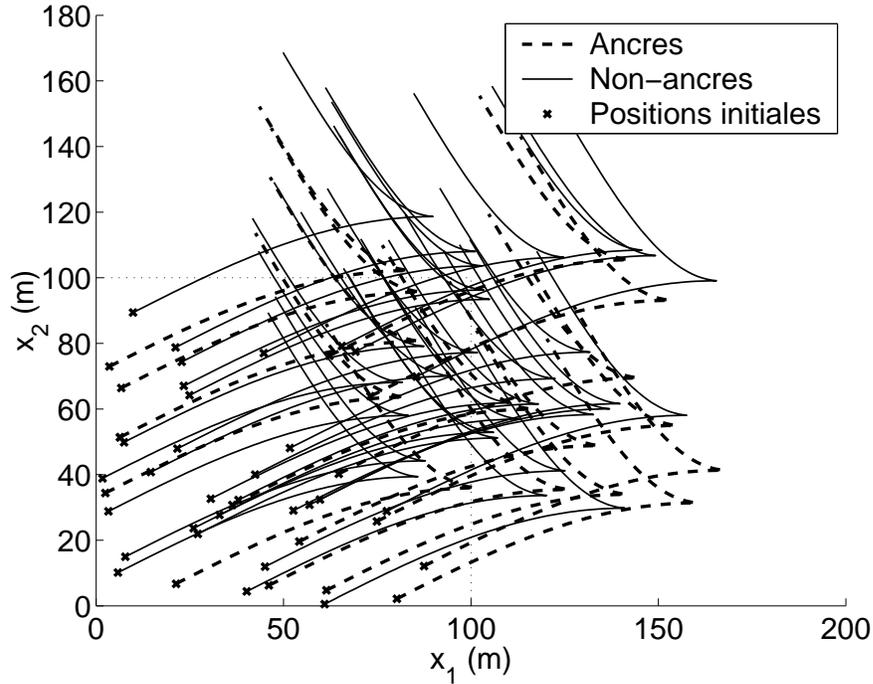


FIG. V.6: Mouvement de groupe de 30 nœuds et 15 ancres.

la méthode aux valeurs de fiabilité accordées aux mesures. Nous comparons par la suite notre approche à la méthode ALCA présentée dans le chapitre III et à une technique de Monte-Carlo utilisant toutes les deux les mesures de connectivités ne tenant pas compte des imperfections.

V.5.1 Evaluation de la méthode

Afin d'évaluer la méthode, nous dispersons aléatoirement 10 ancres dans le réseau de telle sorte que 4 de ces ancres soient initialement dans la portée du nœud considéré. Les autres sont lancées dans l'anneau centré sur le nœud et ayant $10m$ et $20m$ comme rayons interne et externe respectivement. Dans ce cas, les ancres, hors de la portée du nœud, reste très proche de lui conduisant à plus de mesures erronées. En effet, les puissances leur correspondant, étant très proches du seuil ρ_r , peuvent facilement le dépasser en présence d'un faible bruit additif. Les capteurs suivent par la suite le mouvement de groupe illustré dans FIG. V.6. Pour générer des mesures erronées, nous procédons comme suit. Nous générons en premier les valeurs des puissances reçues à partir des distances entre le nœud et les ancres. Cette étape est effectuée à l'aide du modèle d'Okumura-Hata sans bruit avec $\rho_o = 100dBm$, $d_0 = 0.1m$ et $n_P = 3$. Ces paramètres produisent un seuil $\rho_r = 40dBm$. Nous ajoutons ensuite aux puissances calculées un bruit gaussien centré ayant une variance $\sigma^2 = 4dBm^2$. Afin de générer les mesures de connectivités, nous comparons enfin les puissances obtenues au seuil

V. AUTO-LOCALISATION EN PRÉSENCE D'IMPERFECTIONS DE MESURES

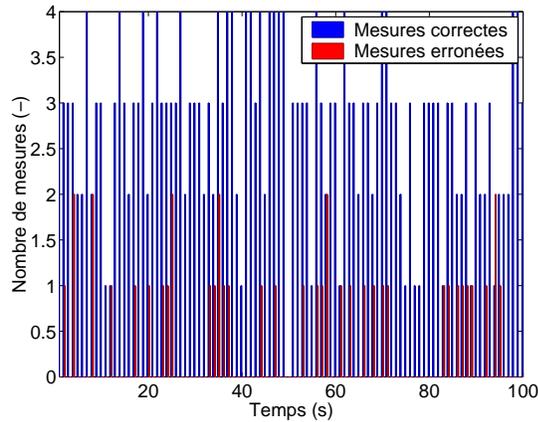


FIG. V.7: Nombres de mesures correctes et erronées obtenues à chaque instant.

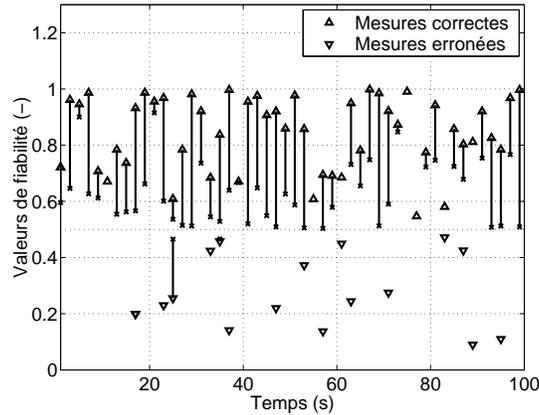


FIG. V.8: Valeurs de fiabilité accordées aux mesures correctes (flèches vers le haut) et erronées (flèches vers le bas), illustrées aux instants impairs.

ρ_r . FIG. V.7 illustre le nombre d'ancres détectées en utilisant les puissances bruitées. La figure montre en bleu le nombre des mesures correctes obtenues et en rouge celui des mesures erronées. Dans cette section, nous associerons aux mesures correctes des valeurs de fiabilité aléatoirement choisies dans l'intervalle $[0.5, 1]$. Les fiabilités données aux mesures erronées sont elles-aussi aléatoires mais choisies dans l'intervalle $[0, 0.5]$. Les valeurs de fiabilité accordées aux mesures sont illustrées dans FIG. V.8. FIG. V.9 montre les pavés obtenus à l'aide de notre méthode, tout en limitant leurs nombres en ne prenant que ceux ayant une masse supérieure à $m_0 = 0.1$. Les pavés en rouge correspondent aux pavés estimés, donc ceux ayant les masses les plus élevées. Il est clair que les positions réelles sont toujours incluses dans les pavés estimés. La garantie est donc gardée à tous les pas de temps même en présence des imperfections. Le nombre de pavés moyen obtenu par pas de temps est égal à 3. A l'instant $t = 6s$, par exemple, trois pavés sont obtenus, ayant des masses égales à 0.6775, 0.1628 et 0.1597. Le temps de calcul moyen nécessaire par pas de temps est égal à $0.02792s$ et l'erreur d'estimation moyenne est de $0.3879m$. La méthode est alors performante, permettant de remédier aux imperfections.

Par ailleurs, nous nous intéressons à la comparaison de deux versions de la méthode : la première étant exacte utilisant les équations de disques, notée ALR (pour Auto-Localisation Robuste), et la deuxième étant approximative, remplaçant les équations de disques par des carrés. Elle sera notée ALRa. Afin de réaliser la comparaison, nous considérons 10 ancres, dont 4 sont dans le voisinage du capteur. Les autres sont dans l'anneau de rayons $10m$ et $20m$ autour du nœud. Nous nous basons sur la même procédure que précédemment pour générer les mesures erronées. Le bruit ajouté aura pareillement une variance de $4dBm^2$. Les mesures de fiabilité sont également similaires. En effet, les mesures correctes ont des fiabilités supérieures à 0.5 et celles erronées ont des fiabilités inférieures à 0.5. FIG. V.10 illustre les pavés estimés à l'aide des méthodes

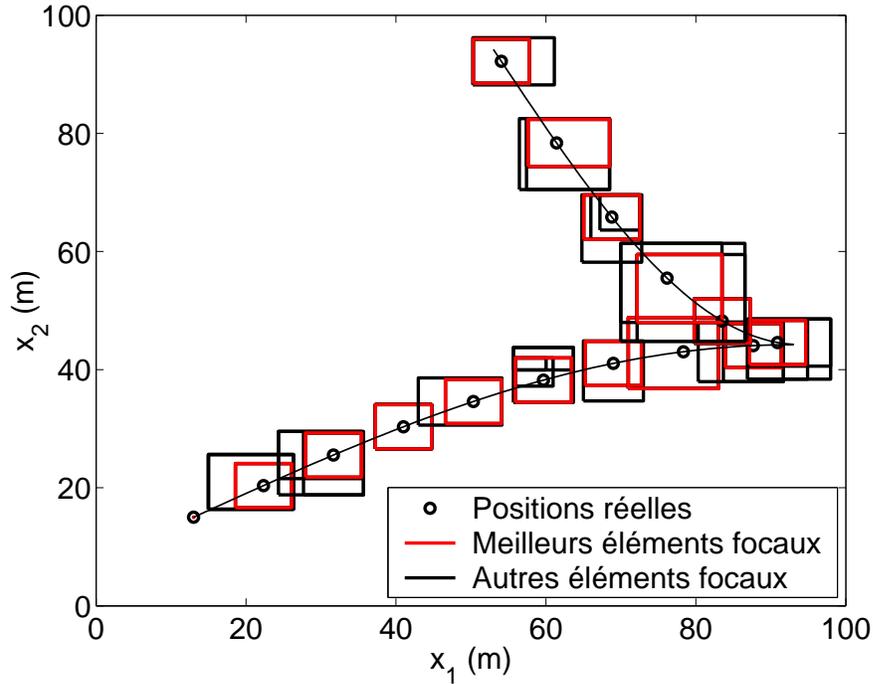


FIG. V.9: Pavés obtenus à l'aide de notre méthode.

Méthode	ALR	ALRa
Temps de calcul	0.01435s	0.00531s
Surface moyenne	29.3218m ²	55.8599m ²
Erreur moyenne	0.9133m	1.0678m

TAB. V.5: Comparaison des deux versions de la méthode ALR.

ALR et ALRa. Il est clair que les pavés obtenus avec la méthode exacte sont plus précis. Toutefois, les deux méthodes gardent l'aspect garanti, ayant les pavés estimés incluant les positions réelles. TAB. V.5 regroupe les valeurs du temps de calcul moyen, de la surface moyenne des pavés estimés et de l'erreur d'estimation moyenne par pas de temps obtenues à l'aide des méthodes ALR et ALRa. Ainsi, la méthode ALR est plus précise, réalisant un gain en précision de 47.51% au prix de l'augmentation du temps de calcul. Toutefois, ce dernier reste relativement faible. Dans la suite des simulations, nous adopterons la version exacte de la méthode.

V.5.2 Sensibilité aux valeurs de fiabilité

Dans la section précédente, nous avons attribué aux mesures correctes des valeurs de fiabilité supérieures à 0.5 alors que celles des mesures erronées étaient inférieures à 0.5. Dans ces conditions, l'aspect garanti de la méthode n'est jamais perdu. Dans cette section, nous supposerons que le modèle de génération des fiabilités est non pertinent. En

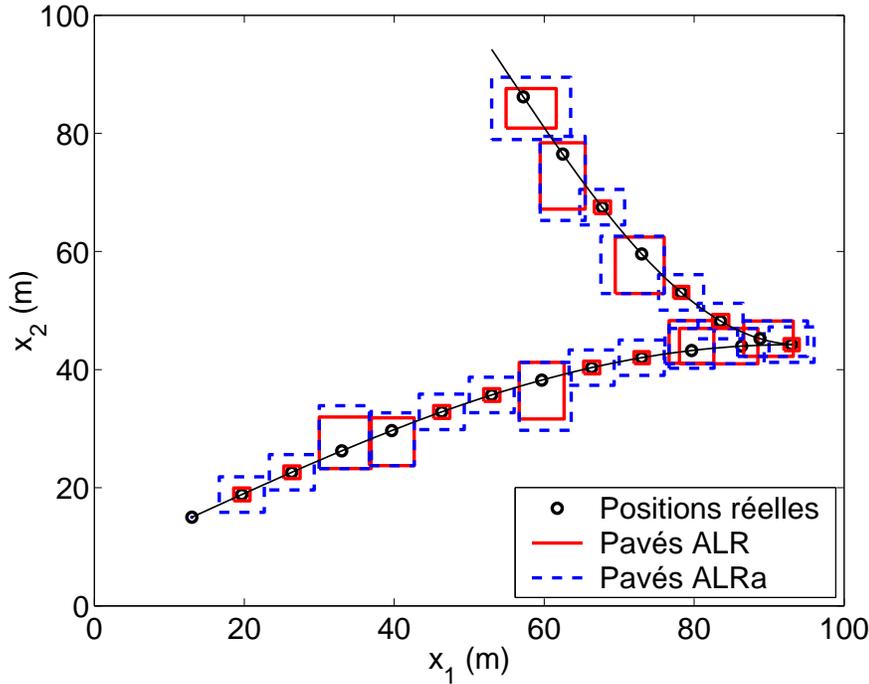


FIG. V.10: Pavés estimés obtenus à l'aide des méthodes ALR et ALRa.

d'autres termes, les fiabilités générées ne suivent pas l'hypothèse précédente. Celles des mesures erronées peuvent alors dépasser le seuil 0.5. Dans la suite, nous envisagerons deux cas : le premier cas où les fiabilités des mesures erronées dépassent le seuil 0.5 mais restent inférieures à celles des mesures correctes, et le second cas où elles peuvent même aller au-delà de celles des mesures correctes. Notons que nous utilisons 10 ancres, toutes étant lancées dans la zone de déploiement. Les mesures erronées sont générées similairement à la section précédente en ajoutant un bruit centré aux puissances avec une variance égale à $4dBm^2$.

Nous commençons tout d'abord par accorder les mesures erronées des valeurs de fiabilités comprises entre 0.5 et 0.75. Les mesures correctes ont en outre des fiabilités comprises entre 0.75 et 1. Toutes les fiabilités sont alors supérieures au seuil 0.5. FIG. V.11 illustre les pavés obtenus dans ce cas à l'aide de notre méthode. Les pavés estimés ne contiennent pas les positions réelles à certains pas de temps. FIG. V.12 montre en haut le nombre des mesures erronées générées à chaque pas de temps. De plus, elle montre en bas l'indicateur de garantie, qui vaut 1 si le pavé estimé contient la position réelle et 0 sinon. La figure montre que des mesures erronées sont générées à 64 pas de temps, la méthode ayant perdu la garantie à 14 parmi eux. Généralement, la méthode perd la garantie quand les disques des observations correctes et erronées ont une intersection non vide. Le schéma gauche de FIG. V.13 illustre un tel cas correspondant à l'instant $t = 13s$. Les disques d'observation ont une intersection non vide ayant la masse la plus élevée. Outre les observations, la figure montre les ensembles focaux de la

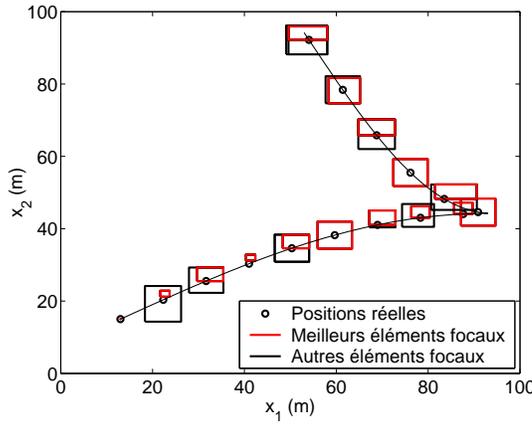


FIG. V.11: Pavés obtenus à l'aide de notre méthode avec des fiabilités supérieures à 0.5.

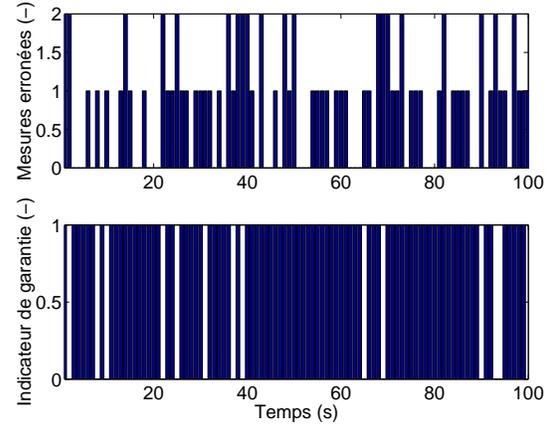


FIG. V.12: Nombre des mesures erronées et indicateur de garantie de notre méthode avec des fiabilités supérieures à 0.5.

fonction de masse de mobilité et le pavé estimé à cet instant. En revanche, le schéma droit de FIG. V.13 illustre les disques d'observation correspondant à l'instant $t = 87s$. Il est clair que le disque erroné possède une intersection vide avec les pavés de mobilité et les autres disques. Ainsi, la méthode peut perdre la garantie dans le cas où les fiabilités des mesures erronées sont supérieures à 0.5. Toutefois, vu que plusieurs pavés sont générés, contenant partiellement la position réelle, notre méthode permet de limiter la propagation de l'erreur aux pas de temps suivants.

Nous considérons en second lieu un autre exemple où les mesures correctes et erronées sont attribuées des valeurs de fiabilité aléatoirement choisies entre 0 et 1. Dans ce cas, lorsque les fiabilités des mesures erronées dépassent celles des mesures correctes, la méthode perd le plus probablement la garantie, sauf si les pavés de mobilité ont une intersection vide avec les mesures erronées. Ainsi, les performances de la méthode dépendent de la manière dont les fiabilités sont accordées aux mesures. Quand les fiabilités des mesures erronées sont inférieures à 0.5, l'aspect garanti est toujours gardé. Dans les autres cas, la garantie dépendra de la disposition des ancres concernées. Toutefois, la méthode possède la particularité de créer plusieurs possibilités de positions, permettant ainsi de limiter l'effet d'une fausse estimation sur les instants suivants.

V.5.3 Comparaison à la méthode ALCA

Dans cette section, nous comparons la méthode ALR à la méthode ALCA proposée dans le chapitre III. Cette méthode se base sur les mesures de connectivités pour estimer la position du nœud dans des conditions parfaites. Dans le cas d'imperfections, lorsque l'intersection des contraintes d'observation est vide, ALCA se contente de garder uniquement le modèle de mobilité tout en omettant les observations. Afin de comparer les deux méthodes, nous utilisons 30 ancres que nous lancerons aléatoirement dans toute la zone de déploiement. Pareillement aux sections précédentes, les mesures erronées sont

V. AUTO-LOCALISATION EN PRÉSENCE D'IMPERFECTIONS DE MESURES

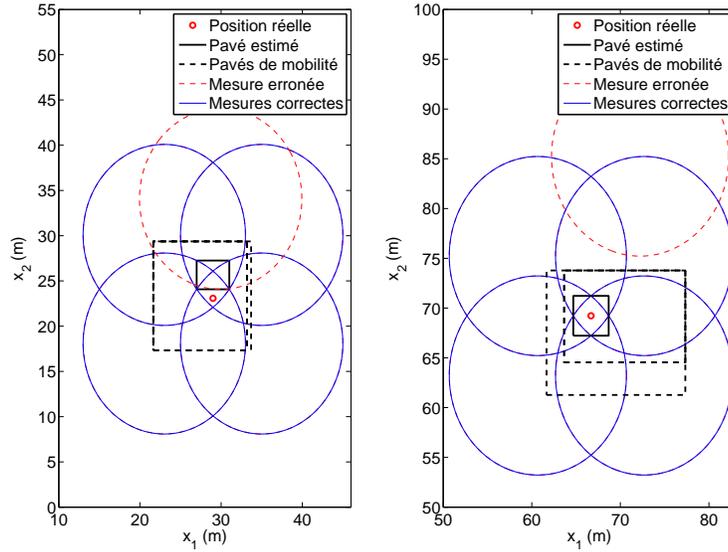


FIG. V.13: Illustrations des contraintes aux instants $t = 13s$ à gauche et $t = 87s$ à droite.

Méthode	ALR	ALCA
Temps de calcul	0.02715s	0.00546s
Surface moyenne	68.8819m ²	140.3218m ²
Erreur moyenne	0.6161m	2.7246m

TAB. V.6: Comparaison des deux méthodes ALR et ALCA.

générées en ajoutant un bruit gaussien centré de variance égale à $4dBm^2$ aux vraies puissances. Notons que les fiabilités accordées aux mesures correctes sont supérieures à 0.5 et celles des mesures erronées sont inférieures à 0.5. FIG. V.14 illustre les pavés estimés à l'aide des deux méthodes. La figure montre que les pavés ALCA sont principalement soit trop petits ne contenant pas les positions réelles, soit trop grands. ALCA perd la garantie à 26 pas de temps, contrairement à la méthode ALR qui fournit toujours un résultat garanti. TAB. V.6 regroupe les valeurs du temps de calcul moyen, de la surface des pavés moyenne et de l'erreur moyenne obtenues à l'aide des deux méthodes. Les résultats montrent que la méthode ALR est beaucoup plus performante que la méthode ALCA en présence d'imperfections, au prix de l'augmentation du temps de calcul.

V.5.4 Comparaison à la méthode MCB

Dans cette section, nous comparons la méthode ALR à une méthode basée sur l'approche séquentielle de Monte-Carlo [Doucet *et al.*, 2000]. Cette méthode, présentée dans [Baggio et Langendoen, 2006], utilise les mesures de connectivités. Notée MCB, elle génère des particules satisfaisant toutes les contraintes. Nous fixons dans cette section

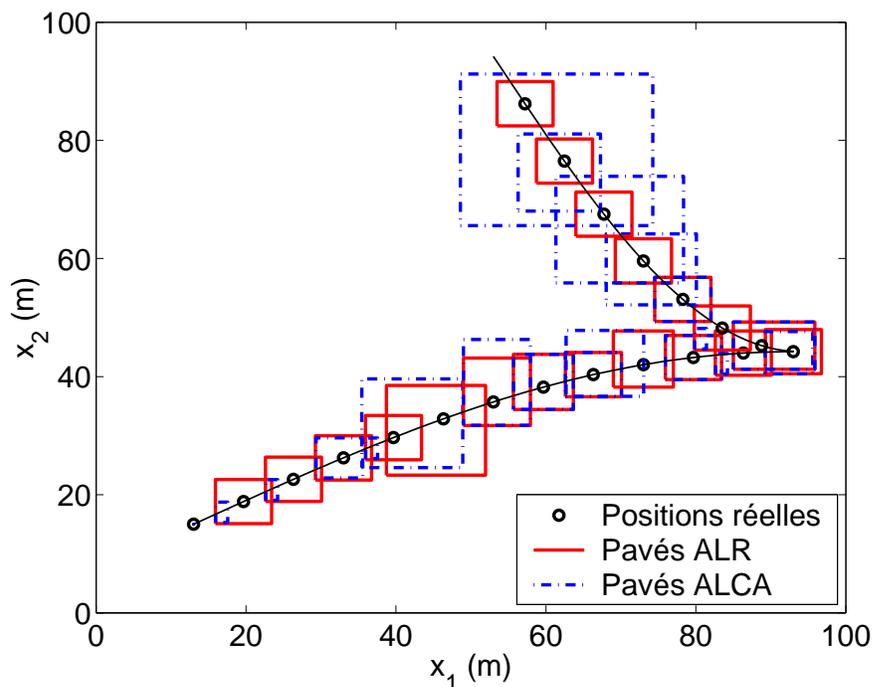


FIG. V.14: Pavés obtenus à l'aide des méthodes ALR et ALCA.

le nombre de particules requises à 50 et le nombre maximal d'itérations pour les obtenir à 100. Si les mesures d'observation ont une intersection vide, seul le modèle de mobilité est utilisé. Afin de comparer les deux méthodes, nous utilisons 20 ancres aléatoirement choisies dans toute la zone de déploiement. Les ancres et le nœud bougent par la suite selon le modèle de groupe précédent. FIG. V.15 montre les particules et les pavés estimés respectivement à l'aide des méthodes MCB et ALR. TAB. V.7 regroupe les valeurs du temps de calcul moyen et de l'erreur d'estimation moyenne des deux méthodes. ALR nécessite ainsi un temps de calcul plus réduit tout en fournissant une estimation bien meilleure que MCB.

Nous comparons ces deux méthodes dans d'autres conditions, où les fiabilités des mesures correctes sont prises entre 0.5 et 1 et celles des mesures erronées entre 0 et 1. Nous utilisons en outre 30 ancres dispersées initialement dans toute la zone de déploiement. FIG. V.16 illustre les pavés et les particules obtenus à l'aide des deux méthodes. Même dans ces conditions où ALR perd sa garantie à 17 pas de temps, elle reste plus performante que la méthode MCB vu que les erreurs moyennes sont de $1.0515m$ pour ALR et $2.1793m$ pour MCB.

V.6 Conclusion

Dans ce chapitre, nous avons présenté une méthode robuste d'auto-localisation. Cette méthode a été développée pour les environnements imparfaits où des mesures erronées

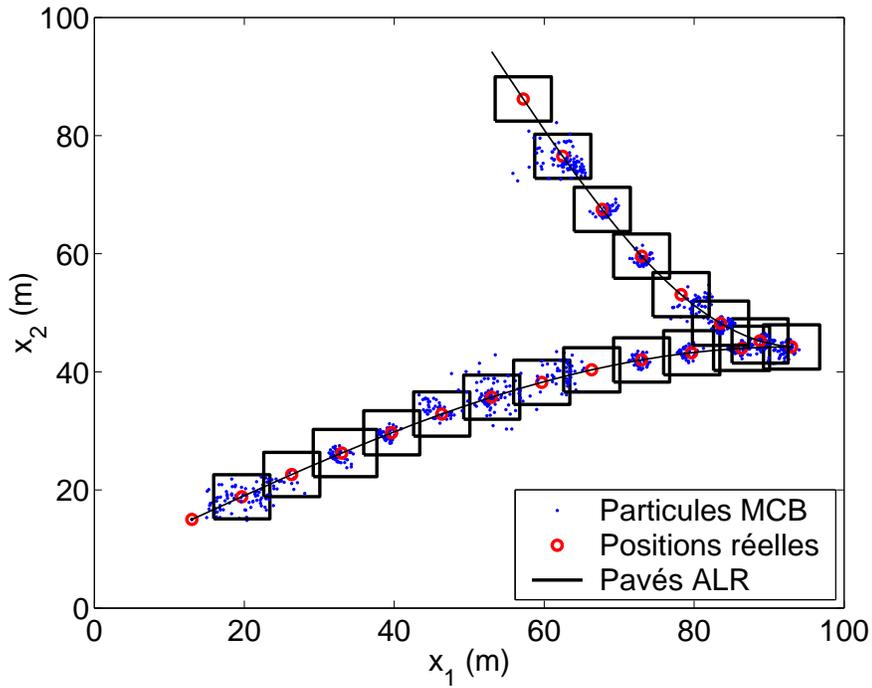


FIG. V.15: Pavés et particules obtenus à l'aide des méthodes ALR et MCB.

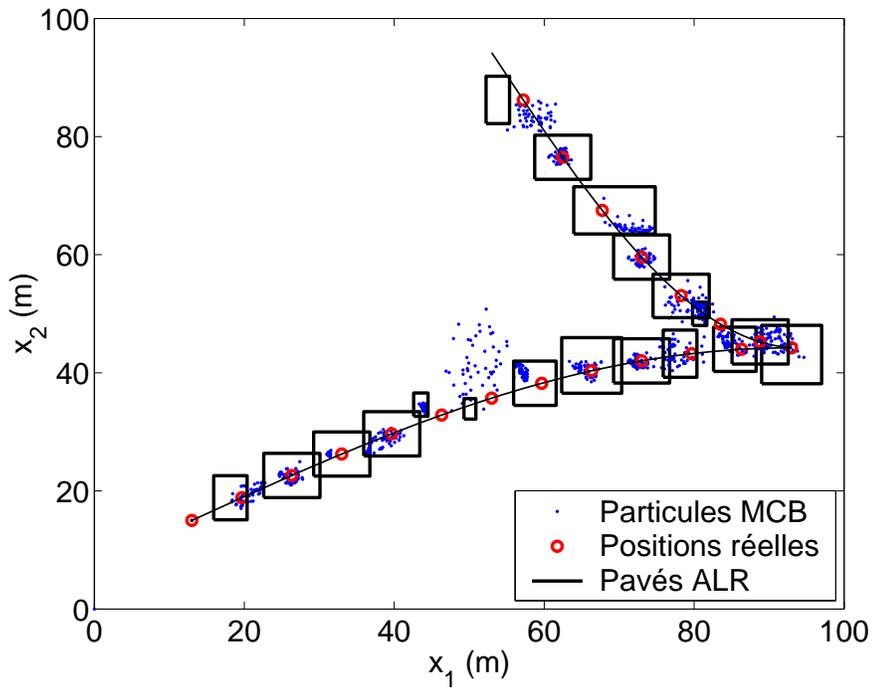


FIG. V.16: Pavés et particules obtenus à l'aide des méthodes ALR et MCB avec des fiabilités des mesures erronées allant de 0 à 1.

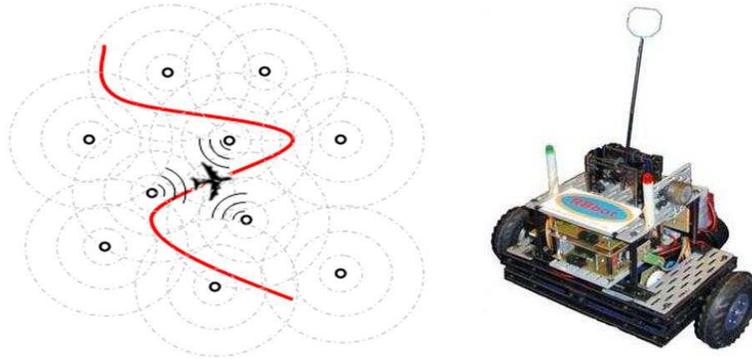
Méthode	ALR	MCB
Temps de calcul	0.02137s	0.1691s
Erreur moyenne	0.2777m	1.9786m

TAB. V.7: Comparaison des deux méthodes ALR et MCB.

peuvent survenir. L'idée de base de la méthode consiste à associer des valeurs de fiabilités aux mesures de connectivités. La résolution du problème est effectuée dans le cadre de la théorie des fonctions de croyance. La méthode fournit alors un ensemble de pavés bidimensionnels ayant chacun une masse spécifique. Les positions estimées correspondent aux centres des pavés ayant les masses les plus élevées. Les performances de la méthode sont évaluées à l'aide de simulations effectuées sur des capteurs mobiles en groupe. Les résultats de simulations confirment l'efficacité de la méthode comparée aux techniques développées pour des environnements parfaits. Toutefois, ces performances dépendent de la manière dont les fiabilités sont associées aux mesures. Il est alors intéressant dans la suite de développer des méthodes pertinentes pour le calcul des fiabilités. Le problème d'auto-localisation peut en outre être traité sous d'autres hypothèses, telles que la connaissance du nombre maximal des mesures erronées.

V. AUTO-LOCALISATION EN PRÉSENCE D'IMPERFECTIONS DE MESURES

Suivi de cibles dans les réseaux mobiles



Sommaire

- VI.1 Introduction
 - VI.2 Estimation de la position de la cible
 - VI.3 Prédiction de la position de la cible
 - VI.4 Définition des positions des capteurs
 - VI.5 Optimisation du repositionnement des capteurs
 - VI.6 Simulations
 - VI.7 Conclusion
-

Dans les réseaux de capteurs mobiles, où la mobilité des capteurs est contrôlable, la gestion du mouvement des capteurs s'avère primordiale. Le contrôle de la mobilité permettra, en effet, l'amélioration des performances du réseau dans la plupart des applications. Dans ce chapitre, nous traitons le problème de suivi de cibles dans les réseaux de capteurs à mobilité contrôlée. Nous proposons ainsi une stratégie de repositionnement des capteurs afin d'optimiser l'estimation de la position de la cible. La méthode est composée de plusieurs étapes : i) l'estimation de la position actuelle de la cible, ii) la prédiction de la position future de la cible, iii) la génération d'un ensemble de positions que doivent prendre les capteurs, et iv) l'attribution d'une position parmi l'ensemble à chacun des capteurs tout en minimisant le déplacement. Cette dernière étape est effectuée à l'aide d'une méthode métaheuristique : l'optimisation par colonies de fourmis. En conséquence, nous commencerons dans ce chapitre par décrire les étapes d'estimation et de prédiction. Nous introduirons ensuite la stratégie de repositionnement des capteurs en utilisant les colonies de fourmis.

VI.1 Introduction

Les Réseaux de Capteurs Mobiles (RCMs) sont composés d'un très grand nombre de dispositifs sans fil pouvant collecter, traiter et transmettre des données [Akyildiz *et al.*, 2002b]. Grâce à l'absence d'infrastructure fixe dans les RCMs, les capteurs ont la possibilité de changer de positions, que ce soit d'une manière contrôlée ou non contrôlée, dite aussi passive. La mobilité passive rend l'utilisation des RCMs plus difficile vu que les capteurs sont inconscients de leurs mouvements et par conséquent doivent être relocalisés régulièrement. En revanche, dans le cas de la mobilité contrôlée, les capteurs se déplacent en réponse à une commande interne ou externe. Une gestion de la mobilité est alors fondamentale pour l'optimisation des performances du réseau face à une certaine application. Toutefois, l'une des contraintes principales des réseaux de capteurs est la limitation de l'énergie consommée vu que leurs sources d'énergie sont généralement non renouvelables. Les stratégies de gestion de mobilité doivent alors optimiser les performances du réseau tout en minimisant la consommation de l'énergie.

Les RCMs ont des applications variées dans différents domaines, tels que le domaine militaire ou la surveillance de l'environnement [Juang *et al.*, 2002; Mainwaring *et al.*, 2002; Gharavi et Kumar, 2003; Hande *et al.*, 2006; Shorey, 2006]. L'une des applications les plus intéressantes des RCMs dans le domaine de surveillance et de sécurité est le suivi de cibles. Elle consiste à estimer instantanément la position d'une cible mobile. Ce problème a été traité principalement dans des réseaux ayant des capteurs fixes [Chen *et al.*, 2004; Djurić *et al.*, 2008; Wang *et al.*, 2009; Teng *et al.*, 2009, 2010]. Dans [Teng *et al.*, 2009], par exemple, une méthode basée sur le filtrage variationnel a été présentée. En revanche, dans [Djurić *et al.*, 2008], Djurić *et al.* ont proposé une méthode basée sur le filtrage particulaire pour le suivi de cibles avec des capteurs binaires. Toutefois, quand les capteurs peuvent se déplacer, il est possible de profiter de cette mobilité pour améliorer l'estimation des positions. Zou et Chakrabarty ont étudié un tel problème dans [Zou et Chakrabarty, 2007]. Ils proposent alors une stratégie de gestion de mobilité basée sur l'estimation bayésienne. Selon cette stratégie, chaque capteur est capable de bouger vers une nouvelle position qu'il choisit parmi une liste de candidats. Ces derniers, au nombre de huit, sont situés à un pas de la position courante. Le plan est alors présenté sous forme d'une grille où chaque case possède huit voisins. Le choix parmi les candidats est fait de façon à optimiser la qualité de l'estimation. Néanmoins, cette méthode impose de nombreuses hypothèses, aussi bien au niveau des positions candidates qu'au niveau du mouvement de la cible et des capteurs (vitesse constante,...).

Dans ce chapitre, nous proposons une nouvelle stratégie de gestion de mobilité pour le suivi de cibles. Cette méthode est composée de quatre étapes consécutives, que nous itérons à chaque pas de temps comme suit,

1. estimation de la position courante de la cible,
2. prédiction de la position suivante de la cible, tout en utilisant les estimés actuel et précédents,
3. génération d'un ensemble de positions que les capteurs doivent prendre pour améliorer l'estimation,

4. attribution d'une nouvelle position à chacun des capteurs, en utilisant l'optimisation par colonies de fourmis.

La phase d'estimation est effectuée à l'aide de la théorie des intervalles [Moore, 1979; Jaulin *et al.*, 2001]. La position de la cible est alors présentée sous forme d'un pavé bidimensionnel contenant la position réelle. La prédiction de la position suivante, basée également sur l'analyse par intervalles, produit un pavé, qu'il faut couvrir. Le repositionnement des capteurs consiste par la suite à déplacer les capteurs afin d'optimiser la phase d'estimation au pas de temps suivant tout en minimisant l'énergie consommée durant le déplacement des capteurs. L'optimisation est réalisée à l'aide de l'algorithme d'optimisation par colonies de fourmis, ou ACO (pour Ant Colony Optimization) [Dorigo et Gambardella, 1997; Dréo *et al.*, 2005]. Outre la minimisation de l'énergie consommée, la méthode possède une contrainte primordiale, correspondant au maintien de la couverture du réseau. En effet, la totalité de la zone de surveillance doit être couverte par des capteurs afin d'être robuste à tout intrus. Pour remédier à ce problème, nous utilisons dans le réseau deux types de capteurs : les capteurs fixes et les capteurs mobiles. Alors que les capteurs mobiles se déplacent pour améliorer l'estimation, les capteurs fixes sont uniformément répartis dans la zone afin d'assurer une couverture continue du réseau indépendamment du mouvement des capteurs mobiles.

Dans la suite, nous commençons par décrire les algorithmes d'estimation et de prédiction. Nous présentons ensuite la stratégie de génération des nouvelles positions des capteurs. Nous exposons par ailleurs l'algorithme ACO que nous appliquons pour attribuer à chaque capteur une des positions générées. Nous illustrons enfin les performances de la méthode par des simulations effectuées sur Matlab.

VI.2 Estimation de la position de la cible

Le suivi de cibles consiste à estimer les positions des cibles à chaque pas de temps. Dans ce chapitre, nous traitons le problème de suivi d'une cible unique. La résolution du problème est effectuée à l'aide de la théorie des intervalles. La cible aura alors une position instantanée sous forme de pavé bidimensionnel. Notons que la méthode proposée suppose que toutes les informations collectées sont acheminées vers un point unique, par exemple une station de base, où est faite toute l'étude. Dans cette section, nous posons tout d'abord le problème d'estimation, que nous résoudrons dans la suite.

VI.2.1 Problématique

Afin d'estimer la position de la cible, chaque capteur collecte, à chaque instant, des informations de connectivités concernant la cible. Un capteur i , $i \in \{1, \dots, N_s\}$, génère ainsi une information sur un bit comme suit,

$$y_i(t) = \begin{cases} 1, & \text{si le capteur } i \text{ détecte la cible,} \\ 0, & \text{sinon.} \end{cases} \quad (\text{VI.1})$$

Soit $I(t)$ l'ensemble des indices des capteurs ayant détecté la cible à l'instant t . Selon le modèle de connectivités introduit dans le chapitre III, la distance séparant la cible

VI. SUIVI DE CIBLES DANS LES RÉSEAUX MOBILES

de tout capteur i , $i \in I(t)$, est inférieure à la portée de communication r . La cible sera alors située dans la zone d'intersection de tous les disques centrés sur les capteurs la détectant et ayant r comme rayons. Soient $\mathbf{x}(t) = (x_1(t), x_2(t))$ et $\mathbf{s}_i(t) = (s_{i,1}(t), s_{i,2}(t))$ les coordonnées respectives de la cible et des capteurs i , $i \in I(t)$, à l'instant t . Les équations d'observation s'écrivent alors de la manière suivante,

$$(x_1(t) - s_{i,1}(t))^2 + (x_2(t) - s_{i,2}(t))^2 \leq r^2, \quad i \in I(t). \quad (\text{VI.2})$$

Dans le cas où le champ de communication n'est pas homogène, la valeur maximale de la portée sera utilisée dans les équations (VI.2).

VI.2.2 Algorithme

La méthode proposée utilise la théorie des intervalles pour résoudre le problème d'estimation. Les différentes variables seront alors remplacées par les intervalles auxquels elles appartiennent. En particulier, la position de la cible sera donnée par le pavé bi-dimensionnel $[\mathbf{x}](t) = [x_1](t) \times [x_2](t)$. Les capteurs sont supposés avoir des positions ponctuelles connues. Dans le cas contraire, leurs positions pourront être remplacées par les pavés $[\mathbf{s}_i](t)$ incluant toute l'incertitude présente sur leurs valeurs. Dans le contexte ensembliste, les équations d'observation sont reformulées comme suit,

$$[[x_1](t) - s_{i,1}(t)]^2 + [[x_2](t) - s_{i,2}(t)]^2 \subseteq [0, r^2], \quad i \in I(t). \quad (\text{VI.3})$$

Si l'on suppose que le domaine initial de $[\mathbf{x}](t)$ correspond à toute la zone de surveillance, le problème peut être défini comme un problème de satisfaction de contraintes. La résolution d'un tel problème à l'aide de la théorie des intervalles est effectuée avec des algorithmes dits contracteurs. Dans ce chapitre, nous utilisons l'algorithme de Waltz [Waltz, 1975; Jaulin *et al.*, 2001] pour son efficacité, sa rapidité et sa simplicité de mise en œuvre. Cet algorithme itère toutes les contraintes, sans aucun ordre a priori, afin de contracter le domaine initial. Chaque contrainte est alors décomposée en deux sous-solveurs exprimant chacune des variables en fonction de l'autre comme suit,

$$\begin{cases} [\psi_{i,1}](t) = [s_{i,1}(t) - \bar{b}_{i,1}(t), s_{i,1}(t) + \bar{b}_{i,1}(t)] \\ [\psi_{i,2}](t) = [s_{i,2}(t) - \bar{b}_{i,2}(t), s_{i,2}(t) + \bar{b}_{i,2}(t)] \end{cases}, \quad i \in I(t), \quad (\text{VI.4})$$

avec

$$\begin{cases} \bar{b}_{i,1}(t) = \sqrt{r^2 - \inf(([x_2](t) - s_{i,2}(t))^2)}, \\ \bar{b}_{i,2}(t) = \sqrt{r^2 - \inf(([x_1](t) - s_{i,1}(t))^2)}. \end{cases}$$

La contraction du pavé $[\mathbf{x}](t)$ se fera alors à l'aide de ces sous-solveurs de la manière suivante,

$$\begin{cases} [x_1](t) = [x_1](t) \cap [\psi_{i,1}](t) \\ [x_2](t) = [x_2](t) \cap [\psi_{i,2}](t) \end{cases}, \quad i \in I(t). \quad (\text{VI.5})$$

Le pavé final englobera la zone d'intersection des disques, contenant ainsi la position réelle. Soit $[\mathbf{x}]_0$ le domaine initial de la position $[\mathbf{x}](t)$, alors le pseudo-code de l'étape d'estimation est donné dans l'algorithme 11. FIG. VI.1 illustre par ailleurs cette étape.

Algorithme 11 : Pseudo-code de l'étape d'estimation de la position de la cible.

Entrées : $s_i(t), i \in I(t), r, [\mathbf{x}]_0$;

Sorties : $[\mathbf{x}](t)$;

$[\mathbf{x}](t) = [\mathbf{x}]_0$;

$A = w([x_1](t)) * w([x_2](t))$;

$A^{old} = A + 1$;

tant que $A < A^{old}$ **faire**

$A^{old} = A$;

pour $i \in I(t)$ **faire**

$[x_2](t) = [x_2](t) \cap [s_{i,2}(t) - r, s_{i,2}(t) + r]$;

$\bar{b}_{i,1}(t) = \sqrt{r^2 - \inf(([x_2](t) - s_{i,2}(t))^2)}$;

$[\psi_{i,1}](t) = [s_{i,1}(t) - \bar{b}_{i,1}(t), s_{i,1}(t) + \bar{b}_{i,1}(t)]$;

$[x_1](t) = [x_1](t) \cap [\psi_{i,1}](t)$;

$[x_1](t) = [x_1](t) \cap [s_{i,1}(t) - r, s_{i,1}(t) + r]$;

$\bar{b}_{i,2}(t) = \sqrt{r^2 - \inf(([x_1](t) - s_{i,1}(t))^2)}$;

$[\psi_{i,2}](t) = [s_{i,2}(t) - \bar{b}_{i,2}(t), s_{i,2}(t) + \bar{b}_{i,2}(t)]$;

$[x_2](t) = [x_2](t) \cap [\psi_{i,2}](t)$;

fin

$A = w([x_1](t)) * w([x_2](t))$;

fin

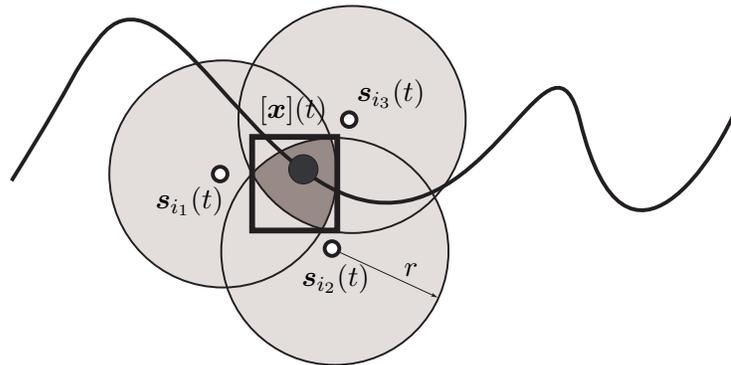


FIG. VI.1: Estimation de la position de la cible.

VI.3 Prédiction de la position de la cible

Une fois la position de la cible est estimée à l'instant courant, cet estimé sera utilisé avec les estimés précédents pour prédire la prochaine position. Soient $\mathbf{x}(1), \dots, \mathbf{x}(t)$ les coordonnées de la cible estimées respectivement aux instants $1, \dots, t$. Un modèle de prédiction d'ordre k est alors donné par

$$\widehat{\mathbf{x}}(t+1) = \mathbf{f}(\mathbf{x}(t), \dots, \mathbf{x}(t-k)), \quad (\text{VI.6})$$

où \mathbf{f} est la fonction de prédiction et $\widehat{\mathbf{x}}(t+1)$ est la position prédite de la cible à l'instant t concernant l'instant $t+1$. Toute information à propos du mouvement de la cible peut être utilisée pour définir ce modèle. Dans cette méthode, nous utilisons le modèle de prédiction du second ordre suivant,

$$\widehat{\mathbf{x}}(t+1) = \mathbf{x}(t) + \Delta t \cdot \mathbf{v}(t) + \frac{\Delta t^2}{2} \cdot \gamma(t), \quad (\text{VI.7})$$

où Δt est la durée entre deux pas de temps consécutifs, $\mathbf{v}(t)$ est la vitesse instantanée de la cible à l'instant t et $\gamma(t)$ est son accélération. $\mathbf{v}(t)$ et $\gamma(t)$ sont alors données par

$$\begin{cases} \mathbf{v}(t) = \frac{\mathbf{x}(t) - \mathbf{x}(t-1)}{\Delta t}, \\ \gamma(t) = \frac{\mathbf{v}(t) - \mathbf{v}(t-1)}{\Delta t}. \end{cases} \quad (\text{VI.8})$$

Dans le contexte ensembliste, le modèle de prédiction est reformulé de la manière suivante,

$$[\widehat{\mathbf{x}}](t+1) = [\mathbf{x}](t) + \Delta t \cdot [\mathbf{v}](t) + \frac{\Delta t^2}{2} \cdot [\gamma](t), \quad (\text{VI.9})$$

avec $[\widehat{\mathbf{x}}](t+1)$ est le pavé prédit représentant la position de la cible à l'instant $t+1$ et

$$\begin{cases} [\mathbf{v}](t) = \frac{[\mathbf{x}](t) - [\mathbf{x}](t-1)}{\Delta t}, \\ [\gamma](t) = \frac{[\mathbf{v}](t) - [\mathbf{v}](t-1)}{\Delta t}. \end{cases}$$

Ainsi, en utilisant les intervalles, l'étape de prédiction produit à l'instant courant un pavé couvrant la position suivante de la cible.

VI.4 Définition des positions des capteurs

L'une des contraintes principales de la mobilité des capteurs est d'assurer une couverture totale permanente de la zone de surveillance. Si nous supposons que le champ de communication des capteurs est circulaire de portée r , alors chaque capteur couvre un disque de rayon r dans la zone de surveillance. Le fait de déplacer les capteurs pour optimiser les performances du réseau peut produire des régions non couvertes. Dans ce cas, le réseau ne sera plus efficace pour la détection des intrus. D'où l'idée d'utiliser deux types de capteurs dans le réseau :

- **Capteurs fixes** : Les capteurs fixes ont pour rôle d'assurer une couverture continue du réseau indépendamment du mouvement des autres capteurs. Ils ne sont donc pas concernés par le mouvement.
- **Capteurs mobiles** : Les capteurs mobiles sont déplacés afin d'assurer une optimisation des performances du réseau.

Dans la suite, nous commençons par définir les positions des capteurs fixes. Nous exposerons ensuite une méthode de génération des nouvelles positions des capteurs mobiles.

VI.4.1 Capteurs fixes

Afin d'obtenir une couverture totale, la zone de surveillance doit être complètement remplie par les disques de communication des capteurs fixes. Vu que l'utilisation des disques n'est pas pratique pour le remplissage, nous les remplacerons par les carrés qui y sont inscrits. Nous supposons alors que chaque capteur fixe couvre un carré centré sur lui et ayant $\sqrt{2}.r$ comme longueur de côté. Afin d'utiliser le minimum de capteurs, il suffit de placer les carrés l'un à côté de l'autre, sur des rangées adjacentes l'une à l'autre, jusqu'à remplissage de la zone de surveillance. Soit $[\mathbf{x}]_0 = [x_1]_0 \times [x_2]_0$ la zone de surveillance et soient $K_{f,1}$ et $K_{f,2}$ les nombres des carrés nécessaires pour garantir la couverture totale respectivement selon la première et la deuxième coordonnée. Alors, $K_{f,1}$ et $K_{f,2}$ sont définis de la manière suivante,

$$\begin{cases} K_{f,1} = \overline{\mathbb{E}}\left(\frac{w([x_1]_0)}{\sqrt{2}.r}\right), \\ K_{f,2} = \overline{\mathbb{E}}\left(\frac{w([x_2]_0)}{\sqrt{2}.r}\right), \end{cases} \quad (\text{VI.10})$$

où $w([x])$ est la largeur de l'intervalle $[x]$ et $\overline{\mathbb{E}}(x)$ produit le plus petit entier supérieur ou égal à x . Ainsi, le nombre de capteurs fixes exigé pour une couverture totale est donné par

$$K_f = K_{f,1} * K_{f,2} = \overline{\mathbb{E}}\left(\frac{w([x_1]_0)}{\sqrt{2}.r}\right) \cdot \overline{\mathbb{E}}\left(\frac{w([x_2]_0)}{\sqrt{2}.r}\right). \quad (\text{VI.11})$$

Afin d'obtenir des coordonnées symétriques par rapport au centre de la zone de surveillance, nous les définissons de la manière suivante,

$$\begin{cases} S_{f,1,p} = \underline{b}_1 + \frac{\sqrt{2}}{2}.r + (p-1).\sqrt{2}.r, & p \in \{1, \dots, K_{f,1}\}, \\ S_{f,2,q} = \underline{b}_2 + \frac{\sqrt{2}}{2}.r + (q-1).\sqrt{2}.r, & q \in \{1, \dots, K_{f,2}\}, \end{cases} \quad (\text{VI.12})$$

avec

$$\begin{cases} \underline{b}_1 = \underline{x}_{1,0} - \frac{K_{f,1}.\sqrt{2}.r - w([x_1]_0)}{2}, \\ \underline{b}_2 = \underline{x}_{2,0} - \frac{K_{f,2}.\sqrt{2}.r - w([x_2]_0)}{2}. \end{cases}$$

Les positions des capteurs fixes seront obtenues suite à des combinaisons de ces coordonnées. Une combinaison $(S_{f,1,p}, S_{f,2,q})$, $p \in \{1, \dots, K_{f,1}\}$ et $q \in \{1, \dots, K_{f,2}\}$, définit alors la position $\mathbf{s}_{f,i} = (S_{f,1,p}, S_{f,2,q})$ d'un capteur fixe i , avec $i = p + K_{f,1} \cdot (q - 1)$, $i \in \{1, \dots, K_f\}$. FIG. VI.2 illustre un arrangement de capteurs fixes permettant de couvrir la zone de surveillance. Notons finalement que ces capteurs interviendront dans l'étape d'estimation, mais ne seront pas concernés dans la stratégie de repositionnement.

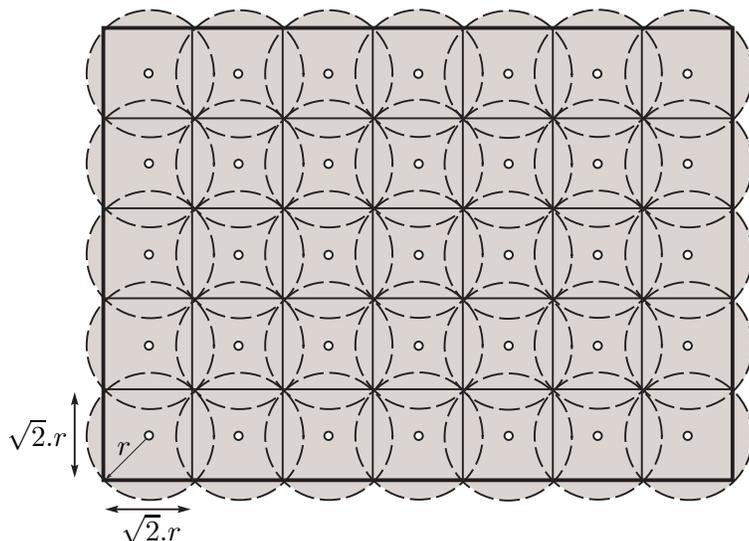


FIG. VI.2: Illustration d'un arrangement de capteurs fixes.

VI.4.2 Capteurs mobiles

Dans cette section, nous proposons une stratégie permettant de définir un ensemble de nouvelles positions qui seront attribuées par la suite aux capteurs mobiles. L'objectif de cette stratégie est de couvrir la zone de prédiction d'une manière optimale, tout en minimisant la consommation d'énergie. Afin de limiter le nombre de capteurs à déplacer, nous proposons d'utiliser ceux qui sont situés à une distance inférieure à une certaine limite d_0 de la surface à couvrir. Soit K_m le nombre maximal de capteurs mobiles que nous déplacerons. Il est alors impératif de définir K_m nouvelles positions dans cette stratégie.

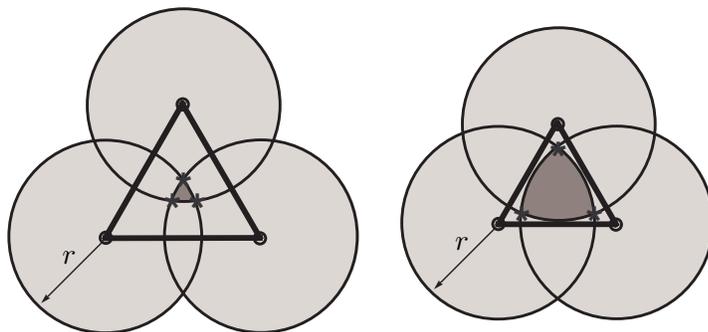


FIG. VI.3: Dépendance de la précision de l'estimation à la longueur des côtés du triangle.

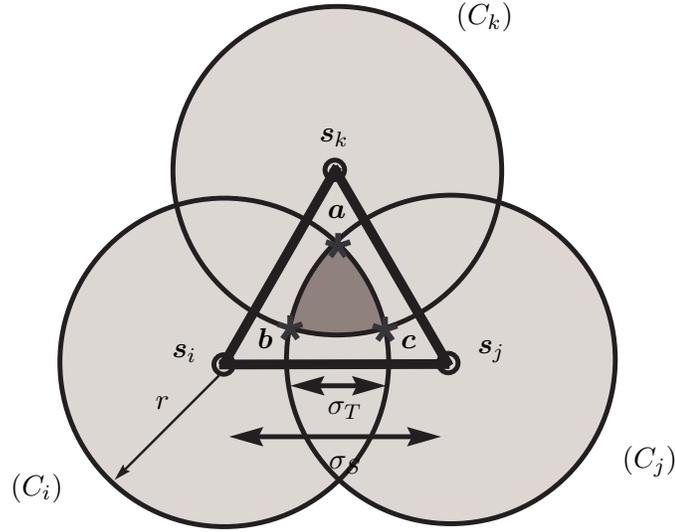


FIG. VI.4: Un illustration d'un triangle de capteurs.

La méthode que nous proposons se base sur le principe de triangulation, où les capteurs mobiles sont disposés sur les sommets de triangles. Supposons le cas d'un problème où trois capteurs sont à déplacer afin de couvrir un point unique. L'idée de base du principe de triangulation consiste à construire, à l'aide des capteurs, un triangle équilatéral, dont le barycentre coïncide avec le point d'intérêt. Si r est la portée de communication des capteurs, la position estimée de la cible correspondra à la zone d'intersection des trois disques centrés sur ces capteurs et de rayons r . La surface de cette zone est d'autant plus petite que la distance inter-capteurs est grande. Une illustration est donnée dans FIG. VI.3. La résolution dépendra alors de la distance séparant les capteurs. Notons **triangle de capteurs**, ou Δ_S , le triangle ayant les capteurs pour sommets et **triangle de cible**, ou Δ_T , le triangle inscrit dans la zone d'intersection des disques de communication. Alors, si $\sigma_S = \gamma_S.r$ et $\sigma_T = \gamma_T.r$ sont respectivement les longueurs des côtés de ces triangles, Δ_T est aussi équilatéral et γ_T est exprimé en fonction de γ_S de la manière suivante,

$$\gamma_T = \frac{\sqrt{3 \cdot (4 - \gamma_S^2)} - \gamma_S}{2}. \quad (\text{voir la preuve VI.1}) \quad (\text{VI.13})$$

Preuve VI.1. Considérons le triangle de capteurs illustré dans FIG. VI.4. Sans perte de généralité, nous montrons les calculs pour une disposition particulière du triangle. Une simple rotation du triangle de capteurs produira une rotation du triangle de la cible sans affecter la longueur de ses côtés. Si $\mathbf{s}_i = (s_{i,1}, s_{i,2})$ est le vecteur de coordonnées du capteur i situé en bas à gauche de la figure VI.4, les coordonnées des autres capteurs

VI. SUIVI DE CIBLES DANS LES RÉSEAUX MOBILES

seront déduites de \mathbf{s}_i comme suit,

$$\mathbf{s}_j \begin{cases} s_{j,1} = s_{i,1} + \sigma_S, \\ s_{j,2} = s_{i,2}, \end{cases} \quad \text{et} \quad \mathbf{s}_k \begin{cases} s_{k,1} = s_{i,1} + \frac{\sigma_S}{2}, \\ s_{k,2} = s_{i,2} + \frac{\sqrt{3}}{2} \cdot \sigma_S, \end{cases} \quad (\text{VI.14})$$

où $\frac{\sqrt{3}}{2} \cdot \sigma_S$ est la hauteur du triangle Δ_S . Les équations des cercles (C_i) , (C_j) et (C_k) sont alors données par,

$$\begin{aligned} (C_i) : & (x_1 - s_{i,1})^2 + (x_2 - s_{i,2})^2 = r^2, \\ (C_j) : & (x_1 - s_{i,1} - \sigma_S)^2 + (x_2 - s_{i,2})^2 = r^2, \\ (C_k) : & (x_1 - s_{i,1} - \frac{\sigma_S}{2})^2 + (x_2 - s_{i,2} - \frac{\sqrt{3}}{2} \cdot \sigma_S)^2 = r^2. \end{aligned} \quad (\text{VI.15})$$

Soient \mathbf{a} , \mathbf{b} et \mathbf{c} respectivement les points d'intersection des cercles (C_i) et (C_j) , (C_j) et (C_k) et (C_i) et (C_k) formant le triangle de la cible. En résolvant les systèmes d'équations des cercles, deux des trois étant pris à la fois, et en remplaçant σ_S par $\gamma_S \cdot r$, nous obtenons les expressions suivantes,

$$\begin{cases} a_1 = s_{i,1} + \frac{\gamma_S}{2} \cdot r, \\ a_2 = s_{i,2} + \frac{\sqrt{4 - \gamma_S^2}}{2} \cdot r, \end{cases} \quad (\text{VI.16})$$

$$\begin{cases} b_1 = s_{i,1} + \frac{\gamma_S + \sqrt{3 \cdot (4 - \gamma_S^2)}}{4} \cdot r, \\ b_2 = s_{i,2} + \frac{\sqrt{4 + 2 \cdot \gamma_S^2 - 2 \cdot \gamma_S \cdot \sqrt{3 \cdot (4 - \gamma_S^2)}}}{4} \cdot r, \end{cases} \quad (\text{VI.17})$$

$$\begin{cases} c_1 = s_{i,1} + \frac{3 \cdot \gamma_S - \sqrt{3 \cdot (4 - \gamma_S^2)}}{4} \cdot r, \\ c_2 = s_{i,2} + \frac{\sqrt{4 + 2 \cdot \gamma_S^2 - 2 \cdot \gamma_S \cdot \sqrt{3 \cdot (4 - \gamma_S^2)}}}{4} \cdot r. \end{cases} \quad (\text{VI.18})$$

Grâce à la symétrie dans le triangle des capteurs et à l'utilisation de disques de communication identiques, le triangle de la cible est équilatéral avec une longueur de côtés donnée par

$$\sigma_T = \gamma_T \cdot r = d(\mathbf{a}, \mathbf{b}) = d(\mathbf{a}, \mathbf{c}) = d(\mathbf{b}, \mathbf{c}) = \frac{\sqrt{3 \cdot (4 - \gamma_S^2)} - \gamma_S}{2} \cdot r. \quad (\text{VI.19})$$

□

Ayant prédit un pavé $[\hat{x}](t + 1)$ dans lequel la cible se trouvera à l'instant suivant, la méthode consiste à couvrir ce pavé d'une manière optimale. L'idée clé revient alors à attribuer des positions aux capteurs de telle sorte que $[\hat{x}](t + 1)$ soit rempli par des triangles de cible Δ_T . Or, le remplissage du pavé par des triangles indépendants ayant chacun ses trois capteurs propres nécessite un très grand nombre de capteurs. Dans ce cas, avec K_m capteurs mobiles disponibles, $\frac{K_m}{3}$ triangles indépendants seulement peuvent être obtenus. Afin de remplir le pavé, il faudrait alors créer de larges triangles de cible, ce qui conduit à une perte dans la précision de l'estimation.

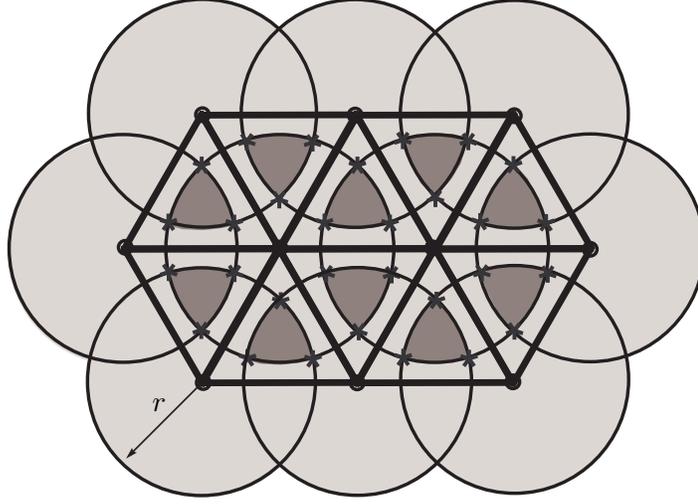


FIG. VI.5: Un exemple d'une structure de dix capteurs.

Dans cette méthode, nous proposons une stratégie alternative consistant à utiliser des structures rigides de triangles de capteurs Δ_S . Cette stratégie est plus avantageuse que l'autre, étant donné qu'elle nécessite moins de capteurs pour un même nombre de triangles de cible. Un exemple d'une structure de dix capteurs est illustré dans FIG. VI.5. Cette structure engendre dix triangles de cible, ce qui aurait nécessité trente capteurs indépendants avec l'ancienne stratégie. FIG. VI.5 montre en gris foncé les zones couvertes par trois capteurs de cette structure. Celles couvertes par un ou deux capteurs sont illustrées en gris clair. L'objectif de la méthode consiste alors à couvrir tous les points du pavé d'intérêt par au moins trois capteurs mobiles à l'aide de structures de Δ_S .

Une première technique revient à définir une structure de triangles de capteurs ayant r comme côtés. Selon la relation (VI.13), avec $\sigma_S = r$, $\sigma_T = r$ également. Si l'on couvre le pavé prédit par cette structure, tous les points du pavé se trouveront à l'intersection d'au moins trois disques de communication. Nous appellerons **structure garantie** ce type de structures. Une illustration d'une telle structure est donnée dans FIG. VI.6.(a). Soient $K_{m,1}^*$ et $K_{m,2}^*$ respectivement le nombre de capteurs formant la première ligne du bas de la structure et le nombre de lignes composant la structure. Toutes les lignes impaires en commençant par le bas seront composées de $K_{m,1}^*$ capteurs, les autres comprendront $K_{m,1}^* + 1$ capteurs mobiles chacune. Afin de couvrir un pavé prédit $[\hat{x}](t+1) = [\hat{x}_1](t+1) \times [\hat{x}_2](t+1)$, $K_{m,1}^*$ et $K_{m,2}^*$ sont définis comme suit,

$$\begin{cases} K_{m,1}^* = \overline{\mathbb{E}}\left\{\frac{w([\hat{x}_1](t+1))}{\sigma_S}\right\} + 1, \\ K_{m,2}^* = \overline{\mathbb{E}}\left\{\frac{w([\hat{x}_2](t+1))}{\frac{\sqrt{3}}{2} \cdot \sigma_S}\right\} + 1, \end{cases} \quad (\text{VI.20})$$

où $\overline{\mathbb{E}}\{x\}$ désigne le plus petit entier supérieur ou égal à x , $\sigma_S = r$, $\frac{\sqrt{3}}{2} \cdot \sigma_S$ est la hauteur

VI. SUIVI DE CIBLES DANS LES RÉSEAUX MOBILES

du triangle de capteurs et $w([x])$ est la largeur de l'intervalle $[x]$. Avec cette technique, l'étape d'estimation donnera des pavés généralement plus larges que les triangles de cible.

Une deuxième technique consiste à utiliser des triangles de capteurs ayant des côtés plus larges. Ceci produit des triangles de cible moins larges, conduisant à plus de précision dans l'estimation de la position de la cible au cas où celle-ci tombe à l'intérieur des triangles de cible générés. Avec cette technique, le pavé prédit ne sera pas entièrement couvert par les triangles de cible. Une illustration d'une telle structure est donnée dans FIG. VI.6.(b). Pour $\sigma_S = \sqrt{3}.r$, nous aurons $\sigma_T = 0$ et par conséquent les triangles de cible seront réduits à des points. Ainsi, pour obtenir théoriquement plus de précision dans l'estimation, il faut choisir une structure de triangles de capteurs avec σ_S comprise entre r et $\sqrt{3}.r$. Une fois σ_S est choisie, les nombres de capteurs par ligne ou par colonne sont définis comme suit,

$$\begin{cases} K_{m,1}^* = \underline{\mathbb{E}}\left\{\frac{w([\hat{x}_1](t+1))}{\sigma_S}\right\} + 1, \\ K_{m,2}^* = \overline{\mathbb{E}}\left\{\frac{w([\hat{x}_2](t+1))}{\frac{\sqrt{3}}{2}.\sigma_S}\right\} + 1, \end{cases} \quad (\text{VI.21})$$

où $\underline{\mathbb{E}}\{x\}$ désigne le plus grand entier inférieur ou égal à x . $\underline{\mathbb{E}}\{x\}$ a la particularité de donner 1 si $x \leq 1$. Notons que dans cette deuxième approche, nous utilisons $\underline{\mathbb{E}}$ pour $K_{m,1}^*$ afin d'avoir tous les triangles de cible le plus probablement inclus dans le pavé prédit.

Le nombre de capteurs nécessaires pour couvrir le pavé prédit dans les deux techniques est égal à,

$$K_m^* = K_{m,1}^* \cdot \frac{K_{m,2}^* + \delta}{2} + (K_{m,1}^* + 1) \cdot \frac{K_{m,2}^* - \delta}{2}, \quad (\text{VI.22})$$

avec $\delta = 1$ si $K_{m,2}^*$ est impair et $\delta = 0$ sinon. Afin d'obtenir des structures symétriques par rapport au centre du pavé prédit, les coordonnées des capteurs mobiles seront données par

$$\begin{cases} S_{m,1,p} = \begin{cases} \hat{b}_{o,1} + (p-1).\sigma_S, & \text{si } q \text{ est impair } (1 \leq p \leq K_{m,1}^*), \\ \hat{b}_{e,1} + (p-1).\sigma_S, & \text{si } q \text{ est pair } (1 \leq p \leq K_{m,1}^* + 1), \end{cases} \\ S_{m,2,q} = \hat{b}_2 + (q-1).\frac{\sqrt{3}}{2}.\sigma_S, \quad 1 \leq q \leq K_{m,2}^*, \end{cases} \quad (\text{VI.23})$$

avec

$$\begin{cases} \hat{b}_{o,1} = \inf([\hat{x}_1](t+1)) - \frac{K_{m,1}^*.\sigma_S - w([\hat{x}_1](t+1))}{2}, \\ \hat{b}_{e,1} = \inf([\hat{x}_1](t+1)) - \frac{(K_{m,1}^* + 1).\sigma_S - w([\hat{x}_1](t+1))}{2}, \\ \hat{b}_2 = \inf([\hat{x}_2](t+1)) - \frac{K_{m,2}^*.\frac{\sqrt{3}}{2}.\sigma_S - w([\hat{x}_2](t+1))}{2}. \end{cases}$$

Les nouvelles positions des capteurs mobiles seront obtenues suite à des combinaisons de ces coordonnées. Une combinaison $(S_{m,1,p}, S_{m,2,q})$ définira alors la position $s_{m,i}(t+1) = (S_{m,1,p}, S_{m,2,q})$ que prendra un capteur mobile i à l'instant $t+1$, avec $i = p + K_{m,1}^* \cdot \frac{q-1+\delta}{2} + (K_{m,1}^* + 1) \cdot \frac{q-1-\delta}{2}$, $\delta = 1$ si $q-1$ est impair et 0 sinon. Notons que si

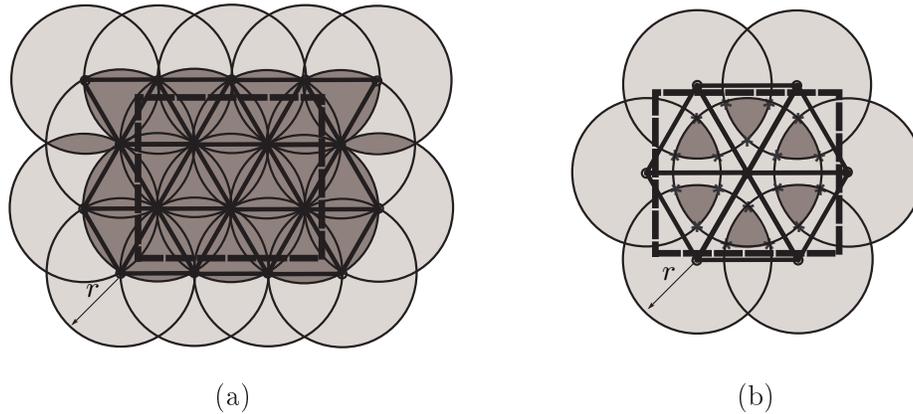


FIG. VI.6: Un exemple d'une structure garantie en (a) et d'une autre plus précise mais moins garantie en (b) permettant de couvrir un même pavé prédit.

$K_m^* > K_m$, nous nous contenterons des K_m positions les plus proches du centre du pavé. En revanche, si $K_m^* < K_m$, nous pouvons nous contenter de déplacer les K_m^* capteurs les plus proches du centre du pavé ou définir $K_m - K_m^*$ positions supplémentaires selon les techniques. Pour la première approche garantie, les positions supplémentaires seront choisies aléatoirement entre les K_m^* positions. Pour la seconde, ces positions seront définies par les barycentres des triangles de capteurs déjà définis. Avec cette technique, nous sommes en mesure de définir des triangles de cible supplémentaires. Si le nombre de positions obtenues reste inférieur à K_m , nous pourrions générer le reste d'une manière aléatoire dans le pavé prédit. Enfin, à l'aide de cette approche, nous nous trouvons en présence de K_m nouvelles positions qu'il faut attribuer aux K_m capteurs mobiles les plus proches du centre du pavé prédit.

VI.5 Optimisation du repositionnement des capteurs

L'étape précédente engendre un ensemble de positions que les capteurs mobiles doivent prendre à l'instant $t + 1$. Le but de cette étape consiste alors à attribuer à chacun des capteurs une position de manière à minimiser la distance totale parcourue par l'ensemble des capteurs. Le problème est alors défini comme un problème d'optimisation que nous résoudrons à l'aide de l'algorithme d'optimisation par colonies de fourmis, ACO (pour *Ant Colony Optimization*) [Dorigo et Gambardella, 1997; Dréo *et al.*, 2005]. Nous introduirons alors dans cette section l'algorithme ACO. Nous l'appliquerons par la suite pour résoudre le problème de repositionnement des capteurs.

VI.5.1 Algorithme d'optimisation par colonies de fourmis

Les algorithmes ACO forment une classe des métaheuristiques récemment présentée pour des problèmes d'optimisation complexe. Le premier algorithme de ce type a été

VI. SUIVI DE CIBLES DANS LES RÉSEAUX MOBILES

proposé pour traiter le problème du voyageur de commerce [Dorigo et Gambardella, 1997]. Toutefois, cet algorithme n'a pas permis de produire des résultats compétitifs. Ultérieurement, de nombreux algorithmes, atteignant des résultats très convaincants, ont été proposés dans différents domaines, notamment ceux des problèmes d'affectation quadratique [Hani *et al.*, 2007], des tournées de véhicules [Donati *et al.*, 2008], des conceptions de lignes d'assemblage [Chehade *et al.*, 2009], etc. Ces algorithmes s'inspirent des comportements collectifs de dépôt et de suivi de piste observés réellement dans les colonies de fourmis. L'idée de base de ces algorithmes consiste à imiter les fourmis réelles dans leur façon à rechercher le trajet le plus court pour arriver aux sources de nourriture. Un trajet est alors généré selon deux éléments de base : une substance chimique déposée appelée phéromone et la visibilité de la fourmi.

Soit $f(x_1, \dots, x_n)$ une fonction de n variables dont les valeurs sont choisies d'un certain ensemble S . Optimiser f consiste à trouver la n -permutation de (x_1, \dots, x_n) , parmi toutes les permutations possibles, produisant la valeur optimale de f . Dans le cadre des algorithmes d'optimisation, la fonction f s'appelle fonction objective ou fonction *fitness*, alors que les variables x_1, \dots, x_n sont dites variables de décision. Soit m le cardinal de S , alors le nombre de toutes les n -permutations possibles est égal à $\frac{m!}{(m-n)!}$, où $m!$ désigne la factorielle de m . L'évaluation de toutes les solutions possibles nécessite un temps calculatoire considérable, particulièrement pour les problèmes de grandes tailles. Dans ces situations, l'utilisation d'algorithmes d'optimisation, tels que les ACO, devient crucial pour réduire le temps de calcul.

Partant d'une solution initiale, les algorithmes ACO atteignent des solutions optimales en utilisant une technique de recherche efficace basée sur la mémoire. La génération des solutions emploie essentiellement deux paramètres : la visibilité et les phéromones. Ces paramètres représentent respectivement les informations a priori et a posteriori concernant les solutions. Alors que la visibilité reste inchangée, les phéromones varient au cours de l'algorithme en fonction des résultats d'évaluation des solutions obtenues. En pratique, l'algorithme ACO considère un nombre fixe de fourmis K_a , chacune d'elles générant une solution à chaque itération. Les solutions sont alors codées tout en affectant chaque variable de décision, l'une après l'autre, une valeur spécifique de S . Soit $S = \{a_1, \dots, a_m\}$. Une fourmi k , $k \in \{1, \dots, K_a\}$, ayant attribuée la valeur a_j à la variable a_i , affectera la variable x_{i+1} la valeur a_l choisie de la manière suivante,

$$l = \begin{cases} \operatorname{argmax}_{u \in J_k(i)} \{ \tau_{(i),j,u}^\alpha \cdot \eta_{(i),j,u}^\beta \}, & \text{si } q \leq q_0, \\ l^*, & \text{sinon,} \end{cases} \quad (\text{VI.24})$$

avec :

- $\tau_{(i),j,u}$: La quantité de phéromone correspondant au fait de choisir $x_{i+1} = a_u$ sachant que $x_i = a_j$,
 $\eta_{(i),j,u}$: La visibilité de la fourmi ou sa désirabilité innée de choisir la valeur a_u pour x_{i+1} en partant de a_j ,
 α et β : Paramètres définissant l'importance du phéromone par rapport à la visibilité,
 q : Un nombre aléatoirement choisi à chaque itération entre 0 et 1,
 q_0 : Un paramètre ($0 < q_0 < 1$) définissant l'importance de l'exploitation par rapport à l'exploration,
 $J_k(i)$: L'ensemble des indices des valeurs de S non encore attribuées par la fourmi k ,
 l^* : Une variable aléatoire générée selon une distribution spécifique de probabilité P^* .

La distribution de probabilité P^* est définie de la manière suivante,

$$P^*(u) = \begin{cases} \frac{\tau_{(i),j,u}^\alpha \cdot \eta_{(i),j,u}^\beta}{\sum_{v \in J_k(i)} \tau_{(i),j,v}^\alpha \cdot \eta_{(i),j,v}^\beta}, & \text{si } u \in J_k(i), \\ 0, & \text{sinon.} \end{cases} \quad (\text{VI.25})$$

Notons que la première variable est attribuée des valeurs aléatoirement choisies de S . A l'aide de la règle pseudo-aléatoire proposée ci-dessus, l'état suivant est soit déterministe dépendant de η et de τ ($q \leq q_0$), soit aléatoirement choisi selon la distribution P^* ($q > q_0$). La méthode fournit alors une manière directe pour équilibrer entre l'exploration de nouveaux états et l'exploitation des informations a priori et collectées au fur et à mesure dans l'algorithme.

A la fin de chaque itération, où chaque fourmi a complété une solution, le phéromone de tout le système s'évapore. Le phéromone correspondant aux solutions est alors localement mis à jour comme suit,

$$\tau_{(i),j,l} = (1 - \rho) \cdot \tau_{(i),j,l} + \rho \cdot \tau_0, \quad (\text{VI.26})$$

où $1 \leq i \leq m-1$, $\{j, l\} \in \text{solutions des fourmis}$, τ_0 est la quantité initiale de phéromone et ρ est un paramètre compris entre 0 et 1. Les solutions sont ensuite évaluées, chacune produisant une valeur de la fonction *fitness*. Seule la meilleure solution, optimisant la fonction, est utilisée pour mettre à jour le phéromone lui correspondant comme suit,

$$\tau_{(i),j,l} = (1 - \rho) \cdot \tau_{(i),j,l} + \rho \cdot \Delta\tau, \quad (\text{VI.27})$$

où $\{j, l\} \in \text{la meilleure solution}$ et $\Delta\tau$ est fonction de la meilleure solution. Ce mécanisme est appelé mise à jour globale.

L'algorithme effectue ainsi un certain nombre d'itérations *MaxIter* fixé à l'avance. Au cours de chaque itération, chacune des fourmis crée une solution. Le phéromone de chacune des solutions est mis à jour localement, puis les solutions sont évaluées. Le phéromone correspondant à celle qui produit le résultat optimal est ensuite mis à jour globalement. Cette solution optimale est par la suite sauvegardée. L'algorithme itère ces mêmes étapes jusqu'à ce qu'il atteigne *MaxIter* où seule la solution donnant le meilleur résultat global est finalement choisie.

VI.5.2 Repositionnement des capteurs

Le problème de repositionnement des capteurs revient à attribuer aux capteurs mobiles de nouvelles positions tout en minimisant la distance totale parcourue par l'ensemble des capteurs. Nous définissons alors la fonction *fitness* par la somme des distances parcourues par les capteurs en mouvement et les variables de décision par les coordonnées des capteurs. Ces variables prennent leurs valeurs dans l'ensemble de nouvelles positions défini dans la section VI.4.2. Soient $\mathbf{s}_{m,1}(t), \dots, \mathbf{s}_{m,K_m}(t)$ les coordonnées des capteurs que l'on déplacera à l'instant t . Alors $\mathbf{s}_{m,1}(t+1), \dots, \mathbf{s}_{m,K_m}(t+1)$ définirons les nouvelles coordonnées. La fonction *fitness* est ainsi formulée de la manière suivante,

$$f(\mathbf{s}_{m,1}, \dots, \mathbf{s}_{m,K_m}) = \sum_{i=1}^{K_m} d(\mathbf{s}_{m,i}(t), \mathbf{s}_{m,i}(t+1)), \quad (\text{VI.28})$$

où $d(\cdot, \cdot)$ est l'opérateur de la distance euclidienne. Soit $S = \{\mathbf{a}_1, \dots, \mathbf{a}_{K_m}\}$ l'ensemble des positions générées à l'étape précédente. Le nombre de toutes les solutions possibles est égal à $K_m!$.

La résolution du problème à l'aide de l'algorithme ACO revient à itérer les équations (VI.24), (VI.25), (VI.26) et (VI.27) jusqu'à ce qu'un nombre maximal d'itérations *MaxIter* soit atteint. Cette étape nécessite la définition de tous les paramètres. La visibilité est alors définie en fonction de la distance parcourue par les capteurs comme suit,

$$\eta_{(i),j,l} = \frac{1}{\|\mathbf{s}_{m,i}(t), \mathbf{a}_j\| + \|\mathbf{s}_{m,i+1}(t), \mathbf{a}_l\|}, \quad (\text{VI.29})$$

avec $\eta_{(i),j,l}$ étant la visibilité de la fourmi de choisir $\mathbf{s}_{m,i+1}(t+1) = \mathbf{a}_l$ après avoir choisi $\mathbf{s}_{m,i}(t+1) = \mathbf{a}_j$. Selon cette définition, la visibilité augmente avec la diminution de la distance totale parcourue. Alors que la visibilité varie d'un état à l'autre, la valeur initiale du phéromone est prise constante pour tous les états. $\Delta\tau$ est défini au cours de chaque itération en fonction de la solution optimale de telle sorte que $\Delta\tau = \frac{1}{f_{min}}$ où f_{min} est la distance totale minimale correspondant à la meilleure solution. Les autres paramètres $\alpha, \beta, \rho, q_0, \tau_0, K_a$ et *MaxIter* sont choisis suite à plusieurs tests.

Après avoir réalisé toutes les itérations, la méthode affecte aux K_m capteurs mobiles la K_m -permutation de positions qui minimise la fonction *fitness*. Le pseudo-code de la technique de repositionnement associé au pas de temps t est illustré dans l'algorithme 12, où V_k représente le vecteur des indices des positions associées aux capteurs par la fourmi k , \setminus est l'opérateur de la différence ensembliste, $\text{random}(A)$ produit un nombre aléatoire choisi dans l'ensemble A selon la loi uniforme et $\text{random}_{P^*}(A)$ produit un nombre aléatoire choisi dans l'ensemble A selon la distribution P^* .

VI.6 Simulations

Dans cette section, nous évaluons les performances de la méthode proposée. Nous supposons alors que la cible se déplace dans une zone carrée de dimensions $100m \times 100m$. Cette zone sera appelée zone de surveillance dans la suite. Si nous fixons la portée de

Algorithme 12 : Algorithme de repositionnement.

Entrées : $s_{m,1}(t), \dots, s_{m,K_m}(t), \mathbf{a}_1, \dots, \mathbf{a}_{K_m}, \alpha, \beta, \rho, q_0, \tau_0, K_a$ et $MaxIter$;

Sorties : $s_{m,1}(t+1), \dots, s_{m,K_m}(t+1)$;

Initialisation :

pour $1 \leq i \leq K_m - 1$ **faire**

pour $1 \leq j \leq K_m$ **faire**

pour $1 \leq l \leq K_m$ **faire**

$$\eta_{(i),j,l} = \frac{1}{\|s_{m,i}(t), \mathbf{a}_j\| + \|s_{m,i+1}(t), \mathbf{a}_l\|};$$

$$\tau_{(i),j,l} = \tau_0;$$

fin

fin

fin

tant que $iter \leq MaxIter$ **faire**

$q = \text{random}([0, 1])$;

pour $1 \leq k \leq K_a$ **faire**

$V_k(1) = \text{random}(\{1, \dots, K_m\})$;

$J_k(1) = \{1, \dots, K_m\} \setminus V_k(1)$;

pour $1 \leq i \leq K_m - 1$ **faire**

si $q \leq q_0$ **alors**

$$V_k(i+1) = \text{argmax}_{u \in J_k(i)} \{\tau_{(i),V_k(i),u}^\alpha \cdot \eta_{(i),V_k(i),u}^\beta\};$$

$$J_k(i+1) = J_k(i) \setminus V_k(i+1);$$

$$\tau_{(i),V_k(i),V_k(i+1)} = (1 - \rho) \cdot \tau_{(i),V_k(i),V_k(i+1)} + \rho \cdot \tau_0;$$

fin

si $q > q_0$ **alors**

pour $u \in J_k(i)$ **faire**

$$P^*(u) = \frac{\tau_{(i),V_k(i),u}^\alpha \cdot \eta_{(i),V_k(i),u}^\beta}{\sum_{v \in J_k(i)} \tau_{(i),V_k(i),v}^\alpha \cdot \eta_{(i),V_k(i),v}^\beta};$$

fin

pour $u \notin J_k(i)$ **faire**

$$P^*(u) = 0;$$

fin

$V_k(i+1) = \text{random}_{P^*}(\{1, \dots, K_m\})$;

$J_k(i+1) = J_k(i) \setminus V_k(i+1)$;

$$\tau_{(i),V_k(i),V_k(i+1)} = (1 - \rho) \cdot \tau_{(i),V_k(i),V_k(i+1)} + \rho \cdot \tau_0;$$

fin

fin

fin

pour $1 \leq k \leq K_a$ **faire**

$$Ob(k) = \sum_{i=1}^{K_m} \|s_{m,i}(t), \mathbf{a}_{V_k(i)}\|;$$

fin

$k^* = \text{argmin}_{1 \leq k \leq K_a} Ob$;

$C_{iter} = V_{k^*}$, $MinOb(iter) = Ob(k^*)$;

$$\Delta\tau = \frac{1}{Ob(k^*)};$$

pour $1 \leq i \leq K_m - 1$ **faire**

$$\tau_{(i),V_{k^*}(i),V_{k^*}(i+1)} = (1 - \rho) \cdot \tau_{(i),V_{k^*}(i),V_{k^*}(i+1)} + \rho \cdot \Delta\tau;$$

fin

fin

$iter^* = \text{argmin}_{1 \leq iter \leq MaxIter} MinOb$;

pour $1 \leq i \leq K_m$ **faire**

$s_{m,i}(t+1) = C_{iter^*}(i)$;

fin

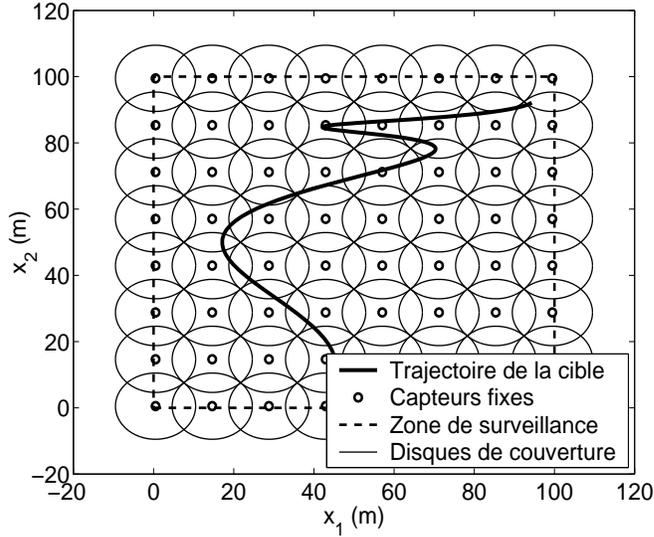


FIG. VI.7: Une illustration de la trajectoire de la cible avec des capteurs fixes uniformément répartis.

communication r à $10m$, le nombre de capteurs fixes nécessaires pour couvrir la totalité de la zone par au moins un capteur est donné par

$$K_f = K_{f,1} * K_{f,2} = \overline{E}\left(\frac{100}{\sqrt{2.10}}\right) \cdot \overline{E}\left(\frac{100}{\sqrt{2.10}}\right) = 8 * 8 = 64.$$

FIG. VI.7 illustre les capteurs fixes ainsi qu'une trajectoire de la cible sur une durée de 100 pas de temps avec une période $\Delta t = 10s$. Nous supposons ainsi que la cible sera localisée toutes les $10s$. Notons que les capteurs fixes n'ont pas nécessairement la même portée de communication que les capteurs mobiles. Ils peuvent alors avoir une portée plus grande, permettant ainsi de réduire le nombre nécessaire pour couvrir la totalité de zone de surveillance. Dans la suite, nous comparons en premier la méthode proposée à une méthode basée sur les intervalles n'employant que des capteurs fixes. Nous étudions ensuite l'effet de la distance inter-capteurs sur les performances de la méthode. Nous évaluons par ailleurs l'efficacité du modèle de prédiction. Nous illustrons enfin les performances de l'algorithme ACO. Il faut noter que toutes les simulations sont réalisées sur Matlab 6.1, installé sur un CPU Intel(R) Core(TM)2 (2.40GHz, 1.00GB RAM).

VI.6.1 Comparaison à une méthode statique basée sur les intervalles

Dans cette section, nous comparons notre méthode à une technique de suivi de cibles développée pour les réseaux de capteurs fixes. Pour cette raison, nous proposons une méthode statique de suivi de cibles basée sur les intervalles. Cette méthode effectue uniquement l'étape d'estimation de notre approche. Dans la suite, la méthode statique sera notée SSt alors que la méthode que nous proposons dans ce chapitre sera notée

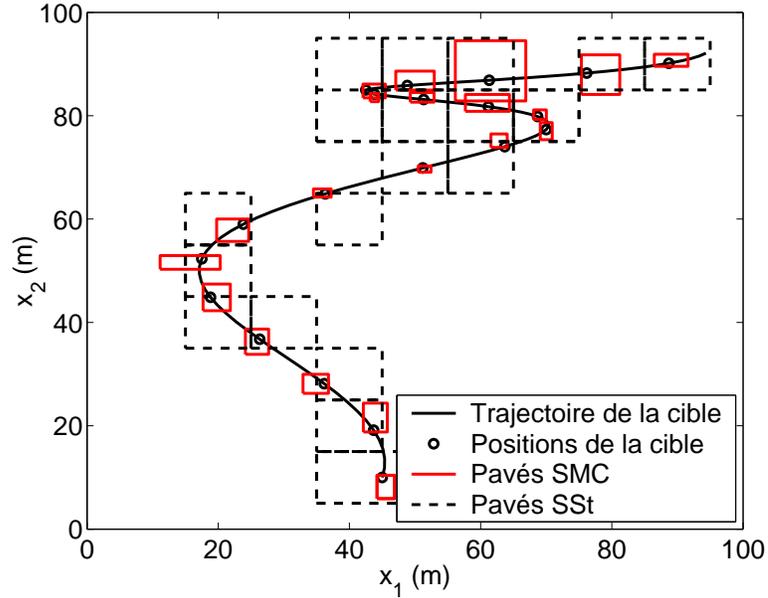


FIG. VI.8: Pavés estimés à l'aide des méthodes SMC et SSt.

SMC (pour Suivi de cibles en cas de Mobilité Contrôlée). Nous utilisons 100 capteurs au total pour les deux méthodes. Dans la méthode statique, les cent capteurs sont uniformément répartis dans la zone de surveillance alors que dans notre méthode, 64 capteurs sont fixes uniformément répartis et 36 sont mobiles. Ces derniers sont lancés initialement d'une manière aléatoire dans la zone de surveillance. Notons que dans cette section, tous les capteurs mobiles seront repositionnés, ce qui conduit à $36! = 3.7199.10^{41}$ solutions possibles. FIG. VI.8 montre les pavés estimés à l'aide des deux méthodes ainsi que la trajectoire de la cible sur 1000s avec $\Delta t = 10s$. Nous considérons pour notre méthode de larges triangles de capteurs avec $\sigma_S = 10.\sqrt{2}m$, $\tau_0 = 10$, $\alpha = 0.7$, $\beta = 0.3$, $\rho = 0.6$, $q_0 = 0.75$ et $K_a = 10$. A l'aide de ces valeurs, la solution optimale est obtenue avec dix itérations au maximum. Le temps de calcul moyen par pas de temps est égal à $0.9419s$ à l'aide de la méthode SMC alors qu'il est de l'ordre de $0.00069s$ à l'aide de SSt. En d'autres termes, le temps de calcul mis par SMC est essentiellement dû à l'étape d'optimisation. La distance parcourue par pas de temps vaut $7.4539m$ par capteur mobile.

Définissons l'erreur d'estimation à un instant donné par la distance entre la position réelle de la cible et le centre du pavé estimé à cet instant. FIG. VI.9 montre les rapports des surfaces des pavés et des erreurs d'estimation obtenues à l'aide de notre méthode (SMC) sur celles obtenues à l'aide de la méthode statique (SSt). La moyenne des rapports des surfaces est égale à 0.1740 alors que celle des rapports des erreurs est égale à 0.4111 . En conséquence, il est évident que le déplacement des capteurs améliore sensiblement la qualité du suivi de cibles.

Par ailleurs, nous varions le nombre total de capteurs que nous utiliserons dans

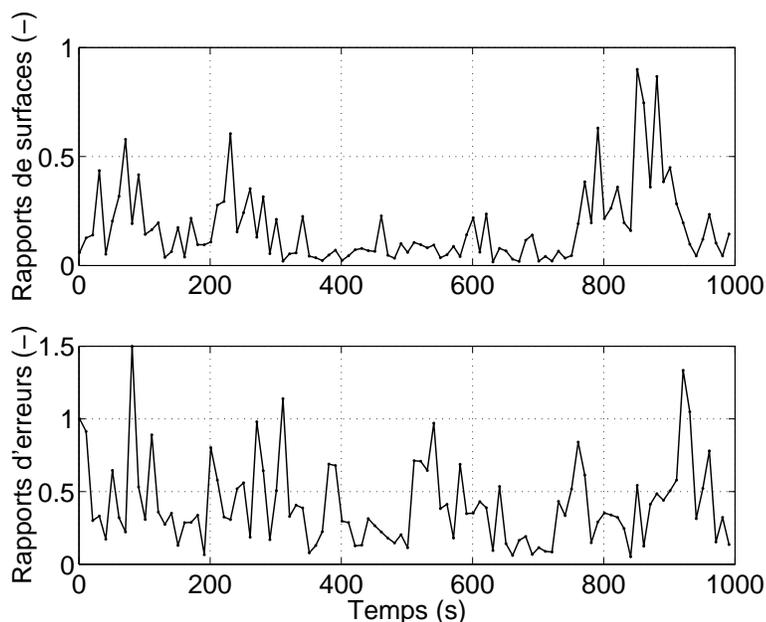


FIG. VI.9: Rapports des surfaces des pavés en haut et des erreurs d'estimation en bas SMC/SSt.

le réseau, 64 étant le nombre inchangeable des capteurs fixes pour la méthode SMC. Nous varions alors le nombre de capteurs considérés de 75 à 125. Le nombre des capteurs mobiles variera ainsi de 11 à 61. FIG. VI.10 montre la variation du temps de calcul moyen avec SMC, la variation du rapport moyen des surfaces des pavés estimés avec SMC sur celles obtenues avec SSt ainsi que la variation du rapport moyen des erreurs d'estimation SMC/SSSt en fonction du nombre total des capteurs. Comparée à la méthode statique, les performances de notre méthode sont de plus en plus meilleures avec l'augmentation du nombre des capteurs mobiles mis en jeu au détriment du temps de calcul.

VI.6.2 Influence de la distance inter-capteurs sur les performances

Dans cette section, nous illustrons l'effet de la distance inter-capteurs sur la précision de l'estimation. Pour cette raison, nous considérons 80 capteurs, 64 parmi eux étant fixes pour assurer la couverture. Nous comparons ainsi notre méthode avec $\sigma_S = 10m$, notée SMCg, à une deuxième version avec $\sigma_S = 10\sqrt{2}m$, notée SMCp. Nous considérons pour le faire la trajectoire de la cible illustrée dans FIG. VI.7 sur 1000s avec $\Delta t = 10s$. FIG. VI.11 illustre les rapports des surfaces des pavés et des erreurs d'estimation obtenues avec SMCp sur SMCg. Notons que pour cette figure, nous utilisons le même nombre de capteurs K_m pour les deux méthodes. Celui-ci est égal, à chaque instant, au nombre minimal de capteurs mobiles nécessaires pour couvrir d'une manière garantie le pavé prédit. La figure montre que les pavés obtenus à l'aide de SMCp sont plus petits que ceux obtenus à l'aide de SMCg à certains pas de temps et vice-versa à d'autres.

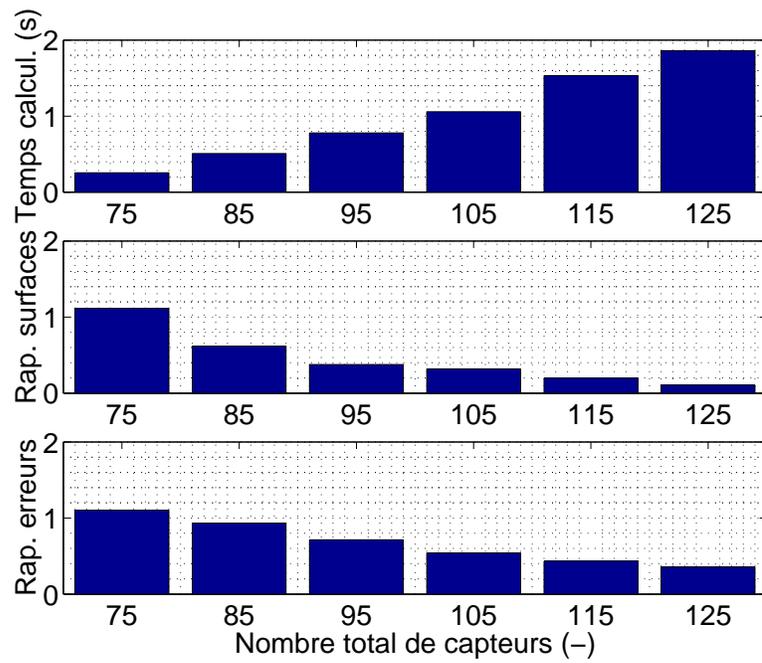


FIG. VI.10: Variations du temps de calcul moyen par pas de temps en haut, du rapport moyen des surfaces des pavés au milieu et du rapport moyen des erreurs d'estimation SMC/SSt en bas en fonction du nombre total des capteurs considérés dans le réseau.

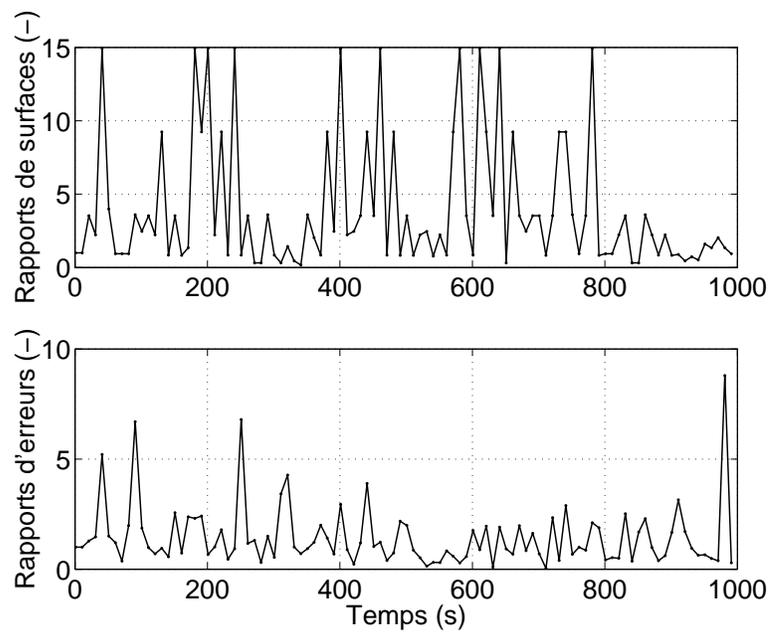


FIG. VI.11: Rapports des surfaces des pavés en haut et des erreurs d'estimation en bas SMC_p/SMC_g.

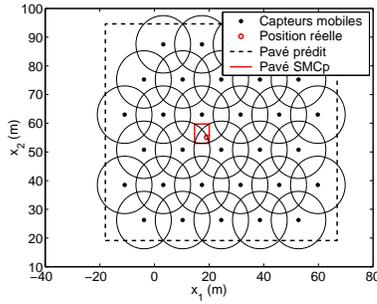


FIG. VI.12: Capteurs mobiles, position réelle et pavé estimé avec SMCp à l'instant $t = 28$.

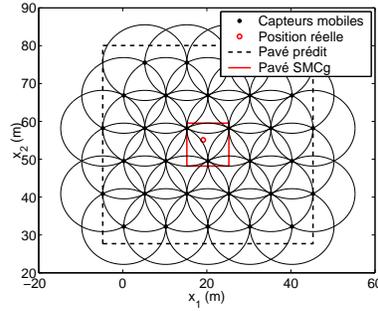


FIG. VI.13: Capteurs mobiles, position réelle et pavé estimé avec SMCg à l'instant $t = 28$.

Pour $t = 28$ par exemple, la méthode SMCp est plus précise que SMCg. FIG. VI.12 et FIG. VI.13 montrent la position réelle de la cible à cet instant ainsi que les capteurs mobiles repositionnés avec leurs disques de couverture à l'aide des méthodes SMCp et SMCg respectivement. Les figures montrent en plus les pavés prédits ainsi que les pavés estimés. Selon ces deux figures, la position réelle de la cible tombe dans les triangles de cible des deux structures, ce qui conduit à plus de précision avec la méthode SMCp, où σ_S est plus grande. En revanche, FIG. VI.14 et FIG. VI.15 montrent les capteurs repositionnés à l'instant $t = 14$ respectivement à l'aide des méthodes SMCp et SMCg. Les figures montrent que la position réelle de la cible tombe à l'extérieur des triangles de cible avec SMCp alors qu'elle tombe dans la zone d'intersection de quatre capteurs mobiles avec SMCg. D'où l'obtention d'un pavé plus petit avec SMCg. Notons que dans cette section, seuls les capteurs mobiles sont utilisés dans l'estimation de la position de la cible, afin de bien illustrer l'effet de la taille des triangles de capteurs. L'utilisation des capteurs fixes produira des pavés plus petits que ceux obtenus uniquement avec les capteurs mobiles. Les rapports moyens des surfaces des pavés et des erreurs d'estimation obtenues avec SMCp sur celles obtenues avec SMCg sont respectivement égaux à 3.963 et 1.4489. Il est alors clair que SMCg est plus performante que SMCp pour cet exemple. En effet, avec un faible nombre de capteurs mobiles, la méthode garantie avec $\sigma_S = r$ couvre mieux le pavé prédit vu qu'avec SMCp, la surface couverte par moins de trois capteurs est très large. Toutefois, avec un nombre de capteurs mobiles plus grand, il est possible de construire des triangles de capteurs plus larges, tout en choisissant la version SMCp. Dans la suite, nous adopterons la version garantie de la méthode.

VI.6.3 Evaluation du modèle de prédiction

Dans cette section, nous étudions la sensibilité de la méthode aux performances du modèle de prédiction. Etant un modèle du second ordre, le modèle de prédiction combine trois positions estimées consécutives afin de définir un pavé prédit. Comparés aux pavés estimés, les pavés prédits sont généralement larges accumulant toute l'incertitude présente sur les estimés précédents. Le modèle que nous utilisons suppose que l'accéléra-

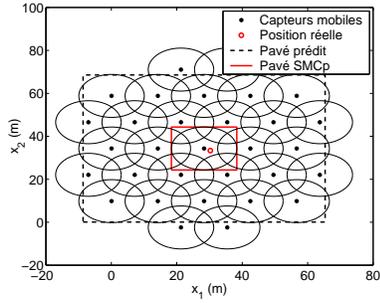


FIG. VI.14: Capteurs mobiles, position réelle et pavé estimé avec SMCp à l'instant $t = 14$.

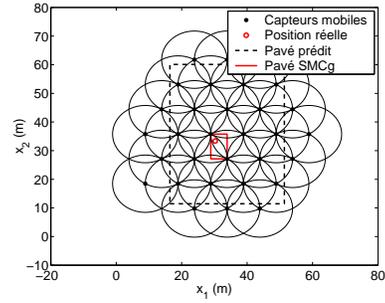


FIG. VI.15: Capteurs mobiles, position réelle et pavé estimé avec SMCg à l'instant $t = 14$.

tion de la cible est constante. Ainsi, le pavé prédit peut ne pas couvrir la position réelle de la cible quand celle-ci a un changement brusque de direction. Afin d'illustrer un cas pareil, nous considérons une cible se déplaçant selon le modèle de mobilité de marche aléatoire ou *Random Walk Mobility Model* [Camp *et al.*, 2002]. Avec ce modèle, la cible possède un mouvement non-prédictible, ayant à chaque instant une vitesse aléatoire choisie entre 0 et $1m.s^{-1}$ et une direction aléatoire choisie entre 0 et $2.\pi$. FIG. VI.16 illustre la trajectoire de la cible sur 1000s ($\Delta t = 10s$) avec les pavés estimés à l'aide de notre méthode. Alors que le modèle de prédiction génère de faux pavés prédits à 35 pas de temps, l'estimation reste le plus souvent très précise. Ceci est principalement dû à l'utilisation des capteurs fixes pour assurer la couverture totale. FIG. VI.17 montre les positions des capteurs mobiles, les pavés prédits et estimés et les positions réelles de la cible correspondant aux instants $t = 18$ (en noir) et $t = 19$ (en rouge). Cette figure montre que malgré le fait que le modèle de prédiction ne fonctionne pas bien à un pas de temps, il pourra réaliser une prédiction correcte à l'instant suivant.

VI.6.4 Efficacité de la méthode ACO

Dans cette section, nous illustrons les performances de l'algorithme d'optimisation par colonies de fourmis ou ACO. Pour cette raison, nous le comparons à la méthode exacte où toutes les permutations possibles sont évaluées. Nous considérons ainsi un réseau composé de 64 capteurs fixes et 10 capteurs mobiles que nous repositionnons tous à chaque pas temps. Le nombre total de solutions possibles est alors égal à $10! = 3\,628\,800$. Ayant un ensemble de 10 positions, la méthode exacte consiste à générer toutes les 10-permutations de solutions où chaque capteur mobile sera attribué une position de l'ensemble. La méthode calcule ensuite la distance totale parcourue par les capteurs pour chacune des permutations. La 10-permutation fournissant la distance minimale sera choisie au final. Afin d'illustrer la comparaison entre les deux méthodes, nous déterminons l'ensemble des nouvelles positions à chaque pas temps, puis nous utilisons la méthode ACO et la méthode exacte pour attribuer à chaque capteur une nouvelle position de l'ensemble.

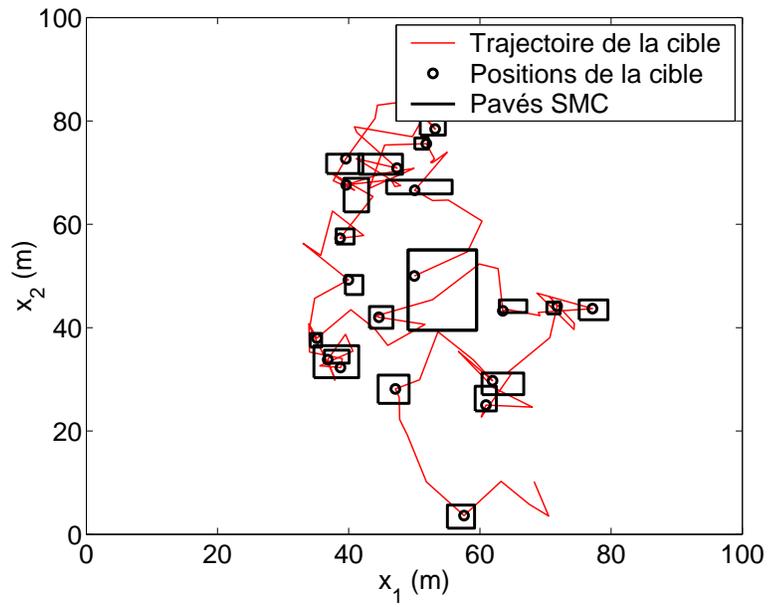


FIG. VI.16: Trajectoire non-prédictible de la cible et pavés estimés.

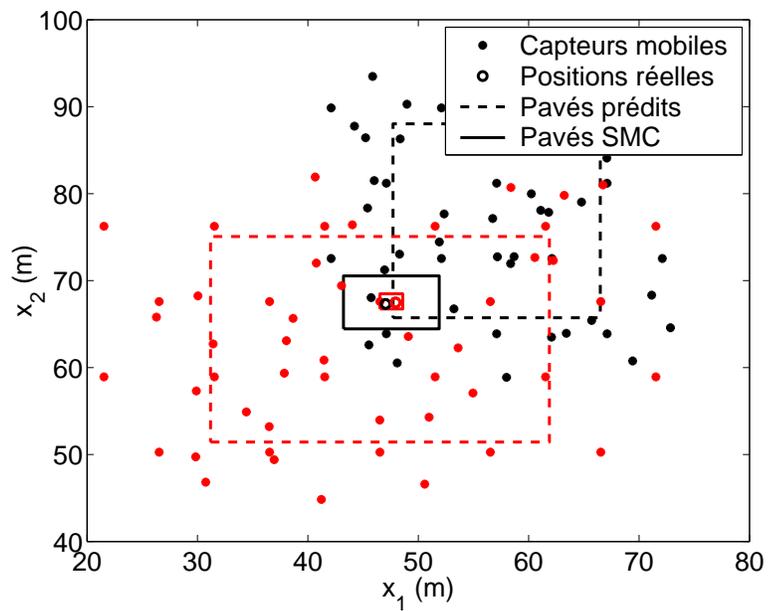


FIG. VI.17: Capteurs mobiles, positions réelles et pavés prédits et estimés aux instants $t = 18$ (en noir) et $t = 19$ (en rouge).

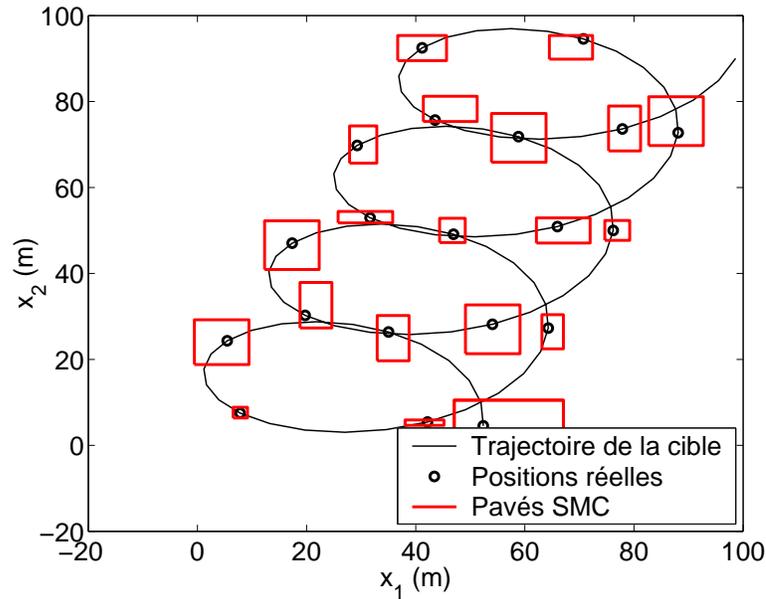


FIG. VI.18: Trajectoire de la cible et pavés estimés obtenus pareillement à l'aide de la méthode ACO ou la méthode exacte.

La trajectoire de la cible, sur 1000s avec $\Delta t = 10s$, que nous utilisons dans cette section est illustrée dans FIG. VI.18 avec les pavés obtenus. Le temps de calcul moyen par pas de temps requis par notre méthode est égale à 0.1973s, alors qu'il vaut 17.9483s pour la méthode exacte. En d'autres termes, l'utilisation de l'algorithme ACO réalise un gain de 98.90% au niveau du temps de calcul. FIG. VI.19 illustre les rapports des temps de calcul et des distances parcourues obtenus avec ACO sur ceux obtenus avec la méthode exacte. Le rapport moyen des distances parcourues avec ACO sur la méthode exacte est de 1.0339, ce qui induit une petite augmentation de la distance parcourue de 3.39% avec ACO par rapport à la méthode exacte. En conséquence, l'utilisation de l'algorithme ACO nous permet de réaliser un gain de 98.9% en temps de calcul pour une légère augmentation de la distance parcourue de 3.39%. Il devient alors crucial d'utiliser l'algorithme ACO surtout lorsque le nombre de capteurs mobiles concernés augmente.

VI.7 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode pour le suivi de cibles dans les réseaux de capteurs à mobilité contrôlée. Ayant une cible mobile dans une zone de surveillance, la méthode consiste à estimer instantanément la position courante de la cible puis à prédire sa position future en utilisant un modèle de prédiction du second ordre. Un repositionnement des capteurs est par la suite effectué afin d'optimiser la localisation de la cible au pas de temps suivant. Un ensemble de nouvelles positions est alors défini en utilisant une méthode basée sur la triangulation. Chaque capteur est

VI. SUIVI DE CIBLES DANS LES RÉSEAUX MOBILES

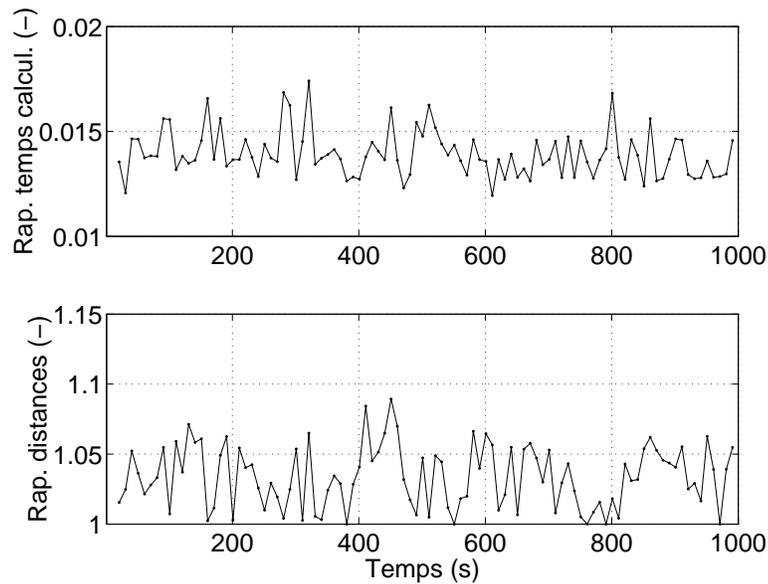
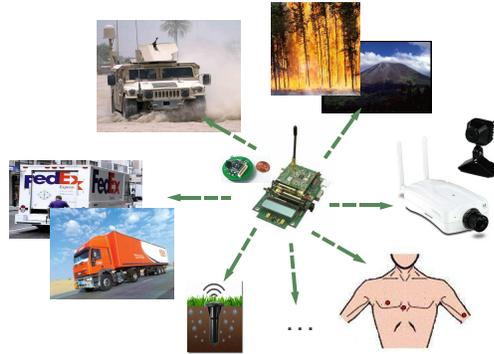


FIG. VI.19: Rapports des temps de calcul et des distances parcourues avec la méthode ACO sur ceux obtenus avec la méthode exacte.

ensuite attribué une position de l'ensemble à l'aide de l'algorithme d'optimisation par colonies de fourmis. Alors que la phase de repositionnement des capteurs est réalisée à l'aide d'une métaheuristique, les étapes d'estimation et de prédiction emploie l'analyse par intervalles où les positions de la cible sont des pavés couvrant les positions réelles. L'approche proposée utilise un réseau hybride comprenant des capteurs fixes et mobiles. Alors que les capteurs mobiles permettent d'optimiser le suivi de la cible, les capteurs statiques assurent la couverture totale du réseau indépendamment des capteurs mobiles. Dans les travaux futurs, nous nous intéresserons à traiter le problème d'une manière distribuée où les décisions sont réalisées localement. Le problème peut être également étendu au cas de cibles multiples.

VII Conclusion et perspectives



Sommaire

VII.1 Résumé et contributions

VII.2 Travaux futurs et suggestions

Dans ce mémoire, nous avons développé des techniques d'estimation d'états faisant appel à la théorie des intervalles dans les réseaux de capteurs mobiles. Tout en respectant les contraintes d'énergie et de mémoire, nous avons traité les problèmes d'auto-localisation et de suivi de cibles. Nous avons alors proposé des méthodes de localisation des capteurs en présence ou non de mesures aberrantes. Nous avons présenté en outre une méthode de gestion de mobilité permettant d'optimiser le tracking d'une cible. Dans les méthodes proposées, nous avons démontré que l'analyse par intervalles constitue un outil très efficace réalisant une estimation garantie des états recherchés. Ce chapitre commence par un récapitulatif des divers thèmes et contributions de ce mémoire. Nous discuterons par la suite quelques perspectives à court terme de nos travaux de recherche.

VII.1 Résumé et contributions

Dans ce mémoire, nous avons abordé les problèmes d'auto-localisation et de suivi de cibles dans les réseaux de capteurs mobiles. Vu l'absence d'une infrastructure fixe dans ces réseaux, les capteurs ont la possibilité de bouger d'une manière contrôlée ou passive. Dans le cas de la mobilité passive, les capteurs ne sont pas alertés de leurs changements de positions et doivent donc être re-localisés régulièrement. L'une des problématiques traitées dans ce mémoire concerne le problème d'auto-localisation des capteurs mobiles ayant des réserves énergétiques limitées. En revanche, dans le cas de la mobilité contrôlée, les capteurs ont la possibilité de se déplacer en réponse à une commande externe ou interne. Il est alors intéressant de gérer leur mobilité pour optimiser les performances du réseau par rapport à des objectifs de traitement de données. L'application que nous avons abordée dans ce mémoire est le suivi de cibles. Nous avons alors proposé une stratégie de repositionnement des capteurs pour optimiser l'estimation de la trajectoire d'une cible traversant le réseau. L'estimation des positions des capteurs ou de la cible se situe dans le cadre de la théorie des intervalles. Cette approche à erreurs bornées permet une modélisation simplifiée de l'incertitude. On résume brièvement ci-dessous les contributions de ce mémoire :

- ◇ Nous avons traité en premier lieu le problème d'auto-localisation avec des mesures de connectivités. Deux types de capteurs sont alors distingués : les ancrs, munies de GPS et donc connaissant leurs positions, et les non-ancres ou nœuds, les capteurs à localiser. Nous avons proposé en premier une méthode basée sur les mesures de proximité vis-à-vis des ancrs. L'estimation des positions est effectuée à l'aide de la théorie des intervalles, où les positions sont représentées par des pavés bidimensionnels. Nous avons également présenté diverses extensions de cette méthode : la méthode *back-propagated*, réalisant une rétropropagation pour corriger les pavés précédents, la méthode *multi-hop*, tenant compte des ancrs voisines ou même plus éloignées et une méthode profitant aussi bien des ancrs que des non-ancres. Toutes les études ont été réalisées dans des environnements parfaits, où les mesures réalisées sont supposées correctes.
- ◇ Nous avons abordé ensuite le problème d'auto-localisation également dans des environnements parfaits. Une méthode sans modèle utilisant les ancrs est alors proposée. Basée sur la comparaison des puissances des signaux échangés, cette méthode est plus robuste que celle basée sur les connectivités. Les positions estimées sont également des pavés bidimensionnels, mais généralement plus précises que celles obtenues avec l'approche précédente. Nous avons proposé en outre une extension de cette méthode, utilisant à la fois les informations échangées avec les ancrs et les non-ancres.
- ◇ Nous avons considéré ensuite le problème d'auto-localisation dans des environnements imparfaits. L'hypothèse sur laquelle nous nous sommes basés est la connaissance des fiabilités des mesures réalisées. Le problème revient alors à combiner toutes les informations collectées à l'aide de la théorie des fonctions de croyance. Les positions déterminées dans cette méthode sont représentées par des ensembles de pavés bidimensionnels, ayant chacun un poids spécifique.

- ◊ Nous avons enfin abordé la problématique de suivi de cibles. Nous nous sommes limités au cas d'une seule cible. L'étude, se situant dans le cadre de la mobilité contrôlée, consiste à gérer le déplacement des capteurs afin d'optimiser les performances du réseau. Nous avons alors proposé une stratégie qui consiste à estimer la position de la cible, prédire sa nouvelle position et ensuite déplacer les capteurs afin de couvrir d'une manière optimale la position prédite. Alors que l'estimation et la prédiction utilisent la théorie des intervalles, la phase de repositionnement emploie une technique de triangulation et une technique d'optimisation par colonies de fourmis permettant de minimiser l'énergie consommée.

Notons que les performances des diverses méthodes proposées ont été démontrées à l'aide de simulations réalisées avec Matlab.

VII.2 Travaux futurs et suggestions

L'un des points que nous avons traités dans ce mémoire est l'estimation des positions des capteurs dans des environnements imparfaits. La méthode que nous avons proposée suppose connaître les fiabilités accordées aux mesures. Nous nous proposons d'étudier le problème sous une hypothèse alternative. Nous supposons alors que le nombre maximal des mesures erronées est connu. Dans la suite, nous commençons par présenter les idées clés de cette approche.

VII.2.1 Auto-localisation avec un maximum de q mesures aberrantes

L'un des problèmes traités dans la littérature par Jaulin *et al.* [Jaulin *et al.*, 1996], et ultérieurement par Leger et Kieffer [Leger et Kieffer, 2010] est l'estimation d'états en présence de mesures aberrantes. Sous l'hypothèse d'avoir au maximum q mesures aberrantes, les méthodes proposées utilisent l'algorithme SIVIA [Jaulin *et al.*, 2001] pour résoudre le problème. Les solutions sont alors définies sous forme de sous-pavages couvrant avec précision toutes les solutions réelles du problème.

L'un de nos travaux futurs sera inspiré de cette approche. En effet, partant de la même hypothèse, nous envisageons l'application de l'algorithme SIVIA au problème d'auto-localisation en présence d'imperfections. Nous comptons en outre comparer cette technique à une méthode alternative que nous sommes en train d'évaluer. Celle-ci sera basée sur des outils combinatoires. En se basant sur le modèle de mobilité et les mesures de connectivités vis-à-vis des ancres présentés dans le chapitre III, il s'agit de satisfaire les équations de ces modèles sachant qu'au maximum, q mesures sont aberrantes. Soit $\mathbf{x}_j(t) = (x_{j,1}(t), x_{j,2}(t))$ le vecteur coordonnées du nœud j à l'instant t . Le problème sera alors modélisé par les équations suivantes :

$$\left\{ \begin{array}{l} (x_{j,1}(t) - x_{j,1}(t-1))^2 + (x_{j,2}(t) - x_{j,2}(t-1))^2 = (\Delta t \cdot v_j)^2, \\ v_j \in [0, v_{max,j}], \\ (x_{j,1}(t) - a_{i,1}(t))^2 + (x_{j,2}(t) - a_{i,2}(t))^2 \leq r^2, \quad i \in I_j(t), \\ \text{où au maximum } q \text{ mesures sur les } |I_j(t)| \text{ sont aberrantes.} \end{array} \right. \quad (\text{VII.1})$$

VII. CONCLUSION ET PERSPECTIVES

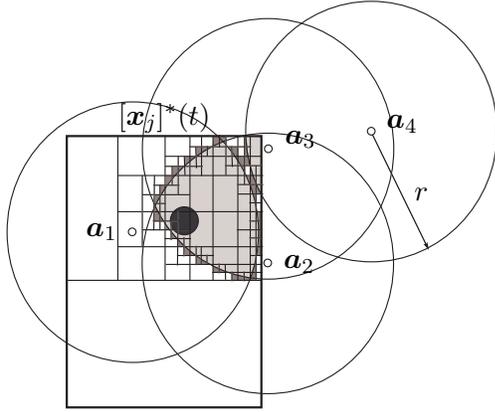


FIG. VII.1: Illustration de la méthode basée sur SIVIA.

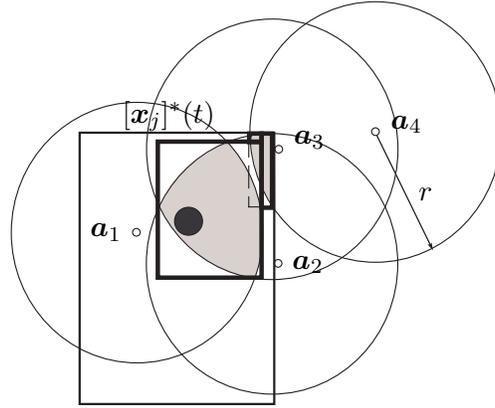


FIG. VII.2: Illustration de la méthode combinatoire.

$I_j(t)$ est l'ensemble des indices des ancrs détectées par le nœud j à l'instant t , $\mathbf{a}_i(t) = (a_{i,1}(t), a_{i,2}(t))$ est le vecteur coordonnées de l'ancre i à l'instant t , r est la portée de communication, $v_{max,j}$ est la vitesse maximale du nœud j , v_j est une variable vitesse et Δt est la durée comprise entre deux pas de temps consécutifs. La solution du problème couvrira alors tous les points satisfaisant à la fois le modèle de mobilité et au minimum $|I_j(t)| - q$ contraintes d'observation.

Une première méthode permettant de résoudre ce problème revient à appliquer l'algorithme SIVIA, présenté dans le chapitre II. La méthode consistera alors en une succession de bisections et de sélections jusqu'à l'obtention d'un sous-pavage final. Ce dernier sera issu de l'union de pavés bidimensionnels inclus entièrement dans la zone solution ou trop petits mais ayant une intersection non vide avec cette zone. En pratique, la méthode consistera en une propagation du sous-pavage précédent à l'aide du modèle de mobilité. L'enveloppe convexe du résultat, notée $[\mathbf{x}_j]^*(t)$, formera le pavé initial aux bisections, dit pavé parent. $[\mathbf{x}_j]^*(t)$ sera ensuite découpé à l'aide de l'algorithme SIVIA de telle sorte à garder les sous-pavés satisfaisant au moins $|I_j(t)| - q$ équations d'observation. Une illustration du sous-pavage résultant (en gris) avec $q = 1$ et $|I_j(t)| = 4$ est donnée dans FIG. VII.1.

La méthode alternative que nous proposerons est basée sur des outils combinatoires. Vu que la solution satisfait au moins $|I_j(t)| - q$ observations, on doit construire l'union de toutes les zones résultant de l'intersection de $(|I_j(t)| - q)$ mesures choisies parmi les $|I_j(t)|$. La méthode consistera alors à définir toutes les combinaisons de $|I_j(t)| - q$ contraintes sur les $|I_j(t)|$ observations relatives aux ancrs détectées. Pour chacune des combinaisons, toutes les contraintes répertoriées seront utilisées pour contracter un domaine initial à l'aide de l'algorithme de Waltz [Waltz, 1975; Jaulin *et al.*, 2001]. Ce domaine initial $[\mathbf{x}_j]^*(t)$ sera issu de la propagation de la solution précédente à l'aide du modèle de mobilité. La solution sera constituée de l'union des pavés obtenus suite aux contractions relatives aux diverses combinaisons. Une illustration de la solution du même

problème que pour FIG. VII.1 avec le même pavé initial est donnée dans FIG. VII.2. La complexité de cette méthode dépendra des valeurs de $|I_j(t)|$ et de q . Comparée à la méthode basée sur SIVIA, elle permettra de réduire la consommation de mémoire nécessaire pour la sauvegarde des solutions au détriment de la précision d'estimation. Notons qu'il sera possible par ailleurs de combiner l'approche de q mesures aberrantes avec l'approche des mesures de fiabilités. Les pavés dont l'union forme la solution seront alors accordés chacun un poids spécifique résultant de la combinaison de toutes les informations de fiabilités éventuellement disponibles.

VII.2.2 Autres perspectives

Les travaux que nous avons réalisés au cours de cette thèse permettent d'ouvrir également bien d'autres perspectives. Nous en citons :

- ◇ Au niveau de la localisation des capteurs :
 - Nous envisageons de développer une méthode pour le calcul des fiabilités des mesures en utilisant une technique de redondance des observations dans le réseau.
 - Nous comptons en outre proposer une méthode permettant d'estimer le nombre de mesures aberrantes collectées dans le réseau.
 - L'approche de robustesse pourra aussi bien être étendue au problème d'auto-localisation sans-modèle proposée dans le chapitre IV.
 - Nous pourrons également appliquer l'approche variationnelle à la localisation des capteurs mobiles, et la comparer par la suite aux approches ensemblistes.
- ◇ Au niveau du suivi de cibles :
 - Nous envisageons d'étendre la méthode que nous avons proposée pour couvrir le cas de cibles multiples. Le défi qu'il faudra relever est de pouvoir détecter la présence de plusieurs cibles et de les distinguer l'une de l'autre (problème d'association de données).
 - Nous pourrons par ailleurs améliorer la fonction objective que nous avons utilisée dans la méthode proposée. Par exemple, au lieu de minimiser la distance totale, nous pourrons considérer la trajectoire optimale pour chaque capteur, tout en tenant compte de sa réserve énergétique.
 - Nous comptons en outre proposer une méthode permettant d'optimiser la définition des positions des capteurs mobiles en ajoutant des degrés de liberté que ce soit au niveau de la taille des triangles ou de leurs directions. Il sera même possible de tenter d'utiliser d'autres formes géométriques autres que la technique de triangulation.
 - Une stratégie alternative pour repositionner les capteurs consistera à définir une grille dans la zone de surveillance permettant ainsi de discrétiser l'ensemble de positions que nous pourrons accorder aux capteurs. Une autre grille sera aussi utile au niveau du pavé prédit, permettant ainsi d'avoir un ensemble de points à couvrir, dits points de couverture. Le problème reviendra ainsi à minimiser le déplacement des capteurs et à maximiser la précision de l'estimation des points de couverture.
 - Il est en outre possible de développer ces méthodes d'une manière distribuée au niveau des capteurs.
- ◇ Au niveau des simulations :

VII. CONCLUSION ET PERSPECTIVES

- Nous envisageons de développer des modèles de mobilité plus réalistes, permettant de reproduire les mouvements des capteurs dans la nature (dans un champ de vent, dans un cours d'eau, etc). Ces modèles seront utiles au niveau des simulations, afin de tester, dans des conditions réalistes, les méthodes dédiées aux applications environnementales à titre d'exemple.
- ◇ Au niveau applicatif :
 - Il sera possible dans la suite d'employer nos outils dans les applications de sécurité routière où les réseaux de capteurs sont destinés à surveiller la circulation, l'état des conducteurs, les structures des automobiles, etc.
 - Les réseaux de capteurs peuvent être appliqués également dans un système d'alarme pour détecter les intrusions.
 - Une autre application très intéressante des réseaux de capteurs concerne la domotique et la surveillance des personnes à domicile. Nous comptons aussi nous intéresser à cette application.

Références

- [Ahmed *et al.*, 2003] A. A. Ahmed, H. Shi et Y. Shang. A survey on network protocols for wireless sensor networks. In *International Conference on Information Technology : Research and Education*, 2003. [3](#)
- [Ahmed et Eskicioglu, 2004] S. Ahmed et R. M. Eskicioglu. Current researches on sensor networks. Rapport interne, Telecommunication Research Labs, Winnipeg, Manitoba, Canada., 2004. [3](#)
- [Akyildiz *et al.*, 2002a] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam et E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40 : 102–114, 2002. [2](#), [5](#)
- [Akyildiz *et al.*, 2002b] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam et E. Cayirci. Wireless sensor networks : a survey. *Computer Networks*, 38 : 393–422, 2002. [15](#), [52](#), [146](#)
- [Al-Karaki et Kamal, 2004] J. N. Al-Karaki et A. E. Kamal. Routing techniques in wireless sensor networks : A survey. *IEEE Wireless Communications*, 11 : 6–28, 2004. [3](#)
- [Ali *et al.*, 2008] M. Ali, A. Böhm et M. Jonsson. Wireless sensor networks for surveillance applications - a comparative survey of mac protocols. In *The 4th International Conference on Wireless and Mobile Communications*, 2008. [6](#)
- [Anderson et Moore, 1979] B. D. O. Anderson et J. B. Moore, éditeurs. *Optimal Filtering*. Prentice-Hall, Elwood Cliffs, NJ, 1979. [11](#)
- [Anderson, 1995] J. G. T. Anderson. Pilot survey of mid-coast maine seabird colonies : an evaluation of techniques. *Report to the State of Maine Dept. of Inland Fisheries and Wildlife*, 1995. [16](#)
- [Arampatzis *et al.*, 2005] T. Arampatzis, J. Lygeros et S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Proceedings of the 13th Mediterranean Conference on Control and Automation*, 2005. [6](#), [116](#)
- [Baggio et Langendoen, 2006] A. Baggio et K. Langendoen. Monte-carlo localization for mobile wireless sensor networks. Hong Kong, China, 2006. [21](#), [22](#), [52](#), [78](#), [81](#), [108](#), [116](#), [140](#)
- [Bandyopadhyay et Coyle, 2003] S. Bandyopadhyay et E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2003. [5](#)

-
- [Basagni *et al.*, 2008] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli et Z. M. Wang. Controlled sink mobility for prolonging wireless sensor networks lifetime. *Wirel. Netw.*, 14 (6) : 831–858, 2008. 22
- [Benhamou *et al.*, 1999] F. Benhamou, F. Goualard, L. Granvilliers et J. Puget. Revising hull and box consistency. In *Proceedings of the International Conference on Logic Programming*, pages 230–244, Las Cruces, NM, 1999. 47
- [Bharathidasan et Ponduru, 2002] A. Bharathidasan et V. A. S. Ponduru. Sensor networks : An overview. Rapport interne, Department of Computer Science, University of California, Davis, CA 95616, 2002. 3
- [Boudec et Vojnovic, 2005] J.-Y. L. Boudec et M. Vojnovic. The Random Trip Model, Part I : Stability. Rapport interne, 2005. 17
- [Brunelli *et al.*, 2008] D. Brunelli, M. Maggiorotti, L. Benini et F. L. Bellifemine. Analysis of audio streaming capability of zigbee networks. In *Proceedings of the 5th European Conference on Wireless Sensor Networks*, 2008. 4
- [Bulusu *et al.*, 2000] N. Bulusu, J. Heidemann et D. Estrin. Gps-less low-cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7 (5) : 28–34, 2000. 21, 52
- [Camp *et al.*, 2002] T. Camp, J. Boleng et V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing : Special issue on Mobile Ad Hoc Networking : Research, Trends and Applications*, 2 : 483–502, 2002. 17, 91, 122, 167
- [Capkun *et al.*, 2001] S. Capkun, M. Hamdi et J.-P. Hubaux. Gps-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, volume 9, 2001. 21, 52
- [CBC, 2004] CBC. Dirty bomb sensors for ottawa airport, 2004. <http://www.cbc.ca/canada/story/2004/12/08/dirtybomb-airport041208.html>. 17
- [Chehade *et al.*, 2009] H. Chehade, L. Amodeo et F. Yalaoui. A new efficient hybrid method for selecting machines and sizing buffers in assembly lines. *Journal of Operations and Logistics*, 2009. 158
- [Chen, 1993] G. Chen, éditeur. *Approximate Kalman filtering*. World Scientific Publishing, 1993. 11
- [Chen *et al.*, 2004] W.-P. Chen, J. C. Hou et L. Sha. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 2004. 146
- [Cheng *et al.*, 2004] X. Cheng, X. H. Huang et D. Du, éditeurs. *Ad hoc wireless networking*. Kluwer Academic Publishers, 2004. 9
- [chit Chin *et al.*, 2010] J. chit Chin, Y. Dong, W.-K. Hon, C. Y. T. Ma et D. K. Y. Yau. Detection of intelligent mobile target in a mobile sensor network. *IEEE/ACM Transactions on Networking*, 18 (1) : 41–52, February 2010. 22
- [Cleary, 1987] J. C. Cleary. Logical arithmetic. *Future Computing Systems*, 2 (2) : 125–149, 1987. 41, 47

-
- [Conner *et al.*, 2004] S. W. Conner, J. Heidemann, L. Krishnamurthy, X. Wang et M. Yarvis. Workplace applications of sensor networks. In N. Bulusu et S. Jha, éditeurs, *Wireless Sensor Networks : A Systems Perspective*. Artech House Publishers, 2004. [6](#)
- [Culler *et al.*, 2004] D. Culler, D. Estrin et M. Srivastava. Guest editors' introduction : Overview of sensor networks. *Computer*, 37 (8) : 41–49, 2004. [2](#)
- [Czubak et Wojtanowski, 2009] A. Czubak et J. Wojtanowski. On applications of wireless sensor networks. In S. B. . Heidelberg, éditeur, *Internet - Technical Development and Applications*, pages 91–99, 2009. [2](#), [116](#)
- [Dantu *et al.*, 2005] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal et G. S. Sukhatme. Robomote : Enabling mobility in sensor networks. In *In IEEE/ACM Fourth International Conference on Information Processing in Sensor Networks (IPSN-SPOTS*, pages 404–409, 2005. [22](#)
- [Davis, 1987] E. Davis. Constraint propagation with interval labels. *Artificial Intelligence*, 32 (3) : 281–331, 1987. [47](#)
- [Demirkol *et al.*, 2006] I. Demirkol, C. Ersoy et F. Alagöz. MAC protocols for wireless sensor networks : A survey. *IEEE Communications Magazine*, 44 : 115–121, 2006. [3](#)
- [Dempster, 1967] A. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38 : 325–339, 1967. [116](#), [117](#)
- [Dencœux, 1997] T. Dencœux. Analysis of evidence-theoretic decision rules for pattern classification. *Pattern Recognition*, 30 : 1095–1107, 1997. [117](#), [121](#)
- [Djurić *et al.*, 2008] P. M. Djurić, M. Vemula et M. F. Bugallo. Target tracking by particle filtering in binary sensor networks. *IEEE Transactions on Signal Processing*, 2008. [146](#)
- [Doherty *et al.*, 2001] L. Doherty, K. S. J. Pister et L. E. Ghaoui. Convex position estimation in wireless sensor networks. Anchorage, AK, 2001. [21](#)
- [Donati *et al.*, 2008] A. Donati, R. Montemanni, N. Casagrande, A. Rizzoli et L. Gambardella. Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research*, 2008. [158](#)
- [Dorigo et Gambardella, 1997] M. Dorigo et L. Gambardella. Ant colonies for the traveling salesman problem. *Biosystems*, 1997. [147](#), [157](#), [158](#)
- [Doucet, 1998] A. Doucet. On sequential Monte Carlo methods for Bayesian filtering. Rapport interne, Department of Engineering, University of Cambridge, UK, 1998. [12](#), [21](#)
- [Doucet *et al.*, 2001] A. Doucet, N. de Freitas et N. Gordon, éditeurs. *Sequential Monte Carlo Methods in Practice*. Springer, 2001. [52](#)
- [Doucet *et al.*, 2000] A. Doucet, S. Godsill et C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10 (3) : 197–208, 2000. [12](#), [21](#), [108](#), [140](#)
- [Doucet *et al.*, 2002] A. Doucet, B.-N. Vo, C. Andrieu et M. Davy. Particle filtering for multi-target tracking and sensor management. In *Proceedings of the Fifth International Conference on Information Fusion*, 2002. [11](#)

-
- [Dréo *et al.*, 2005] J. Dréo, A. Pérowski, P. Siarry et E. Taillard. *Métaheuristiques pour l'optimisation difficile*. Eyrolles, 2005. 147, 157
- [El-Rabbany, 2006] A. El-Rabbany. *Introduction to GPS : The Global Positioning System, 2nd edition*. Artech House Publishers, 2006. 9, 52
- [Elson *et al.*, 2002] J. Elson, L. Girod et D. Estrin. Fine-grained network time synchronization using reference broadcasts. Rapport interne, University of California, Los Angeles, May 2002. 10
- [Elson et Romer, 2002] J. Elson et K. Romer. Wireless sensor networks : A new regime for time synchronization. In *Proceedings of the First Workshop on Hot Topics In Networks (HotNets-I)*, Princeton, NJ, USA, October 2002. 10
- [Estrin *et al.*, 2001] D. Estrin, L. Girod, G. Pottie et M. Srivastava. Instrumenting the world with wireless sensor networks. In *International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 2033–2036, 2001. 9
- [Fleury et Simplot-Ryl, 2009] E. Fleury et D. Simplot-Ryl. *Réseaux de capteurs : Théorie et modélisation*. 2009. 5
- [Gharavi et Kumar, 2003] H. Gharavi et S. Kumar. Special issue on sensor networks and applications. *Proceedings of the IEEE*, 91 (8), 2003. 6, 116, 146
- [Hande *et al.*, 2006] A. Hande, T. Polk, W. Walker et D. Bhatia. Self-powered wireless sensor networks for remote patient monitoring in hospitals. *Sensors*, 6 (9) : 1102–1117, 2006. 16, 146
- [Hani *et al.*, 2007] Y. Hani, L. Amodeo, F. Yalaoui et H. Chen. Ant colony optimization for solving an industrial layout problem. *European Journal of Operational Research*, 2007. 158
- [Haroud *et al.*, 1995] D. Haroud, S. Boulanger, E. M. Gelle et I. F. C. Smith. Management of conflicts for preliminary engineering design tasks. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 9 (4) : 313–323, 1995. 41
- [Hasu et Koivo, 2005] V. Hasu et H. Koivo. Decentralized kalman filter in wireless sensor networks, case studies. In *Proceedings of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering*, 2005. 11
- [Hickey *et al.*, 2007] T. Hickey, Q. Ju et M. H. V. Emden. Interval arithmetic : From principles to implementation. *Journal of the ACM*, 48 (5) : 1038–1068, 2007. 28
- [Hofmann-Wellenhof *et al.*, 2004] B. Hofmann-Wellenhof, H. Lichtenegger et J. Collins. *Global Positioning System : Theory and Practice*. Springer, September 2004. 9, 21, 23, 52, 88, 122
- [Hong *et al.*, 1999] X. Hong, M. Gerla, G. Pei et C.-C. Chiang. A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'99)*, 1999. 17
- [Hu et Evans, 2004] L. Hu et D. Evans. Localization for mobile sensor networks. Philadelphia, USA, 2004. 21, 22, 52, 116
- [Jaakkola, 2000] T. S. Jaakkola. Tutorial on variational approximation methods. In *In Advanced Mean Field Methods : Theory and Practice*, pages 129–159. MIT Press, 2000. 13

-
- [Jaakkola et Jordan, 1999] T. S. Jaakkola et M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10 : 25–37, 1999. [13](#)
- [Jardosh et Ranjan, 2008] S. Jardosh et P. Ranjan. A survey : topology control for wireless sensor networks. In *IEEE International Conference on Signal processing, Communications and Networking*, 2008. [5](#)
- [Jaulin, 1994] L. Jaulin. *Solution globale et garantie de problèmes ensemblistes ; application à l'estimation non linéaire et à la commande robuste*. Thèse de doctorat, Université Paris-Sud, Orsay, France, 1994. [36](#), [37](#)
- [Jaulin et al., 2001] L. Jaulin, M. Kieffer, O. Didrit et E. Walter. *Applied interval analysis*. Springer, 2001. [23](#), [28](#), [29](#), [31](#), [33](#), [34](#), [35](#), [36](#), [44](#), [88](#), [118](#), [125](#), [131](#), [147](#), [148](#), [173](#), [174](#)
- [Jaulin et Walter, 1993a] L. Jaulin et E. Walter. Guaranteed nonlinear parameter estimation from bounded-error data via interval analysis. *Mathematics and Computers in Simulation*, 35 (2) : 123–137, 1993. [38](#)
- [Jaulin et Walter, 1993b] L. Jaulin et E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29 (4) : 1053–1064, 1993. [38](#)
- [Jaulin et al., 1996] L. Jaulin, E. Walter et O. Didrit. Guaranteed robust nonlinear parameter bounding. In *Proceedings of CESA '96 IMACS Multiconference*, pages 1156–1161, Lille, France, 1996. [173](#)
- [Juang et al., 2002] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. shiuan Peh et D. Rubenstein. Energy-efficient computing for wildlife tracking : Design tradeoffs and early experiences with zebnet. In *ASPLPS'02*, 2002. [15](#), [146](#)
- [Kalman, 1960] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineerings*, 82 : 35–45, 1960. [11](#)
- [Kansal et al., 2004] A. Kansal, M. Rahimi, W. J. Kaiser, M. B. Srivastava, G. J. Pottie et D. Estrin. Controlled mobility for sustainable wireless networks. In *IEEE Sensor and Ad Hoc Communications and Networks (SECON)*, pages 1–6, Santa Clara, CA, October 2004. [22](#)
- [Karl et Willig, 2003] H. Karl et A. Willig. A short survey of wireless sensor networks. Rapport interne, october 2003. [2](#)
- [Kearfott et Kreinovich, 1996] R. B. Kearfott et V. Kreinovich. *Applications of interval computations*. Kluwer academic publishers, 1996. [29](#), [31](#)
- [Kieffer, 1999] M. Kieffer. *Estimation ensembliste par analyse par intervalles, application à la localisation d'un véhicule*. Thèse de doctorat, Université Paris-Sud, Orsay, France, 1999. [36](#)
- [Klukas et Fattouche, 1998] R. Klukas et M. Fattouche. Line-of-sight angle of arrival estimation in the outdoor multipath environment. *IEEE Transactions on Vehicular Technology*, 47, February 1998. [10](#)
- [Kotecha et Djurić, 2003a] J. H. Kotecha et P. M. Djurić. Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51 : 2592–2601, 2003. [12](#)

-
- [Kotecha et Djurić, 2003b] J. H. Kotecha et P. M. Djurić. Gaussian sum particle filtering. *IEEE Transactions on Signal Processing*, 51 : 2602 – 2612, 2003. [12](#), [13](#)
- [Labiod, 2006] H. Labiod. *Réseaux mobiles ad hoc et réseaux de capteurs sans fil*. Hermes Science Publications, 2006. [14](#), [52](#)
- [Lambrou et al., 2009] T. P. Lambrou, C. G. Panayiotou, S. Felici et B. Beferull. Exploiting mobility for efficient coverage in sparse wireless sensor networks. In S. Netherlands, éditeur, *Wireless Personal Communications*, April 2009. [22](#)
- [Langendoen et Reijers, 2003] K. Langendoen et N. Reijers. Distributed localization in wireless sensor networks : a quantitative comparison. *Computer Networks : The International Journal of Computer and Telecommunications Networking*, 43 : 499–518, 2003. [10](#)
- [Leger et Kieffer, 2010] J. Leger et M. Kieffer. Guaranteed robust distributed estimation in a network of sensors. *ICASSP*, 2010. [173](#)
- [Levis, 2006] P. Levis. Tinyos programming. Rapport interne, 2006. [3](#)
- [Lhomme, 1993] O. Lhomme. Consistency techniques for numeric csp. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 232–238, Chambéry, France, 1993. [41](#), [47](#)
- [Li, 2006] X. Li. Rss-based location estimation with unknown pathloss model. *IEEE transactions on wireless communications*, 5, 2006. [88](#)
- [Liu et al., 2004] C. Liu, K. Wu et T. He. Sensor localization with ring overlapping based on comparison of received signal strength indicator. *IEEE international conference on mobile ad hoc and sensor systems*, pages 516–518, 2004. [88](#)
- [MacKay, 2003] D. J. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. [13](#)
- [Mainwaring et al., 2002] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler et J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA'02*, Atlanta, Georgia, USA, September 2002. [16](#), [146](#)
- [Medeisis et Kajackas, 2000] A. Medeisis et A. Kajackas. On the use of the universal okumura-hata propagation prediction model in rural areas. In *Vehicular Technology Conference Proceedings*, volume 3, 2000. [55](#), [88](#), [123](#)
- [Moore et al., 2004] D. Moore, J. Leonard, D. Rus et S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the second ACM conference on embedded networked sensor systems (SenSys'04)*, pages 50–61, November 2004. [21](#), [52](#)
- [Moore, 1966] R. E. Moore. *Interval analysis*. Prentice-Hall, 1966. [23](#), [28](#), [88](#)
- [Moore, 1979] R. E. Moore. *Methods and applications of interval analysis*. Siam, Philadelphia, USA, 1979. [28](#), [29](#), [31](#), [88](#), [118](#), [125](#), [147](#)
- [Moore et al., 2009] R. E. Moore, R. B. Kearfott et M. J. Cloud. *Introduction to interval analysis*. Siam, Philadelphia, USA, 2009. [28](#)

-
- [Moses *et al.*, 2003] R. L. Moses, D. Krishnamurthy et R. Patterson. A self-localization method for wireless sensor networks. *EURASIP Journal on Applied Signal Processing*, 4 : 348–358, 2003. [10](#)
- [Mostafaei *et al.*, 2009] R. Mostafaei, A. Habiboghli et M. R. Meybodi. Target tracking in sensor networks : A distributed constraint satisfaction approach. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 55, pages 97–101, Oslo, Norway, July 2009. [10](#)
- [Nabae et Olfat, 2008] M. Nabae et A. Olfat. Enhanced rss based location estimation with disk overlapping in wireless sensor networks. *IEEE international conference on telecommunications (ICT'08)*, 2008. [88](#)
- [Nadir *et al.*, 2008] Z. Nadir, N. Elfadhil et F. Touati. Pathloss determination using okumura-hata model and spline interpolation for missing data for oman. In *Proceedings of the World Congress on Engineering*, volume 1, London, UK, July 2008. [55](#), [88](#), [123](#)
- [Nawaz et Jha, 2007] S. Nawaz et S. Jha. Collaborative localization for wireless sensor networks. In *IEEE PIMRC*, Athens, Greece, September 2007. [10](#)
- [Nguyen et Rattenbury, 2003] X. Nguyen et T. Rattenbury. Localization algorithms for sensor networks using rf signal strength. Rapport interne, University of California at Berkeley, May 2003. [10](#)
- [Pages-Zamora *et al.*, 2002] A. Pages-Zamora, J. Vidal et D. Brooks. Closed-form solution for positioning based on angle of arrival measurements. In *The 13th IEEE Int. Symposium Personal, Indoor and Mobile Radio Communications*, September 2002. [10](#)
- [Pan *et al.*, 2003] J. Pan, Y. T. Hou, L. Cai, Y. Shi et S. X. Shen. Topology control for wireless sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 286–299, 2003. [5](#), [8](#)
- [Peng et Sichitiu, 2006] R. Peng et M. L. Sichitiu. Angle of arrival localization for wireless sensor networks. *IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, 1 : 374–382, September 2006. [10](#)
- [Petit-Renaud et Dencœux, 2004] S. Petit-Renaud et T. Dencœux. Nonparametric regression analysis of uncertain and imprecise data using belief functions. *International Journal of Approximate Reasoning*, 35 : 1–28, 2004. [118](#)
- [Rajaraman, 2002] R. Rajaraman. Topology control and routing in ad hoc networks : a survey. In *ACM SIGACT News*, pages 60–73, 2002. [8](#)
- [Ramachandra, 2000] K. V. Ramachandra. *Kalman Filtering Techniques for Radar Tracking*. Marcel Dekker, 2000. [11](#)
- [Rappaport *et al.*, 1996] T. Rappaport, J. Reed et B. Woerner. Position location using wireless communications on highways of the future. *IEEE Communication Magazine*, October 1996. [10](#)
- [Ribeiro *et al.*, 2006] A. Ribeiro, G. B. Giannakis et S. I. Roumeliotis. SOI-KF : Distributed Kalman filtering with low-cost communications using the sign of innovations. *IEEE Transactions on Signal Processing*, 54 : 4782–4795, 2006. [12](#)

-
- [Ribeiro, 2004] M. I. Ribeiro. Kalman and extended kalman filters : Concept, derivation and properties. Rapport interne, Institute for Systems and Robotics, Instituto Superior Técnico, February 2004. [11](#)
- [Ristic *et al.*, 2004] B. Ristic, S. Arulampalam et N. Gordon. *Beyond the Kalman filter : particle filters for tracking applications*. Artech House Publishers, 2004. [11](#)
- [Ristic et Smets, 2005] B. Ristic et P. Smets. Target identification using belief functions and implication rules. *IEEE Transactions on Aerospace and Electronic Systems*, 41 : 1097–1103, 2005. [117](#)
- [Sam-Haroud, 1995] D. Sam-Haroud. *Constraint consistency techniques for continuous domains*. Thèse de doctorat, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1995. [41](#)
- [Sam-Haroud et Faltings, 1996] D. Sam-Haroud et B. Faltings. Consistency techniques for continuous constraints. *Constraints*, 1 (1-2) : 85–118, 1996. [37](#), [41](#)
- [Savvides *et al.*, 2001] A. Savvides, C. chieh Han et M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. of the 7th ACM MobiCom*, pages 166–179, Rome, Italy, 2001. [21](#), [52](#)
- [Shafer, 1976] G. Shafer. A mathematical theory of evidence. *Princeton University Press*, 1976. [116](#), [117](#)
- [Sheng *et al.*, 2005] X. Sheng, Y. Hu et P. Ramanathan. Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 24, Piscataway, NJ, USA, 2005. IEEE Press. [12](#), [13](#)
- [Shorey, 2006] R. Shorey, éditeur. *Mobile, Wireless, and Sensor Networks : Technology, Applications, and Future Directions*. John Wiley & Sons, 2006. [14](#), [52](#), [146](#)
- [Smets, 1993] P. Smets. Belief functions : The disjunctive rule of combination and the generalized bayesian theorem. *International Journal of Approximate Reasoning*, 9 : 1–35, 1993. [120](#)
- [Smets et Kennes, 1994] P. Smets et R. Kennes. The transferable belief model. *Artificial Intelligence*, 66 : 191–234, 1994. [116](#), [117](#), [120](#), [121](#)
- [Snoussi et Richard, 2006] H. Snoussi et C. Richard. Ensemble learning online filtering in wireless sensor networks. In *Proceedings of the 10th IEEE International Conference on Communications Systems*, 2006. [13](#), [14](#)
- [Sohraby *et al.*, 2007] K. Sohraby, D. Minoli et T. Znati. *Wireless sensor networks : technology, protocols, and applications*. John Wiley & Sons, 2007. [2](#)
- [Song *et al.*, 2008] G. Song, Y. Zhou, F. Ding et A. Song. A mobile sensor network system for monitoring of unfriendly environments. *Sensors*, 8 : 7259–7274, November 2008. [22](#)
- [Spanos *et al.*, 2005] D. P. Spanos, R. Olfati-Saber et R. M. Murray. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 2005. [12](#)
- [Teng *et al.*, 2007a] J. Teng, H. Snoussi et C. Richard. Binary variational filtering for target tracking in sensor networks. In *IEEE Workshop on Statistical Signal Processing*, 2007. [14](#)

-
- [Teng *et al.*, 2007b] J. Teng, H. Snoussi et C. Richard. Prediction-based proactive cluster target tracking protocol for binary sensor networks. In *Proceedings of the 7th IEEE International Symposium on Signal Processing and Information Technology*, 2007. 14
- [Teng *et al.*, 2008] J. Teng, H. Snoussi et C. Richard. Variational filtering algorithm for independent target tracking and sensor localization in wireless sensor network. Lausanne, August 2008. 14
- [Teng *et al.*, 2009] J. Teng, H. Snoussi et C. Richard. Decentralized variational filtering for simultaneous sensor localization and target tracking in binary sensor networks. In *ICASSP'09*, Taipei, Taiwan, April 2009. 14, 146
- [Teng *et al.*, 2010] J. Teng, H. Snoussi et C. Richard. Decentralized variational filtering for target tracking in binary sensor networks. *IEEE Transactions on Mobile Computing*, 2010. 146
- [Tirta *et al.*, 2004] Y. Tirta, B. Lau, N. Malhotra, S. Bagchi, Z. Li et Y. H. Lu. Controlled mobility for efficient data gathering in sensor networks with passively mobile nodes. *IEEE Transactions on Mobile Computing Special Issue on Sensor Networks*, 2004. 22
- [Tran et Yang, 2006] S. P. M. Tran et T. A. Yang. Evaluations of target tracking in wireless sensor networks. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 97–101, Houston, Texas, USA, 2006. 10
- [Tzikas *et al.*, 2008] D. G. Tzikas, A. C. Likas et N. P. Galatsanos. The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25 : 131–146, 2008. 13
- [Vermaak *et al.*, 2003] J. Vermaak, N. D. Lawrence et P. Pérez. Variational inference for visual tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003. 13
- [Vieira *et al.*, 2003] M. A. M. Vieira, J. Coelho, Claudionor N., D. C. da Silva et J. M. da Mata. Survey on wireless sensor network devices. In *IEEE Conference on Emerging Technologies and Factory Automation*, 2003. 3
- [Vivekanandan et Wong, 2007] V. Vivekanandan et V. Wong. Concentric anchor beacon localization algorithm for wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 56 : 2733–2744, 2007. 21, 52
- [Waltz, 1972] D. L. Waltz. Generating semantic descriptions from drawings of scenes with shadows. Rapport interne, Massachusetts Institute of Technology, Cambridge, MA, USA, 1972. 48
- [Waltz, 1975] D. L. Waltz. Generating semantic descriptions from drawings of scenes with shadows. In P. H. Winston, éditeur, *The Psychology of Computer Vision*, pages 19–91, McGraw-Hill, New York, NY, 1975. 47, 48, 148, 174
- [Wang et Xiao, 2007] C. Wang et L. Xiao. Sensor localization under limited measurement capabilities. *IEEE Network*, 21 (3) : 16–23, 2007. 9
- [Wang *et al.*, 2009] X. Wang, J. Ma, S. Wang et D. Bi. Distributed energy optimization for target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 2009. 146

-
- [Ward *et al.*, 1997] A. Ward, A. Jones et A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4 : 42–47, 1997. [10](#)
- [Wu, 2006] J. Wu, éditeur. *Handbook on theoretical and algorithmic aspect of sensor, ad hoc wireless, and peer-to-peer networks*. Auerbach Publications, 2006. [9](#)
- [Yager, 2004] R. Yager. Cumulative distribution functions from dempster-shafer belief structures. *IEEE Transactions on Systems Man and Cybernetics B*, 34 : 2080–2087, 2004. [118](#)
- [Yang et Sikdar, 2003] H. Yang et B. Sikdar. A protocol for tracking mobile targets using sensor networks. In *The 1st IEEE International Workshop on Sensor Network Protocols and Applications*, 2003. [10](#)
- [Yi *et al.*, 2007] J. Yi, S. Yang et H. Cha. Multi-hop-based monte carlo localization for mobile sensor networks. San Diego, CA, USA, 2007. [21](#), [22](#), [52](#), [116](#)
- [Zanca *et al.*, 2008] G. Zanca, F. Zorzi, A. Zanella et M. Zorzi. Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks. In *REALWSN*, Glasgow, United Kingdom, April 2008. [9](#)
- [Zeinalipour-Yazti et Chrysanthis, 2009] D. Zeinalipour-Yazti et P. K. Chrysanthis. Mobile sensor network data management. In *Encyclopedia of Database Systems*, pages 1755–1759, 2009. [14](#)
- [Zhang *et al.*, 1998] W. Zhang, J. Luo et N. Mandayam. Mobile location estimation using time of arrivals in cdma systems. Rapport interne, Rutgers University, 1998. [21](#), [52](#)
- [Zou et Chakrabarty, 2007] Y. Zou et K. Chakrabarty. Distributed mobility management for target tracking in mobile sensor networks. *IEEE Transactions on Mobile Computing*, 2007. [22](#), [146](#)