



HAL
open science

Localisation de robots mobiles en coopération mutuelle par observation d'état distribué

Khaoula Lassoued

► **To cite this version:**

Khaoula Lassoued. Localisation de robots mobiles en coopération mutuelle par observation d'état distribué. Autre. Université de Technologie de Compiègne, 2016. Français. NNT : 2016COMP2289 . tel-01468454

HAL Id: tel-01468454

<https://theses.hal.science/tel-01468454>

Submitted on 15 Feb 2017

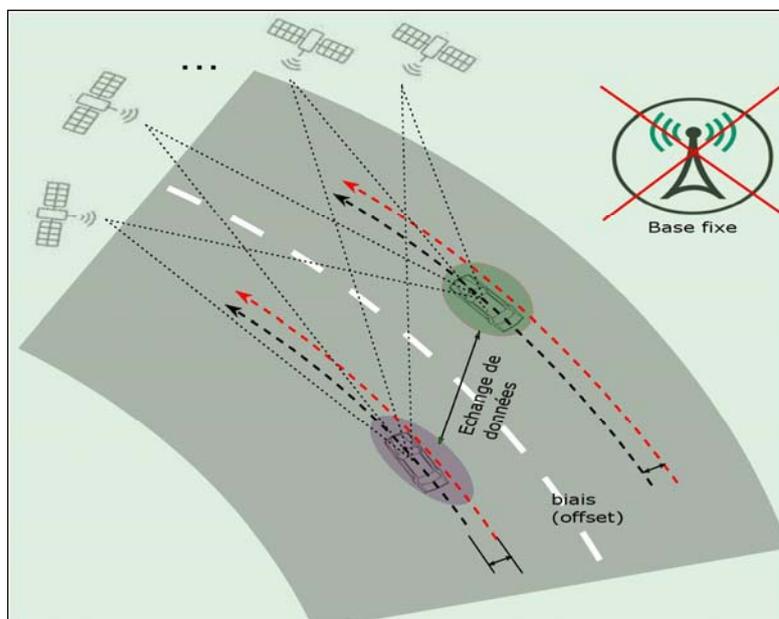
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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

Par Khaoula LASSOUED

Localisation de robots mobiles en coopération mutuelle par observation d'état distribuée

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 11 juillet 2016
Spécialité : Technologies de l'Information et des Systèmes

D2289

Localisation de robots mobiles en coopération mutuelle par observation d'état distribuée

Khaoula Lassoued

Spécialité : Technologies de l'Information et des Systèmes

Soutenance soutenue le 11 juillet 2016 devant le jury composé de :

M. Luc Jaulin	<i>rapporteur</i>
M. Maan El Badaoui El Najjar	<i>rapporteur</i>
M. Philippe Bonnifait	<i>directeur de thèse</i>
Mme. Isabelle Fantoni	<i>directrice de thèse</i>
M. Ali Charara	<i>examineur</i>
M. Dominique Meizel	<i>examineur</i>



Remerciements

Tout d'abord, je tiens à exprimer mes remerciements et ma gratitude à mes directeurs de thèse, Philippe BONNIFAIT et Isabelle FANTONI, pour leur soutien, présence et leurs conseils précieux au cours de ces trois années.

C'était un vrai plaisir de mener cette recherche sous leur supervision tout en bénéficiant de leur connaissance scientifique étendue. J'espère que j'aurai la possibilité d'échanger et de travailler d'avantage avec eux.

Je remercie Ali CHARARA, directeur de l'unité mixte de recherche CNRS 7253 *Heu-DiaSyc*, de m'avoir accueillie dans son laboratoire, au sein de l'équipe *Automatique, Systèmes Embarqués, Robotique*.

Mes remerciements vont également à Luc JAULIN et Maan EL BADAOUI EL NAJJAR, pour le temps passé à étudier ce travail en tant que rapporteurs, ainsi qu'aux membres du jury Ali CHARARA et Dominique MEIZEL pour la grande pertinence de leurs questions et l'intérêt de la discussion que j'ai pu avoir avec eux.

Je tiens à exprimer ma profonde gratitude à mon encadrant de Master, Tonino SOPHY, je lui dois mon intérêt pour la recherche et la volonté de poursuivre un doctorat.

Je remercie tous les amis et collègues du laboratoire, qui ont été compréhensifs, patients, soucieux et attentionnés durant cette thèse. Je suis reconnaissante à Gérald DHERBOMEZ et Thierry MONGLON pour m'avoir aidée et accompagnée lors des essais expérimentaux.

Une pensée particulière va à Nadia, Carine, Ahmed, Dhoha, Hajer, Asma, Bacet, Mohamed, Hoda, Hafida, Zui, Vincent, Nicolas, Enrico, Samuel, Vanessa qui m'ont encouragée et apporté la joie et le réconfort durant les moments difficiles.

Merci à mes parents pour leur soutien et leur confiance.

Table des matières

1	Introduction générale	1
1.1	Systèmes multi-robots	1
1.2	Contexte du travail	3
1.3	Localisation coopérative pour les systèmes multi-robots	6
1.3.1	Estimation coopérative centralisée	6
1.3.2	Estimation coopérative distribuée	8
1.3.3	Méthodes ensemblistes pour l'estimation	10
1.4	Technologies considérées pour le positionnement	10
1.4.1	Systèmes de positionnement par balises fixes	12
1.4.1.1	Définition de l'UWB et son évolution historique	12
1.4.1.2	Avantages de l'UWB	13
1.4.1.3	Localisation avec mesure UWB	14
1.4.2	Système Global de Navigation par Satellite	15
1.4.2.1	Concepts de base de GPS	17
1.4.2.2	Observables GPS	19
1.4.2.3	Équation d'observation associée au code	21
1.4.2.4	GPS Différentiel	24
1.4.2.5	GPS RTK	25
1.5	Problématique et organisation du manuscrit	26
1.5.1	Problématique	26
1.5.2	Organisation du manuscrit	26
1.6	Conclusion	27
2	Étude de l'observabilité de quelques systèmes robotiques coopératifs	29
2.1	Introduction	29
2.2	Problème de la localisation coopérative avec des mesures de distance sur balises	30
2.3	Problème de la localisation coopérative avec des mesures GNSS	35
2.3.1	Cas statique	35
2.3.1.1	Robots dans un voisinage	37
2.3.1.2	Robots distants	42
2.3.2	Cas dynamique	46
2.3.2.1	Observabilité des systèmes non-linéaires	46
2.3.2.2	Dérivée de Lie et observabilité non-linéaire au sens du rang	50

2.3.2.3	Observabilité de la localisation coopérative avec des mesures GNSS	51
2.4	Conclusion	59
3	Localisation coopérative avec une approche ensembliste	61
3.1	Introduction	61
3.2	Formulation du problème de coopération avec des mesures de distance	62
3.2.1	Nature des données	62
3.2.1.1	Mesures proprioceptives (DR)	63
3.2.1.2	Mesures extéroceptives	63
3.2.2	Modélisation des systèmes	63
3.2.2.1	Modèle d'observation	64
3.2.2.2	Modèle d'évolution	64
3.3	Techniques ensemblistes pour la localisation	65
3.4	Propagation de contraintes	66
3.4.1	Problèmes de Satisfacation de Contraintes (CSP)	66
3.4.2	Contracteur par propagation-rétropropagation	68
3.4.3	Inversion ensembliste par intervalles (SIVIA)	71
3.4.3.1	Sous pavage	71
3.4.3.2	SIVIA	73
3.4.3.3	SIVIA avec contracteur	75
3.5	Localisation coopérative avec des mesures des distances	76
3.5.1	Simulation	76
3.5.1.1	Données partagées	79
3.5.1.2	Description des scénarios	79
3.5.2	Algorithme coopératif	80
3.5.3	Résultats	85
3.5.4	Amélioration de l'algorithme coopératif	92
3.5.4.1	Amélioration avec SIVIAP et augmentation du nombre de robots	93
3.5.4.2	Comparaison des algorithmes C_p , $C_{Sviap2R}$ et $C_{Sviap10R}$	94
3.6	Localisation coopérative avec des mesures de pseudodistances	96
3.6.1	Modèles	97
3.6.2	Formulation du problème	98
3.6.3	Algorithme de localisation coopérative	100
3.6.4	Résultats expérimentaux	101
3.7	Conclusion	108
4	Comparaison avec une approche Bayésienne dans le cas des mesures de pseudodistance	109
4.1	Introduction	109
4.2	Méthodes Bayésiennes	110
4.3	Algorithmes de fusion distribuée	110
4.3.1	Combinaison Convexe Simple	111

4.3.2	Intersection de Covariance	112
4.4	Algorithme de localisation coopérative	114
4.5	Résultats expérimentaux de méthode Bayésienne	120
4.5.1	Résultats de la fusion avec SCC	123
4.5.2	Résultats de la fusion avec CI	131
4.5.3	Comparaison entre les fusions SCC et CI	136
4.6	Comparaison entre la méthode Bayésienne CI et l'approche ensembliste	146
4.6.1	Travaux sur la comparaison des deux approches	147
4.6.2	Critères utilisés pour la comparaison	148
4.6.3	Comparaison des méthodes C-SIVIAP et C-EKF-CI	149
4.6.3.1	Exactitude et incertitude	149
4.6.3.2	Consistance	157
4.7	Conclusion	158
5	Conclusion et perspectives	161
5.1	Conclusion	161
5.2	Contributions	163
5.3	Perspectives	164
	Annexes	169
A	Éléments d'analyse par intervalles	171
A.1	Définition des intervalles	171
A.2	Arithmétique sur les intervalles	172
A.3	Vecteurs et Matrices d'intervalles	174
A.3.1	Vecteurs intervalles	174
A.3.2	Matrices intervalles	175
A.4	Fonctions d'inclusion	175
A.4.1	Propriétés spécifiques de fonctions d'inclusion	176
A.4.2	Fonctions d'inclusion naturelles	177
B	Éléments de filtrage Bayésien	179
B.1	Le filtrage optimal	179
B.2	Le filtre de Kalman	181
B.3	Le filtre de Kalman étendu EKF	183
	Bibliographie	187
	Liste des symboles	197

Table des figures

1.1	Véhicules terrestres et aérien du laboratoire Heudiasyc issus du projet Robotex.	5
1.2	Localisation avec 3 balises par le principe de trilatération. La croix rouge matérialise la position inconnue.	15
1.3	Constellation des satellites du système GPS [1].	18
2.1	Erreur de distance due à une erreur sur la position de la balise.	31
2.2	Système coopératif de 2 robots avec 4 mesures de distances. Repère fixe (R_0) et repères mobiles (R_1 et R_2).	32
2.3	Deux états indiscernables (\mathbf{x}_0 et \mathbf{x}_1) qui produise la même sortie	47
2.4	Illustration de l'observabilité : les sorties sont différentes	47
2.5	Illustration de l'observabilité locale	48
2.6	Illustration de l'observabilité faible	49
2.7	Illustration de l'observabilité faible locale : deux états initiaux très proches engendrent des sorties différentes	50
2.8	Système coopératif de 2 robots avec 4 mesures de GNSS. Repère fixe (R_0) et repères mobiles (R_1 et R_2).	52
3.1	Dérive de l'estimation	63
3.2	Localisation ensembliste avec 3 balises sous hypothèse d'erreurs bornées	66
3.3	Différentes représentations ensemblistes de la solution du problème de localisation à trois balises [2].	67
3.4	Décomposition en contraintes primitives de la contrainte associée à la mesure de distance d'une balise B^j [2].	70
3.5	Encadrement d'un ensemble entre deux sous-pavages	72
3.6	Test d'inclusion pour l'inversion ensembliste.	74
3.7	Modèle bicyclette du véhicule. Le véhicule est représenté en gris clair, et l'approximation en bicyclette est en gris foncé. vR et oR sont les repères mobile et fixe. v est la vitesse en direction de x et γ est l'angle de braquage.	78
3.8	Architecture de la commande de véhicule (v et γ sont les entrées de la commande).	78
3.9	Scénario de localisation coopérative (en réalité la distance entre les véhicules et les balises est grande par rapport à la zone d'évolution).	79
3.10	Scénario de la localisation coopérative.	80
3.11	Architecture de localisation d'un véhicule.	81

3.12	Contraction de la position du véhicule 2 estimée dans le véhicule 1 et la position reçue du véhicule 2 estimée dans le véhicule 2.	85
3.13	Trajectoires des 2 véhicules.	86
3.14	Bornes de positions de R_1	87
3.15	Bornes de positions de R_2	87
3.16	Bornes des erreurs de positions de R_1	88
3.17	Bornes des erreurs de positions de R_2	88
3.18	Erreurs HPE absolue de R_1	89
3.19	Erreurs HPE absolue de R_2	89
3.20	Distance relative de véhicules.	90
3.21	Volume de domaine de confiance de R_1	90
3.22	Volume de domaine de confiance de R_2	91
3.23	Les 10 premiers biais de R_1 (unités des axes x et y sont secondes (s) et mètre (m)).	92
3.24	Boîtes estimées de position des 10 véhicules.	94
3.25	Erreurs HPE absolues de R_1 des méthodes coopératives.	95
3.26	Biais de R_1 sur la mesure de la balise 2. Comparaison des méthodes coopératives.	95
3.27	Les véhicules expérimentaux et les données partagées.	97
3.28	Borne supérieure et inférieure relative (Δ_{min} , Δ_{max}). C'est un calcul simplifié de la distance relative.	102
3.29	Trajectoires et boîtes estimées de positions de véhicules.	102
3.30	Bornes des erreurs de positions de R_1	103
3.31	Bornes des erreurs de positions de R_2	103
3.32	Erreurs de distance relative RDE de deux véhicules.	104
3.33	CDF des CDS de véhicules en considérant la mesure de $\text{cap}\psi$	105
3.34	Estimations et bornes de la correction de pseudodistances (les unités sont les échantillons et les mètres).	106
3.35	Bornes d'erreurs de positions de R_2 en utilisant les mesures de la vitesse angulaire.	108
4.1	Illustration de l'Intersection de Covariance (CI).	113
4.2	Scénario de la coopération	115
4.3	Étapes principales du processus de l'estimation coopérative	115
4.4	Trajectoires de véhicules estimées par la méthode coopérative	122
4.5	CDS (i.e, $k\sigma_{HPE}$) et HPE d'une estimation absolue.	123
4.6	CDS (i.e, $k\sigma_{RDE}$) et RDE d'une estimation relative.	124
4.7	Erreurs HPE absolues et domaines de confiance de R_1 . Fusion SCC.	124
4.8	Erreurs HPE absolues et domaines de confiance de R_2 . Fusion SCC.	125
4.9	Erreurs de distance relative RDE et leurs domaines de confiance. Fusion SCC.	125
4.10	Distribution cumulative des erreurs HPE de R_1 . Fusion SCC.	126
4.11	Distribution cumulative des erreurs HPE de R_2 . Fusion SCC.	127
4.12	Distribution cumulative de CDS de R_1 . Fusion SCC.	127

4.13	Distribution cumulative de CDS de R_2 . Fusion SCC.	128
4.14	Estimation des biais avec la fusion SCC (les unités sont secondes en abscisses et mètres en ordonnées).	129
4.15	Nombre de satellites utilisés par les véhicules durant l'essai.	130
4.16	Nombre de satellites en vue durant l'essai.	131
4.17	Variation des SNR des satellites 4,11,22 pendant l'essai pour le véhicule R_1	132
4.18	Décalage d'horloge estimé (\times la vitesse de la lumière).	132
4.19	Erreurs HPE absolues et domaines de confiance de R_1 . Fusion CI. . .	133
4.20	Erreurs HPE absolues et domaines de confiance de R_2 . Fusion CI. . .	134
4.21	Erreurs relatives RDE et leurs domaines de confiance. Fusion CI. . .	134
4.22	Distribution cumulative des erreurs HPE de R_1 . Fusion CI.	135
4.23	Distribution cumulative des erreurs HPE de R_2 . Fusion CI.	135
4.24	Distribution cumulative de CDS de R_1 . Fusion CI.	136
4.25	Distribution cumulative de CDS de R_2 . Fusion CI.	137
4.26	Estimation des biais avec la fusion CI (les unités sont secondes et mètres).	137
4.27	Erreurs HPE absolues et domaines de confiance de R_1 avec les deux méthodes de fusion.	138
4.28	Erreurs HPE absolues et domaines de confiance de R_2 avec les deux méthodes de fusion.	139
4.29	Erreurs relatives RDE et leurs domaines de confiance relatifs avec les deux méthodes de fusion.	139
4.30	Distribution cumulative des erreurs absolues HPE de R_1 avec les deux méthodes de fusion.	140
4.31	Distribution cumulative des erreurs absolues HPE de R_2 avec les deux méthodes de fusion.	140
4.32	Distribution cumulative des CDS de R_1	142
4.33	Distribution cumulative des CDS de R_2	142
4.34	Diagramme de Stanford simplifié : Consistance des filtres en utilisant les 2 fusions pour R_1 et R_2 . (a) C-EKF-SCC pour R_1 , (b) C-EKF-CI pour R_1 , (c) C-EKF-SCC pour R_2 , (d) C-EKF-CI pour R_2	143
4.35	Illustration du problème de sur-convergence dû à la consanguinité des données dans la fusion.	144
4.36	Fusion dans R_1	144
4.37	Trajectoires des véhicules. (a) Références et enveloppes de position en utilisant la méthode C-SIVIAP, (b) Trajectoires estimées en utilisant la méthode EKF. La vitesse moyenne des véhicules était d'environ 30 <i>Kmph</i>	147
4.38	HPE et $k\sigma_{HPE}$ de la méthode C-SIVIAP	150
4.39	HPE et CDS de la position absolue de R_1 par les méthodes C-EKF-CI et C-SIVIAP.	151
4.40	Trajectoire de référence de R_1 . Les zones encadrées présentent des sauts de référence.	152

Table des figures

4.41	HPE et CDS de la position absolue estimée en 2D de R_2	152
4.42	Statistiques des HPE des véhicules R_1 et R_2	153
4.43	Accumulation des HPE pour les 2 véhicules	154
4.44	CDS des véhicules R_1 et R_2	155
4.45	Distribution Cumulative des domaines de confiance des HPE pour les 2 véhicules	156
4.46	Diagramme de Stanford simplifié. Consistance de filtre ensembliste de chaque véhicule.	158
A.1	Représentation d'une boîte $[1, 2] \times [1, 3]$	174
A.2	Images d'une boîte par des fonction.	176

Chapitre 1

Introduction générale

Sommaire

1.1	Systèmes multi-robots	1
1.2	Contexte du travail	3
1.3	Localisation coopérative pour les systèmes multi-robots	6
1.4	Technologies considérées pour le positionnement	10
1.5	Problématique et organisation du manuscrit	26
1.6	Conclusion	27

1.1 Systèmes multi-robots

La recherche dans le domaine des systèmes multi-robots (i.e. Multi-Robot Systems MRS) a débuté dans les années 90, dans des travaux regroupant des robots mobiles rassembleurs d'objets [3] et des colonies de robots marcheurs [4, 5]. Dudek et al. ont présenté une taxonomie qui classe les systèmes multi-agents (Multi-Agent Systems MAS) selon la communication et la capacité de calcul [6]. Ils ont démontré que la performance d'un ensemble multi-robots est meilleure que celle d'un seul robot dans certains cas. Cao et al. ont fourni un extrait des travaux dans la littérature sur la robotique mobile coopérative jusqu'à la moitié des années 90 [7]. Ils classent ces travaux selon cinq catégories : l'architecture du groupe, le conflit de ressources, l'origine de la coopération, l'apprentissage et la géométrie. Ils mentionnent notamment les limites liées à la technologie et les lacunes des travaux existants. Arai et al. [8] ont défini sept thèmes de recherche primaires dans les systèmes MRS : 1) les systèmes d'inspirations biologiques dont la plupart des MRS font suite aux travaux des chercheurs qui se sont intéressés à la modélisation des sociétés d'insectes ou d'animaux et ont reproduit avec succès leurs comportements. 2) Les systèmes qui étudient la communication : ces systèmes traitent le problème de communication soit entre robots ou entre un utilisateur et les robots. 3) Les systèmes qui s'intéressent aux architectures, l'allocation de tâches et le contrôle : les problèmes de ces systèmes sont communs aux systèmes décentralisés : allocation et planification de tâches, structure du système de communication, homogénéité ou hétérogénéité des

robots. 4) Les systèmes orientés vers la localisation, la cartographie, l'exploration. 5) Les systèmes dont le but est le transport et la manipulation d'objet : ces systèmes proposent une méthode de planification du mouvement d'une équipe de robots pour le transport collectif d'un objet. 6) Les systèmes de coordination de mouvements : ces systèmes étudient la planification des trajectoires des robots, la génération et le maintien de la formation ainsi que le contrôle de trafic. 7) Les systèmes dont l'objet est la conception de robots reconfigurables. Farinelli et al. ont présenté des travaux plus récents réalisés jusqu'au début des années 2000 qui portent sur l'aspect de la coopération et la coordination des MRS [9]. Par ailleurs, ils proposent une taxonomie pour la classification des MRS basée sur la notion de coordination, dont la complexité se caractérise selon deux dimensions : les dimensions de l'espace de coordination et les dimensions de l'espace des configurations du système utilisé.

Un système multi robots peut effectuer des tâches difficiles ou impossibles à accomplir par un seul robot. Une équipe de robots fournit une certaine redondance qui contribue à l'accomplissement d'une tâche de manière collaborative. L'utilisation d'un système multi-robots offre donc plusieurs avantages par rapport à un système mono-robot. Les motivations les plus communes pour le développement de solutions de systèmes multi-robots selon Parker [10] sont :

- La robustesse : les performances de contrôle dans un MRS ne sont pas trop affectées en cas de défaillance d'un robot. Les systèmes multi-robots introduisent une notion de robustesse résultante de la fusion de données et du partage d'informations entre les robots. Cette notion de robustesse est synonyme de tolérance aux pannes résultante de la redondance de l'information. Plusieurs robots peuvent se localiser plus efficacement s'ils échangent des renseignements sur leur position lors de l'exécution d'une mission.
- La rapidité : les tâches sont effectuées en parallèle pour obtenir un niveau de performance élevé. A titre d'exemple : l'exploration parallèle d'un environnement inconnu par un ensemble de robots mobiles, dans le but de faire une cartographie de l'environnement. Ces deux tâches peuvent être réalisées par un seul robot, mais l'ajout d'autres robots permet d'accélérer l'exécution des tâches en question.
- Le faible coût : les systèmes MRS présentent dans certains cas des coûts inférieurs, dans le sens où la construction, la mise au point, le déploiement et la maintenance d'un seul système complexe peut devenir plus difficile que pour un ensemble de robots simples.

La coopération est l'interaction entre des robots qui travaillent dans un intérêt commun [11]. Cette coopération devient utile quand le résultat de l'exécution de la tâche est considéré meilleur avec plusieurs robots. Parker [12] a introduit le concept de « couverture » comme un indicateur du niveau de coopération requis pour l'accomplissement d'une mission : si la couverture est grande, alors les tâches peuvent être accomplies sans coopération et si la couverture est faible, alors une coopération est nécessaire. Une des questions centrales qu'on doit traiter dans un système multi-robots, quel que soit le domaine d'application, est comment faire coopérer efficacement les robots pendant une mission, soit automatiquement, soit éventuelle-

ment par l'intervention d'un opérateur externe. La coopération signifie que les robots doivent communiquer pour échanger des informations et coordonner leurs actions dans le but d'accomplir une mission commune globale [13]. La coopération est le point clé pour exploiter le potentiel des systèmes multirobots [14]. La communication est une condition préalable et indispensable pour n'importe quel algorithme de coopération. La présence d'un canal de communication sûr et de débit suffisant permet de mettre en place un mécanisme de coopération sophistiqué et efficace. Avec l'avancement des technologies de communication, les robots récents peuvent être équipés d'interfaces de communication sans fil haut-débit qui leur fournissent un moyen de communication assez sûr.

La plupart des premiers travaux sur les systèmes MRS ont porté sur les systèmes homogènes (SWARM) constitués de robots dotés de capacités individuelles identiques même si leur structure physique est différente. Des travaux significatifs utilisant des robots homogènes sont présentés dans [15, 16, 17, 18, 19, 20]. Cependant, ces dernières années la communauté robotique a orienté ses travaux vers les systèmes MRS hétérogènes. Il s'agit de systèmes constitués de robots dotés de capacités différentes notamment au niveau physique : caractéristiques mécaniques, mode de locomotion, capteurs, actionneurs. Quelques travaux pertinents utilisant des robots hétérogènes sont fournis dans [21, 22, 23, 24].

Les grands problèmes coopératifs s'expriment dans les travaux de localisation multi-robots [25, 23, 26, 27], d'exploration multi-robots [28, 29, 30, 16, 31, 32], de recherche et de sauvetage multi-robots [33, 34, 17, 35] et de transport multi-robots [36, 37, 38].

1.2 Contexte du travail

Cette recherche s'inscrit dans le cadre du projet ROBOTEX (N° ANR-10-EQPX-44-01). C'est un réseau national de plates-formes robotiques en France qui constitue un « équipement d'excellence (EQUIPEX) » financé dans le cadre des investissements d'avenir. Cet instrument scientifique offre aux chercheurs en robotique un environnement de travail répondant aux critères internationaux de qualité les plus exigeants.

L'objectif principal de ROBOTEX est la constitution d'une infrastructure cohérente d'équipements techniques à laquelle est associée une expertise scientifique et technologique .

ROBOTEX regroupe quinze laboratoires dont les domaines de recherche se classifient en cinq grandes catégories de la robotique :

1. Robotique humanoïde et interactions naturelles.
2. Robotique mobile terrestre et aérienne.
3. Robotique médicale.
4. Robotique de production.

5. Micro-nano robotique.

Le laboratoire est impliqué dans le domaine de la robotique mobile (terrestre et aérienne). L'EQUIPEX Robotex nous a permis d'acquérir des moyens expérimentaux de qualité en robotique mobile : trois véhicules électriques (deux Zoe et une Fluence) et une flotte de mini-drones (octorotors et quadrirotors), ainsi que le support expérimental associé : une arène de vol extérieure, un véhicule Master, une machine à usiner, une imprimante 3D, des capteurs haut de gamme, etc. Ces équipements ont permis de tester et de valider plusieurs travaux de recherche. Par exemple, les véhicules électriques Fluence et Zoe ont servi de support expérimental pour cette thèse. L'utilisation de mini-drones serait aussi possible avec des capteurs adaptés à nos besoins. Les robots qui nous intéressent dans cette recherche sont principalement les « véhicules » montrés sur la figure 1.1. Enfin, notre thème d'investigation est lié particulièrement à la localisation des systèmes multi-robots mobiles (Multiple Mobile Robot Systems MMRS).

Pour qu'un robot mobile navigue de manière autonome, il doit être capable de se localiser [39], i.e. connaître sa pose (position et orientation). La localisation a toujours été un problème pour les robots mobiles intérieurs (indoor) et extérieurs (outdoor). Différents types de capteurs [40] et de techniques ont été utilisés pour traiter ce problème (e.g. [41], [42], [43], [44]). L'idée de base derrière la plupart des systèmes de localisation actuels est de combiner des capteurs proprioceptifs qui mesurent le déplacement du véhicule avec l'information recueillie par les capteurs extéroceptifs qui interagissent avec l'environnement. De nombreuses applications robotiques exigent la coopération de robots afin d'effectuer une certaine tâche [45], [46]. La plupart des approches de localisation existantes se réfèrent au cas d'un seul robot. Même quand un groupe de N robots est considéré, le problème de localisation du groupe se transforme ainsi en un problème d'estimation de N poses d'une façon indépendante. Chaque robot estime sa position en fonction de son expérience individuelle (mesures de capteurs proprioceptifs et extéroceptifs). Les connaissances de différents agents de l'équipe ne sont pas combinées et chaque membre doit compter uniquement sur ses propres ressources (détection, perception et capacités de traitement). Cette approche est relativement simple, puisqu'on évite de traiter le problème complexe de la fusion des informations à partir d'un grand nombre de sources souvent interdépendantes. D'autre part, si on considère une stratégie de localisation plus coordonnée, on obtient un certain nombre d'avantages qui peuvent compenser la complexité ajoutée par la fusion des données. L'une des tâches de fusion de données les plus fondamentales est l'estimation de l'état dynamique de robots mobiles. Quelques solutions proposées dans le contexte d'estimation coopérative pour le positionnement de multi-robots seront citées dans la suite.



(a) Zoe Apache commandable par ordinateur permettant un contrôle coopératif avec le safety driver.



(b) Fluence Iris commandable par ordinateur.



(c) Octorotor.

Figure 1.1 – Véhicules terrestres et aérien du laboratoire Heudiasyc issus du projet Robotex.

1.3 Localisation coopérative pour les systèmes multi-robots

Le problème d'estimation coopérative a été considéré depuis plus que dix ans. En 1994, Kurazume et al. ont proposé un des premiers algorithmes de localisation multi-véhicules pour résoudre le problème d'accumulation des erreurs de mesures de capteurs proprioceptifs [47]. Ils ont proposé une solution de balises portables pour améliorer le positionnement mais aucune fusion des informations des robots du groupe n'a été réalisée. Dans ce cas, il ne s'agit pas d'une localisation coopérative multi-véhicules mais d'une localisation multi-véhicules. Depuis, de nombreuses approches de localisation coopérative de systèmes multi-robots ont été proposées en considérant qu'il est possible de combiner non seulement les informations propres à chaque robot, mais aussi les informations provenant des autres robots du groupe.

Le gain en précision d'un système de localisation coopératif par rapport à un système de localisation classique à un seul robot est dû au fait de l'utilisation des autres robots du groupe comme capteurs extéroceptifs. Les données provenant des autres membres fournissent des informations supplémentaires sur la pose du robot dans un repère absolu. L'information reçue peut aussi être considérée comme une localisation relative par rapport à un point de référence de position connue qui est, dans ce cas, le robot qui a envoyé cette information. Les approches de localisation coopérative proposées dans la littérature peuvent être classifiées selon leur architecture en deux grandes catégories : systèmes basés sur une architecture centralisée et systèmes basés sur une architecture distribuée.

Cette section introduit les deux principales techniques de localisation coopérative centralisée et distribuée. On cite les principaux avantages et inconvénients des approches décrites dans la littérature selon leur architecture. Cette présentation est inspirée de travaux de Karam [48]. Dans le dernier paragraphe, on présente l'architecture et la méthode présentée dans ce mémoire.

1.3.1 Estimation coopérative centralisée

De nombreuses variantes d'un système de positionnement coopératif sont issues essentiellement des systèmes centralisés, où le traitement de l'estimation d'état est effectué par une unité centrale qui communique avec tous les robots du groupe. Le principe de l'estimation centralisée est de contenir l'estimation de l'état de chaque robot dans un seul vecteur d'état du groupe et le met à jour en traitant les données de capteurs issues de chaque membre. Ces données peuvent être égocentriques telles que les déplacements des robots et la pose absolue ou bien des observations inter-robot comme la pose relative entre deux robots. L'un des progrès les plus notables dans le positionnement coopératif a été atteint par Roumeliotis et Bekey qui présentent un système coopératif pour le positionnement d'un groupe de robots mobiles par une approche centralisée [49]. Ils utilisent dans cette approche un Filtre de Kalman

Étendu pour mettre à jour l'état du groupe. Plusieurs approches coopératives ont fait l'extension de l'étude de Roumeliotis pour étudier par exemple les effets d'utilisation des informations relatives comme les mesures d'orientation et les distances relatives dans la mise à jour d'état du groupe [50] ou encore la propagation des incertitudes [51]. Les méthodes de fusion de données ne se limitent pas au filtre de Kalman. D'autres outils sont proposés tels que l'estimateur par maximum a posteriori [52] et l'estimation par maximum de vraisemblance [53]. Certaines applications utilisent d'autres types de capteurs, d'où l'apparition d'autres travaux proposant une mise à jour de l'état par des capteurs de vision ou d'onde acoustique. Une caméra qui mesure l'orientation relative a aussi été utilisée dans une méthode de localisation coopérative dans [54]. Des ondes acoustiques ont été utilisées pour mesurer la pose relative entre les robots [55].

Les approches centralisées présentent des avantages tels que la simplicité de la fusion centrale des données collectées de chaque robot permettant une mise à jour optimale de l'état du groupe. Les interdépendances sont gérées d'une manière naturelle du fait que les données sont fusionnées une seule fois dans l'unité centrale et donc cette approche ne présente aucun risque de réutilisation des mêmes informations dans la fusion. Cependant, les approches centralisées en pratique présentent des inconvénients liés aux contraintes de communication avec l'unité centrale. Les robots devront maintenir un contact permanent avec l'infrastructure, ils doivent être proches vu que la portée des systèmes de communication est souvent courte, ce qui limite la zone d'évolution des robots. Afin de surmonter ces inconvénients, l'architecture centralisée a évolué vers une architecture multi-centralisée. Pour élargir la zone d'évolution des robots, des agents mobiles collecteurs d'informations sont utilisées comme des intermédiaires entre les robots mobiles et l'infrastructure. Dans le contexte de localisation multi-centralisée, Worrall and Nebot ont utilisé des communications multibonds pour transmettre des informations sur l'état du véhicule en temps réel à d'autres véhicules [56]. Cependant, un problème d'agent isolé se produit lorsque l'environnement est vaste et faiblement peuplé. Une solution possible à ce problème est d'utiliser des agents mobiles supplémentaires qui naviguent à proximité de l'agent isolé pour recueillir les mises à jour de l'information de l'état et les amener au réseau principal [57]. Cependant, dans ces solutions proposées, le système centralisé dépend toujours de l'infrastructure et donc une défaillance de l'infrastructure compromet tout le système.

Compte tenu de l'évolution technologique et la disponibilité de plusieurs ordinateurs embarqués dans plusieurs types de robots, les chercheurs ont commencé à examiner la façon de s'affranchir du système central et de répartir l'estimation de la pose du groupe sur ses membres. Ce type d'architecture est appelé *architecture distribuée* (ou décentralisée).

1.3.2 Estimation coopérative distribuée

Dans un système de fusion distribuée, chaque robot traite ses mesures et communique les résultats avec les autres robots. L'architecture distribuée ne possède pas de serveur central communiquant avec les membres du groupe. Le principe de base de la localisation coopérative distribuée est le même que celui des approches fondées sur une architecture centralisée. Les robots du groupe sont considérés comme un système de systèmes qui utilise les données issues de chaque membre. Cependant, dans ce cas chaque robot possède une version de l'état du groupe qu'il met à jour avec ses propres informations et avec les données partagées avec ses partenaires. Ce type d'architecture a été introduit dans [25] par Roumeliotis et al.. L'auteur propose un système de localisation collective basé sur un filtre de Kalman étendu. Cette approche est basée sur la modification des équations du filtre de Kalman pour les répartir sur les différents membres du groupe. Cette méthode est en mesure de produire le même résultat qu'un estimateur centralisé. D'autres travaux s'inspirent de cette approche et proposent des solutions basées sur différents capteurs et différents outils mathématiques pour une fusion des informations entièrement distribuée.

Les approches distribuées peuvent être divisées en deux grandes catégories selon le type de données partagées entre les robots :

- Les approches basées sur l'échange des mesures de capteurs.
- Les approches basées sur l'échange d'états fusionnés ou de parties des états fusionnés.

Dans la suite, chaque type d'approche sera décrit en fournissant quelques exemples de la littérature.

Le principe des approches basées sur l'échange des mesures de capteurs est que chaque robot doit envoyer ses mesures à tous ses partenaires, afin d'avoir une version de mise à jour de l'état dans chaque robot.

Par exemple, Madhavan et al. ont présenté un système distribué de localisation coopérative des robots mobiles hétérogènes dans [23]. L'approche proposée a été appliquée sur une flotte de deux robots mobiles. La coopération est réalisée par le partage des informations sur la pose relative qui contiennent les distances entre les deux robots et l'orientation relative ainsi que les informations de la position absolue contenant des mesures GPS et proprioceptives. Chaque robot doit avoir une version de l'état du groupe grâce au partage des données d'où la nécessité d'une disponibilité permanente de la communication entre les robots. De surcroît, la quantité de données à échanger augmente selon le nombre de robots utilisés dans le groupe et les cadences de capteurs. Les estimations produites par chaque robot sont sous-optimales par rapport à un estimateur centralisé. Fox et al. [27] ont adopté une approche similaire à [23] dans le sens où chaque robot doit maintenir une estimation de sa propre pose. Ils ont utilisé un filtre à particules à la place d'un filtre de Kalman Étendu et une approche d'arbre de densité pour la fusion des données relatives partagées. Cette approche permet de combiner temporairement les fonctions de densité de probabilité

(i.e, les particules) provenant des deux robots pour effectuer un échantillonnage d'importance.

Dans les approches basées sur l'échange d'états fusionnés, chaque robot n'utilise que ses propres capteurs pour mettre à jour l'état du groupe et ne partage que son résultat final. Ce type d'approche permet de limiter la quantité de données échangées. Cependant, ces approches font face au problème de la consanguinité des données (en anglais *Data Incest*). Ce problème est notamment dû à la réutilisation de la même information dans la fusion. Les estimées obtenues dans ce cas sont biaisées et sur-confiantes. En effet, le problème de consanguinité des données engendre un problème de sur-convergence qui provient de l'ignorance des corrélations entre les estimations de robots. Ce problème sera illustré dans le chapitre 4.

Plusieurs solutions du problème de consanguinité des données ont été proposées. À titre d'exemple, Howard et al. ont utilisé une approche basée sur l'échange des distributions de poses mesurées, où chaque robot estime les positions des autres robots dans le groupe par rapport à sa propre position à travers les mesures de caméra et d'un télémètre laser [58]. Leur méthode de fusion des informations partagées est fondée sur un filtre particulaire. Après l'échange des données, chaque robot met à jour la distribution de sa pose avec celle qu'il a reçue du robot qui l'a détecté. Afin d'éviter le problème des mises à jour circulaires, l'auteur tient à jour un arbre de dépendances dans le but de ne mettre à jour que les descendants avec des données qui n'ont pas été fusionnées auparavant. D'autres solutions sont proposées par Roumeliotis et al. et Karam et al. [51, 59]. Dans [51], les robots ne partagent qu'une partie de leur état et, lors de la fusion, ils ne considèrent qu'une partie des informations collectées. Dans ce cas, l'état global fusionné contient des estimations sous-optimales. Dans [59], l'état global du système combine tous les états locaux collectés par tous les membres du groupe et l'état global n'est utilisé que par les robots en local mais jamais communiqué à nouveau aux autres membres. Ainsi, l'état global fusionné contient des estimations optimales des poses de tous les robots. Cependant, ces travaux ne traitent pas la consistance/intégrité des estimations. La méthode d'intersection de covariance représente une solution pour le problème de consanguinité des données et, en même temps, elle gère les dépendances entre les estimations en offrant des domaines de confiance assez fiables.

Les méthodes sus-cités sont toutes basées sur des approches Bayésiennes (filtre de Kalman Étendu) ou filtre particulaire. D'autres approches de fusion peuvent éviter le problème de sur-convergence tout en garantissant des résultats avec des domaines de confiance fiables, telles que les approches ensemblistes [60, 61, 62, 63]. Ce sera l'approche privilégiée dans cette recherche en utilisant des méthodes ensemblistes par analyse d'intervalles.

1.3.3 Méthodes ensemblistes pour l'estimation

L'approche ensembliste est basée sur l'hypothèse que les erreurs de modèle et de mesure sont bornées. Elle a été appliquée avec succès pour l'estimation des paramètres de modèle [64] et l'estimation des positions des robots lorsque des domaines de confiance fiables sont nécessaires [65]. Meizel et al. [66] ont développé une méthode d'inversion ensembliste en utilisant les techniques d'analyse par intervalles (SIVIA¹) basées sur des observateurs à erreurs bornées pour la localisation d'un seul robot. Cependant, SIVIA ne convient pas aux applications en temps réel lorsque le nombre de bisections est élevé. Une solution consiste à utiliser SIVIA tout en résolvant simultanément un problème de satisfaction de contraintes (CSP²) pour limiter le temps de calcul des bisections [67].

En ce qui concerne la localisation multi-robots en temps réel, plusieurs études récentes basées sur l'inversion ensembliste avec la technique des CSP ont été étudiées. Drevelle et al. [68] ont exploité un groupe de véhicules autonomes sous-marins (AUV³) pour explorer un grand espace. Dans leur application, ils ont utilisé des capteurs de distance pour mesurer l'inter-distance entre les robots. En outre, Bethencourt [69] a utilisé des algorithmes ensemblistes distribués pour accomplir une mission coopérative d'un groupe d'AUVs en utilisant des mesures inter-temporelles. Kyoung-Hwan et Jihong ont également étudié une méthode de localisation coopérative pour plusieurs robots terrestres basée sur des techniques de propagation de contraintes [70]. Dans ce travail, la fusion des données de capteurs proprioceptifs et de capteurs extéroceptifs a été utilisée.

En résumé, un travail considérable a été fait pour la simulation de systèmes coopératifs, mais très peu de résultats utilisant des données réelles ont été rapportés. Enfin, peu de travaux ont été fait pour évaluer les performances de l'intégrité des estimations à partir de capteurs de faible coût.

1.4 Technologies considérées pour le positionnement

La localisation est utilisée dans de nombreux domaines comme les marins qui ont besoin de connaître leur position absolue régulièrement. Aujourd'hui, de nombreuses applications grand public sont apparues. La liste des applications est très large et de nouveaux besoins apparaissent régulièrement ces dernières années. Dans le domaine de la robotique, la localisation permet de guider les robots dans des endroits difficilement accessibles ou bien contaminés pour exécuter différentes tâches de façon autonome. Dans les Systèmes de Transport Intelligent (ITS), la localisation est une tâche primordiale pour la navigation de véhicules intelligents. Qu'il s'agisse d'infor-

1. Set Inversion Via Interval Analysis
2. Constraint Satisfaction Problem
3. Autonomous Underwater Vehicle

mer l'utilisateur de sa position en vue de le guider ou de contrôler un véhicule de façon autonome, le véhicule doit pouvoir se localiser dans son environnement.

Il existe différentes technologies pour localiser et/ou positionner des objets mobiles en espace libre. Elles sont fondées soit sur des systèmes communicants qui mettent en œuvre un bouquet d'émetteurs (infrastructure terrestre ou balises) et un récepteur (GPS, UWB : *Ultra Wide Band*, WiFi, Bluetooth, RFID⁴, etc), soit sur des capteurs embarqués pour la navigation (MEMS⁵, caméra, etc) sans dépendance à une infrastructure ou soit sur l'association des systèmes communicants avec des capteurs autonomes.

Parmi les technologies les plus répandues basées sur une infrastructure terrestre, il existe des systèmes exploitant les techniques de radiolocalisation fondées sur les réseaux cellulaires ou les réseaux sans fil (UWB, WiFi...), où l'objet mobile est localisé à l'aide des signaux qu'il transmet au réseau (2G, 3G, WiFi, RFID...) ou reçus depuis des stations de base ou des points d'accès de ce réseau. Le principal désavantage de ces méthodes est que leur zone de couverture est limitée. La géolocalisation par des systèmes de navigation par satellites (GNSS : *Global Navigation Satellite System*) offre l'avantage d'une couverture globale avec une constellation de satellites en orbite (bien que certains systèmes, basés sur des satellites géostationnaires, n'ont qu'une couverture régionale). Depuis le premier système satellitaire global TRANSIT mis en service en 1967, d'autres systèmes ont été développés. Les systèmes NAVSTAR GPS, GALILEO, GLONASS et BEIDOU sont ainsi actuellement opérationnels ou en cours de déploiement. Ces systèmes fournissent un service de positionnement absolu avec une précision d'une dizaine de mètres, pour un coût raisonnable. Cependant, les techniques satellitaires ne sont pas efficaces dans les milieux urbains et les environnements couverts amenant des erreurs de plusieurs dizaines de mètres. Une méthode alternative est l'hybridation du positionnement GNSS avec d'autres sources d'information comme les mesures de vitesse et d'orientation (e.g. des capteurs inertiels), la localisation relative basée sur la détection de balises (e.g. UWB, radar) ou encore des éléments géométriques de la route (e.g. des cartes). Les capteurs de navigation à l'estime (en anglais DR comme *Dead Reckoning*) et les capteurs inertiels sont les technologies alternatives fréquemment utilisées pour déterminer la position, la vitesse et l'attitude. Ils sont très communs et les mesures peuvent être facilement récupérées, par exemple à partir d'un bus CAN à bord des voitures modernes.

Nous nous intéressons à une méthode hybride qui associe les systèmes de radionavigation terrestres ou satellitaires avec des capteurs de navigation à l'estime. Les technologies considérées dans cette thèse pour le positionnement des véhicules sont la fusion des mesures DR avec des mesures radio sur balises (UWB, 802.11) ou avec des mesures GNSS. Cette section présente ces deux technologies de radionavigation considérées pour le positionnement des véhicules. Les systèmes de positionnement

4. Radio Frequency IDentification

5. Micro Electro Mechanical System

par balises feront l'objet de la première section en prenant comme exemple les balises UWB. Ensuite, dans la deuxième section, nous mettrons l'accent sur les systèmes de positionnement par satellites.

1.4.1 Systèmes de positionnement par balises fixes

Plusieurs technologies du positionnement par ondes radio existent actuellement comme le WiFi (802.11), l'Ultra Wide Band (UWB ou ULB en français, IEEE 802.15.3a, IEEE 802.15.4a) ou encore le Bluetooth (IEEE 802.15). Les ondes radio possèdent la propriété de pouvoir traverser certains obstacles rencontrés habituellement dans les bâtiments, de sorte qu'une localisation peut être envisagée sans avoir accès, dans toutes les positions de la zone à couvrir, à une vision directe avec les émetteurs. Dans cette section, nous prenons l'exemple de l'UWB car il est largement utilisé pour la localisation des robots mobiles.

La recherche et le développement de la technologie radio UWB ont débuté dans les années 60. Ces recherches ont conduit à l'utilisation de l'UWB comme une mesure radio de courte distance et comme un système de communication, de géoradar ou encore un système précis de localisation [71]. L'UWB a été initialement proposé comme une technique de communication à courte portée caractérisée par une consommation faible de puissance, un haut débit de données et par une forte résistance aux multi-trajets. Cependant, la communication de l'UWB n'a pas été autant populaire que le positionnement par UWB. Un intérêt particulier est la haute résolution temporelle du signal UWB qui est l'origine de la haute précision du positionnement.

Cette section aborde brièvement la définition du signal UWB et son évolution historique [72]. Ensuite, ses avantages et son principe pour la localisation seront présentés.

1.4.1.1 Définition de l'UWB et son évolution historique

La technique radio UWB désigne une approche particulière dans le domaine des télécommunications sans fil. L'UWB est basée sur l'envoi d'impulsions ultra-courtes (typiquement $< 1ns$). Un signal UWB est défini par sa largeur de bande relative donnée par l'équation 1.1.

$$B_f = 2 \frac{f_H - f_L}{f_H + f_L} \quad (1.1)$$

où f_H et f_L représentent respectivement les fréquences de coupure supérieure et inférieure à - 3 dB. Un signal UWB possède un rapport largeur de bande sur fréquence centrale de 20% au minimum, ou une largeur de bande de 250 MHz ou plus.

La définition retenue aujourd'hui est celle de la FCC⁶, autorité américaine de réglementation de fréquences qui a étendu la définition pour inclure les signaux dont le rapport largeur de bande sur fréquence centrale est supérieure à 20% ou présentant une bande de fréquence supérieure à 500 MHz (et non pas 250 MHz) [73].

Les premières recherches dans le domaine de l'UWB ont débuté il y a plus d'une cinquantaine d'années et se sont focalisées sur les applications radar en raison de la nature large bande des signaux qui autorisent un fort pouvoir de résolution temporelle. Ultérieurement, Taylor a présenté les résultats de ses travaux sur les bases de la technologie UWB appliqués au radar [74]. En 1990, le département de la défense des États-Unis a publié les résultats de son aperçu sur la technologie UWB basée particulièrement sur les systèmes radar. A cette époque, aucune application de l'UWB aux systèmes de communication n'était envisagée. Les recherches dans le domaine de communication radio en utilisant les signaux UWB sont apparues avec les travaux de Sholtz [75]. Dans ces travaux, une attention particulière est donnée aux principales caractéristiques de la technologie UWB qui simplifie l'architecture des systèmes radio : une résolution temporelle de l'ordre de la nanoseconde due à la largeur de la bande de fréquences, un faible rapport cyclique permettant des modulations de type « saut temporel » et la gestion des usagers multiples ainsi qu'une transmission possible sans porteuse.

Avant la fin du 20^{ème} siècle, l'usage de l'UWB pour la communication n'avait pas été concrètement envisagé. Cependant, face au potentiel affiché par l'UWB, les industriels civils ont poussé les autorités de réglementation américaines à prendre des mesures normatives visant à permettre l'introduction de cette technologie dans le marché des applications civiles. La FCC a donc entamé une étude en 1998 qui a abouti à la loi « Part 15 » et à la publication du rapport « First Report and Order » le 14 Février 2002 [73]. Cette loi a autorisé l'émission sans licence dans la bande 3.1-10.6 GHz, sous contrainte de respecter une limitation de puissance égale à -41.3 dBm/MHz. Cette loi américaine a permis, dans un premier temps, d'initier des travaux de recherche partout dans le monde, notamment aux USA, afin d'introduire la technologie UWB sur le marché mondial, de façon harmonisée.

1.4.1.2 Avantages de l'UWB

Nous citons ci-après les principales caractéristiques qui ont attiré un intérêt croissant par la technologie UWB selon [76].

- Très haut débit : grâce à sa très grande largeur de bande, un système UWB peut atteindre un débit de 480 Mbps sur une portée inférieure à 10 m.
- Robustesse face aux multitrajets : un système UWB est capable de résoudre un nombre important de trajets multiples, et de combattre ainsi la sélectivité fréquentielle du canal de propagation, source de multiple trajets des signaux

6. Federal Communication Commission

et de dégradation de performances. Cette résolution des trajets multiples augmente cependant la complexité d'implémentation.

- Excellente résolution temporelle : un système UWB dispose d'une résolution sub-centimétrique lui permettant une localisation et une estimation de distance précise [77]. La résolution temporelle peut être évaluée grâce à l'inégalité de Cramer-Rao qui donne une estimation de la borne inférieure d'une mesure de distance.
- La sécurité : sans avoir besoin d'un codage quelconque, le spectre du signal UWB se situe en-dessous du niveau du bruit. Ce signal est donc, naturellement discret et difficilement détectable par un autre système. Cette faible densité spectrale de puissance du signal améliore la sécurité des communications radio UWB.
- Franchissement d'obstacles : un signal UWB s'étale sur une large gamme de fréquence, donc des applications telles que la vision aux travers des murs, la poursuite d'objet, ou la localisation, peuvent bénéficier d'une excellente résolution en particulier aux fréquences basses.

1.4.1.3 Localisation avec mesure UWB

La technologie UWB est très sollicitée dans des applications de localisation et de poursuite de personnes et d'objets. En effet, l'Ultra Large Bande permet une grande précision dans la mesure des distances car la résolution obtenue est inversement proportionnelle à la durée de l'impulsion, ce qui la rend adaptée aux systèmes de localisation [77]. De ce fait, la technologie UWB offre une opportunité unique pour hybrider les fonctionnalités de localisation et communications numériques sans fil. Ceci répond à des exigences fortes liées à de nombreuses applications émergentes. Rappelons que l'UWB se propose d'adresser de façon privilégiée deux standards WPAN. D'une part, un standard haut débit HDR-UWB11 dit standard IEEE 802.15.3a (de 1Mbps jusqu'à plusieurs centaines de Mbps) et d'autre part, un standard bas-débit LDR-UWB12 dit standard IEEE 802.15.4a (de quelques kbps à 1Mbps).

Le principe de positionnement par UWB est la trilatération (cf. figure 1.2). Des balises UWB situées sur les mobiles émettent des données, qui sont reçues par des récepteurs fixes répartis dans l'environnement. Ces derniers évaluent la distance des balises par mesure du temps de propagation des signaux, ce qui permet au système de calculer la position des mobiles par triangulation. Le modèle d'observation de mesure de distance sur balise UWB est présenté dans l'équation (1.2) :

$$r = R + b + \beta \tag{1.2}$$

où r désigne la mesure de distance UWB, R représente la distance géométrique entre le récepteur et l'émetteur UWB, b et β sont respectivement le biais et le bruit de la mesure.

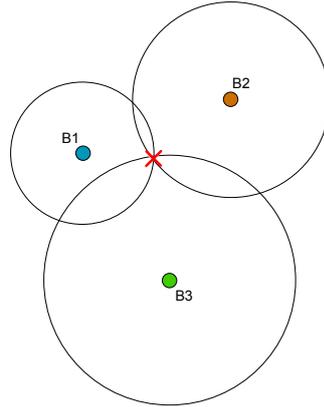


Figure 1.2 – Localisation avec 3 balises par le principe de trilatération. La croix rouge matérialise la position inconnue.

On peut également concevoir des systèmes de positionnement dans lesquels il n'y a que des mobiles, tous équipés d'un émetteur-récepteur UWB. Dans ce cas les mobiles se communiquent de proche en proche les informations de distance relevées et parviennent à calculer leurs distances respectives également par trilatération. Des systèmes de localisation par UWB sont définis dans [78]. Des travaux récents qui portent sur la localisation coopérative des robots par UWB sont présentés dans [79]. Les auteurs ont présenté un système de localisation par UWB pour qu'un robot puisse se localiser tout seul et coopérer avec plusieurs robots en présence de communication. Les performances du système de localisation UWB ont été évaluées sur un quadricoptère en comparaison avec un système de localisation de capture de mouvement. Dans [80], un système coopératif du positionnement des véhicules intégrant des mesures UWB et GNSS a été présenté. L'objectif est de surmonter le problème de perte de signaux GNSS par l'assistance de l'UWB et par la coopération entre les véhicules.

1.4.2 Système Global de Navigation par Satellite

Les systèmes de positionnement par satellites comportent essentiellement une constellation de satellites de navigation situés en orbite à environ vingt mille kilomètres d'altitude sur la surface de la terre, fournissant un service de géolocalisation couvrant le monde entier.

L'idée de base est que les récepteurs mesurent le temps d'arrivée des signaux satellites et le comparent au temps de transmission pour calculer le temps de propagation des signaux. Le temps de propagation est utilisé ensuite pour estimer les distances entre le récepteur GNSS et les satellites. La distance estimée est connue sous le nom de pseudodistance. En utilisant les estimations de pseudodistances, les récepteurs GNSS calculent la position fixe de l'utilisateur à travers des techniques de multilatération

basées sur plusieurs mesures satellitaires. La précision des estimations de position dépend à la fois de la précision des mesures de pseudodistances et de la géométrie des satellites utilisés dans la multilatération.

Actuellement, plusieurs systèmes sont entièrement ou partiellement opérationnels : le système américain NAVSTAR ⁷ Global Positioning System (GPS), le système russe Globalnaya Navigatsionnaya Sputnikovaya Sistema (GLONASS) qui est de nouveau opérationnel suite à sa restauration après une période d'abandon, le système de positionnement européen Galileo et le système chinois Beidou-2, aussi appelé Compass (BDS).

Le système GPS est aujourd'hui le plus développé des systèmes de positionnement par satellites. Permettant le calcul PVT (Position, Vitesse et Temps), il est utilisé dans le monde entier pour des applications aussi diverses que l'aide à la navigation, la gestion de flotte, le guidage d'engins ou encore la synchronisation d'horloges et d'émetteurs. Le GPS a été développé par le Département de la Défense (DoD) américain en 1973. Il n'est devenu entièrement opérationnel qu'à partir du 27 avril 1995. Le GPS a été initialement conçu pour des applications exclusivement militaires. Ensuite, des applications civiles sont devenues possibles mais avec une dégradation volontaire de la qualité des signaux civils par la mise en place de la *Selective Availability* (SA), où la précision était limitée à une centaine de mètres. Depuis l'année 2000, la SA a été désactivée afin d'améliorer la précision jusqu'à une dizaine de mètres. Le système GLONASS a été développé simultanément avec le GPS et déclaré opérationnel en 1997. GLONASS a souffert de dégradations et d'interruptions de service en raison de l'absence d'entretien de la constellation par les autorités russes, au point de n'avoir plus que 6 satellites fonctionnels en 2000. De nouveaux satellites ont été mis en orbite depuis le début des années 2000 afin de restaurer le fonctionnement du système. Le système GNSS européen Galileo est en cours de développement par l'UE et l'Agence Spatiale Européenne (ESA). Il a été officiellement lancé le 26 mars 2002. L'objectif de Galileo est de fournir un système alternatif du GPS de haute précision propre à la communauté Européenne et qui sera géré par un organisme civil. Galileo commence à offrir ses premiers services. L'achèvement complet du système des 30 satellites Galileo est prévu en 2020. Le système BDS de la Chine est encore en construction. Il est devenu opérationnel en Chine en Décembre 2011. Ensuite en Décembre 2012, le BDS a commencé à offrir ses services à la région d'Asie-Pacifique. Il est prévu de passer à une couverture mondiale à son achèvement avec 35 satellites à l'horizon 2020.

Tous les GNSS partagent une certaine similitude comme, par exemple, leur haute disponibilité et leur service continu. Nous allons présenter les GNSS en prenant le GPS comme exemple, puisque le GPS est le seul GNSS qui est pleinement opérationnel et largement utilisé. Dans une première partie, des éléments principaux de GPS seront décrits. Ensuite, nous allons expliquer les principes de fonctionnement et les différentes techniques utilisées pour calculer une position à partir de mesures

7. NAVigation System by Timing And Ranging

GPS.

1.4.2.1 Concepts de base de GPS

L'utilisation d'un système de positionnement global nécessite le choix de référentiels communs de temps et d'espace, définis sur toute la surface du globe.

Le référentiel de temps pour les calculs GPS est le temps GPS. Le temps GPS a été calé sur le Temps universel coordonné (UTC) en 1980. En raison des secondes intercalaires ajoutées régulièrement pour garder l'UTC synchronisé avec la rotation de la Terre. En 2011, le temps GPS avait un écart de 15 s par rapport à l'UTC.

Le système géodésique associé au GPS est le WGS 84 (*World Geodetic System 1984*), ou Système géodésique mondial, révision de 1984, qui s'est rapidement imposé comme une référence pour la cartographie. L'origine de WGS 84 se trouve au centre de masse de la Terre, avec une erreur supposée inférieure à 2 cm. Il définit également les paramètres d'un ellipsoïde de révolution aplati aux pôles approximant l'ellipsoïde terrestre, ainsi qu'un méridien de longitude zéro situé à 102,5 m à l'est du méridien de Greenwich.

Le système GPS est constitué de trois parties principales nommées « segments » : le segment spatial, le segment de contrôle et le segment utilisateur.

Segment spatial Le segment spatial contient initialement une constellation d'un ensemble de 24 satellites répartis sur six plans orbitaux (Figure 1.3) pour fournir une capacité de positionnement global continu. Actuellement, la constellation comporte 31 satellites. Ces satellites évoluent à une altitude d'environ 20 000 km et mettent un demi-jour sidéral, soit 11 h 58 min 2 s pour effectuer une rotation autour de la terre. La disposition de la constellation assure une visibilité minimale de 4 satellites à une élévation minimale de 15°.

Chaque satellite possède plusieurs horloges atomiques qui asservissent un oscillateur à une fréquence fondamentale de 10,23 MHz. L'émetteur génère deux porteuses (L1 et L2) de fréquences respectives 1575,42 MHz et 1227,60 MHz.

Chaque satellite émet :

- des signaux horaires (codes) sur les porteuses L1 et L2,
- des almanachs (données grossières sur les orbites de l'ensemble des satellites de la constellation),
- et des éphémérides fournissant une description fine de l'orbite du satellite, son état, le décalage d'horloge, etc.

Plus de détails sur l'orbite, la couverture et les performances de la constellation de satellites GPS sont documentées dans les normes de performance GPS [81].

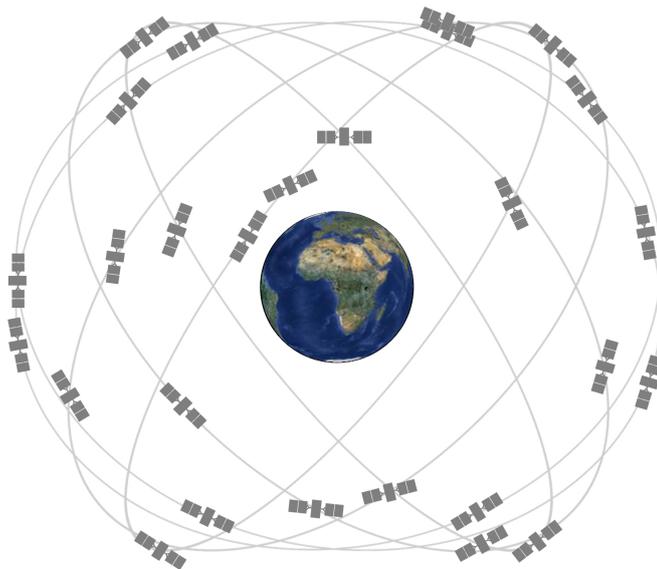


Figure 1.3 – Constellation des satellites du système GPS [1].

Segment de contrôle Le segment de contrôle du GPS est composée d'un réseau mondial de stations au sol réparties le long de l'équateur qui permettent de suivre les satellites GPS, surveiller leurs transmissions, effectuer des analyses, et envoyer des commandes et des données à la constellation. Actuellement, le segment de contrôle opérationnel comprend une station maîtresse de commande (MCS), une station maîtresse de contrôle alternative, 12 antennes de commande et de contrôle et 16 sites de surveillance.

Les antennes au sol sont utilisées pour communiquer avec les satellites GPS. Les stations de surveillance permettent de suivre les satellites GPS et de recueillir des données atmosphériques, les mesures de code / phase et les signaux de navigation. Les stations de surveillance envoient leurs observations à la station maîtresse de contrôle à Colorado. Ensuite, la station MCS génère et télécharge des messages de navigation et s'assure de la santé et de la précision des constellations de satellites.

Segment utilisateur Le segment utilisateur comprend l'ensemble des récepteurs civils et militaires qui ne font que recevoir les informations des satellites. Le nombre d'utilisateurs est ainsi illimité. Pour déterminer la position, la vitesse et le temps, la réception simultanée des signaux de quatre satellites est nécessaire.

On trouve une large variété de récepteurs, allant du récepteur mono-fréquence bon marché au récepteur dit de géodésie, beaucoup plus précis et réservé aux applications de topographie. Le type de récepteur employé dépend fortement de l'application qui exploite le GPS. Ainsi, pour une application dans le domaine de l'automobile, les constructeurs se contentent généralement de récepteurs mono-fréquence (L1C) à faible coût.

Selon le rapport du marché GNSS⁸ fourni par l'agence européenne de GNSS, l'industrie automobile envisage l'amélioration du positionnement pour les différentes applications innovantes par la fusion des capteurs avec des récepteurs GNSS qui seront intégrés dans les véhicules.

1.4.2.2 Observables GPS

Le terme observable est utilisé dans la communauté GPS pour parler des mesures brutes d'un récepteur. Sur chaque satellite en vue, un récepteur mono-fréquence civil mesure toujours le code C/A et parfois la phase et le Doppler sur L1. Un récepteur bi-fréquence peut quant à lui mesurer, le code C/A sur L1, la phase et le Doppler sur L1 et L2. Un récepteur militaire mesure en plus le code P sur L1 et L2. Il est à noter que certains récepteurs civils performants, grâce à des méthodes de traitement du signal d'extraction de phase sur L2, délivrent également des observables bruités de code P sur L1 et L2. Les erreurs d'observable d'une même porteuse sont souvent considérées comme peu corrélées du fait que la traversée dans l'atmosphère n'induit pas les mêmes effets sur le code et la phase. Le Doppler, indispensable pour la démodulation des signaux des satellites au niveau des boucles à verrouillage de phase, est mis à profit notamment pour la mesure de vitesse.

Sur chaque porteuse (L1 ou L2) les récepteurs GPS peuvent fournir trois types d'observables : la mesure de pseudodistance, qui est une mesure de temps de vol obtenue à partir des codes C/A ou P(Y) ; la mesure de Doppler, la fréquence des signaux reçus qui varie en fonction de la vitesse relative entre les satellites et le récepteur ; et enfin la mesure de phase, qui compte le nombre de cycles de la porteuse (à une ambiguïté inconnue près).

Nous allons nous concentrer sur le calcul d'une position à partir des mesures de pseudodistance. Les méthodes de positionnement à partir des mesures de phase et de Doppler, et par combinaison d'observables sur les fréquences L1 et L2 sont détaillées dans [82, 83].

Structure du signal GPS La structure du signal est basée sur la fréquence fondamentale $f_0 = 10.23$ MHz. La stabilité des signaux GPS est assurée grâce aux plusieurs horloges atomiques sur chaque satellite. À partir de cette fréquence fondamentale, chaque satellite génère deux ondes porteuses L1 et L2 :

$$\begin{cases} L1 & = 154 \cdot f_0 = 1575.42 \text{ MHz} \\ L2 & = 120 \cdot f_0 = 1227.60 \text{ MHz} \end{cases} \quad (1.3)$$

Le système GPS émet sur deux fréquences afin d'éliminer les effets de l'ionosphère. Ces deux porteuses constituent la base du signal GPS et sont modulées par le message de navigation ainsi que des séquences pseudo-aléatoires différentes pour chaque

8. http://www.navipedia.net/index.php/GNSS_Market_Report

satellite appelées *codes PRN*. Chacun des codes est transmis à des débits différents [83] :

- **Code C/A (coarse/acquisition)**. Séquence pseudo-aléatoire de 1023 bits émise à la fréquence de 1,023 MHz. La séquence se répète à l'identique toutes les millisecondes. La rapidité d'émission de ce code permet une acquisition rapide du signal par le récepteur. Le C/A code n'est transmis que sur L1 et n'est pas crypté.
- **Code P(Y) (precise)**. Ce code est réservé au service *PPS*⁹. Il est constitué d'une séquence unique de 2.3547×10^{14} bits émis à la fréquence de 10,23 MHz, soit une période de 266,41 jours. Il est modulé sur les deux fréquences L1 et L2. Les caractéristiques de code P(Y) sont inconnues et il est donc impossible de le reproduire. Ce code n'est accessible qu'aux utilisateurs possédant des clés de décryptage comme l'armée américaine et certains de ses alliées. Le message de navigation, de fréquence de 50 Hz est modulé sur les deux fréquences. Ce message contient les informations spécifiques au satellite ainsi que des données communes à l'ensemble des satellites et des informations générales sur le système.
- **Message de navigation**. Le message de navigation est transmis à la vitesse de 50 bits par seconde et se décompose en cinq sous-messages (*subframes*) de 300 bits. Les deux premiers mots de chaque portion sont identiques : ils contiennent un préambule fixe, le *time-of-week*, qui permet de passer du code C/A au code P, et un bit d'alerte indiquant un fonctionnement erroné. Le message de navigation contient des informations sur le satellite telles que son éphéméride (qui contient les paramètres képlériens de l'orbite), les paramètres d'horloge, des indicateurs sur la santé du satellite, etc. Les sous-messages 4 et 5 contenant des informations de moindre importance (en particulier les almanachs, qui décrivent de façon grossière les orbites de tous les satellites de la constellation).

Dans le cas d'un récepteur mono-fréquence, il est possible d'extraire la mesure de code C/A, la mesure de phase du signal, ainsi que le décalage Doppler de la fréquence mesurée par rapport à la fréquence L1.

L'utilisation d'un récepteur bi-fréquence permet d'accéder à des observables supplémentaires : la mesure de phase sur L2, le décalage Doppler sur L2 ainsi que la mesure de code P(Y) si le récepteur sait la décoder. Les récepteurs bi-fréquences sont généralement des récepteurs haut de gamme orientés vers des applications nécessitant un positionnement précis. En effet, le bi-fréquence fournit une redondance d'information permettant de réaliser des combinaisons d'observables de façon à éliminer l'allongement ionosphérique par exemple [84, p.35]. Ceci permet d'augmenter sensiblement la qualité de la solution calculée. La résolution n'est possible cependant que si les mesures sur L1 et L2 sont disponibles à l'instant du calcul.

9. *Precise Point Service*

1.4.2.3 Équation d'observation associée au code

Le principe de mesure de code C/A repose sur une mesure du temps de vol. Les séquences C/A générées par les satellites sont émises à des instants connus dans le système temps GPS. Le récepteur génère à tout instant une réplique des codes pseudo-aléatoires des différents satellites ; les signaux ainsi générés sont comparés au signal reçu. Comme les codes des satellites sont orthogonaux, quand le signal répliqué et le signal reçu sont fortement corrélés, la signature du satellite recherché est présente dans le signal reçu.

Le récepteur est capable de mesurer le temps de vol Δt en connaissant parfaitement son instant d'émission t_e (dans le système de temps GPS) et en mesurant son instant d'arrivée t_r :

$$\Delta t = t_r - t_e$$

On notera dans la suite les paramètres dépendant des récepteur et satellite respectivement par les indices u et s .

En pratique, comme les horloges dérivent, ni le récepteur ni le satellite ne sont parfaitement à l'heure GPS. Ils présentent tous les deux un décalage respectif dt_u et dt_s par rapport au temps GPS.

La date où le satellite transmet le signal dans son propre référence de temps en fonction de l'instant t_e est :

$$t_s(t_e) = t_e + dt_s(t_e) \tag{1.4}$$

La date où le récepteur effectue la mesure dans son propre référence de temps en fonction de l'instant t_r est :

$$t_u(t_r) = t_r + dt_u(t_r) \tag{1.5}$$

Le temps de vol mesuré est donc

$$\begin{aligned} \Delta t_m &= t_u(t_r) - t_s(t_e) \\ &= t_r - t_e + dt_u(t_r) - dt_s(t_e) \end{aligned} \tag{1.6}$$

Sous l'hypothèse que le signal se propage à la vitesse c de la lumière et en négligeant toute perturbation, la pseudodistance ρ entre le satellite et le récepteur s'écrit :

$$\begin{aligned}
 \rho &= c.\Delta t_m \\
 &= c(t_r - t_e) + c.dt_u(t_r) - c.dt_s(t_e) \\
 &= R + c.dt_u(t_r) - c.dt_s(t_e)
 \end{aligned}
 \tag{1.7}$$

où R est la distance géométrique entre le satellite et le récepteur.

On parle de mesure de pseudodistance car l'estimation de la distance fournie par le récepteur est entachée des erreurs d'horloge à la fois du satellite et du récepteur.

L'expression de pseudodistance de l'équation 1.7 n'est vraie que dans le cas où un signal évolue dans le vide et en ligne droite. Cependant, le signal GPS se propage dans l'atmosphère de la Terre, ce qui engendre une dégradation de la mesure de pseudodistance. Le signal peut également être retardé ou avancé par de nombreux phénomènes. Nous nous intéressons par la suite à en présenter les principaux.

Effets Ionosphériques Située entre 50 et quelques centaines de kilomètres d'altitude, la ionosphère est une couche haute de l'atmosphère ionisée par l'action des radiations solaires. Cette ionisation dépend de la latitude, de l'activité solaire, de la période de l'année, du jour et de la nuit. Lorsque l'onde du signal traverse ce milieu, tout se passe comme si la distance mesurée (i.e. la pseudodistance) était plus longue que la vraie distance géométrique satellite-récepteur. Ceci est dû au fait que les ondes électromagnétiques traversant ce milieu sont retardées. L'effet sur la mesure (noté δ_{iono}) induit un allongement qui peut varier de 0 à 50m.

Effets Troposphériques Située entre le sol et 8 à 17 kilomètres d'altitude, la troposphère est la couche basse de l'atmosphère. Son influence dépend de la température, de la pression, de l'humidité et de l'élévation du satellite mais pas de la fréquence. On peut recourir à l'utilisation de modèles pour la réduire. Les retards du signal sont dus à deux phénomènes :

- le trajet des ondes passant dans ce milieu est courbé et donc la longueur du trajet est augmentée,
- la vitesse de propagation décroît.

L'effet sur la mesure (noté δ_{tropo}) peut varier de 2 à 30 m suivant l'élévation du satellite.

Effets Relativistes Les effets relativistes (noté δ_{rel}) sont dus au fait qu'il y a une différence de champ gravitationnel et surtout une grande différence de vitesse entre le satellite et le récepteur. Cela se traduit par une dilatation du temps et un décalage des fréquences [85]. Ainsi, vont être modifiés :

- le décalage d'horloge du satellite par rapport au temps GPS,

- le décalage d’horloge du récepteur par rapport au temps GPS,
- le signal satellite par rapport au récepteur (courbures et retards dus à la gravitation terrestre).

Le bruit de mesure Le signal provenant du satellite étant fortement atténué, le récepteur procède à une amplification ce qui amplifie également le bruit. On considère que le bruit de la mesure induit une erreur minimum de 1% de la longueur d’onde du code sur lequel on fait la mesure ; par exemple, pour le code C/A de longueur d’onde 300 m, on a une erreur de 1 à 3 m. Ce bruit affecte la mesure du retard temporel et donc le calcul de la pseudodistance. On le note β .

Modèle complet de la mesure de code En prenant en compte les diverses dégradations que nous venons d’évoquer, on obtient :

$$\rho = R + c.dt_u - c.dt_s + \delta_{rel} + \delta_{tropo} + \delta_{iono} + \beta$$

Correction des mesures avec les données de navigation Nous avons vu précédemment que les données du message de navigation fournissent des indications telles que l’état de santé du satellite, son identification, son éphéméride, sa correction d’horloge, la correction ionosphérique, etc.

Grâce au message de navigation, on peut calculer :

- le décalage d’horloge du satellite \hat{dt}_s
- l’allongement ionosphérique $\hat{\delta}_{iono}$
- l’allongement troposphérique $\hat{\delta}_{tropo}$
- les erreurs relativistes $\hat{\delta}_{rel}$

La bibliothèque logicielle GPSTk [86] de l’Université du Texas contient de nombreuses fonctions qui permettent d’estimer ces paramètres. On peut noter que les allongements ionosphérique et troposphérique dépendent de la position du satellite par rapport au récepteur. Ces corrections ne peuvent donc être appliquées qu’une fois qu’une position grossière a été déterminée.

Quand on applique ces corrections, on obtient la pseudodistance corrigée

$$\begin{aligned} \rho &= R + c.dt_u(t_r) - c.dt_s(t_e) + \delta_{Rel} + \delta_{Tropo} + \delta_{Iono} + \beta \\ &\quad + c.\hat{dt}_s(t_e) - \hat{\delta}_{Rel} - \hat{\delta}_{Tropo} - \hat{\delta}_{Iono} \\ &= R + c.dt_u(t_r) + \beta + \text{erreurs résiduelles} \end{aligned}$$

Il existe d’autres phénomènes induisant des imprécisions sur la mesure de pseudodistance et qui sont difficile à corriger. On cite ci-après les plus connus.

Les multitrajets Un multitrajet provient de la réflexion du signal avant réception. Le signal capté par l'antenne apparaît comme la somme du signal direct et des multitrajets. Si l'antenne (par une analyse de la polarité de l'onde) ou le corrélateur ne parviennent pas à éliminer le ou les multitrajets, il apparaît un allongement. Dans le cas extrême, le trajet direct n'est pas présent ; la détection bas niveau du multitrajet est alors impossible. Des multitrajets éphémères ne sont pas trop problématiques car ils sont filtrés par les boucles à verrouillage de phase des corrélateurs. Ce cas peut se produire en milieu urbain, par exemple, si le mobile est en mouvement : les multitrajets apparaissent et disparaissent rapidement. Par contre, des multitrajets persistants sont détectables s'ils sont importants en étudiant la cohérence du signal avec les autres. Remarquons que plus les satellites sont situés à la verticale du récepteur, moins ils sont sensibles aux multitrajets.

Mauvaise connaissance de la position des satellites Les paramètres d'éphéméride qui permettent de retrouver les coordonnées des satellites \mathbf{x}_s^{ECEF} sont estimés à partir d'observations réalisées par les stations permanentes liées au segment de contrôle. Ces valeurs sont envoyées toutes les deux heures aux satellites qui les répètent dans leurs données de navigation.

Lorsque l'on fait du calcul en temps réel, le récepteur doit donc extrapoler ces paramètres pour retrouver la position du satellite. Il existe ainsi une erreur résiduelle, notée δ_{X_S} , qui affecte la mesure de pseudodistance. D'après [83], δ_{X_S} est de l'ordre du mètre à 1σ .

On distingue trois composantes pour l'erreur en position du satellite : l'erreur radiale, l'erreur en position le long de la trajectoire (*along-track*) et perpendiculairement à cette dernière (*cross-track*). L'erreur radiale est la composante ayant l'influence la plus importante sur l'erreur de mesure mais c'est celle qui peut être la plus aisément observée et corrigée par le centre de contrôle. Inversement, les erreurs transversale et le long de la trajectoire ont un impact faible sur l'erreur en position.

L'utilisation des éphémérides précises permet de réduire l'influence de ces termes d'erreurs. En général, elles ne sont pas disponibles en temps-réel mais peuvent être utilisées en traitement différé.

1.4.2.4 GPS Différentiel

Le GPS différentiel (DGPS) est une amélioration de GPS qui permet d'avoir un positionnement de qualité supérieure. Il utilise une station fixe ou un réseau de stations fixes de référence qui transmet des corrections. Des agences gouvernementales ou des sociétés privées fournissent ou revendent les corrections. Depuis l'an 2000, le procédé DGPS a perdu un peu de son intérêt mais conserve tout de même un avantage sur le GPS en offrant une résolution de l'ordre du mètre. Cette technologie est très employée en navigation aussi bien terrestre que maritime.

Une autre solution consiste à avoir, sur un support informatique, l'ensemble des corrections propres aux stations de référence et de traiter les corrections de coordonnées en post-traitement.

En dehors des corrections émises par des stations terrestres, des satellites ont été lancés afin de créer une correction des données directement depuis l'espace. C'est ce qu'on appelle les systèmes SBAS (Satellite Based Augmentation System). Les satellites sont géostationnaires et émettent les corrections pour une zone du globe. Le système n'est plus alors considéré comme du DGPS. En effet, un satellite envoie ces corrections pour une vaste zone, contrairement au DGPS, où la station envoie des corrections locales. Les satellites envoient des corrections d'horloge, d'orbites des satellites GPS, ionosphériques et troposphériques qui sont calculées, puis transmises en même temps que des informations d'intégrité relatives à la santé des satellites GPS. Pour les Etats-Unis (agence fédérale de l'aviation), c'est le système WAAS (Wide Area Augmentation System) ; pour l'Europe (Commission européenne, Agence spatiale européenne et organisme européen chargé de la sécurité de la navigation aérienne), c'est le système EGNOS (European Geostationary Navigation Overlay System) et enfin pour le Japon, c'est le système MSAS (Multi-functional Satellite Augmentation System). Au sol, des stations récupèrent les données SBAS pour diffuser les corrections. Les trois systèmes SBAS sont compatibles entre eux. La précision obtenue avec ce système est d'environ 3m.

Malgré cette correction importante, la précision recherchée pour certaines applications n'est pas atteinte. Afin d'améliorer le positionnement au niveau du centimètre, on a recours au GPS RTK.

1.4.2.5 GPS RTK

Le GPS RTK (Real Time kinematic ou cinématique temps réel) utilise le même principe de correction différentielle que le DGPS. Une base dont la position est connue transmet les corrections. Son avantage provient de la différence de phase utilisée pour la correction. Alors que le DGPS utilise le code (environ 1,023MHz), le GPS RTK utilise la phase de la porteuse du signal (oscillant à 1575,42MHz) apportant ainsi une précision proportionnellement plus importante. Cette amélioration a un prix ; le récepteur ne doit pas être à plus de 10km de la base. Une solution est alors d'avoir sa propre base correctrice qui envoie les corrections par liaison radio au mobile. La mesure de phase de l'onde porteuse consiste à comparer la phase de l'onde reçue par le récepteur avec la phase de l'onde générée à l'intérieur du récepteur. Théoriquement, cette différence de phase oscille entre 0 et 2π . Cette mesure de phase peut être convertie en mètres puisqu'on connaît la longueur d'onde de l'onde porteuse (autour de 20cm). Malheureusement, le nombre entier de longueur d'onde initial contenu dans la distance récepteur-satellite n'est pas mesurable par le récepteur. Cette inconnue est appelée l'ambiguïté de phase initiale. Par contre, le récepteur est à même de compter le nombre entier de cycles (ainsi que la partie fractionnaire)

cumulé depuis l'époque (ou le temps) d'observation initiale, s'il n'y a pas d'interruption dans la réception du signal. Les interruptions provoquent des sauts de cycles et sont principalement causées par les obstructions (édifices, montagnes, arbres, ...) entre les satellites et le récepteur. La mesure de phase peut être interprétée comme une mesure précise de la variation de la distance récepteur-satellite depuis l'époque initiale. Si l'ambiguïté de phase initiale peut être résolue, la mesure de phase ainsi corrigée représente une mesure précise de la distance récepteur-satellite. La résolution d'une mesure de phase est de quelques millimètres. En pratique, l'incertitude de positionnement est de l'ordre de quelques centimètres.

1.5 Problématique et organisation du manuscrit

1.5.1 Problématique

Au cours de ces travaux de thèse, des systèmes robotiques mobiles effectuant des mesures radio sur des balises actives ou des satellites sont considérés. Comme nous l'avons vu, ces mesures de distances ou de pseudodistances sont entachées d'erreurs systématiques connues aussi sous le nom de « biais ». Ces biais peuvent introduire des erreurs importantes sur le positionnement, ce qui n'est pas souhaitable pour des applications nécessitant une haute précision.

Deux grandes problématiques sont abordées dans cette thèse. La première consiste à étudier comment la coopération par échange des données peut corriger les erreurs systématiques des mesures et diminuer l'imprécision et l'incertitude de la position en absolu et en relatif, sans utiliser de station fixe et sans avoir des mesures directes entre les robots. L'imprécision correspond à l'écart entre une estimée et la vérité terrain. L'incertitude est relative à un domaine de confiance autour de l'estimée.

La deuxième problématique consiste à déterminer des domaines de confiance fiable contenant la position réelle des robots ce qui est indispensable pour éviter tout risque de collision présent dans certaines tâches en robotique comme la navigation à proximité. C'est le cas pour de nombreuses missions robotiques comme explorer un environnement inconnu, effectuer un rendez-vous entre deux robots ou encore naviguer en flotte.

Un premier objectif lié à ces travaux est d'étudier s'il est possible d'observer les biais des mesures et de déterminer sous quelles conditions un système coopératif peut effectivement les observer. Il conviendra ensuite de proposer des observateurs d'état distribués et de les comparer notamment avec des données réelles.

1.5.2 Organisation du manuscrit

Dans le deuxième chapitre, nous présenterons une étude d'observabilité du système coopératif de n_r robots dans les cas des mesures de distance sur balise et de pseudo-

distance sur satellites sous des hypothèses simplifiées. Les cas statique et dynamique seront notamment étudiés pour le positionnement par pseudodistances. L'étude des propriétés structurelles du système coopératif est nécessaire mais n'est pas suffisante pour confirmer si l'estimateur peut déterminer les positions des robots. Toutefois, l'observabilité du système reste un critère indispensable à vérifier avant le développement d'une approche de localisation coopérative.

Le troisième chapitre abordera l'approche distribuée de localisation coopérative basée sur des techniques ensemblistes de propagation de contraintes sur intervalles et d'inversion ensembliste. L'objectif de la méthode est d'améliorer le positionnement en absolu et en relatif, en corrigeant les erreurs communes de mesures de GNSS pseudodistances (i.e. biais). Dans la première partie, quelques travaux sur la localisation coopérative dans la littérature seront présentés. Dans la deuxième partie, une formulation du problème coopératif de l'estimation sera décrite. Dans la troisième partie, les principes de localisation ensembliste, les techniques de propagation de contraintes et d'inversion ensembliste seront exposés. Ensuite, les résultats de notre approche de localisation coopérative seront présentés dans deux cas : avec des données simulées de distance sur balise en utilisant d'abord deux robots puis dix robots et avec des données réelles de pseudodistance en ne considérant que deux robots. Dans chaque cas, les résultats de l'approche coopérative seront comparés avec une méthode non-coopérative classique¹⁰ pour illustrer les améliorations apportées par la coopération.

Dans le dernier chapitre, nous développerons une approche coopérative Bayésienne dans le but de la comparer avec la méthode ensembliste proposée dans le chapitre précédent en utilisant deux robots. L'objectif de la méthode Bayésienne est aussi d'améliorer le positionnement en absolu et en relatif, en corrigeant les biais de mesures de GNSS pseudodistances. Les opérateurs de fusion de Combinaison Convexe Simple (SCC) et d'Intersection de Covariance (CI) seront employés pour la fusion des biais. Nous y rappellerons tout d'abord les principes des approches Bayésiennes classiques. Ensuite, un filtre Bayésien coopératif basé sur l'EKF sera présenté en utilisant deux types de fusion des biais (SCC et CI). Les effets de ces fusions sur le positionnement absolu et relatif seront évalués et comparés. De cette comparaison, la méthode Bayésienne basée sur la fusion fournissant les meilleurs performances sera retenue pour une comparaison avec la méthode ensembliste coopérative.

1.6 Conclusion

Dans ce chapitre, nous avons d'abord présenté les systèmes multi-robots et particulièrement les différents travaux sur l'estimation coopérative des robots mobiles. Nous avons classifié selon la topologie utilisée, deux grandes catégories d'approches :

10. Une méthode non-coopérative sera appelée *standalone* car la traduction française (autonome) n'est pas suffisamment claire et précise

les approches basées sur une architecture centralisée et les approches basées sur une architecture distribuée. L'architecture centralisée présente une solution simple et rigoureuse aux problèmes liés à la localisation coopérative. Cependant, elle est souvent inadaptée aux applications réelles avec un grand nombre de robots naviguant dans une vaste zone. Ces inconvénients sont principalement liés aux contraintes de la communication et à la dépendance à l'infrastructure. De ce fait, nous nous sommes orientés vers une architecture distribuée. Ensuite, nous avons classifié les approches distribuées en deux catégories selon le type d'informations partagées : les approches fondées sur le partage des mesures de capteurs et les approches fondées sur l'échange de l'état. En effet, ces approches limitent la quantité d'informations à échanger et elles sont indépendantes de l'infrastructure mais, en même temps, elles sont sensibles aux problèmes de consanguinité des données. Quelques solutions à ce problème ont été présentées parmi lesquelles nous avons choisi les méthodes ensemblistes. Ces méthodes sont bien adaptées à nos problématiques puisqu'elles évitent le problème de consanguinité des données et elles permettent d'avoir des domaines de confiance fiables contenant les positions des robots avec une forte probabilité.

Dans la deuxième section, nous avons présenté les technologies de positionnement considérées dans cette recherche. Nous avons mis l'accent sur les systèmes de positionnement par satellites et les systèmes de positionnement par balises UWB qui souvent utilisent des observations biaisées dont il faut tenir compte dans l'estimation coopérative.

Dans le chapitre suivant, nous allons étudier l'observabilité du système coopératif, afin de s'assurer qu'il est possible de localiser les robots avec des mesures biaisées.

Chapitre 2

Étude de l'observabilité de quelques systèmes robotiques coopératifs

Sommaire

2.1	Introduction	29
2.2	Problème de la localisation coopérative avec des mesures de distance sur balises	30
2.3	Problème de la localisation coopérative avec des mesures GNSS	35
2.4	Conclusion	59

2.1 Introduction

L'observabilité dans le cadre de la localisation des robots est une condition nécessaire mais pas suffisante pour qu'un estimateur de la pose converge. Cela signifie que : (a) le système peut être observable alors que l'estimation peut ne pas converger, (b) si le système n'est pas observable, il n'y a aucun moyen de déterminer la position des robots, quel que soit l'algorithme d'estimation utilisé. Pour pouvoir estimer la position des robots de façon coopérative, le système doit être donc observable. Les états peuvent être observés en fusionnant les mesures DR¹ avec les mesures de distances. Dans ce chapitre, l'observabilité de deux problèmes de localisation de systèmes coopératifs est traitée selon le système de positionnement utilisé. Les deux exemples traités dans ce chapitre correspondent à des mesures de distances sur des balises et de pseudodistances GNSS. Ces deux types de mesures sont affectés par des biais.

1. Dead-Reckoning

2.2 Problème de la localisation coopérative avec des mesures de distance sur balises

Le modèle d'observation des distances d'un système robotique coopératif est en général non linéaire (2.5). Par conséquent, les tests conçus pour les systèmes linéaires invariants dans le temps (par exemple, le rang de la matrice Gramienne [87]) ne peuvent pas être utilisés pour vérifier l'observabilité dans notre étude. Cependant, il est possible de linéariser le système et d'étudier l'observabilité faible locale au sens du rang. Si la matrice d'observabilité est de rang plein alors le système est faiblement localement observable. Par exemple, Antonelli *et al.* ont étudié l'observabilité faible locale de la localisation relative de deux robots sous marins [88]. Ils ont employé des mesures de distances relatives et des mesures de profondeur des robots sous la surface de l'eau. Ils ont notamment considéré le cas dynamique. Cependant, nous nous intéressons ici au cas statique par souci de simplification.

Notons r_i^j la distance qui sépare le robot i ($i = \{1, \dots, n_r\}$) situé aux coordonnées $\mathbf{p}_i = (x_i, y_i, z_i)$ de la balise j ($j = 1, \dots, n_b$) située aux coordonnées $\mathbf{p}^j = (x^j, y^j, z^j)$. n_r et n_b sont respectivement le nombre de robots et le nombre total de balises. De plus, la position des balises est connue avec une erreur $\Delta^j = (\delta_{x^j}, \delta_{y^j}, \delta_{z^j})$ dans le repère de travail. L'équation (2.1) représente l'expression de r_i^j :

$$r_i^j = \sqrt{(x_i - (x^j + \delta_{x^j}))^2 + (y_i - (y^j + \delta_{y^j}))^2 + (z_i - (z^j + \delta_{z^j}))^2} \quad (2.1)$$

Étant donné que les balises sont très éloignées de la zone d'évolution des robots, le modèle (2.1) peut être linéarisé. On peut considérer donc l'erreur Δ^j comme un biais additif sur la distance représenté par b^j :

Si on note :

$$\begin{cases} r_i^j = \sqrt{(x_i - (x^j + \delta_{x^j}))^2 + (y_i - (y^j + \delta_{y^j}))^2 + (z_i - (z^j + \delta_{z^j}))^2} \\ r_{i,real}^j = \sqrt{(x_i - x^j)^2 + (y_i - y^j)^2 + (z_i - z^j)^2} \end{cases}$$

De la figure 2.1, on a :

$$\cos \alpha = \frac{r_i^j - r_{i,real}^j \cos \theta}{b^j} \quad (2.2)$$

On peut négliger la composante transversale de l'erreur de position entre la position réelle de la balise j et sa position erronée par rapport aux distances r_i^j et $r_{i,real}^j$. Les deux vecteurs r_i^j et $r_{i,real}^j$ sont presque parallèles. Cela signifie que $\theta \approx 0$ par la suite

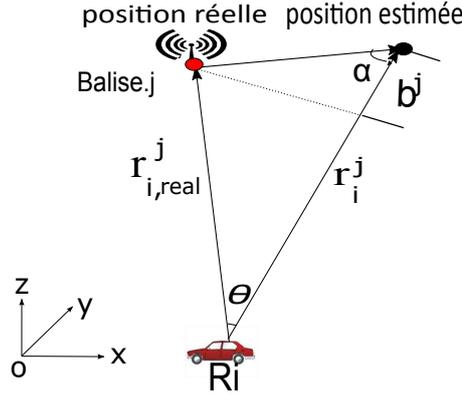


Figure 2.1 – Erreur de distance due à une erreur sur la position de la balise.

on a $\alpha \approx 0$. En prenant l'équation (2.2) en considérant que $\alpha \approx 0$ et $\theta \approx 0$ on obtient :

$$r_i^j = r_{i,real}^j + b^j \quad (2.3)$$

En tenant compte de l'équation (2.3), l'équation (2.1) devient :

$$r_i^j = \sqrt{(x_i - x^j)^2 + (y_i - y^j)^2 + (z_i - z^j)^2} + b^j \quad (2.4)$$

où b^j représente le biais de la balise j . On ajoute un terme β^j qui modélise le bruit de la mesure.

Le modèle de mesure de distance final est :

$$r_i^j = \|\mathbf{p}_i - \mathbf{p}^j\| + b^j + \beta^j \quad (2.5)$$

où $\|\mathbf{p}_i - \mathbf{p}^j\| = \sqrt{(x_i - x^j)^2 + (y_i - y^j)^2 + (z_i - z^j)^2}$. Le scénario de notre système coopératif est présenté dans la figure (2.2) (en particulier 2 robots et 4 balises sont considérés). \mathbf{p}_{R_1} et \mathbf{p}_{R_2} représentent respectivement le vecteur position \mathbf{p}_1 du robot 1 et le vecteur position \mathbf{p}_2 du robot 2 qui sont exprimés dans le repère du travail R_0 . Dans ces travaux, on considère que les biais doivent être les mêmes entre les robots. Si les biais sont dus aux erreurs de localisation des balises, on considère qu'elles soient éloignées du domaine d'évolution des robots. Dans le cas où les biais sont dus à un mauvais étalonnage du système UWB, les balises peuvent être proches mais elles doivent être bien localisées.

En théorie, il est possible de mettre en place un problème de moindres carrés non linéaires itératifs pour résoudre le problème d'estimation des biais. On obtient alors

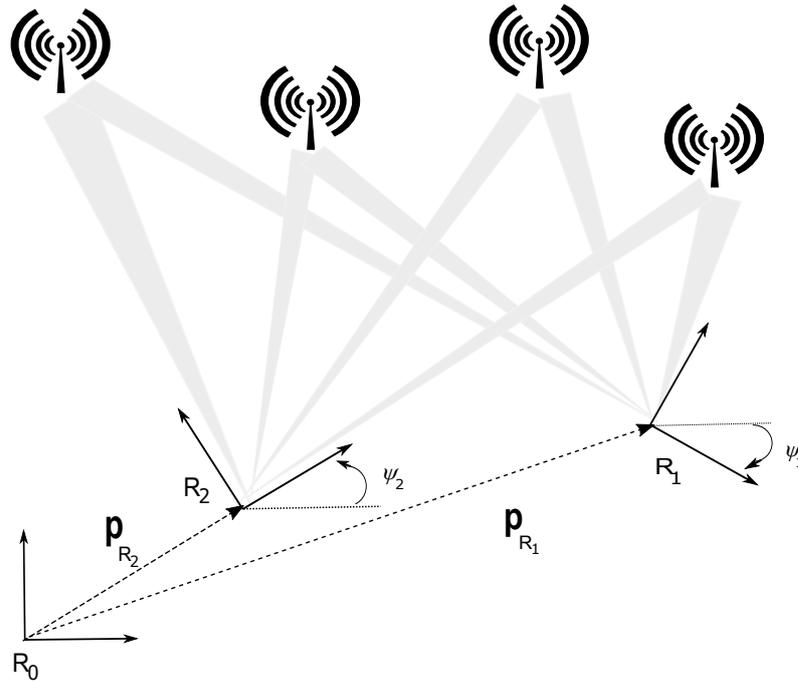


Figure 2.2 – Système coopératif de 2 robots avec 4 mesures de distances. Repère fixe (R_0) et repères mobiles (R_1 et R_2).

pour chaque estimation de biais b^j , une estimation de la position de chaque robot. Par souci de simplification, nous négligeons dans un premier temps le terme de bruit aléatoire de mesure β^j .

Afin de mettre en place le problème des moindres carrés, une linéarisation du modèle de mesure (équation 2.5) en fonction de δr^j et des états du système ($\delta \hat{\mathbf{p}}_i$ et $\delta \hat{b}^j$) est nécessaire.

$$\delta r^j = (\mathbf{u}_i^j)^T \cdot \delta \hat{\mathbf{p}}_i + \delta \hat{b}^j \quad (2.6)$$

Le vecteur \mathbf{u}_i^j (ligne de vue de la balise) est égal à :

$$\mathbf{u}_i^j = (\mathbf{p}_i - \mathbf{p}^j) / \|\mathbf{p}_i - \mathbf{p}^j\| \quad (2.7)$$

Considérons \mathbf{x} le vecteur d'état du système coopératif contenant les positions des n_r robots et les biais de toutes les n_s distances des balises.

$$\hat{\mathbf{x}} = [\hat{\mathbf{p}}_1, \dots, \hat{\mathbf{p}}_{n_r}, \hat{b}^1, \dots, \hat{b}^{n_s}]^T \quad (2.8)$$

2.2 Problème de la localisation coopérative avec des mesures de distance sur balises

avec $\dim(\hat{\mathbf{x}}) = 3n_r + n_s$.

Notons \mathbf{y} le vecteur de n_s mesures disponibles pour observer l'état global du système coopératif de n_r robots. Le vecteur \mathbf{y} est décrit comme suit :

$$\mathbf{y} = [r_1^1, r_1^2, \dots, r_1^{n_s}, \dots, r_{n_r}^1, r_{n_r}^2, \dots, r_{n_r}^{n_s}]^T \quad (2.9)$$

avec $\dim(\mathbf{y}) = n_s \times n_r$.

Les résidus du vecteur de mesure (2.9) sont liés aux variations du vecteur d'état (2.8) en utilisant le modèle linéarisé de mesure (2.6). Le système peut alors être linéarisé et présenté sous la forme matricielle suivante :

$$\delta \mathbf{y} = \mathcal{G} \delta \hat{\mathbf{x}} \quad (2.10)$$

la matrice \mathcal{G} est donnée par :

$$\mathcal{G} = \begin{matrix} & \overbrace{\hspace{10em}}^{3n_r + n_s} \\ n_r \times n_s \left\{ \begin{array}{cccc|c} \mathbf{G}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{G}_2 & \cdots & \mathbf{0} & \mathbf{I} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_{n_r} & \mathbf{I} \end{array} \right. \end{matrix} \quad (2.11)$$

où \mathbf{I} est la matrice identité et $\dim(\mathbf{I}) = n_s \times n_s$. Les $\mathbf{G}_1, \dots, \mathbf{G}_{n_r}$ sont les matrices qui contiennent les lignes de vue. Puisque $\dim(\mathbf{G}_i) = n_s \times 3$, on peut alors écrire :

$$\mathbf{G}_i = \begin{bmatrix} (\mathbf{u}_i^1)^T \\ (\mathbf{u}_i^2)^T \\ \vdots \\ (\mathbf{u}_i^{n_s})^T \end{bmatrix}. \quad (2.12)$$

L'ensemble d'équations du système matriciel semble être solvable, si le nombre d'équations ($n_r n_s$) croît plus rapidement que le nombre d'inconnues $n_I = 3n_r + n_s$. Donc si le nombre des mesures $n_M = n_r n_s$ est insuffisant (i.e, $n_r n_s < 3n_r + n_s$), le système est sous-contraint et le rang de la matrice \mathcal{G} est déficient. Dans ce cas, le système n'est pas observable. Il est évident que le rang de \mathcal{G} dans le cas où $n_M < n_I$ (i.e. $n_s < \frac{3n_r}{n_r - 1}$) est déficient, puisque le rang de \mathcal{G} est toujours inférieur ou égal au $\min(n_M, n_I)$, alors on peut donc écrire :

$$\text{rang}(\mathcal{G}) \leq n_M < n_I.$$

Étudions le cas où le système semble être solvable (*i.e.* $n_s \geq \frac{3n_r}{n_r-1}$).

Une façon de déterminer le rang de \mathcal{G} est de combiner linéairement ses lignes, en soustrayant le premier bloc de ligne de toutes les autres rangées de blocs :

$$\mathcal{G}_{sd} = \left[\begin{array}{cccc|c} \mathbf{G}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ -\mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\mathbf{G}_1 & \mathbf{0} & \cdots & \mathbf{G}_{n_r} & \mathbf{0} \end{array} \right]. \quad (2.13)$$

La différenciation des lignes matricielles ne change pas le rang d'une matrice, donc le rang de (2.13) est le même que celui de (2.11). Le rang de la nouvelle matrice \mathcal{G}_{sd} peut être déduit en additionnant le rang de chaque bloc de ligne BL_i .

le rang de BL_1 :

$$\text{rang}(BL_1) = \left[\overbrace{\mathbf{G}_1 \ \mathbf{0} \ \cdots \ \mathbf{0}}^{3n_r+n_s} \mid \mathbf{I} \right] = n_s.$$

Le rang du deuxième bloc de ligne :

$$\text{rang}(BL_2) = \left[\overbrace{-\mathbf{G}_1 \ \mathbf{G}_2 \ \cdots \ \mathbf{0}}^{3n_r+n_s} \mid \mathbf{0} \right] = 6.$$

Ce bloc représente une simple différence entre 2 robots, ce qui permet d'annuler les erreurs communes b^j . Il contient 6 équations linéairement indépendantes (on raisonne selon le nombre de colonnes car il est inférieur au nombre de lignes). Donc $\text{rang}(BL_2) = 6$.

Le rang de chaque bloc postérieur $BL_{i=3 \dots n_r}$ est égal à 3 puisque les nombres de colonnes indépendants ne sont contenus que dans les $\mathbf{G}_{3 \dots n_r}$:

$$BL_{3 \dots n_r} = \left[\overbrace{\begin{array}{cccc|c} -\mathbf{G}_1 & \mathbf{0} & \mathbf{G}_3 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \mathbf{0} & \ddots & \vdots & \vdots \\ -\mathbf{G}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_{n_r} & \mathbf{0} \end{array}}^{3n_r+n_s} \right].$$

Donc le rang($BL_{3\dots n_r}$) = $3(n_r - 2)$.

d'où :

$$\text{rang}(\mathcal{G}_{sd}) = 3(n_r - 2) + 6 + n_s = 3n_r + n_s. \quad (2.14)$$

Le nombre d'équations linéairement indépendantes de \mathcal{G}_{sd} est égal à $\dim(\hat{\mathbf{x}}) = 3n_r + n_s$, donc le système est observable si le nombre des balises est supérieur ou égal à $E\left(\frac{3n_r}{n_r-1}\right)$.

En particulier, si on a 2 robots dans le groupe, il faut au moins 6 balises pour que le système puisse être observable.

2.3 Problème de la localisation coopérative avec des mesures GNSS

2.3.1 Cas statique

On cherche à estimer les erreurs communes de mesures GNSS (i.e. biais) afin d'améliorer le positionnement par GNSS de véhicules coopératifs et de réduire les incertitudes. Nous ne disposons que des mesures brutes de GNSS sous forme de pseudodistances. Les erreurs sur les mesures de pseudodistances sont spatialement corrélées pour des utilisateurs proches.

Il est donc essentiel d'étudier théoriquement l'observabilité de ces erreurs pour évaluer la faisabilité de leur estimation. En effet, dans un contexte de localisation coopérative, Rife *et al.* [89] ont étudié l'observabilité des biais en se basant sur le calcul du rang de la matrice jacobienne du modèle linéarisé du système dans deux cas. D'abord, en utilisant uniquement le GNSS comme capteur. Dans ce cas, ils n'ont utilisé que des mesures brutes de pseudodistances. Dans le deuxième cas, ils ont ajouté des capteurs supplémentaires comme des caméras pour détecter le bord de la route et ajouter des informations géoréférencées.

Nous nous inspirons des travaux de Rife *et al.* pour étudier l'observabilité. Étudions donc le rang de la matrice jacobienne de notre système :

Soit i ($i = \{1, \dots, n_r\}$) l'indice des véhicules dans le groupe. Chaque agent est situé à la position $\mathbf{p}_i = (x_i, y_i, z_i)$. Soit j ($j = 1, \dots, n_s$) l'indice de satellites de positions $\mathbf{p}^j = (x^j, y^j, z^j)$ en commun pour tous les agents.

Le modèle de mesure d'une pseudodistance est :

$$\rho_i^j = \|\mathbf{p}_i - \mathbf{p}^j\| + c.dt_i + I^j + T^j + E^j + \varepsilon_i^j \quad (2.15)$$

où $\|\mathbf{p}_i - \mathbf{p}^j\| = \sqrt{(x_i - x^j)^2 + (y_i - y^j)^2 + (z_i - z^j)^2}$ représente la distance réelle entre la position du récepteur et la position de satellite commun en vue. $I^j + T^j + E^j$ sont respectivement les erreurs ionosphériques, troposphériques et d'estimation de la position des satellites qui ne sont pas compensables avec les données transmises dans les éphémérides. ε_i^j modélise les erreurs dues aux bruits et aux multi-trajets de la propagation des signaux des satellites. Le terme dt_i qui représente le décalage d'horloge entre le récepteur et le temps GNSS est converti en une unité de distance basée sur la célérité de la lumière c .

En GNSS, on peut classier les erreurs communes en 2 types :

- erreur spécifique du récepteur : $d_i = c.dt_i$ qui représente le décalage de l'horloge du récepteur.
- erreur spécifique de satellite : $b^j = I^j + T^j + E^j$ qui représente le biais du satellite j (ces biais sont spatialement corrélés). Les erreurs (b^j) sont presque les mêmes pour des utilisateurs proches même situés dans une région de plusieurs kilomètres de rayon. Dans ce cas, l'équation (2.15) peut être réécrite de la façon suivante :

$$\rho_i^j = \|\mathbf{p}_i - \mathbf{p}^j\| + d_i + b^j + \varepsilon_i^j \quad (2.16)$$

La linéarisation du système se fait comme dans la section (2.2). Comme nous faisons une étude d'observabilité, nous négligeons le terme de bruit aléatoire des mesures ε_i^j .

On linéarise le modèle de mesure (équation (2.16)) en termes de $\delta\rho_i^j$ et d'états du système ($\delta\hat{\mathbf{p}}_i$, δd_i et $\delta\hat{b}^j$) comme suit :

$$\delta\rho_i^j = (\mathbf{u}_i^j)^T \cdot \delta\hat{\mathbf{p}}_i + \delta d_i + \delta\hat{b}^j \quad (2.17)$$

Dans l'équation (2.17), le vecteur \mathbf{u}_i^j représente l'estimation de la ligne de vue (LOS²) entre les récepteurs des utilisateurs et chaque satellite j . Le vecteur \mathbf{u}_i^j est égal à :

$$\mathbf{u}_i^j = (\mathbf{p}_i - \mathbf{p}^j) / \|\mathbf{p}_i - \mathbf{p}^j\| \quad (2.18)$$

Soit \mathbf{x} le vecteur d'état du système contenant les positions et les décalages d'horloge des n_r agents et les biais des n_s satellites en commun.

$$\hat{\mathbf{x}} = [\hat{\mathbf{p}}_1, \hat{d}_1, \dots, \hat{\mathbf{p}}_{n_r}, \hat{d}_{n_r}, \hat{b}^1, \dots, \hat{b}^{n_s}]^T \quad (2.19)$$

avec $\dim(\hat{\mathbf{x}}) = 4n_r + n_s$.

2. Line Of Sight

Notons \mathbf{y} le vecteur de n_s mesures disponibles comme suit :

$$\mathbf{y} = [\rho_1^1, \rho_1^2, \dots, \rho_1^{n_s}, \dots, \rho_{n_r}^1, \rho_{n_r}^2, \dots, \rho_{n_r}^{n_s}]^T \quad (2.20)$$

avec $\dim(\mathbf{y}) = n_s \times n_r$.

Le système linéarisé peut être écrit sous la forme matricielle suivante :

$$\delta\mathbf{y} = \mathcal{G}\delta\hat{\mathbf{x}} \quad (2.21)$$

\mathcal{G} représente la matrice des coefficients du modèle linéarisé. Dans le cas spécifique où tous les utilisateurs visualisent le même ensemble de satellites, la matrice \mathcal{G} est donnée par :

$$\mathcal{G} = \left[\begin{array}{cccc|c} \mathbf{G}_1 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{G}_2 & \dots & \mathbf{0} & \mathbf{I} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{G}_{n_r} & \mathbf{I} \end{array} \right] \quad (2.22)$$

où \mathbf{I} est la matrice identité et $\dim(\mathbf{I}) = n_s \times n_s$. Les $\mathbf{G}_1, \dots, \mathbf{G}_{n_r}$ sont les matrices qui contiennent tous les coefficients des positions \mathbf{p}_i et les décalages d'horloges. Dans ce cas, $\dim(\mathbf{G}_i) = n_s \times 4$ et on peut alors écrire :

$$\mathbf{G}_i = \begin{bmatrix} (\mathbf{u}_i^1)^T & 1 \\ (\mathbf{u}_i^2)^T & 1 \\ \vdots & \vdots \\ (\mathbf{u}_i^{n_s})^T & 1 \end{bmatrix}. \quad (2.23)$$

Les vecteurs $\mathbf{u}^{j=1 \dots n_s}$ sont les mêmes pour tous les utilisateurs i si les distances entre chacun d'eux sont négligeables par rapport aux distances.

Deux cas sont considérés pour l'étude de l'observabilité :

- Cas où les robots sont à proximité.
- Cas où les robots sont éloignés.

2.3.1.1 Robots dans un voisinage

Sous l'hypothèse que les utilisateurs sont à proximité (dans un rayon de $10km$), les satellites sont très éloignés, on peut dire que $\mathbf{G}_1 = \mathbf{G}_2 \dots = \mathbf{G}_{n_r}$ et que l'équation

(2.22) peut être réécrite :

$$\mathcal{G} = \begin{matrix} n_r \times n_s \\ \left\{ \begin{array}{c} \overbrace{\left[\begin{array}{cccc|c} \mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{G} & \cdots & \mathbf{0} & \mathbf{I} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G} & \mathbf{I} \end{array} \right]}^{4n_r+n_s} \end{array} \right. \end{matrix} \quad (2.24)$$

En considérant le cas où $n_M \geq n_I$ (i.e $n_s \geq \frac{4n_r}{n_r-1}$), est-il possible de résoudre le système matriciel (2.21) ?

En suivant la même démarche présentée dans la section (2.2), on détermine le rang de \mathcal{G} , en soustrayant le premier bloc de ligne de toutes les autres rangées de blocs :

$$\mathcal{G}_{sd} = \left[\begin{array}{cccc|c} \mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ -\mathbf{G} & \mathbf{G} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\mathbf{G} & \mathbf{0} & \cdots & \mathbf{G} & \mathbf{0} \end{array} \right]. \quad (2.25)$$

Le rang de la matrice \mathcal{G}_{sd} peut être déduit en additionnant le rang de chaque bloc de ligne BL_i .

Le rang de la matrice

$$BL_1 = \left[\overbrace{\left[\begin{array}{cccc|c} \mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{array} \right]}^{4n_r+n_s} \right]$$

doit être inférieure ou égal à n_s et $4n_r + n_s$, soit inférieure au $\min(n_s, 4n_r + n_s) = n_s$. La matrice d'identité fournit un ensemble complet de n_s équations linéairement indépendantes (i.e. $\text{rang}(BL_1) = n_s$).

Le deuxième bloc de ligne est :

$$BL_2 = \left[\overbrace{\left[\begin{array}{cccc|c} -\mathbf{G} & \mathbf{G} & \cdots & \mathbf{0} & \mathbf{0} \end{array} \right]}^{4n_r+n_s} \right]$$

Après la soustraction du premier bloc, les positions relatives et la différence de dé-

calage d'horloge entre R_1 et R_2 seront les 4 variables d'état $\begin{pmatrix} \hat{x}_2 - \hat{x}_1 \\ \hat{y}_2 - \hat{y}_1 \\ \hat{z}_2 - \hat{z}_1 \\ \hat{d}_2 - \hat{d}_1 \end{pmatrix}$ dans le

système. En effet, chaque bloc postérieur au deuxième représente une simple différence entre 2 récepteurs ce qui annule les erreurs communes b^j . Ce bloc représente 4 équations linéairement indépendantes. $\text{Rang}(BL_2) = 4$.

De même, on détermine les rangs de l'empilement des blocs de ligne $BL_{i=2 \dots n_r}$.

$$BL_{2 \dots n_r} = \overbrace{\begin{bmatrix} -\mathbf{G} & \mathbf{G} & \cdots & \mathbf{0} & | & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & | & \vdots \\ -\mathbf{G} & \mathbf{0} & \cdots & \mathbf{G} & | & \mathbf{0} \end{bmatrix}}^{4n_r + n_s},$$

Chaque bloc ligne $BL_{2 \dots n_r}$ contient 4 équations linéairement indépendantes et on a $n_r - 1$ blocs donc le $\text{rang}(BL_{2 \dots n_r}) = 4(n_r - 1)$.

d'où le rang de la matrice globale :

$$\text{rang}(\mathbf{G}_{sd}) = 4(n_r - 1) + n_s \quad (2.26)$$

Le rang de \mathbf{G}_{sd} est donc déficient et le nombre d'équations linéairement indépendantes est égal à $4(n_r - 1) + n_s$ ce que n'est pas suffisant pour résoudre le nombre total des inconnues ($4n_r + n_s$).

Ce calcul du rang a été validé par l'implémentation de quelques exemples sur Matlab en utilisant des mesures réelles de pseudodistances.

Exemple 1. Soit $n_r = 2$ et $n_s = 8$. $n_M = n_s n_r = 16$ et $n_I = 4n_r + n_s = 16$.

Soit :

$$\mathbf{G} = \begin{bmatrix} 0.3159 & 0.0958 & -0.9439 & 1 \\ -0.6119 & 0.2898 & -0.7358 & 1 \\ -0.2670 & 0.4422 & -0.8562 & 1 \\ 0.6536 & -0.4781 & -0.5866 & 1 \\ 0.1273 & 0.6840 & -0.7182 & 1 \\ 0.0547 & 0.9017 & -0.4288 & 1 \\ -0.1972 & 0.0516 & -0.9789 & 1 \\ -0.1766 & 0.9644 & -0.1967 & 1 \end{bmatrix}$$

La matrice \mathbf{G} est :

$$\mathbf{G} = \underset{16}{=} \left\{ \begin{bmatrix} \overbrace{\infty \{ \mathbf{G} \}}^4 & \mathbf{0} & \overbrace{\infty \{ \mathbf{I} \}}^8 \\ \mathbf{0} & \overbrace{\infty \{ \mathbf{G} \}}^4 & \overbrace{\infty \{ \mathbf{I} \}}^8 \end{bmatrix} \right.$$

$$\mathcal{G} =_{20} \left\{ \left[\begin{array}{cc|c} \overbrace{\mathbf{G}}^4 & \mathbf{0} & \overbrace{\mathbf{I}}^{10} \\ \mathbf{0} & \overbrace{\mathbf{G}}^4 & \overbrace{\mathbf{I}}^{10} \end{array} \right] \right\}$$

$$\mathcal{G}_{sd} =_{20} \left\{ \left[\begin{array}{cc|c} \overbrace{\mathbf{G}}^4 & \mathbf{0} & \overbrace{\mathbf{I}}^{10} \\ \overbrace{-\mathbf{G}}^4 & \overbrace{\mathbf{G}}^4 & \mathbf{0} \end{array} \right] \right\}$$

$$\mathcal{G}_{sd} =_{20} \left\{ \left[\begin{array}{ccc|c} \overbrace{\begin{matrix} -0.2138 & 0.0659 & -0.9746 & 1 \\ 0.3000 & 0.0816 & -0.9504 & 1 \\ -0.6209 & 0.3044 & -0.7222 & 1 \\ -0.1776 & 0.9679 & -0.1774 & 1 \\ -0.2763 & 0.4588 & -0.8444 & 1 \\ 0.6547 & -0.4626 & -0.5977 & 1 \\ 0.1248 & 0.7002 & -0.7029 & 1 \\ 0.0531 & 0.8927 & -0.4474 & 1 \\ 0.9645 & 0.1660 & -0.2050 & 1 \\ -0.9789 & -0.0536 & -0.1968 & 1 \end{matrix}}^4 & \mathbf{0} & \overbrace{\mathbf{I}}^{10} \\ \overbrace{\begin{matrix} 0.2138 & -0.0659 & 0.9746 & -1 \\ -0.3000 & -0.0816 & 0.9504 & -1 \\ 0.6209 & -0.3044 & 0.7222 & -1 \\ 0.1776 & -0.9679 & 0.1774 & -1 \\ 0.2763 & -0.4588 & 0.8444 & -1 \\ -0.6547 & 0.4626 & 0.5977 & -1 \\ -0.1248 & -0.7002 & 0.7029 & -1 \\ -0.0531 & -0.8927 & 0.4474 & -1 \\ -0.9645 & -0.1660 & 0.2050 & -1 \\ 0.9789 & 0.0536 & 0.1968 & -1 \end{matrix}}^4 & \overbrace{\begin{matrix} -0.2138 & 0.0659 & -0.9746 & 1 \\ 0.3000 & 0.0816 & -0.9504 & 1 \\ -0.6209 & 0.3044 & -0.7222 & 1 \\ -0.1776 & 0.9679 & -0.1774 & 1 \\ -0.2763 & 0.4588 & -0.8444 & 1 \\ 0.6547 & -0.4626 & -0.5977 & 1 \\ 0.1248 & 0.7002 & -0.7029 & 1 \\ 0.0531 & 0.8927 & -0.4474 & 1 \\ 0.9645 & 0.1660 & -0.2050 & 1 \\ -0.9789 & -0.0536 & -0.1968 & 1 \end{matrix}}^4 & \mathbf{0} \end{array} \right\}$$

Quand on utilise la fonction « rank » de Matlab le résultat obtenu est de 14.

Avec la formule $rank(\mathcal{G}) = rank(\mathcal{G}_{sd}) = 4(n_r - 1) + n_s = 14$. On obtient donc le même résultat (le rang de \mathcal{G} est inférieur au nombre des inconnues $n_I = 18$ et le système n'est donc pas observable).

En effet, quel que soit le nombre de mesures, il n'est pas possible de résoudre (2.21)

pour les différents états de $\hat{\mathbf{x}}$ parce que la matrice \mathcal{G} n'est pas de rang plein.

On peut se demander si la non observabilité est dû au fait que les \mathbf{G} sont quasiment les mêmes, si les robots ne sont pas à proximité et que chacun d'eux est distant des autres (de plus de $10km$). Peut-on augmenter le nombre des équations linéairement indépendants dans \mathcal{G} ?

2.3.1.2 Robots distants

Reprenons la même étude sous l'hypothèse que la distance entre les robots est supérieure à $10km$. Dans ce cas, les matrices \mathbf{G} ne sont pas les mêmes car les lignes de vue de satellites \mathbf{u}^j sont différentes.

La matrice jacobienne de notre système linéarisé (2.21) est celle de l'équation (2.23) :

$$\mathcal{G} = \left[\begin{array}{cccc|c} \mathbf{G}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{G}_2 & \cdots & \mathbf{0} & \mathbf{I} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_{n_r} & \mathbf{I} \end{array} \right]$$

Étudions le rang de \mathcal{G} .

En soustrayant le premier bloc de ligne de toutes les autres rangées de blocs, on obtient la matrice \mathcal{G}_{sd} :

$$\mathcal{G}_{sd} = \left[\begin{array}{cccc|c} \mathbf{G}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ -\mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\mathbf{G}_1 & \mathbf{0} & \cdots & \mathbf{G}_{n_r} & \mathbf{0} \end{array} \right]. \quad (2.27)$$

Pour trouver le rang de cette matrice, il suffit d'additionner le rang de chaque bloc de ligne BL_i .

Comme précédemment, le rang de BL_1 est :

$$\text{rang}(BL_1) = \overbrace{\left[\begin{array}{cccc|c} \mathbf{G}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{array} \right]}^{4n_r+n_s} = n_s$$

Le rang du deuxième bloc de ligne ne vaut pas 4 cette fois, car les satellites ont des lignes de vue différentes. Ce bloc représente une simple différence entre 2 récepteurs ce qui annule les erreurs communes b^j .

$$BL_2 = \left[\begin{array}{cccc|c} \overbrace{\hspace{10em}}^{4n_r+n_s} & & & & \\ -\mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{0} & \mathbf{0} \end{array} \right]$$

Raisonnons sur les colonnes. On a $\mathbf{G}_1 \neq \mathbf{G}_2$. \mathbf{G}_1 et \mathbf{G}_2 possèdent chacune 4 colonnes dont une colonne de 1. Donc le rang(BL_2) = 4 + 4 - 1 = 7. Donc,

$$\text{rang}(BL_2) = 7$$

Considérons maintenant les blocs $BL_{i=3\dots n_r}$:

$$BL_{3\dots n_r} = \left[\begin{array}{cccc|c} \overbrace{\hspace{10em}}^{4n_r+n_s} & & & & \\ -\mathbf{G}_1 & \mathbf{0} & \mathbf{G}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \mathbf{0} & \ddots & \vdots \\ -\mathbf{G}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_{n_r} \\ \hline & & & & \mathbf{0} \end{array} \right].$$

Le rang de chaque bloc $BL_{i=3\dots n_r}$ est égal à 4 car le nombre de colonnes de \mathbf{G}_1 linéairement indépendantes ont déjà été prises en compte dans le calcul du rang de BL_2 .

Ainsi, le rang($BL_{3\dots n_r}$) = 4($n_r - 2$).

Au total, le rang est :

$$\text{rang}(\mathbf{G}_{sd}) = 4(n_r - 2) + 7 + n_s = 4n_r + n_s - 1 \quad (2.28)$$

Même si le rang est supérieur au cas des robots proches, la matrice \mathbf{G}_{sd} a encore un rang déficient puisqu'on n'obtient pas $4n_r + n_s$ et donc le système n'est donc pas observable.

Exemple 3. Soit $n_r = 2$ et $n_s = 8$. $n_M = n_s n_r = 16$ et $n_I = 4n_r + n_s = 16$.

soit :

$$\mathbf{G}_1 = \begin{bmatrix} 0.3159 & 0.0958 & -0.9439 & 1 \\ -0.6119 & 0.2898 & -0.7358 & 1 \\ -0.2670 & 0.4422 & -0.8562 & 1 \\ 0.6536 & -0.4781 & -0.5866 & 1 \\ 0.1273 & 0.6840 & -0.7182 & 1 \\ 0.0547 & 0.9017 & -0.4288 & 1 \\ -0.1972 & 0.0516 & -0.9789 & 1 \\ -0.1766 & 0.9644 & -0.1967 & 1 \end{bmatrix}$$

$$\mathbf{G}_2 = \begin{bmatrix} -0.1969 & 0.05139 & -0.9790 & 1 \\ 0.3161 & 0.0960 & -0.9438 & 1 \\ -0.6117 & 0.2895 & -0.7361 & 1 \\ -0.1766 & 0.9643 & -0.1970 & 1 \\ -0.2668 & 0.4419 & -0.8564 & 1 \\ -0.6038 & -0.7499 & -0.2699 & 1 \\ 0.12737 & 0.6837 & -0.7185 & 1 \\ 0.0548 & 0.9018 & -0.4284 & 1 \end{bmatrix}$$

d'où

$$\mathcal{G} =_{16} \left\{ \left[\begin{array}{cc|c} \overbrace{\mathbf{G}_1}^4 & \mathbf{0} & \overbrace{\mathbf{I}}^8 \\ \mathbf{0} & \underbrace{\mathbf{G}_2}_4 & \underbrace{\mathbf{I}}_8 \end{array} \right] \right\}$$

$$\mathcal{G}_{sd} =_{16} \left\{ \left[\begin{array}{cc|c} \overbrace{\mathbf{G}_1}^4 & \mathbf{0} & \overbrace{\mathbf{I}}^8 \\ \underbrace{-\mathbf{G}_1}_4 & \underbrace{\mathbf{G}_2}_4 & \mathbf{0} \end{array} \right] \right\}$$

$$\mathcal{G}_{sd} = \begin{bmatrix}
 \begin{array}{c|cccc}
 \hline
 \begin{array}{cccc}
 0.3159 & 0.0958 & -0.9439 & 1 \\
 -0.6119 & 0.2898 & -0.7358 & 1 \\
 -0.2670 & 0.4422 & -0.8562 & 1 \\
 0.6536 & -0.4781 & -0.5866 & 1 \\
 0.1273 & 0.6840 & -0.7182 & 1 \\
 0.0547 & 0.9017 & -0.4288 & 1 \\
 -0.1972 & 0.0516 & -0.9789 & 1 \\
 -0.1766 & 0.9644 & -0.1967 & 1 \\
 \hline
 \end{array}
 & \begin{array}{c} 4 \\ \hline \end{array}
 & \begin{array}{c} 0 \\ \hline \end{array}
 & \begin{array}{c} 8 \\ \hline \mathbf{I} \\ \hline \end{array} \\
 \hline
 \begin{array}{cccc}
 -0.3159 & -0.0958 & 0.9439 & -1 \\
 0.6119 & -0.2898 & 0.7358 & -1 \\
 0.2670 & -0.4422 & 0.8562 & -1 \\
 -0.6536 & 0.4781 & 0.5866 & -1 \\
 -0.1273 & -0.6840 & 0.7182 & -1 \\
 -0.0547 & -0.9017 & 0.4288 & -1 \\
 0.1972 & -0.0516 & 0.9789 & -1 \\
 0.1766 & -0.9644 & 0.1967 & -1 \\
 \hline
 \end{array}
 & \begin{array}{c} 4 \\ \hline \end{array}
 & \begin{array}{c} 4 \\ \hline \end{array}
 & \begin{array}{c} 0 \\ \hline \end{array} \\
 \hline
 \end{array}
 \end{bmatrix}$$

Quand on utilise la fonction « rank » de Matlab le résultat obtenu est de 15.

Avec la formule $rank(\mathcal{G}) = rank(\mathcal{G}_{sd}) = 4n_r + n_s - 1 = 15$. On obtient donc le même résultat (le rang de \mathcal{G} reste toujours inférieur au nombre des inconnues $n_I = 16$ même si le nombre des équations linéairement indépendantes est augmenté).

On peut maintenant conclure, quel que soit le nombre de mesures et quel que soit la distance entre les robots, le système est non observable. Dans le cas statique, l'observation de toutes les erreurs communes de satellites n'est pas possible en utilisant uniquement des mesures brutes de GNSS (pseudodistances). Rappelons que Rife *et al.* ont apporté une solution à ce problème [89] en ajoutant des mesures de caméra [90].

Supposons qu'un satellite soit sans biais alors on n'estime que $n_s - 1$ biais et le système devient observable. On ne considère dans le système présenté dans (2.19) que $n_s - 1$ biais i.e :

$$\hat{\mathbf{x}} = [\hat{\mathbf{p}}_1, \hat{d}_1, \dots, \hat{\mathbf{p}}_{n_r}, \hat{d}_{n_r}, \hat{b}^1, \dots, \hat{b}^{n_s-1}]^T. \quad (2.29)$$

Le rang de la matrice \mathcal{G} est toujours égal à $4n_r + n_s - 1$. Le nombre des états du système (2.29) est égal à $4n_r + n_s - 1$. Le système devient observable en observant un biais de moins. De même, dans le cas où les robots sont proches, le système devient observable en n'observant que $n_s - 4$ biais.

La question qui se pose maintenant est la suivante : est ce que le système complet est observable dans le cas dynamique ?

En rajoutant des informations sur l'évolution de l'état du système au cours du temps, le système complet peut-il être observable ? En effet un système qui n'est pas observable en statique, peut l'être en dynamique.

2.3.2 Cas dynamique

Le critère de l'observabilité au sens du rang à base de dérivées de Lie [91] est utilisé. Ci-après, une étude à base de ce critère est détaillée, pour déterminer sous quelles conditions le système est localement faiblement observable. Dans le cadre de systèmes robotiques coopératifs, plusieurs tests d'observabilité non-linéaire ont été basés sur le critère de dérivées de Lie. Par exemple, *Mariottini et al.* [92] ont utilisé ce critère afin de démontrer l'observabilité d'un système de formation de leader suiveurs en 2D, en utilisant uniquement des mesures de distances. *Martinelli et Siegwart* [93] ont également utilisé le même test pour déterminer les conditions suffisantes pour l'observabilité d'un système de localisation coopérative de robots mobiles naviguant en 2D. Récemment, *Zhou et al.* [94] ont employé les dérivées de Lie pour étudier l'observabilité de la pose relative en 2D d'un système coopératif de 2 robots. Ils ont décrit l'évolution de l'état du système au cours du temps et le modèle de mesures de l'inter-distance. Ils ont pu déterminer les conditions pour que l'état de leur système soit observable. En toute rigueur, on ne sait démontrer que les pertes d'observabilité.

Dans la suite, un bref examen des concepts de base de l'observabilité et les dérivées de Lie sont d'abord présentés. Ensuite, une étude de l'observabilité de la localisation coopérative avec GNSS basée sur les dérivée de Lie est considérée.

La question à laquelle on cherche une réponse est :

Est-ce qu'il existe des critères d'observabilité nécessaires et suffisants sous lesquels le système peut être localement faiblement observable ?

2.3.2.1 Observabilité des systèmes non-linéaires

Considérons la représentation d'état du système non-linéaire (S) suivant [95] :

$$(S) = \begin{cases} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}), & \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}) \end{cases} \quad (2.30)$$

où $\mathbf{x} \in M \subseteq \mathbb{R}^n$ (M , une variété de classe C^∞ de dimension n) est le vecteur d'état, $\mathbf{u} = [u_1 \dots u_l]^T \in \mathbb{R}^l$ est le vecteur d'entrées, et $\mathbf{y} = [y_1 \dots y_m]^T \in \mathbb{R}^m$ est le vecteur de mesures, où :

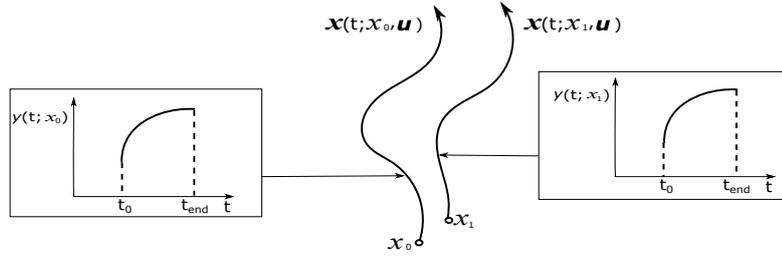


Figure 2.3 – Deux états indiscernables (\mathbf{x}_0 et \mathbf{x}_1) qui produisent la même sortie

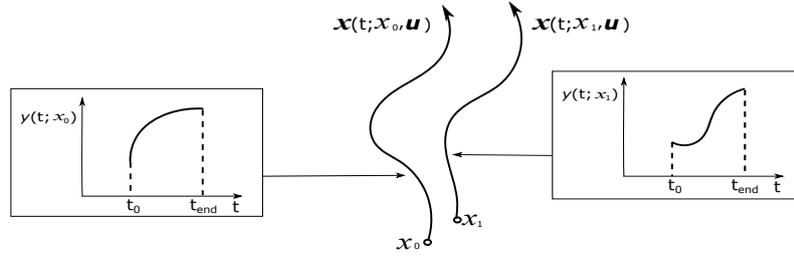


Figure 2.4 – Illustration de l'observabilité : les sorties sont différentes

$$y_k = h_k(\mathbf{x}), \quad k = 1, \dots, m.$$

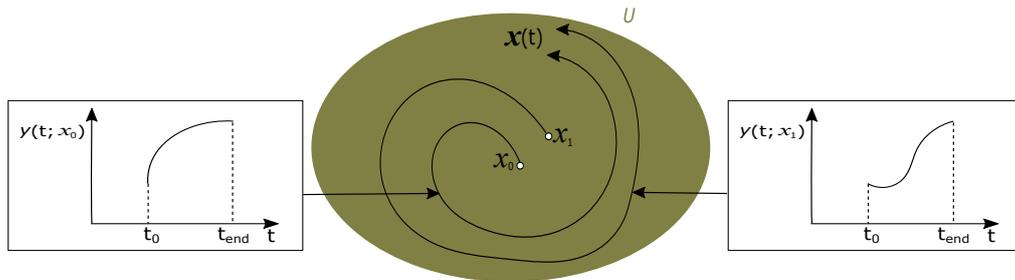
Définition 1. Deux états initiaux \mathbf{x}_0 et \mathbf{x}_1 sont indiscernables si en donnant la même entrée $u(t)$, le système (S) produit la même sortie $y(t)$ pour les deux états initiaux \mathbf{x}_0 et \mathbf{x}_1 (Figure (2.3)). On dit que le système est *observable* si pour tout $\mathbf{x} \in M$, le seul état indiscernable de \mathbf{x} est \mathbf{x} lui-même (Figure (2.4)).

Notons que l'observabilité sus-citée est un concept global. Cependant, il peut être nécessaire pour l'état de voyager une longue distance ou pendant une longue période afin de distinguer deux points de M . Pour cette raison, le concept local suivant est introduit [95].

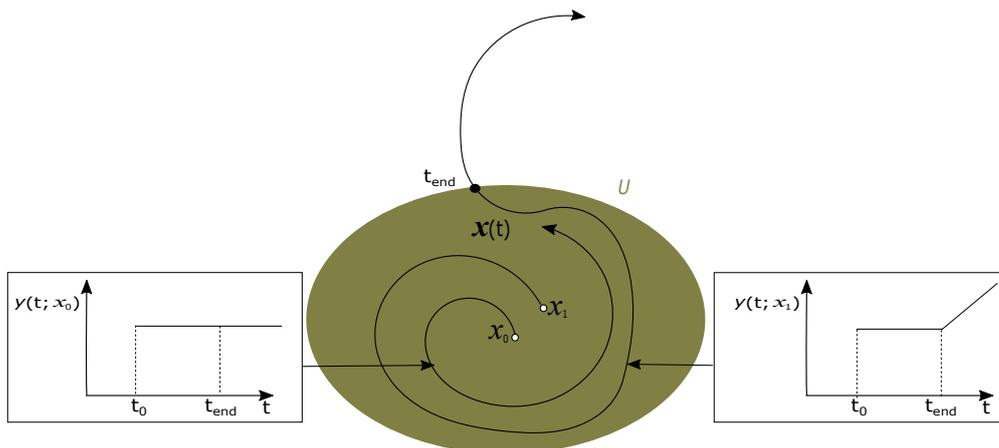
Définition 2. Le système (S) est *localement observable* en $\mathbf{x}_0 \in M$, si pour tout voisinage ouvert U de \mathbf{x}_0 , l'ensemble des points indiscernables de \mathbf{x}_0 par des trajectoires dans U est \mathbf{x}_0 lui-même. (S) est *localement observable* s'il est localement observable pour tout $\mathbf{x} \in M$ (Figure (2.5)).

Dans le cas pratique, il est souvent suffisant de distinguer \mathbf{x}_0 seulement de ses voisins (e.g. quand une certaine connaissance préalable de \mathbf{x}_0 est disponible). Cependant, il est possible que \mathbf{x}_0 soit indiscernable des autres états qui sont loin. Cela conduit à la notion de faible observabilité [95].

Définition 3. (S) est *faiblement observable* en \mathbf{x}_0 s'il existe un voisinage ouvert V de \mathbf{x}_0 tel que le seul point dans V qui est indiscernable de \mathbf{x}_0 est \mathbf{x}_0 lui-même. Le système (S) est *faiblement observable* s'il est faiblement observable pour tout $\mathbf{x} \in M$ (Figure (2.6)).

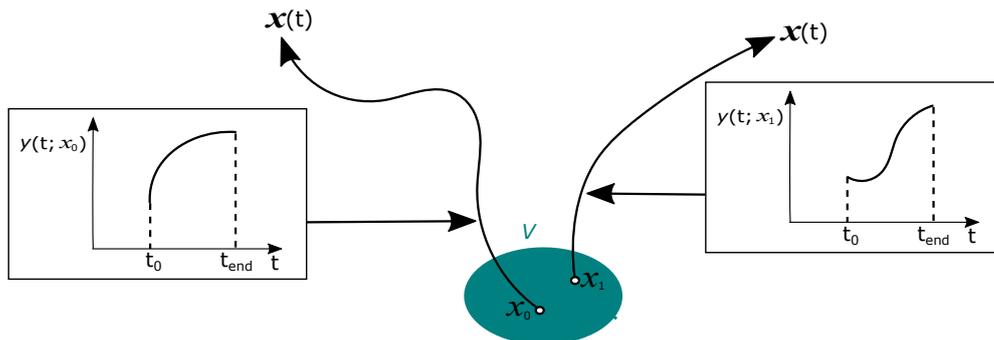


(a) Deux états (x_0 et x_1) localement observables : les sorties sont différentes dans le voisinage.

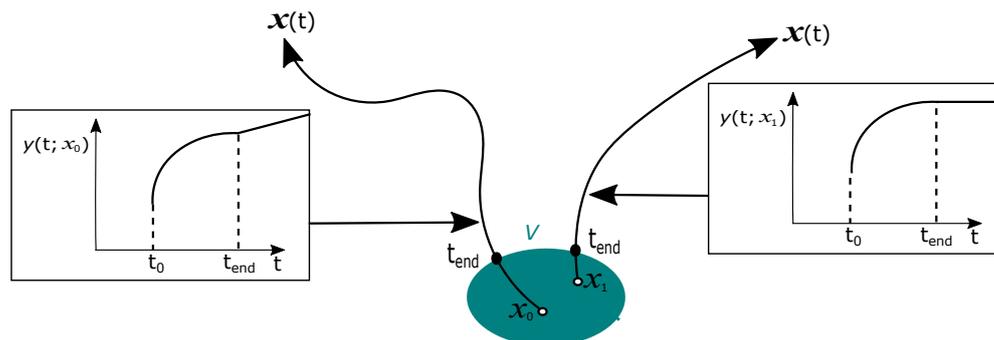


(b) Deux états (x_0 et x_1) non localement observables : les sorties sont identiques mais différent où bout d'un moment.

Figure 2.5 – Illustration de l'observabilité locale



(a) Deux états (x_0 et x_1) faiblement observables



(b) x_0 et x_1 non faiblement observables (x_0 et x_1 sont non discernables dans V , x_0 et x_1 sont discernables en dehors de V)

Figure 2.6 – Illustration de l'observabilité faible

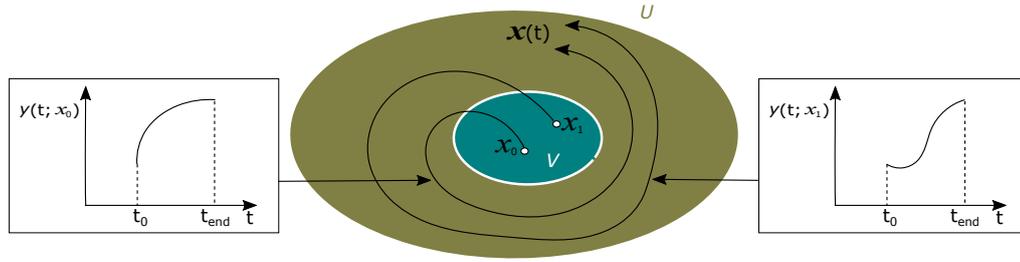
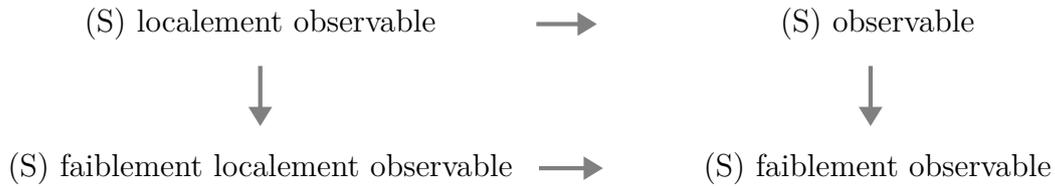


Figure 2.7 – Illustration de l'observabilité faible locale : deux états initiaux très proches engendrent des sorties différentes

Définition 4. (S) est *localement faiblement observable* en \mathbf{x}_0 s'il existe un voisinage ouvert U de \mathbf{x}_0 tel que pour tout voisinage ouvert V de \mathbf{x}_0 contenu dans U , l'ensemble des points indiscernables de \mathbf{x}_0 dans U par des trajectoires dans V est \mathbf{x}_0 lui-même. Le système (S) est *localement faiblement observable* s'il est localement faiblement observable pour tout $\mathbf{x} \in M$ (Figure (2.7)).

Les diverses notions de l'observabilité peuvent être reliées de la manière suivante :



En général, il n'y a pas d'autres implications. Pour les systèmes linéaires, il peut être démontré que les quatre notions sont équivalentes. L'avantage de l'observabilité faible locale par rapport aux autres concepts est qu'il possède un test algébrique simple : ce sont les dérivées de Lie qui sont décrites par la suite.

2.3.2.2 Dérivée de Lie et observabilité non-linéaire au sens du rang

Considérons le cas particulier du système (2.30) où la fonction de processus, \mathbf{f} , peut être présentée par une sommation des fonctions indépendantes, dont chacune est multipliée par un composant différent du vecteur d'entrées de la commande comme suit :

$$\begin{cases} \dot{\mathbf{x}} = f_0(\mathbf{x}) + f_1(\mathbf{x}) u_1 + \cdots + f_l(\mathbf{x}) u_l \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases} \quad (2.31)$$

où f_0 est la fonction d'entrée zéro du modèle de processus. La dérivée d'ordre zéro de Lie de toute fonction (scalaire) est la fonction elle-même, *i.e.*, $\mathcal{L}_0 h_k = h_k$. Le

premier ordre de la dérivée de Lie de la fonction h_k dans la direction de f_i est défini comme suit :

$$\begin{aligned}\mathcal{L}_{f_i}^1 h_k(\mathbf{x}) &= \frac{\partial h(\mathbf{x})}{\partial x_1} f_{i1}(X) + \dots + \frac{\partial h(\mathbf{x})}{\partial x_n} f_{in}(\mathbf{x}) \\ &= \nabla h_k(\mathbf{x}) \cdot f_i(\mathbf{x})\end{aligned}\quad (2.32)$$

où $f_i(X) = [f_{i1} \dots f_{in}]^T$, ∇ représente l'opérateur de gradient et $\nabla h_k \cdot f_i$ est le produit scalaire du gradient ∇h_k de h_k avec f_i .

Le second ordre de la dérivée de Lie de la fonction h_k est le suivant :

$$\begin{aligned}\mathcal{L}_{f_i}^2 h_k(\mathbf{x}) &= \mathcal{L}_{f_i}^1(\mathcal{L}_{f_i}^1 h_k(\mathbf{x})) \\ &= \nabla \mathcal{L}_{f_i}^1 h_k(\mathbf{x}) \cdot f_i(\mathbf{x})\end{aligned}\quad (2.33)$$

Les dérivées d'ordre supérieur de Lie sont calculées de la même façon. La matrice d'observabilité du système est la matrice \mathcal{O} de lignes :

$$\mathcal{O} \triangleq \{\nabla \mathcal{L}_{f_i}^l h_k(\mathbf{x}) \mid i = 0 \dots l, k = 1 \dots m; l \in \mathbb{N}\} \quad (2.34)$$

Maintenant, nous sommes en mesure de donner une définition "faible locale" de la notion d'observabilité comme dans [91] (Théorème 2.3.2.2, 2.3.2.2). Ces conditions d'observabilité sont basées sur l'observabilité au sens du rang.

Définition 5. (*Condition de rang de l'observabilité*) : La condition de rang de l'observabilité est satisfaite si la matrice d'observabilité est de rang plein.

Théorème 1. (*Condition suffisante de l'observabilité*) : Si le système dynamique (S) satisfait la condition de rang de l'observabilité alors il est faiblement localement observable.

Théorème 2. Si le système dynamique (S) est faiblement localement observable alors la condition d'observabilité est satisfaite.

La condition de rang de l'observabilité est donc nécessaire et suffisante pour l'observabilité faible locale.

L'étude de l'observabilité au sens de rang devient considérablement plus facile si l'expression de la fonction h est simple.

2.3.2.3 Observabilité de la localisation coopérative avec des mesures GNSS

Cas1. Considérons 2 robots (i.e. $n_r = 2$) et 4 biais (i.e. $n_s = 4$) dans le système représenté dans la figure (2.8). \mathbf{p}_{R_1} et \mathbf{p}_{R_2} représentent respectivement le vecteur

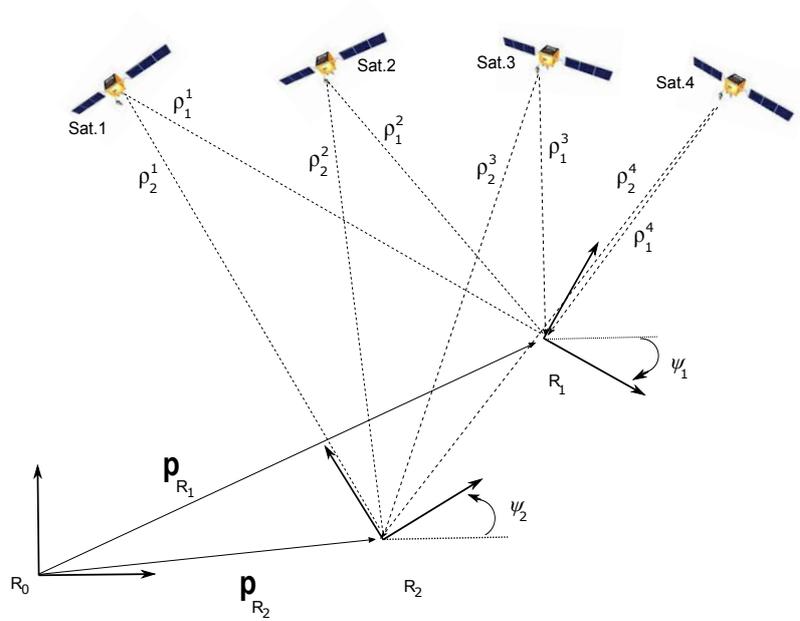


Figure 2.8 – Système coopératif de 2 robots avec 4 mesures de GNSS. Repère fixe (R_0) et repères mobiles (R_1 et R_2).

position \mathbf{p}_1 du robot 1 et le vecteur position \mathbf{p}_2 du robot 2 dans le repère du travail R_0 . La figure (2.8) présente des distortions rendant les pseudodistances figurant non conformes aux dimensions réelles.

Le vecteur d'état du système considéré est :

$$\hat{\mathbf{x}} = [\hat{\mathbf{p}}_1 \ \hat{d}_1 \ \hat{\mathbf{p}}_2 \ \hat{d}_2 \ \hat{b}^1 \ \hat{b}^2 \ \hat{b}^3 \ \hat{b}^4]^T \quad (2.35)$$

avec $\dim(\hat{\mathbf{x}})=12$.

où $\hat{\mathbf{p}}_1 = [\hat{x}_1, \hat{y}_1, \hat{z}_1]$, $\hat{\mathbf{p}}_2 = [\hat{x}_2, \hat{y}_2, \hat{z}_2]$, \hat{d}_1 et \hat{d}_2 représentent le décalage d'horloge du robot 1 et 2. ($\hat{b}^{1 \dots 4}$) représentent les biais sur les mesures de pseudodistances de quatre satellites en commun entre les 2 robots.

En 2D, le modèle d'évolution de chaque robot est :

$$\begin{cases} \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} = \begin{bmatrix} v_i \cos(\psi_i) \\ v_i \sin(\psi_i) \\ 0 \end{bmatrix} \\ \dot{d}_i = dr_i \end{cases} \quad (2.36)$$

où dr_i présente la dérive de l'horloge pour chaque récepteur qui sera estimée. On

suppose qu'il est possible de mesurer le cap ψ_i (avec une boussole par exemple).

Le modèle d'évolution des biais est :

$$\dot{b}^j = a b^j \quad (2.37)$$

d'où a est une constante du temps du modèle auto-régressif des biais ($a \neq 0$).

Considérons $u = [v_1 \ v_2]^T$ l'entrée du système (2.35). Le modèle d'évolution du système (2.35) peut être écrit selon (2.31) comme suit :

$$\dot{X} = \begin{array}{c} \overbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ dr_1 \\ 0 \\ 0 \\ 0 \\ dr_2 \\ ab^1 \\ ab^2 \\ ab^3 \\ ab^4 \end{bmatrix}}^{f_0} + \begin{array}{c} \overbrace{\begin{bmatrix} \cos(\psi_1) \\ \sin(\psi_1) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}^{f_1} \\ v_1 + \end{array} \begin{array}{c} \overbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \cos(\psi_2) \\ \sin(\psi_2) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}^{f_2} \\ v_2 \end{array} \quad (2.38)$$

Le modèle d'observation est :

$$\mathbf{y} = \begin{bmatrix} \sqrt{(x_1 - x^1)^2 + (y_1 - y^1)^2 + (z_1 - z^1)^2} + d_1 + b^1 \\ \vdots \\ \sqrt{(x_1 - x^1)^2 + (y_1 - y^1)^2 + (z_1 - z^1)^2} + d_1 + b^4 \\ \sqrt{(x_2 - x^1)^2 + (y_2 - y^1)^2 + (z_2 - z^1)^2} + d_2 + b^1 \\ \vdots \\ \sqrt{(x_2 - x^1)^2 + (y_2 - y^1)^2 + (z_2 - z^1)^2} + d_2 + b^4 \end{bmatrix} \quad (2.39)$$

Calculons ci-après les dérivées de Lie nécessaires de \mathbf{h} et leurs gradients :

— L'ordre zéro de la dérivée de Lie :

$$\mathcal{L}^0 \mathbf{h}(X) = \mathbf{h}(X) = \begin{bmatrix} \|\mathbf{p}_1 - \mathbf{p}^1\| + d_1 + b^1 \\ \|\mathbf{p}_1 - \mathbf{p}^2\| + d_1 + b^2 \\ \|\mathbf{p}_1 - \mathbf{p}^3\| + d_1 + b^3 \\ \|\mathbf{p}_1 - \mathbf{p}^4\| + d_1 + b^4 \\ \|\mathbf{p}_2 - \mathbf{p}^1\| + d_2 + b^1 \\ \|\mathbf{p}_2 - \mathbf{p}^2\| + d_2 + b^2 \\ \|\mathbf{p}_2 - \mathbf{p}^3\| + d_2 + b^3 \\ \|\mathbf{p}_2 - \mathbf{p}^4\| + d_2 + b^4 \end{bmatrix}$$

où $\|\mathbf{p}_i - \mathbf{p}^j\| = \sqrt{(x_i - x^j)^2 + (y_i - y^j)^2 + (z_i - z^j)^2}$, $i = \{1, 2\}$ et $j = \{1, 2, 3, 4\}$.

le gradient de $\mathcal{L}^0 \mathbf{h}$:

$$\begin{aligned} \nabla \mathcal{L}^0 \mathbf{h} &= \text{jacobian}(\mathbf{h}) \\ &= \begin{bmatrix} \frac{x_1 - x^1}{\|\mathbf{p}_1 - \mathbf{p}^1\|} & \frac{y_1 - y^1}{\|\mathbf{p}_1 - \mathbf{p}^1\|} & \frac{z_1 - z^1}{\|\mathbf{p}_1 - \mathbf{p}^1\|} & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{x_1 - x^2}{\|\mathbf{p}_1 - \mathbf{p}^2\|} & \frac{y_1 - y^2}{\|\mathbf{p}_1 - \mathbf{p}^2\|} & \frac{z_1 - z^2}{\|\mathbf{p}_1 - \mathbf{p}^2\|} & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{x_1 - x^3}{\|\mathbf{p}_1 - \mathbf{p}^3\|} & \frac{y_1 - y^3}{\|\mathbf{p}_1 - \mathbf{p}^3\|} & \frac{z_1 - z^3}{\|\mathbf{p}_1 - \mathbf{p}^3\|} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{x_1 - x^4}{\|\mathbf{p}_1 - \mathbf{p}^4\|} & \frac{y_1 - y^4}{\|\mathbf{p}_1 - \mathbf{p}^4\|} & \frac{z_1 - z^4}{\|\mathbf{p}_1 - \mathbf{p}^4\|} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{x_2 - x^1}{\|\mathbf{p}_2 - \mathbf{p}^1\|} & \frac{y_2 - y^1}{\|\mathbf{p}_2 - \mathbf{p}^1\|} & \frac{z_2 - z^1}{\|\mathbf{p}_2 - \mathbf{p}^1\|} & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{x_2 - x^2}{\|\mathbf{p}_2 - \mathbf{p}^2\|} & \frac{y_2 - y^2}{\|\mathbf{p}_2 - \mathbf{p}^2\|} & \frac{z_2 - z^2}{\|\mathbf{p}_2 - \mathbf{p}^2\|} & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{x_2 - x^3}{\|\mathbf{p}_2 - \mathbf{p}^3\|} & \frac{y_2 - y^3}{\|\mathbf{p}_2 - \mathbf{p}^3\|} & \frac{z_2 - z^3}{\|\mathbf{p}_2 - \mathbf{p}^3\|} & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{x_2 - x^4}{\|\mathbf{p}_2 - \mathbf{p}^4\|} & \frac{y_2 - y^4}{\|\mathbf{p}_2 - \mathbf{p}^4\|} & \frac{z_2 - z^4}{\|\mathbf{p}_2 - \mathbf{p}^4\|} & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \infty \left\{ \begin{bmatrix} \widehat{\mathbf{G}}_1 & \mathbf{0} & \widehat{\mathbf{I}} \\ \mathbf{0} & \widehat{\mathbf{G}}_2 & \widehat{\mathbf{I}} \end{bmatrix} \right\} \end{aligned} \tag{2.40}$$

— 1^{er} ordre de la dérivée de Lie :

$$\mathcal{L}_{f_0}^1 \mathbf{h} = \nabla \mathcal{L}^0 \mathbf{h} \cdot f_0 = \begin{bmatrix} a b^1 + dr_1 \\ a b^2 + dr_1 \\ a b^3 + dr_1 \\ a b^4 + dr_1 \\ a b^1 + dr_2 \\ a b^2 + dr_2 \\ a b^3 + dr_2 \\ a b^4 + dr_2 \end{bmatrix}$$

$$\mathcal{L}_{f_1}^1 \mathbf{h} = \nabla \mathcal{L}^0 \mathbf{h} \cdot f_1 = \begin{bmatrix} \frac{x_1 \cos(\psi_1) - x^1 \cos(\psi_1) + y_1 \sin(\psi_1) - y^1 \sin(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^1\|} \\ \frac{x_1 \cos(\psi_1) - x^2 \cos(\psi_1) + y_1 \sin(\psi_1) - y^2 \sin(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^2\|} \\ \frac{x_1 \cos(\psi_1) - x^3 \cos(\psi_1) + y_1 \sin(\psi_1) - y^3 \sin(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^3\|} \\ \frac{x_1 \cos(\psi_1) - x^4 \cos(\psi_1) + y_1 \sin(\psi_1) - y^4 \sin(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^4\|} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathcal{L}_{f_2}^1 \mathbf{h} = \nabla \mathcal{L}^0 \mathbf{h} \cdot f_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{x_2 \cos(\psi_2) - x^1 \cos(\psi_2) + y_2 \sin(\psi_2) - y^1 \sin(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^1\|} \\ \frac{x_2 \cos(\psi_2) - x^2 \cos(\psi_2) + y_2 \sin(\psi_2) - y^2 \sin(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^2\|} \\ \frac{x_2 \cos(\psi_2) - x^3 \cos(\psi_2) + y_2 \sin(\psi_2) - y^3 \sin(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^3\|} \\ \frac{x_2 \cos(\psi_2) - x^4 \cos(\psi_2) + y_2 \sin(\psi_2) - y^4 \sin(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^4\|} \end{bmatrix}$$

les gradients de $\mathcal{L}^1 \mathbf{h}$:

$$\nabla \mathcal{L}_{f_0}^1 \mathbf{h} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a \end{bmatrix}$$

$$\nabla \mathcal{L}_{f_1}^1 \mathbf{h} = \begin{bmatrix} \frac{\cos(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^1\|} - \frac{(x_1 - x^1) A_1^1}{\|\mathbf{p}_1 - \mathbf{p}^1\|^3} & \frac{\sin(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^1\|} - \frac{(y_1 - y^1) A_1^1}{\|\mathbf{p}_1 - \mathbf{p}^1\|^3} & -\frac{(z_1 - z^1) A_1^1}{\|\mathbf{p}_1 - \mathbf{p}^1\|^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\cos(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^2\|} - \frac{(x_1 - x^2) A_1^2}{\|\mathbf{p}_1 - \mathbf{p}^2\|^3} & \frac{\sin(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^2\|} - \frac{(y_1 - y^2) A_1^2}{\|\mathbf{p}_1 - \mathbf{p}^2\|^3} & -\frac{(z_1 - z^2) A_1^2}{\|\mathbf{p}_1 - \mathbf{p}^2\|^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\cos(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^3\|} - \frac{(x_1 - x^3) A_1^3}{\|\mathbf{p}_1 - \mathbf{p}^3\|^3} & \frac{\sin(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^3\|} - \frac{(y_1 - y^3) A_1^3}{\|\mathbf{p}_1 - \mathbf{p}^3\|^3} & -\frac{(z_1 - z^3) A_1^3}{\|\mathbf{p}_1 - \mathbf{p}^3\|^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\cos(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^4\|} - \frac{(x_1 - x^4) A_1^4}{\|\mathbf{p}_1 - \mathbf{p}^4\|^3} & \frac{\sin(\psi_1)}{\|\mathbf{p}_1 - \mathbf{p}^4\|} - \frac{(y_1 - y^4) A_1^4}{\|\mathbf{p}_1 - \mathbf{p}^4\|^4} & -\frac{(z_1 - z^4) A_1^4}{\|\mathbf{p}_1 - \mathbf{p}^4\|^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\nabla \mathcal{L}_{f_2}^1 \mathbf{h} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\cos(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^1\|} - \frac{(x_2 - x^1) A_2^1}{\|\mathbf{p}_2 - \mathbf{p}^1\|^3} & \frac{\sin(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^1\|} - \frac{(y_2 - y^1) A_2^1}{\|\mathbf{p}_2 - \mathbf{p}^1\|^3} & -\frac{(z_2 - z^1) A_2^1}{\|\mathbf{p}_2 - \mathbf{p}^1\|^3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\cos(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^2\|} - \frac{(x_2 - x^2) A_2^2}{\|\mathbf{p}_2 - \mathbf{p}^2\|^3} & \frac{\sin(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^2\|} - \frac{(y_2 - y^2) A_2^2}{\|\mathbf{p}_2 - \mathbf{p}^2\|^3} & -\frac{(z_2 - z^2) A_2^2}{\|\mathbf{p}_2 - \mathbf{p}^2\|^3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\cos(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^3\|} - \frac{(x_2 - x^3) A_2^3}{\|\mathbf{p}_2 - \mathbf{p}^3\|^3} & \frac{\sin(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^3\|} - \frac{(y_2 - y^3) A_2^3}{\|\mathbf{p}_2 - \mathbf{p}^3\|^3} & -\frac{(z_2 - z^3) A_2^3}{\|\mathbf{p}_2 - \mathbf{p}^3\|^3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\cos(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^4\|} - \frac{(x_2 - x^4) A_2^4}{\|\mathbf{p}_2 - \mathbf{p}^4\|^3} & \frac{\sin(\psi_2)}{\|\mathbf{p}_2 - \mathbf{p}^4\|} - \frac{(y_2 - y^4) A_2^4}{\|\mathbf{p}_2 - \mathbf{p}^4\|^4} & -\frac{(z_2 - z^4) A_2^4}{\|\mathbf{p}_2 - \mathbf{p}^4\|^3} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

où $A_i^j = (x_i \cos(\psi_i) - x^j \cos(\psi_i) + y_i \sin(\psi_i) - y^j \sin(\psi_i))$, $i = \{1, 2\}$ et $j = \{1, 2, 3, 4\}$.

Lemme. 1 (conditions suffisantes) : Le système (2.38, 2.39) est faiblement localement observable si les conditions suivantes sont satisfaites :

- (i) $v_1 \neq 0$,
- (ii) $v_2 \neq 0$,
- (iii) $\psi_1 \neq \{0, \pi/2\}$ ou $\psi_2 \neq \{0, \pi/2\}$,
- (iv) $a \neq 0$.

Démonstration. Si les conditions (i)-(iv) sont tous satisfaites, la matrice d'observabilité est : □

$$\mathcal{O} = \begin{bmatrix} \nabla \mathcal{L}^0 \mathbf{h} \\ \nabla \mathcal{L}_{f_0}^1 \mathbf{h} \\ \nabla \mathcal{L}_{f_1}^1 \mathbf{h} \\ \nabla \mathcal{L}_{f_2}^1 \mathbf{h} \end{bmatrix} \quad (2.41)$$

Examinons le rang de chaque bloc de ligne de la matrice \mathcal{O} . $\text{Rang}(\nabla\mathcal{L}^0\mathbf{h})=8$ quelque soit la position des robots (distants ou proches). $\text{Rang}(\nabla\mathcal{L}_{f_0}^1\mathbf{h})=4$. On s'arrête ici car le rang de la matrice \mathcal{O} est déjà plein ($\text{Rang}(\mathcal{O})=12$).

Donc le système est faiblement localement observable.

Lemme. 2 (conditions nécessaires) : Le système (2.38,2.39) n'est pas faiblement localement observable si la condition suivante n'est pas satisfaite :

(iv) $a \neq 0$.

Démonstration. Si le modèle des biais est constant (i.e. $a = 0$). Le rang de la matrice \mathcal{O} est toujours déficient donc le système n'est pas faiblement localement observable. En effet le rang de la matrice \mathcal{O} dans ce cas est égal à 11. \square

$$\mathcal{O} = \begin{bmatrix} \nabla\mathcal{L}^0\mathbf{h} \\ \nabla\mathcal{L}_{f_1}^1\mathbf{h} \\ \nabla\mathcal{L}_{f_2}^1\mathbf{h} \end{bmatrix}. \quad (2.42)$$

En outre, les conditions (i)-(iii) ne sont pas nécessaires, puisque si l'une entre elles n'est pas satisfaite la matrice \mathcal{O} reste de rang plein.

Par exemple, la condition (i) est suffisante mais pas nécessaire. Si $v_1 = 0$, la matrice d'observabilité devient :

$$\mathcal{O} = \begin{bmatrix} \nabla\mathcal{L}^0\mathbf{h} \\ \nabla\mathcal{L}_{f_0}^1\mathbf{h} \\ \nabla\mathcal{L}_{f_2}^1\mathbf{h} \end{bmatrix}. \quad (2.43)$$

$\text{Rang}(\mathcal{O})=12$. $\nabla\mathcal{L}^0\mathbf{h}$ et $\nabla\mathcal{L}_{f_0}^1\mathbf{h}$ possèdent un nombre complet d'équations linéairement indépendantes. Donc, le rang de la matrice \mathcal{O} est toujours égal à la dimension de notre système quelles que soient les valeurs de v_1 , v_2 , ψ_1 et ψ_2 .

Cas 2. Reprenons maintenant l'étude du système pour le cas général où le nombre des robots est égal à n_r et le nombre des satellites en commun est égal à n_s . L'état du système complet est :

$$\hat{\mathbf{x}} = [\hat{\mathbf{p}}_1, \hat{d}_1, \dots, \hat{\mathbf{p}}_{n_r}, \hat{d}_{n_r}, \hat{b}^1, \dots, \hat{b}^{n_s}]^T \quad (2.44)$$

avec $\dim(\hat{\mathbf{x}})=4n_r + n_s$.

On cherche à étudier l'observabilité du système (2.44) quelques soient les valeurs des variables v_1, v_2, ψ_1 et ψ_2 . En considérant uniquement que le modèle des biais est

auto-régressif (i.e la condition (iv) $a \neq 0$ est respectée). Pour cela, on ne considère que $\nabla \mathcal{L}^0 \mathbf{h}$ et $\nabla \mathcal{L}_{f_0}^1 \mathbf{h}$ dans la matrice \mathcal{O} .

Le système (2.44) est faiblement localement observable si la matrice \mathcal{O} est de rang plein. Combien faut-il au moins des satellites en commun pour que le nombre des équations linéairement indépendantes dans $\nabla \mathcal{L}^0 \mathbf{h}$ et $\nabla \mathcal{L}_{f_0}^1 \mathbf{h}$ soit complet (i.e. égal à $4n_r + n_s$ la dimension du système (2.44)) ?

soit :

$$\mathcal{O} = \begin{bmatrix} \nabla \mathcal{L}^0 \mathbf{h} \\ \nabla \mathcal{L}_{f_0}^1 \mathbf{h} \end{bmatrix}. \quad (2.45)$$

$$Rang(\mathcal{O}) = rang(\nabla \mathcal{L}^0 \mathbf{h}) + rang(\nabla \mathcal{L}_{f_0}^1 \mathbf{h}).$$

Le nombre d'équations linéairement indépendantes dans $\nabla \mathcal{L}_{f_0}^1 \mathbf{h}$ est égal à n_s . $Rang(\nabla \mathcal{L}_{f_0}^1 \mathbf{h}) = n_s$. Le rang de $\nabla \mathcal{L}^0 \mathbf{h}$ est déterminé selon le nombre de colonnes (NC) et de lignes (NL). On distingue deux cas si (NL \geq NC) alors ($n_s \geq \frac{4n_r}{n_r-1}$) et si (NL < NC) alors ($n_s < \frac{4n_r}{n_r-1}$).

D'après les études en cas statique (2.3.1) , si $n_s \geq \frac{4n_r}{n_r-1}$:

$$Rang(\nabla \mathcal{L}^0 \mathbf{h}) = \begin{cases} 4(n_r - 1) + n_s & \text{si } \mathbf{G} \text{ sont les mêmes} \\ 4n_r + n_s - 1 & \text{si } \mathbf{G} \text{ sont différents} \end{cases} \quad (2.46)$$

donc :

$$Rang(\mathcal{O}) = \begin{cases} 4(n_r - 1) + 2n_s & \text{si } \mathbf{G} \text{ sont les mêmes} \\ 4n_r + 2n_s - 1 & \text{si } \mathbf{G} \text{ sont différents} \end{cases} \quad (2.47)$$

D'après des tests on a pu constater que si $n_s < \frac{4n_r}{n_r-1}$:

$$Rang(\nabla \mathcal{L}^0 \mathbf{h}) = n_s n_r \quad (2.48)$$

donc :

$$Rang(\mathcal{O}) = n_s (n_r + 1) \quad (2.49)$$

Pour que le rang(\mathcal{O}) soit plein combien faut-il au moins de satellites en commun ?

— Si $n_s \geq \frac{4n_r}{n_r-1}$:

On sait que le rang(\mathcal{O}) $\leq 4n_r + n_s$, donc pour que le rang de \mathcal{O} soit plein (i.e rang(\mathcal{O}) = $4n_r + n_s$), il faut que rang(\mathcal{O}) soit en même temps supérieur ou égal $4n_r + n_s$.

Pour que $4n_r + 2n_s - 4 \geq 4n_r + n_s$ il faut que $n_s \geq 4$

$\Rightarrow \forall n_s \geq 4$, min(rang(\mathcal{O})) dans l'équation (2.47) $\geq 4n_r + n_s$. Donc, dans les 2 cas de l'équation (2.47) le rang(\mathcal{O}) = $4n_r + n_s$. $\forall n_s \geq 4$ le système est faiblement localement observable $\forall n_r$.

— Si $n_s < \frac{4n_r}{n_r-1}$:

De même, on sait que rang(\mathcal{O}) $\leq 4n_r + n_s$, donc pour que le rang(\mathcal{O}) = $4n_r + n_s$, il faut que $n_s(n_r + 1) \geq 4n_r + n_s \Rightarrow n_r(n_s - 4) \geq 0 \Rightarrow n_s \geq 4$.

\Rightarrow Il faut au moins 4 satellites en commun pour que le système soit faiblement localement observable $\forall n_r$.

On peut donc conclure que dans le cas général, pour que le système soit faiblement localement observable il faut au moins 4 satellites en commun.

2.4 Conclusion

L'observabilité de deux problèmes de la localisation coopératives des robots avec des mesures biaisées (mesures de distances sur des balises et de pseudodistances sur des satellites) a été étudiée dans le cas général. D'après ces études, dans le cadre de la localisation avec balises il est possible d'observer l'état du système, s'il existe au moins $E(\frac{3n_r}{n_r-1})$ balises (en particulier, pour 2 robots il faut au moins 6 balises). En ce qui concerne le problème de la localisation coopérative avec GNSS, il est possible d'observer les positions des robots même si les robots ne se déplacent pas mais sous la condition que le modèle des biais ne soit pas constant et que les robots possèdent au moins 4 satellites en commun quel que soit le nombre de robots dans le système.

Chapitre 3

Localisation coopérative avec une approche ensembliste

Sommaire

3.1	Introduction	61
3.2	Formulation du problème de coopération avec des mesures de distance	62
3.3	Techniques ensemblistes pour la localisation	65
3.4	Propagation de contraintes	66
3.5	Localisation coopérative avec des mesures des distances	76
3.6	Localisation coopérative avec des mesures de pseudodistances	96
3.7	Conclusion	108

3.1 Introduction

Au cours de ce chapitre, une approche distribuée de la localisation coopérative présentant une analogie avec le GPS différentiel mais sans base fixe sera détaillée. Cette approche est basée sur des techniques ensemblistes comme la propagation de contraintes sur des intervalles et l'inversion ensembliste. Les méthodes ensemblistes connues aussi sous le nom des méthodes à erreurs bornées permettent d'obtenir des domaines de confiance fiables contenant les positions des robots. Une position garantie des robots est essentielle pour certaines tâches coopératives afin d'assurer leur sécurité.

Dans le but d'évaluer l'algorithme coopératif, les technologies de positionnement par mesures radio sur balises sont d'abord utilisées. Ensuite, l'algorithme est testé en utilisant les technologies de positionnement par GNSS.

Dans une première partie, une formulation du problème coopératif de l'estimation sera décrite. Dans une deuxième partie, les principes de la localisation ensembliste, les techniques de propagation des contraintes et l'inversion ensembliste seront rappelés.

Ensuite, les résultats de l'implémentation de l'approche proposée avec deux robots seront présentés pour illustrer la performance de l'approche coopérative en comparaison avec une méthode non-coopérative classique (i.e. standalone) basée sur le positionnement hybride en fusionnant des données proprioceptives et extéroceptives. Afin de prouver la faisabilité de cette méthode, nous avons simplifié l'étude en considérant que les robots sont homogènes (i.e. deux robots terrestres). Cependant, l'extension à un système hétérogène de N robots est assez facile à obtenir car l'hétérogénéité est liée aux capteurs de navigation à l'estime utilisés.

3.2 Formulation du problème de coopération avec des mesures de distance

Nous sommes dans le contexte où des robots mobiles naviguent ensemble dans un environnement particulier afin d'assurer une mission coopérative. Tout d'abord, nous sommes intéressés à trouver les positions réelles de chacun des robots avec des domaines de confiance fiables. Ensuite, l'objectif est de fournir aux robots des capacités de coopération, de sorte que chaque robot puisse déterminer les positions de ses partenaires dans un repère commun. Dès l'instant où chaque robot parvient à bien se localiser et à déterminer la position de ses partenaires dans le groupe, il ne reste qu'à contrôler les robots afin d'exécuter la mission souhaitée. On peut imaginer par exemple une tâche de rendez-vous de navigation en formation tout en évitant la collision.

Les méthodes distribuées de localisation sans serveur central sont des solutions qui existent déjà et qui sont très pertinentes pour les robots mobiles coopératifs. Ce type de méthode, étant adapté à notre problème, est utilisé durant nos travaux.

Généralement, la coopération exige le partage des mesures proprioceptives ou extéroceptives. Souvent ces dernières (mesures extéroceptives) souffrent d'erreurs communes aussi connues sous le nom de "biais" comme nous l'avons déjà vu. Les méthodes usuelles ne tiennent souvent pas compte de ces erreurs.

Une contribution de cette thèse est la prise en compte du fait que les robots mobiles partagent leurs mesures proprioceptives mais aussi leurs biais. Nous cherchons à estimer ces biais afin d'améliorer la localisation des robots en absolu et en relatif.

3.2.1 Nature des données

On cherche à estimer un vecteur de paramètres à partir d'un ensemble de mesures et de connaissances *a priori*, comme par exemple, l'estimation de la position d'un véhicule terrestre à partir des données de capteurs proprioceptifs et extéroceptifs.

3.2.1.1 Mesures proprioceptives (DR)

Chaque véhicule est équipé d'un gyromètre de lacet (système ESP dans une voiture) qui mesure la vitesse angulaire du véhicule (ω). Une estimation de la pose du véhicule est assurée par l'intégration de ces mesures à partir d'une première pose connue. La vitesse linéaire de chaque roue arrière est mesurée par des capteurs de vitesse (système ABS). La vitesse linéaire est calculée par

$$v = (v_{rl} + v_{rr}) / 2$$

où v_{rl} et v_{rr} sont respectivement les vitesses des roues gauche et droite. Ces mesures sont souvent affectées par des bruits ($\omega = \omega + \gamma^\omega$, $v = v + \gamma^v$). C'est la raison pour laquelle l'estimée par DR dérive à cause de l'accumulation de glissements de roue durant de longues distances ou en cas de surfaces irrégulières (Figure 3.1).

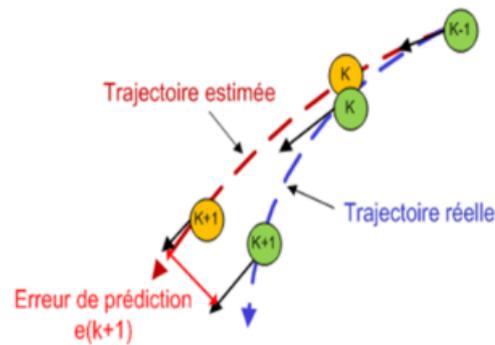


Figure 3.1 – Dérive de l'estimation

3.2.1.2 Mesures extéroceptives

Les mesures extéroceptives considérées ici, sont soit des distances entre les robots et des balises d'emplacements connus, soit des pseudodistances entre les robots et des satellites (GNSS). Les biais qui affectent ces mesures sont dus aux conditions environnementales particulières ou aux connaissances inexactes des positions des balises ou des satellites.

3.2.2 Modélisation des systèmes

Les modèles utilisés sont des modèles non linéaires qui peuvent contenir des observations redondantes (s'il y a plus que trois mesures) et aberrantes (dans le cas où le modèle d'observation ne correspond pas à la mesure réelle à cause d'un modèle physique inexact ou à cause d'un signal très bruité). Les modèles d'évolution des robots considérés sont classiques.

3.2.2.1 Modèle d'observation

Le modèle d'observation est un moyen d'accéder à la position des robots à partir des mesures et des connaissances du système.

Le problème de localisation d'un objet dans un plan à partir de mesures de distances faites par des balises dont les positions sont connues devient un problème d'estimation si les balises sont discernables. Ces mesures de distance peuvent être, par exemple, fournies par radio-communication par une technique de double temps de vol. Dans [96], les auteurs ont utilisé ce système dans des entrepôts industriels afin de localiser des robots mobiles.

Il faut tout de même noter que dans le cas des balises, quand ces dernières sont très éloignées de la zone d'évolution des robots, on retrouve le cas des mesures GNSS avec erreur additionnelle. L'unique différence est qu'on a un biais d'horloge de récepteur avec les mesures de GNSS.

Le modèle d'observation utilisé lors de nos travaux a été décrit dans le chapitre 2 précédent. On considère ici le modèle bidimensionnel de la façon suivante.

Soit r_i^j la distance qui sépare le robot i ($i = \{1, \dots, n_r\}$) situé aux coordonnées $\mathbf{p}_i = (x_i, y_i)$ de la balise j ($j = 1, \dots, n_b$) située aux coordonnées $\mathbf{p}^j = (x^j, y^j)$. n_r et n_b sont respectivement le nombre de robots et le nombre total de balises.

$$r_i^j = \sqrt{(x_i - x^j)^2 + (y_i - y^j)^2} + b^j \quad (3.1)$$

où b^j représente le biais de la balise j . On ajoute un terme ${}^i\beta^j$ qui modélise le bruit de la mesure.

Le modèle d'observation présenté dans l'équation (3.1) est similaire pour chaque robot, le vecteur de mesure des distances est alors :

$${}^i\mathbf{y} = \begin{bmatrix} r_i^1 & r_i^2 & \dots & r_i^{n_b} \end{bmatrix}^T$$

3.2.2.2 Modèle d'évolution

Un modèle uni-cycle classique f est souvent utilisé afin de prédire la position et l'orientation des robots. On suppose que chaque robot est équipé d'un gyromètre de lacet et d'encodeurs de roues pour mesurer la vitesse linéaire et angulaire des roues. Ces mesures constituent le vecteur d'entrée de notre système ${}^i\mathbf{u}_k = [v_{i,k} \ \omega_{i,k}]^T$. Le modèle d'évolution du robot i et un modèle constant des biais à chaque échantillon

k sont décrits par :

$$f = \begin{cases} x_{i,k} = x_{i,k-1} + T_e v_{i,k-1} \cos \psi_{i,k-1} \\ y_{i,k} = y_{i,k-1} + T_e v_{i,k-1} \sin \psi_{i,k-1} \\ \psi_{i,k} = \psi_{i,k-1} + T_e \omega_{i,k-1} \end{cases} \quad (3.2)$$

$$b_k^j = b_{k-1}^j \quad (3.3)$$

où T_e est la période d'échantillonnage et ψ_i le cap du robot à estimer.

Dans certains cas, l'angle de cap est mesuré. Les équations du modèle d'évolution 3.2 sont simplifiées en ne gardant que les deux premières. Nous utiliserons parfois un tel modèle par la suite.

3.3 Techniques ensemblistes pour la localisation

Pour certaines tâches en robotique la garantie d'un domaine de confiance fiable contenant la position des robots est parfois aussi importante que la position elle-même. Il est donc intéressant d'avoir recours aux méthodes ensemblistes qui sont bien adaptées à ce type de tâche. L'approche ensembliste permet d'obtenir l'ensemble des paramètres en adéquation avec les observations et leur intervalle d'erreur associé.

Les bornes des erreurs étant supposées connues, le problème consiste à déterminer un ensemble de vecteurs de paramètres compatibles avec les observations au lieu de faire une minimisation de la variance de l'erreur.

En tenant compte des bornes, chaque distance mesurée peut être écrite sous la forme $r_i^j \in r_{i,meas}^j + [e_r]$. La mesure des distances est une contrainte sur la position du robot mobile. En prenant l'exemple de la mesure de la distance entre un robot et une balise, la position du robot se situe dans une couronne dont le centre est la balise concernée. En considérant toutes les mesures, l'engin se trouve ainsi dans l'intersection des couronnes qui correspondent à chaque mesure de distance (Figure 3.2).

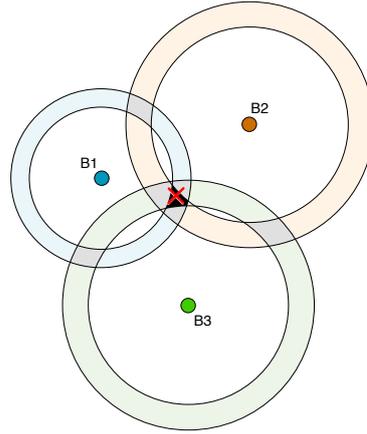


Figure 3.2 – Localisation ensembliste avec 3 balises sous hypothèse d’erreurs bornées. La croix rouge matérialise la position inconnue. La zone noire représente la zone d’incertitude dans laquelle se trouve la solution [2].

Afin de caractériser l’ensemble des solutions, différentes méthodes ensemblistes ont été développées. Par exemple, on trouve dans la littérature des méthodes basées sur des boîtes, des polytopes, des ellipses, ou encore des zonotopes. La figure 3.3 montre quelques représentations de la solution de localisation à trois balises.

Les notions de base de l’analyse par intervalles sont détaillées dans l’Annexe A. Dans la suite, nous nous intéressons aux méthodes de propagation de contraintes et d’inversion ensembliste par intervalles.

3.4 Propagation de contraintes

Il s’agit de résoudre un problème de satisfaction de contraintes (avec un ensemble de contraintes) en appliquant l’algorithme de Waltz [97].

3.4.1 Problèmes de Satisfaction de Contraintes (CSP)

Considérons n_x variables $x_i \in \mathbb{R}$, $i \in \{1, \dots, n_x\}$ reliées par n_f relations (ou contraintes) de la forme :

$$f_j(x_1, x_2, \dots, x_{n_x}) = 0, j \in \{1, \dots, n_f\}. \quad (3.4)$$

Chaque variable x_i appartient à un *domaine* \mathbb{X}_i qui peut être par souci de simplification un intervalle ou une union d’intervalles, notés $[x_i]$. On définit le vecteur \mathbf{x} comme suit :

$$\mathbf{x} = (x_1, x_2, \dots, x_{n_x})^T \quad (3.5)$$

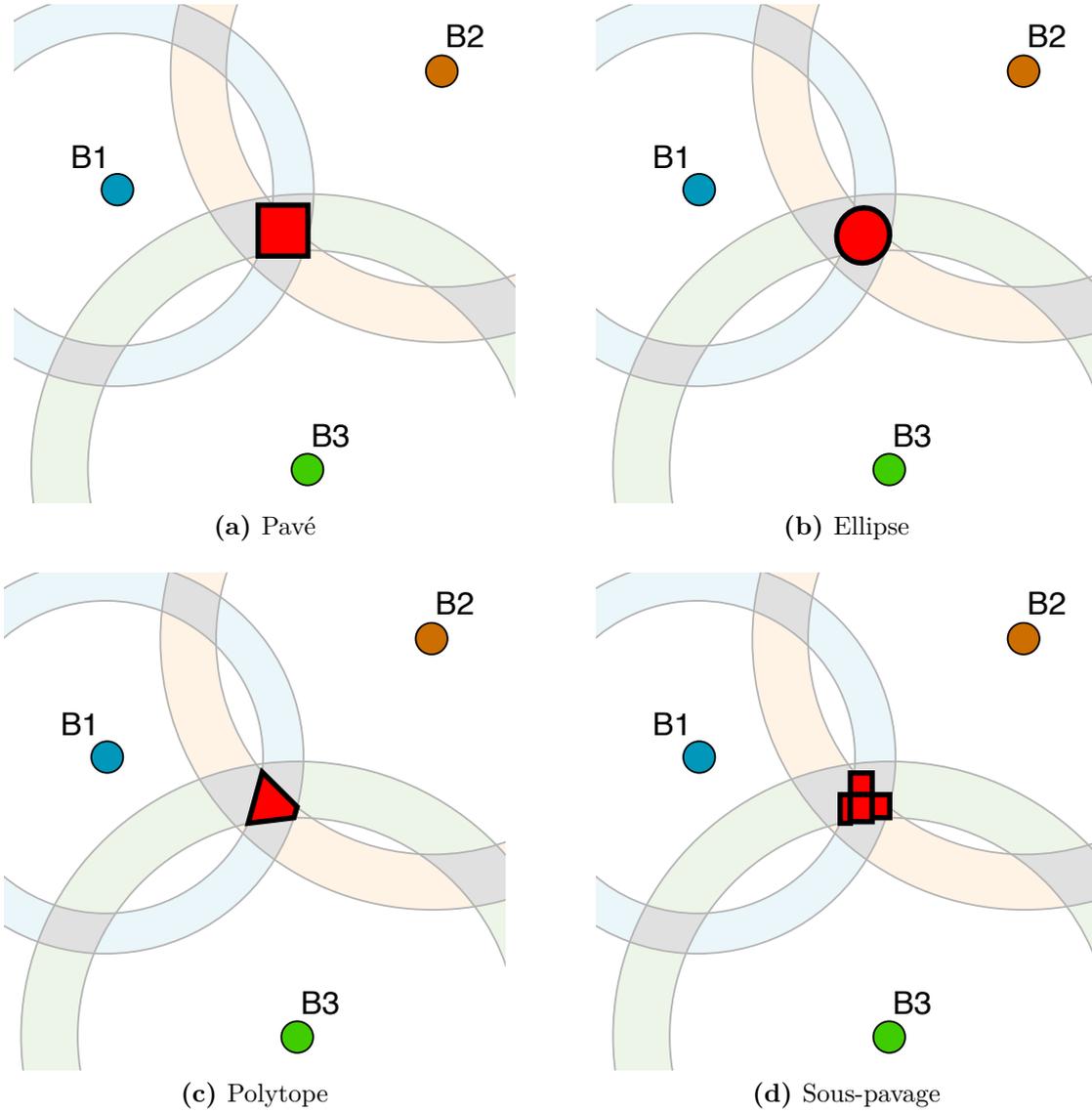


Figure 3.3 – Différentes représentations ensemblistes de la solution du problème de localisation à trois balises [2].

Le domaine initial de \mathbf{x} est noté :

$$[\mathbf{x}] = [x_1] \times \cdots \times [x_{n_x}]. \quad (3.6)$$

Notons par f_{1,\dots,n_f} les n_f fonctions coordonnées de la fonction vectorielle \mathbf{f} . L'équation (3.4) peut être écrite sous la forme vectorielle $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. On peut maintenant définir un *Problème de Satisfaction de Contraintes* ou CSP (Constraint Satisfaction Problem) \mathcal{H} défini comme suit [98] :

$$\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}]). \quad (3.7)$$

Un CSP est principalement constitué de variables, de domaines contenant ces variables et de contraintes.

L'ensemble solution \mathbb{S} de \mathcal{H} est défini comme

$$\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) = \mathbf{0}\}. \quad (3.8)$$

Les problèmes de satisfaction de contraintes peuvent contenir des contraintes d'égalité comme des contraintes d'inégalité. En CSP, les inégalités sont manipulées comme des égalités en introduisant des variables positives nommées *variables d'écart*.

Par exemple, à partir des contraintes suivantes :

$$\begin{cases} x_1 - \cos(x_2) \leq 0, \\ x_1 + x_2 = 2, \end{cases} \quad (3.9)$$

on peut définir un CSP noté \mathcal{H} en introduisant la variable d'écart x_3 :

$$\mathcal{H} = \begin{cases} x_1 - \cos(x_2) + x_3 = 0, \\ x_1 + x_2 - 2 = 0, \end{cases} \quad (3.10)$$

où les domaines des variables sont $[x_3] = [0, \infty[$ et $[x_1] = [x_2] = \mathbb{R}$. Les fonctions coordonnées sont $f_1(\mathbf{x}) = x_1 - \cos(x_2) + x_3$ et $f_2(\mathbf{x}) = x_1 + x_2 - 2$.

Résoudre un CSP \mathcal{H} c'est à dire *Contracter* un CSP \mathcal{H} ce qui consiste à remplacer le domaine initial $[\mathbf{x}]$ par un plus petit domaine $[\mathbf{x}']$ contenant toujours l'ensemble solution. Définissons la contraction de \mathcal{H} telle que $\mathbb{S} \subset [\mathbf{x}'] \subset [\mathbf{x}]$. Une contraction de \mathcal{H} est dite optimale si le domaine initial $[\mathbf{x}]$ peut être remplacé par la plus petite boîte contenant \mathbb{S} . Un *contracteur* noté \mathcal{C} est défini comme un opérateur permettant de contracter \mathcal{H} . Il existe différents contracteurs basiques [99]. Plusieurs méthodes peuvent être utilisées selon la nature du problème, par exemple l'élimination de Gauss, l'algorithme de Gauss-Seidel et l'algorithme de Newton qui sont détaillés dans [98]. Nous nous intéressons ici uniquement au contracteur par propagation-rétropropagation de contraintes noté $\mathcal{C}_{\downarrow\uparrow}$ pour sa simplicité et son efficacité dans le cas des problèmes à grande dimension.

3.4.2 Contracteur par propagation-rétropropagation

Le contracteur par propagation-rétropropagation $\mathcal{C}_{\downarrow\uparrow}$, également appelé « *forward-backward* » dans la littérature [100, 101], est basé sur la propagation de contraintes. Ce contracteur réduit les domaines de l'ensemble des contraintes du CSP $\mathcal{H} : \{(\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}])\}$ en traitant de façon isolée chacune des contraintes.

On suppose que chaque contrainte est de la forme $f_i(x_1, \dots, x_{n_x}) = 0$ et que la fonction f_i peut se décomposer en une suite d'opérations mettant en œuvre des

opérateurs et fonctions élémentaires telles que $(+, -, *, /, \sin, \cos \dots)$. On peut alors décomposer la contrainte en *contraintes primitives*.

Dans l'exemple de localisation par mesure de distances avec trois balises (Eq. (A.12)), la contrainte associée à une balise B^j située aux coordonnées (x^j, y^j) où $j \in \{1, 2, 3\}$ peut s'écrire

$$r^j = \sqrt{(x - x^j)^2 + (y - y^j)^2},$$

et peut ainsi se décomposer en contraintes primitives en introduisant des variables intermédiaires :

$$\begin{cases} i_1 = -x^j \\ i_2 = x + i_1 \\ i_3 = i_2^2 \\ i_4 = -y^j \\ i_5 = y + i_4 \\ i_6 = i_5^2 \\ i_7 = i_3 + i_6 \\ r^j = \sqrt{i_7} \end{cases}$$

Les domaines associés aux variables intermédiaires i_k introduites dans la décomposition sont $] - \infty, \infty[$. Le principe de la propagation de contraintes introduite par Waltz [97] consiste à contracter chaque contrainte primitive jusqu'à ce que les domaines de variables ne se réduisent plus. Pour les contraintes mettant en jeu deux variables et une fonction, telle que la racine carrée, deux étapes de contraction sont produites en reformulant la contrainte : une à partir de l'image directe par la fonction, l'autre à partir de l'image réciproque. Ainsi, dans notre exemple, la contrainte $r^j = \sqrt{i_7}$ peut être réécrite de deux façons :

$$\begin{cases} r^j = \sqrt{i_7} \\ i_7 = (r^j)^2 \end{cases} .$$

Les étapes de la contraction sont :

$$\begin{aligned} [r^j] &:= \sqrt{[i_7]} \cap [r^j] \\ [i_7] &:= [r^j]^2 \cap [i_7]. \end{aligned}$$

Pour les contraintes liant trois variables par un opérateur binaire (tel que l'addition), trois réécritures sont possibles. La contrainte $i_7 = i_3 + i_6$ peut ainsi s'exprimer sous

les formes

$$\begin{cases} i_3 = i_7 - i_6 \\ i_6 = i_7 - i_3 \\ i_7 = i_3 + i_6 \end{cases}$$

ce qui permet d'obtenir les étapes de contraction suivantes :

$$\begin{aligned} [i_3] &:= ([i_7] - [i_6]) \cap [i_3] \\ [i_6] &:= ([i_7] - [i_3]) \cap [i_6] \\ [i_7] &:= ([i_3] + [i_6]) \cap [i_7]. \end{aligned}$$

L'algorithme de Waltz consiste à appliquer l'ensemble des étapes de contraction, sans ordre défini, jusqu'à ce que toutes les contraintes soient inefficaces. Cette stratégie peut nécessiter d'effectuer plusieurs tours de contraction avant d'atteindre un point fixe.

La contraction par propagation et rétropropagation permet de choisir les contraintes primitives utilisées dans un ordre optimal afin de maximiser la contraction obtenue [100].

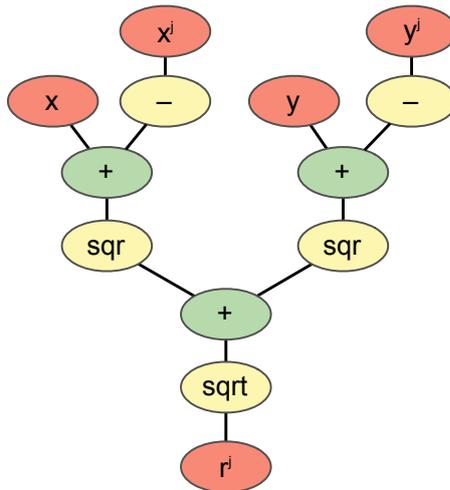


Figure 3.4 – Décomposition en contraintes primitives de la contrainte associée à la mesure de distance d'une balise B^j [2].

Quand la décomposition en contraintes primitives mène à un graphe de contraintes acyclique (Figure 3.4), la contraction optimale par une contrainte $f_i = 0$ est obtenue en descendant puis remontant les nœuds de l'arbre. On peut ainsi écrire l'algorithme 3.1 du contracteur par propagation et rétropropagation pour la contrainte associée à la mesure de distance d'une balise B^j .

Algorithme 3.1 Contracteur par propagation et rétropropagation pour une balise

Propagation

Rétropropagation

$$[i_1] := -[x^j]$$

$$[i_2] := [x] + [i_1]$$

$$[i_3] := [i_2]^2$$

$$[i_4] := -[y^j]$$

$$[i_5] := [y] + [i_4]$$

$$[i_6] := [i_5]^2$$

$$[i_7] := [i_3] + [i_6]$$

$$[r^j] := \sqrt{[i_7]} \cap [r^j]$$

$$[i_7] := [r^j]^2 \cap [i_7]$$

$$[i_3] := ([i_7] - [i_6]) \cap [i_3]$$

$$[i_6] := ([i_7] - [i_3]) \cap [i_6]$$

$$[i_5] := (\text{sqr}^{-1}[i_6]) \cap [i_5]$$

$$[y] := ([i_5] - [i_4]) \cap [y]$$

$$[i_4] := ([i_5] - [y]) \cap [i_4]$$

$$[y^j] := -[i_4] \cap [y^j]$$

$$[i_2] := (\text{sqr}^{-1}[i_3]) \cap [i_2]$$

$$[x] := ([i_2] - [i_1]) \cap [x]$$

$$[i_1] := ([i_2] - [x]) \cap [i_1]$$

$$[x^j] := -[i_1] \cap [x^j]$$

Le contracteur que nous venons de présenter ne traite qu'une contrainte à la fois. Quand il s'agit de contracter un domaine en tenant compte de toutes les contraintes du problème, on va appliquer successivement les contracteurs liés à chaque contrainte. Cependant, les dépendances entre contraintes impliquent que le contracteur n'arrive pas nécessairement à un point fixe dès sa première application. Le contracteur va donc être appelé plusieurs fois, jusqu'à l'obtention d'un point fixe.

3.4.3 Inversion ensembliste par intervalles (SIVIA)

Les intervalles et les boîtes sont des objets faciles à manipuler grâce à l'arithmétique intervalle et aux fonctions d'inclusion. Les contracteurs définis précédemment permettent également de réduire un pavé en fonction d'une ou plusieurs contraintes.

En général, les solutions des problèmes traités par l'approche ensembliste sont des ensembles quelconques. Afin de caractériser l'ensemble solution, l'inclusion des sous-pavages dans une boîte englobante sera utilisée dans ces travaux. Cette représentation est parfois pessimiste, en particulier lorsque l'ensemble solution n'est pas connexe ou convexe. Cependant, le calcul du volume et la détermination du centre d'une boîte englobante est plus rapide que le calcul du volume et du centre de gravité des sous-pavages.

3.4.3.1 Sous pavage

Un sous-pavage d'une boîte $[\mathbf{x}] \subset \mathbb{R}^n$ est l'union de boîtes de $[\mathbf{x}]$ qui ne se chevauchent pas et de largeur non nulle. Deux boîtes d'un même sous-pavage peuvent

avoir une intersection non vide contenant juste une frontière, mais leurs intérieurs doivent être disjoints. Un sous-pavage de $[\mathbf{x}]$ qui couvre \mathbb{X} est appelé un *pavage* de $[\mathbf{x}]$.

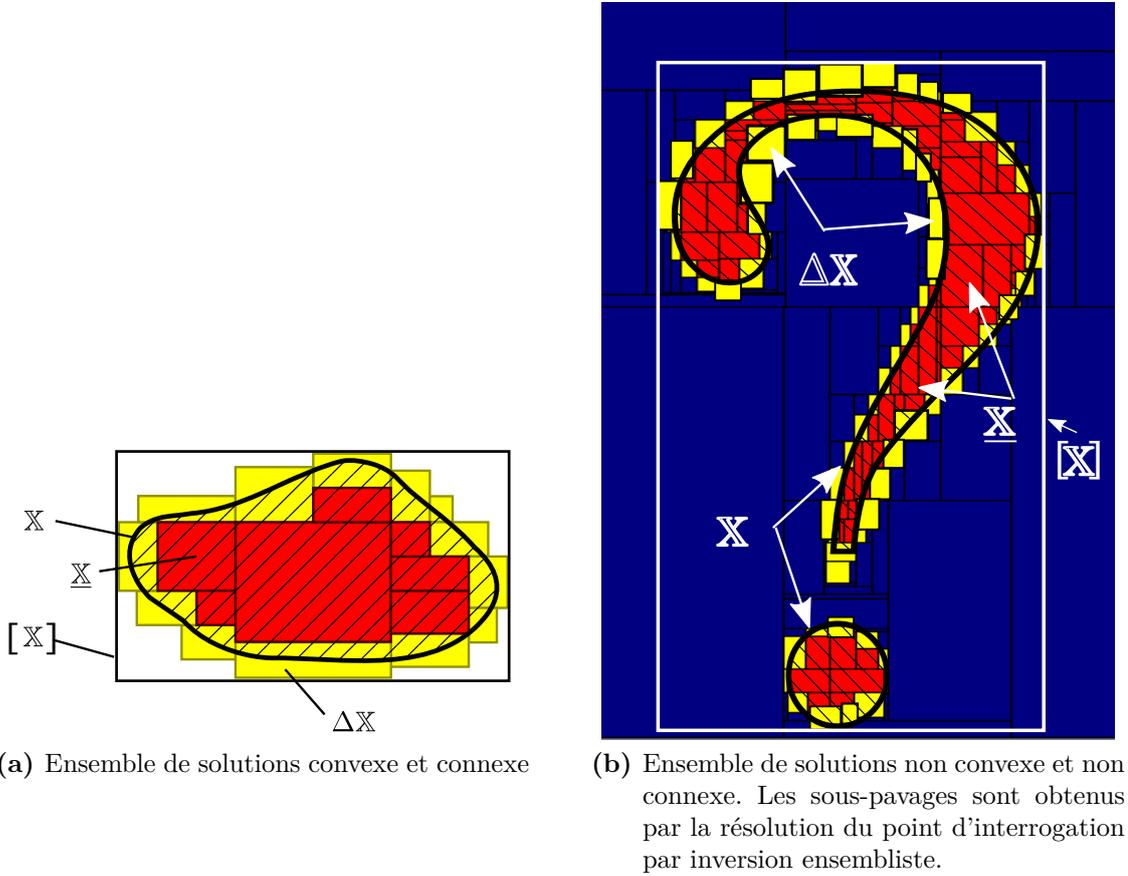


Figure 3.5 – Encadrement de l'ensemble \mathbb{X} entre deux sous-pavages $\underline{\mathbb{X}}$ et $\overline{\mathbb{X}} = \underline{\mathbb{X}} \cup \Delta\mathbb{X}$. $[\mathbb{X}]$ est la boîte englobante de la solution.

On peut encadrer l'ensemble solution \mathbb{X} entre deux sous-pavages (Fig. 3.5) : une approximation intérieure $\underline{\mathbb{X}}$ et une approximation extérieure $\overline{\mathbb{X}} = \underline{\mathbb{X}} \cup \Delta\mathbb{X}$ (où $\Delta\mathbb{X}$ est le sous-pavage frontière) telles que

$$\underline{\mathbb{X}} \subset \mathbb{X} \subset \overline{\mathbb{X}}. \quad (3.11)$$

Le volume du sous-pavage frontière $\Delta\mathbb{X}$ permet de caractériser la précision de l'approximation.

Un sous-pavage *régulier* d'une boîte $[\mathbf{x}]$ peut être obtenu par bisections (division en deux boîtes selon un plan médian) et sélections successives à partir de $[\mathbf{x}]$. Les sous-pavages réguliers présentent l'avantage de pouvoir être représentés par des arbres binaires, ce qui limite leur occupation mémoire et permet d'implémenter facilement les opérations telles que l'union, l'intersection ou l'inclusion [98].

3.4.3.2 SIVIA

L'algorithme récursif d'inversion ensembliste par intervalles SIVIA (Set Inversion via Interval Analysis) permet d'encadrer l'ensemble solution \mathbb{X} entre 2 sous-pavages tels que $\underline{\mathbb{X}} \subset \mathbb{X} \subset \overline{\mathbb{X}}$ [102]. L'objectif est de caractériser l'ensemble \mathbb{X} tel que $\mathbf{f}(\mathbb{X}) = \mathbb{Y}$. Autrement dit, il s'agit de déterminer l'image réciproque $\mathbb{X} = \mathbf{f}^{-1}(\mathbb{Y})$. Une boîte de recherche $[\mathbf{x}]$ contenant l'ensemble solution, un ensemble des mesures \mathbb{Y} et une fonction d'inclusion convergente $[\mathbf{f}]$ sont requises pour l'évaluation de SIVIA. La boîte initiale $[\mathbf{x}]$ peut être choisie très grande si on ne dispose pas de connaissance a priori sur la solution. L'algorithme ne traite que des pavés.

L'algorithme SIVIA est du type *branch and bound* (séparation et évaluation, une méthode générique de résolution de problèmes d'optimisation combinatoire). Tout d'abord, un test d'inclusion est effectué à l'aide de la fonction d'inclusion $[\mathbf{f}]$.

- Si l'image de la boîte $[\mathbf{x}]$ par $[\mathbf{f}]$ est disjointe de \mathbb{Y} , $[\mathbf{x}]$ est garanti de ne pas appartenir à l'ensemble solution. La boîte est donc rejetée.
- Si l'image de $[\mathbf{x}]$ par $[\mathbf{f}]$ est incluse dans l'ensemble \mathbb{Y} , alors, on a la garantie que $[\mathbf{x}]$ appartient à l'ensemble solution \mathbb{X} . La boîte $[\mathbf{x}]$ est donc ajoutée au sous-pavage intérieur $\underline{\mathbb{X}}$ et au sous-pavage extérieur $\overline{\mathbb{X}}$.
- Sinon $[\mathbf{x}]$ est indéterminé. Dans ce cas, $[\mathbf{x}]$ est bissecté en deux boîtes $[\mathbf{x}_1]$ et $[\mathbf{x}_2]$, et l'algorithme SIVIA est appliqué récursivement sur les deux sous-boîtes obtenues.

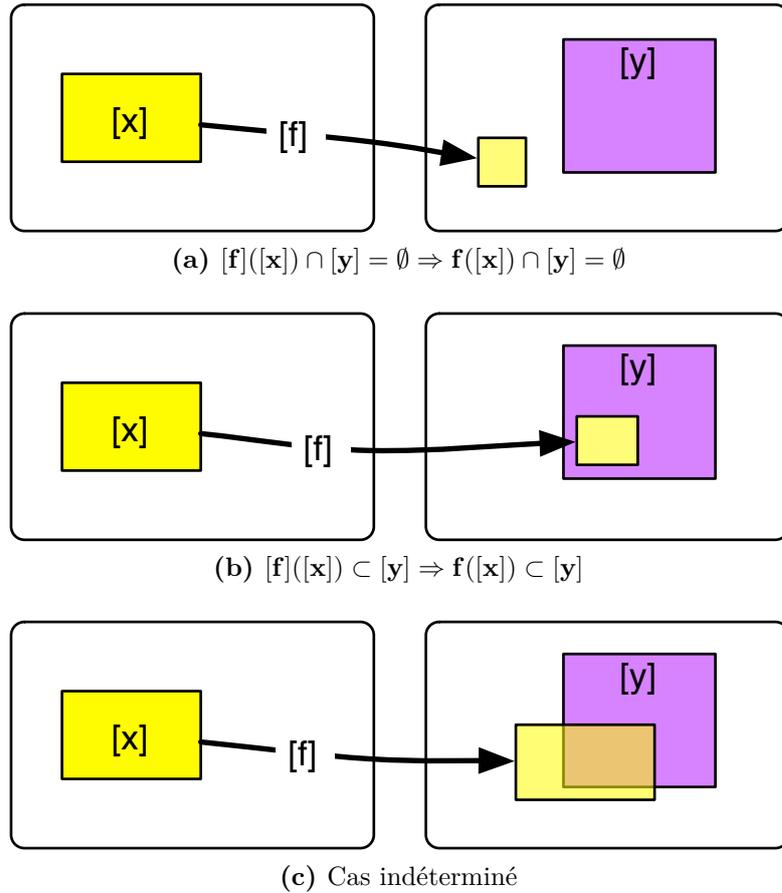


Figure 3.6 – Test d'inclusion pour l'inversion ensembliste.

Les boîtes indéterminées dont la largeur est inférieure à une précision ε ne sont pas bissectées mais ajoutées au sous pavage extérieur $\bar{\mathbb{X}}$. SIVIA est présenté dans l'algorithme 3.2, où les sous-pavages $\underline{\mathbb{X}}$ et $\bar{\mathbb{X}}$ sont initialement l'ensemble vide.

Algorithme 3.2 SIVIA(in : $[\mathbf{x}], \mathbf{f}, \mathbb{Y}, \varepsilon$; inout : $\underline{\mathbb{S}}_p, \overline{\mathbb{S}}_p$)

```

1: if  $[\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$  then
2:   return
3: end if
4: if  $[\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y}$  then
5:    $\underline{\mathbb{S}}_p := \underline{\mathbb{S}}_p \cup [\mathbf{x}]; \overline{\mathbb{S}}_p := \overline{\mathbb{S}}_p \cup [\mathbf{x}];$  return
6: end if
7: if  $w([\mathbf{x}]) < \varepsilon$  then  $\{\omega([\mathbf{x}])$  est la largeur de la boîte $\}$ 
8:    $\overline{\mathbb{S}}_p := \overline{\mathbb{S}}_p \cup [\mathbf{x}];$  return
9: end if
10: bissecter  $[\mathbf{x}]$  en  $[\mathbf{x}_1]$  et  $[\mathbf{x}_2]$ 
11: SIVIA( $[\mathbf{x}_1], \mathbf{f}, \mathbb{Y}, \varepsilon, \underline{\mathbb{S}}_p, \overline{\mathbb{S}}_p$ )
12: SIVIA( $[\mathbf{x}_2], \mathbf{f}, \mathbb{Y}, \varepsilon, \underline{\mathbb{S}}_p, \overline{\mathbb{S}}_p$ )

```

3.4.3.3 SIVIA avec contracteur

Le nombre de bisections dans l'algorithme SIVIA (3.2) risque d'augmenter exponentiellement la complexité de calcul, en particulier si la boîte initiale est de grande dimension. Dans ce cas SIVIA est peu adapté à des applications temps réel. L'ajout d'un contracteur avec propagation à l'algorithme classique SIVIA avant la bisection de la boîte permet de réduire le nombre des bisections et d'obtenir un sous-pavage solution plus précis. On peut définir l'algorithme SIVIAP (3.3) permettant l'estimation de l'ensemble \mathbb{S}_p des paramètres d'un vecteur \mathbf{p} tel que $\mathbf{g}(\mathbf{p}) \in [\mathbf{y}]$ à l'aide d'un contracteur $\mathcal{C}_{\mathbb{S}_p}$. L'utilisation d'un contracteur permet la réduction du domaine de \mathbf{p} avant l'opération de bisection. On peut appliquer n'importe quel type de contracteur ou même une combinaison de contracteurs selon la complexité du problème [98]. Le contracteur le plus simple pour \mathbb{S}_p est le contracteur $\mathcal{C}_{\mathbb{S}_p}^0$ tel que

$$\mathcal{C}_{\mathbb{S}_p}^0 : \begin{cases} \mathbb{I}\mathbb{R}^{n_p} & \rightarrow & \mathbb{I}\mathbb{R}^{n_p} \\ [\mathbf{p}] & \mapsto & \begin{cases} \emptyset & \text{si } [\mathbf{g}]([\mathbf{p}]) \cap [\mathbf{y}] = \emptyset \\ [\mathbf{p}] & \text{sinon} \end{cases} \end{cases} .$$

L'utilisation du contracteur $\mathcal{C}_{\mathbb{S}_p}^0$ conduit au même résultat que l'algorithme SIVIA basé sur une fonction d'inclusion. Une combinaison de $\mathcal{C}_{\mathbb{S}_p}^0$ avec un contracteur par propagation-rétropropagation $\mathcal{C}_{\downarrow\uparrow}$ est souvent utilisé. La contrepartie de l'utilisation d'un contracteur est que le sous-pavage solution obtenu n'est plus nécessairement régulier.

Algorithme 3.3 SIVIAP(in : $[\mathbf{p}]$, $\mathcal{C}_{\mathbb{S}_p}$, \mathbf{g} , $[\mathbf{y}]$, ε ; inout : $\underline{\mathbb{S}}_p$, $\overline{\mathbb{S}}_p$)

```

1:  $[\mathbf{p}] := \mathcal{C}_{\mathbb{S}_p}([\mathbf{p}])$ 
2: if  $[\mathbf{p}] = \emptyset$  then
3:   return
4: end if
5: if  $[\mathbf{g}]([\mathbf{p}]) \subset [\mathbf{y}]$  then
6:    $\underline{\mathbb{S}}_p := \underline{\mathbb{S}}_p \cup [\mathbf{p}]$ ;  $\overline{\mathbb{S}}_p := \overline{\mathbb{S}}_p \cup [\mathbf{p}]$ ; return
7: end if
8: if  $w([\mathbf{p}]) < \varepsilon$  then
9:    $\overline{\mathbb{S}}_p := \overline{\mathbb{S}}_p \cup [\mathbf{p}]$ ; return
10: end if
11: bissecter  $[\mathbf{p}]$  en  $[\mathbf{p}_1]$  et  $[\mathbf{p}_2]$ 
12: SIVIAP( $[\mathbf{p}_1]$ ,  $\mathcal{C}_{\mathbb{S}_p}$ ,  $\mathbf{g}$ ,  $[\mathbf{y}]$ ,  $\varepsilon$ ,  $\underline{\mathbb{S}}_p$ ,  $\overline{\mathbb{S}}_p$ )
13: SIVIAP( $[\mathbf{p}_2]$ ,  $\mathcal{C}_{\mathbb{S}_p}$ ,  $\mathbf{g}$ ,  $[\mathbf{y}]$ ,  $\varepsilon$ ,  $\underline{\mathbb{S}}_p$ ,  $\overline{\mathbb{S}}_p$ )

```

3.5 Localisation coopérative avec des mesures des distances

Nous proposons une méthodologie pour la localisation coopérative basée sur des techniques de propagation des contraintes adaptée aux applications temps réel. Les méthodes classiques de localisation avec balises reposent sur la fusion des mesures de DR et de distance afin d'améliorer le positionnement individuel des robots. Notre objectif est de corriger les erreurs des mesures de distances grâce à la coopération. L'idée est inspirée du principe du GPS différentiel qui utilise un réseau de stations fixes de référence qui transmet l'écart entre les positions indiquées par les satellites et leurs positions réelles connues. Cependant, notre approche n'utilise pas de bases fixes vu que les robots sont mobiles et partagent les informations entre eux indépendamment d'un serveur fixe. Par souci de simplification, l'algorithme est d'abord testé sur deux véhicules en utilisant des mesures de DR et des distances simulées. Il sera ensuite testé avec plus de véhicules.

3.5.1 Simulation

Les données simulées contiennent des mesures de DR qui sont les vitesses linéaires et l'angle d'orientation de chaque véhicule, et des mesures extéroceptives qui sont les distances entre les véhicules et des balises fixes. Nous considérons un environnement où les balises de positions connues sont assez éloignées de la zone d'évolution des véhicules pour prendre en compte le fait qu'il peut y avoir des erreurs de positionnement des balises. Les véhicules communiquent en permanence afin de partager leurs informations.

3.5 Localisation coopérative avec des mesures des distances

Nous avons développé un simulateur sous Matlab-Simulink qui nous permet de simuler des trajectoires très diverses à différentes vitesses. Afin de générer les données DR, qui sont la vitesse linéaire de roues v et le cap ψ , chaque véhicule terrestre est commandé en vitesse v à l'aide d'un contrôleur Proportionnel Intégral (PI) et en angle de braquage γ à l'aide d'un contrôleur proportionnel (P) afin de suivre une trajectoire de référence. La figure 3.7 illustre les différents paramètres du modèle bicyclette utilisé pour chaque véhicule. Le modèle de la commande résumé dans la Fig. 3.8 est décrit par :

$$\begin{aligned}\dot{x} &= v \cos \psi \\ \dot{y} &= v \sin \psi \\ \dot{\psi} &= \frac{v}{L} \tan \gamma\end{aligned}\tag{3.12}$$

v et γ sont les entrées de la commande qui corrigent les erreurs de la position et l'angle par rapport à la référence. Plusieurs exemples de commande utilisant le même modèle de véhicules sont détaillés dans [103].

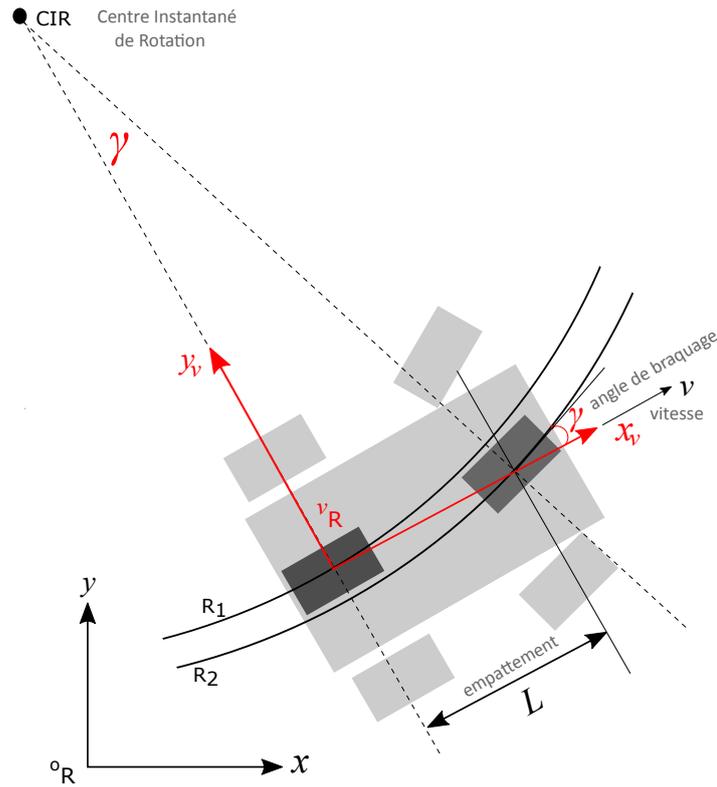


Figure 3.7 – Modèle bicyclette du véhicule. Le véhicule est représenté en gris clair, et l'approximation en bicyclette est en gris foncé. v_R et o_R sont les repères mobile et fixe. v est la vitesse en direction de x et γ est l'angle de braquage.



Figure 3.8 – Architecture de la commande de véhicule (v et γ sont les entrées de la commande).

La génération des mesures de distances est effectuée en utilisant le modèle d'observation défini dans l'Eq (3.13), où (x_i, y_i) sont les coordonnées de chaque véhicule d'indice i ($i = \{1, 2\}$) et b_i^j représente le biais de la mesure de distance r_i^j entre le véhicule i et la balise j ($j = \{1 \dots m\}$) de coordonnées (x^j, y^j) .

$$r_i^j = \sqrt{(x_i - x^j)^2 + (y_i - y^j)^2} + b_i^j + \beta^j \quad (3.13)$$

Nous supposons dans cette partie que chaque véhicule effectue une trajectoire circulaire de rayon différent de l'autre.

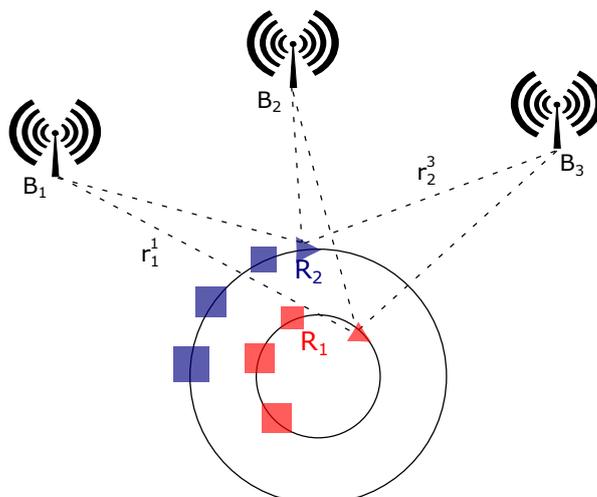


Figure 3.9 – Scénario de localisation coopérative (en réalité la distance entre les véhicules et les balises est grande par rapport à la zone d'évolution).

3.5.1.1 Données partagées

Nous supposons que les véhicules ont synchronisé leurs horloges dans le même référentiel du temps. Par exemple, dans notre cas, la synchronisation peut être effectuée en utilisant le *Network Time Protocol* (NTP), ou par GPS. Dans notre problème, la précision de la synchronisation n'est pas cruciale. Chaque véhicule partage les informations suivantes à chaque période d'échantillonnage :

- la dernière estimation des biais,
- la dernière estimation de sa propre position absolue,
- le vecteur vitesse (les mesures courantes de DR).

Chaque information est traitée comme un intervalle ou une boîte. La méthode d'estimation basée sur la propagation des contraintes est étendue afin de développer des algorithmes de fusion de données pour les applications de localisation coopérative. Les informations sont considérées publiques et il n'est pas nécessaire de manipuler des copies privées de l'information, comme dans [59] et [104].

3.5.1.2 Description des scénarios

Deux scénarii ont été simulés où : (a) chaque véhicule doit se localiser en mode standalone, (b) chaque véhicule doit se localiser en mode coopératif grâce à l'échange des informations. Dans ce dernier cas, les délais de la communication ne sont pas traités. On suppose que la durée de latence des données est inférieure à la durée de traitement des données.

La localisation standalone signifie que chaque véhicule estime sa position en fusionnant les données des capteurs extéroceptifs recueillies à partir des balises fixes (situées dans la zone d'évolution des robots) avec les mesures DR. La coopération des deux véhicules signifie que chaque véhicule estime sa position par rapport à des balises et prédit également la position des partenaires dans le repère de travail. La différence entre les deux méthodes consiste dans le partage des informations qui peut améliorer les estimations des positions. La figure 3.10 illustre le scénario de la coopération en précisant les informations partagées. La distance relative entre les véhicules est notée Δ .

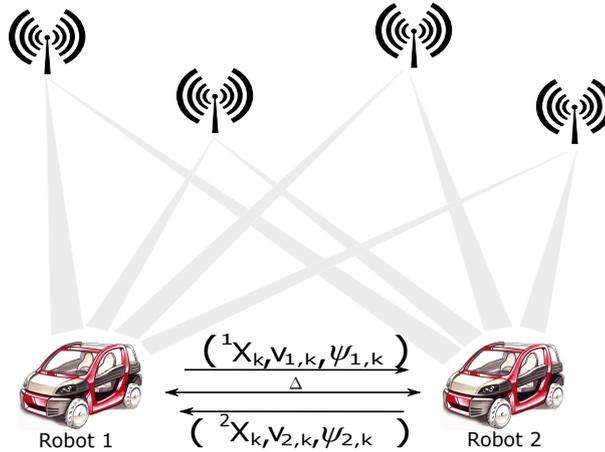


Figure 3.10 – Scénario de la localisation coopérative.

Soit ${}^i \mathbf{x}$ le vecteur d'état du véhicule i , dans notre cas $i = \{1, 2\}$. $(x_{1,k}, y_{1,k})$ et $(x_{2,k}, y_{2,k})$ sont les positions des véhicules 1 et 2 à chaque instant k et ${}^i b^1 \dots {}^i b^j$ sont les biais qui affectent les mesures disponibles de distances :

$${}^i \mathbf{x}_k = [{}^i x_{1,k} \quad {}^i y_{1,k} \quad {}^i x_{2,k} \quad {}^i y_{2,k} \quad {}^i b^1 \quad \dots \quad {}^i b^m]^T \quad (3.14)$$

Tant que la communication est établie, chaque véhicule peut fusionner sa propre estimation des biais de mesures de distances avec l'estimation reçue et mettre à jour la position prédite de son partenaire. Mais en cas de perte de signal, le véhicule estime sa position avec ses propres mesures et continue de prédire la position de l'autre avec les dernières mesures reçues afin d'avoir une idée sur l'emplacement de l'autre véhicule. Cela est nécessaire dans des missions robotiques d'approche pour que les engins ne rentrent pas en collision.

3.5.2 Algorithme coopératif

L'architecture proposée est un mécanisme prédicteur/estimateur qui est illustré dans la figure 3.11. C'est un problème de positionnement en $2D$. À chaque instant k , une

étape de prédiction de l'état par le modèle d'évolution est faite en utilisant comme informations les données DR (la vitesse et le cap) et la position estimée à l'instant précédent. Les biais suivent un modèle d'évolution constant. Ensuite, si des mesures de distances sont disponibles, une étape de mise à jour peut être effectuée pour fournir à la fin la position absolue de chaque véhicule et les biais estimés.

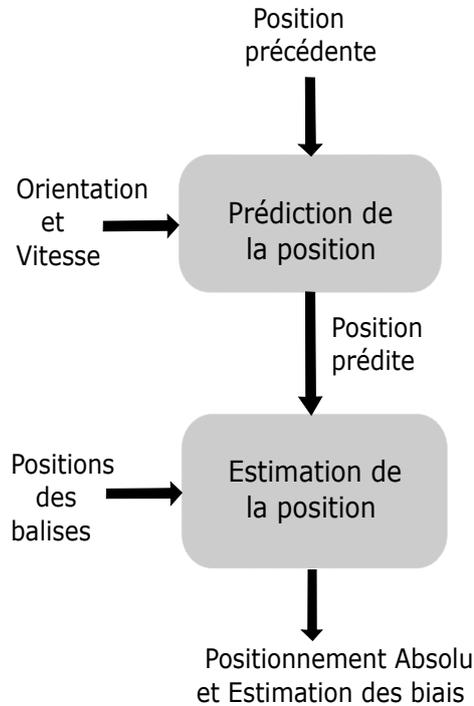


Figure 3.11 – Architecture de localisation d'un véhicule.

L'algorithme prédictif/estimateur est présenté avec deux robots de la façon suivante.

Prédiction

Soit ${}^i\mathbf{u}_k$ le vecteur des mesures proprioceptives qui contient respectivement la vitesse linéaire des roues $v_{i,k}$ et le cap $\psi_{i,k}$ du véhicule considéré et de son partenaire.

$${}^i\mathbf{u}_k = \begin{bmatrix} {}^i v_{1,k} \\ {}^i \psi_{1,k} \\ {}^i v_{2,k} \\ {}^i \psi_{2,k} \end{bmatrix} \quad (3.15)$$

Le modèle d'évolution utilisé est le suivant où T_e est la période d'échantillonnage et

k est le pas du temps :

$$\begin{cases} x_{i,k} = x_{i,k-1} + T_e v_{i,k-1} \cos \psi_{i,k-1} \\ y_{i,k} = y_{i,k-1} + T_e v_{i,k-1} \sin \psi_{i,k-1} \end{cases} \quad (3.16)$$

En ce qui concerne l'évolution des biais, un modèle constant est utilisé car les balises sont fixes et très éloignées :

$$b_k^j = b_{k-1}^j \quad (3.17)$$

Estimation

Soit ${}^i \mathbf{y}$ le vecteur des mesures extéroceptives qui fournit une information partielle sur la position de chaque véhicule en considérant les balises de positions fixes et connues. Ce vecteur d'observation contient les mesures de distance r^j où $j \in \{1, \dots, n_b\}$:

$${}^i \mathbf{y} = \begin{bmatrix} i_{r^1} \\ i_{r^2} \\ \vdots \\ i_{r^{n_b}} \end{bmatrix} \quad (3.18)$$

Le problème de fusion des données est traité comme un problème d'observation d'état. Les données considérées lors de la fusion sont redondantes puisqu'à chaque étape de fusion on réutilise des biais qui ont déjà été fusionnés à l'instant précédent.

Le principe est que chaque véhicule se localise tout seul en estimant la position et les biais et coopère avec l'autre en partageant ses estimations (positions et biais) pour la fusion.

Les étapes de localisation coopérative sont détaillées dans l'algorithme suivant 3.4 :

Algorithme 3.4 Traitement de l'information dans le véhicule i

Le véhicule

1. Fait ses mesures proprioceptives
2. Prédit sa position
3. Prédit les biais (qui sont constants ici)
4. Fait ses mesures extéroceptives
5. Met à jour sa propre position
6. Prédit la position du partenaire en utilisant les dernières mesures de DR reçues
7. Diffuse ses informations (positions estimées + mesures DR + biais estimés) au partenaire
8. Écoute s'il a reçu des informations de son partenaire. Si oui, alors la coopération devient possible :
 - a) Met à jour la prédiction de la position de partenaire en utilisant la position estimée reçue
 - b) Met à jour les biais

Toutes ces étapes sont exécutées par chaque véhicule et la coopération est obtenue par la contraction (i.e. intersection) de leurs estimations des biais. Dans cette mise en œuvre de l'algorithme, les retards de communication doivent être plus petits que la durée de traitement. Sinon, l'étape 8 de l'algorithme peut être étendue avec des buffers de données pour faire les mises à jour dans le passé et les propager jusqu'à l'instant actuel.

Dans ce cas, le Problème de Satisfaction de Contraintes CSP noté \mathcal{H} est le suivant (3.19) :

$$\mathcal{H} : \{F(\mathbf{x}) = \sqrt{(x_i - x^j)^2 + (y_i - y^j)^2} + b^j + \beta_i^j - r_i^j = \mathbf{0}\} \quad (3.19)$$

où j est l'indice de la balise $\{j \in 1 \dots n_b\}$ et F est une fonction vectorielle qui contient les j contraintes de distance. L'objectif est de contracter \mathcal{H} afin d'éliminer les parties de chaque boîte contenant les paramètres à estimer qui sont inconsistantes avec les mesures r_i^j . Tant que les mesures sont compatibles avec le modèle d'erreur borné choisi pour chaque domaine des variables, l'ensemble solution garantit de contenir la position réelle du véhicule.

Le contracteur utilisé pour chaque contrainte de l'ensemble \mathcal{H} est le contracteur de propagation et rétropropagation $\mathcal{C}_{\downarrow\uparrow}$ décrit dans la sous-section 3.4.2.

Un algorithme de Waltz est aussi appliqué sur toutes les contraintes jusqu'à que les

boîtes de positions et des biais ne se contractent plus. Son principe, comme il était indiqué avant, est de répéter l'étape de propagation et rétropropagation jusqu'à ce que les domaines des variables ne se contractent plus.

L'algorithme 3.5 présente les étapes détaillées de la méthode coopérative proposée.

Algorithme 3.5 Algorithme avec propagation de contraintes (CP) implémenté dans le véhicule 1

```

1:  $[x_{1,k-1}] = [x_{1,ref}], [y_{1,k-1}] = [y_{1,ref}],$ 
2:  $[v_{1,k-1}] = [v_{1,ref}(0)], [\psi_{1,k-1}] = [\psi_{1,ref}(0)]$ 
3:  $[{}^1b_{k-1}^{1\dots n_b}] = [0, 5], \varepsilon$  (Tolérance de la contraction)
4: for  $t = 1, \dots, T_{max}$  do
5:    $[x_{2,k-1}], [y_{2,k-1}], [v_{2,k-1}], [\psi_{2,k-1}]$ 
6:    $[x_{1,k}] = [x_{1,k-1}] + T_e[v_{1,k-1}] \cos[\psi_{1,k-1}]$ 
7:    $[y_{1,k}] = [y_{1,k-1}] + T_e[v_{1,k-1}] \sin[\psi_{1,k-1}]$ 
8:    $[x_{2,k}] = [x_{2,k-1}] + T_e[v_{2,k-1}] \cos[\psi_{2,k-1}]$ 
9:    $[y_{2,k}] = [y_{2,k-1}] + T_e[v_{2,k-1}] \sin[\psi_{2,k-1}]$ 
10:  for  $j = 1, \dots, n_b$  do
11:     $[b_k^j] = [b_{k-1}^j]$ 
12:  end for
13:  while  $max(width([{}^1\mathbf{x}_k])) < \varepsilon$  do
14:    for  $j = 1, \dots, n_b$  do
15:       $[{}^1\mathbf{x}_k] = \text{Solve Forward-Backward CSP} : \{\mathcal{H} : F(\mathbf{x}) = 0\}$ 
16:    end for
17:  end while
18:  Broadcast ( $[{}^1\mathbf{x}_k], [v_{1,k}], [\psi_{1,k}]$ )
19:  if Received ( $[{}^2\mathbf{x}_k], [v_{2,k}], [\psi_{2,k}]$ ) then
20:     $[{}^1\mathbf{x}_k] = [{}^1\mathbf{x}_k] \cap [{}^2\mathbf{x}_k]$  (Contraction des positions et biais)
21:  end if
22: end for

```

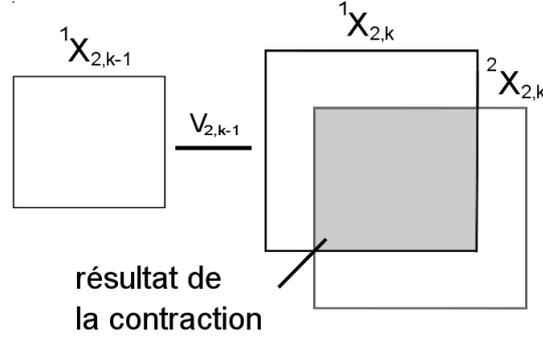


Figure 3.12 – Contraction de la position du véhicule 2 estimée dans le véhicule 1 et la position reçue du véhicule 2 estimée dans le véhicule 2.

3.5.3 Résultats

On présente les résultats des deux scénarii : Standalone (S) et Coopératif (C). L'objectif est de quantifier le gain dû à la coopération en termes d'exactitude et d'incertitude des méthodes en comparant respectivement les erreurs horizontales de positionnement (*Horizontal Positioning Errors* HPE) et la taille de domaine de confiance (*Confidence Domain Size* CDS). Les HPE sont calculées en utilisant les centres des boîtes de positions comme suit :

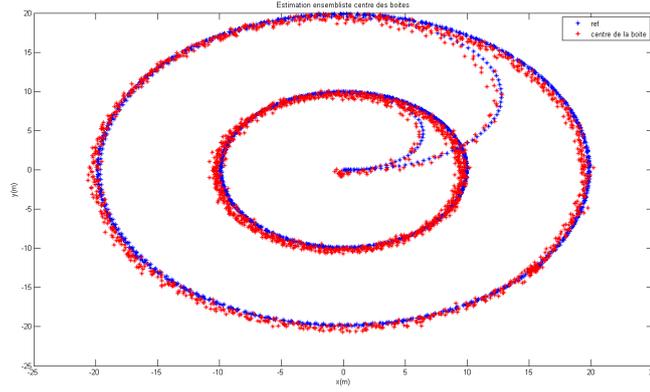
$$HPE = \sqrt{(\hat{x}c_i - x_{i,ref})^2 + (\hat{y}c_i - y_{i,ref})^2} \quad (3.20)$$

où $(\hat{x}c_i, \hat{y}c_i)$ et $(x_{i,ref}, y_{i,ref})$ sont respectivement les coordonnées de position estimée du centre de la boîte et les coordonnées de la position référence du robot considéré R_i .

Le CDS est le volume de la boîte estimée (Eq. 3.21). Il est considéré comme un indicateur de niveau d'incertitude de chaque méthode.

$$CDS = \text{width } [x_i, \bar{x}_i] \times \text{width } [y_i, \bar{y}_i] \quad (3.21)$$

L'algorithme est testé sur deux véhicules effectuant deux trajectoires circulaires de rayons différents ($r_1 = 10m$, $r_2 = 20m$). La figure 3.13 montre les trajectoires des véhicules en précisant la référence et les centres des boîtes de positions estimées.


Figure 3.13 – Trajectoires des 2 véhicules.

L'implémentation de l'algorithme est réalisée par la librairie d'analyse par intervalle IBEX en respectant la chronologie du temps réel, en C++ [105]. Le nombre des balises n_b est égale à 40. Les biais sont considérés comme des valeurs aléatoires positives entre 0 et 5. Le temps de la simulation choisi est 1000 secondes. Les valeurs des bruits sur les mesures sont indiquées dans le tableau 3.1 :

Variables	Bornes des variables mesurées
$[v_k]$	$[v_k - \bar{\gamma}_v , v_k + \bar{\gamma}_v]$; $\bar{\gamma}_v = \text{bruit uniforme} \times 0,28 \text{ m/s}$
$[\psi_k]$	$[\psi_k - \bar{\gamma}_\psi , \psi_k + \bar{\gamma}_\psi]$; $\bar{\gamma}_\psi = \text{bruit uniforme} \times 0.1 \text{ rad}$
$[r_i^j]$	$[r_i^j - \beta , r_i^j + \beta]$; $\beta = \text{bruit uniforme} \times 1 \text{ m}$

Table 3.1 – Distribution des variables de bruit.

Les fonctions de distribution cumulatives empiriques F des erreurs de la position sont présentées dans les figures 3.18-3.19 et les domaines de confiance (CDS) de positions sont présentées dans les figures 3.21-3.22. La position et le domaine de confiance d'un véhicule sont respectivement le centre et le volume de la boîte de position estimée. La méthode est garantie et la vraie valeur de toutes les variables (position des véhicules et biais) est toujours incluse dans les boîtes estimées. Les bornes des erreurs selon les axes x et y sont respectivement $\bar{x}_i - x_{i,ref}$ et $\bar{y}_i - y_{i,ref}$. Les figures des erreurs de positions 3.16-3.17 prouvent que le filtre ensembliste coopératif est intègre puisque la valeur "0" est toujours incluse dans les bornes des erreurs de positions. Cela signifie que les boîtes estimées de la position avec la méthode coopérative garantissent les positions réelles des véhicules. En ce qui concerne les y_i estimations de la méthode standalone, des pertes d'intégrité sont observées à quelques instants dues au fait que la méthode ne tient pas compte des biais des mesures. Les résultats sont obtenus avec des petites valeurs de bruit comme il est indiqué dans le tableau (3.1). Les mêmes paramètres de bruit sont employés pour les deux filtres.

3.5 Localisation coopérative avec des mesures des distances

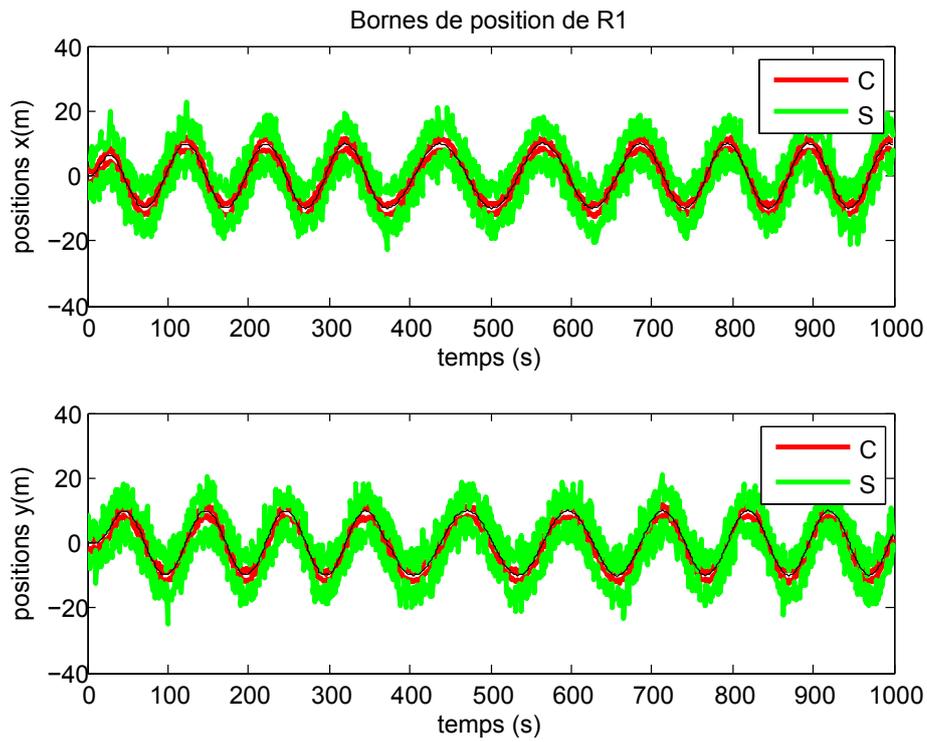


Figure 3.14 – Bornes de positions de R_1 .

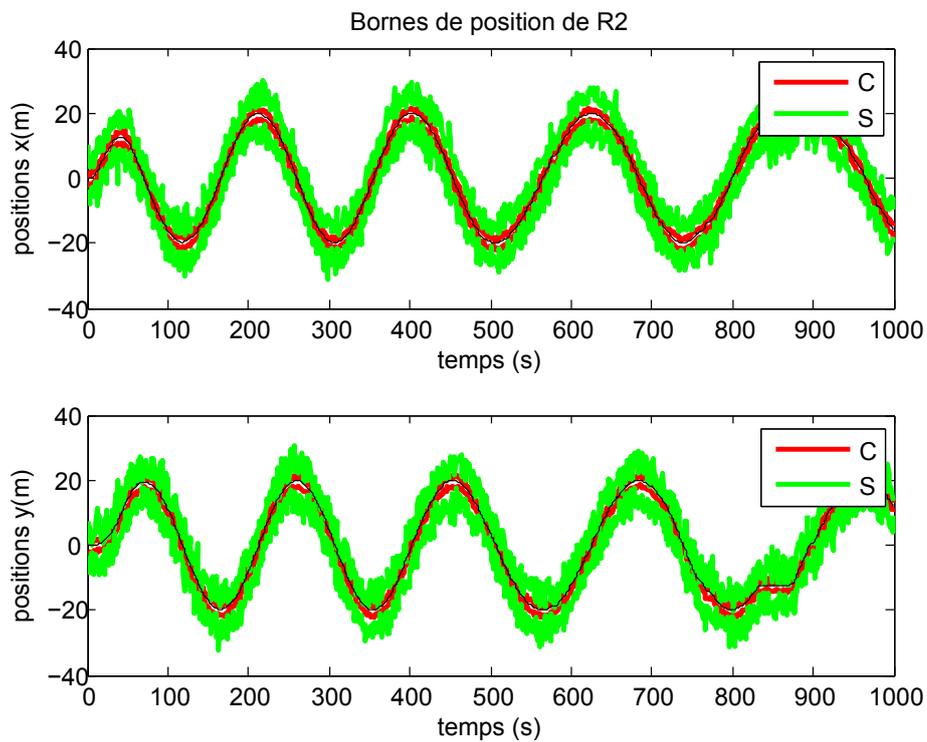


Figure 3.15 – Bornes de positions de R_2 .

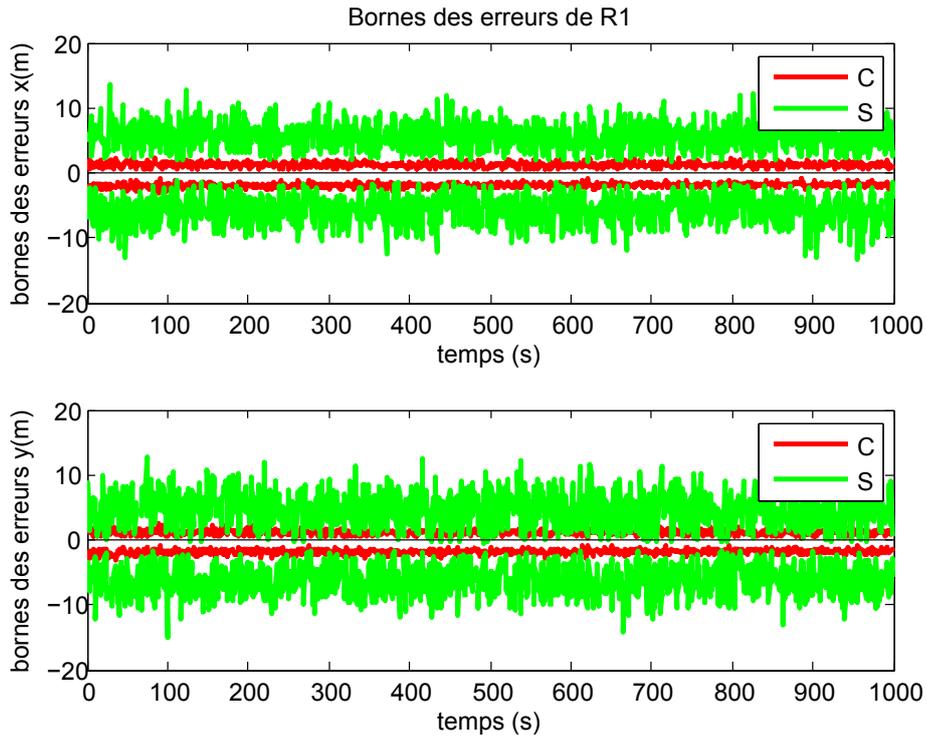


Figure 3.16 – Bornes des erreurs de positions de R_1 .

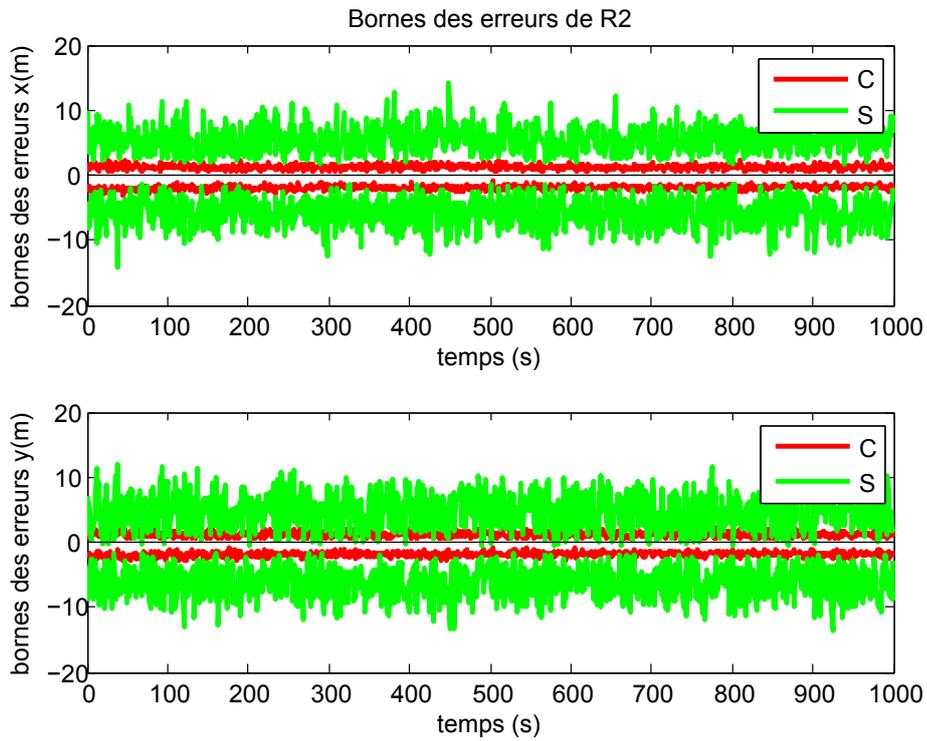


Figure 3.17 – Bornes des erreurs de positions de R_2 .

3.5 Localisation coopérative avec des mesures des distances

Comme représenté dans la figure 3.18, 95% des erreurs de position de véhicule 1 sont inférieures à 0.93 m et 4.06 m en utilisant respectivement la méthode (C) et la méthode (S). Dans la figure 3.19, 95% des erreurs de position de véhicule 2 sont inférieures à 0.68 m et 1.13 m lorsqu'on utilise respectivement les méthodes (C) et (S). Ceci montre que la coopération apporte des améliorations significatives à la méthode (S) en terme d'incertitude pour les 2 véhicules considérés.

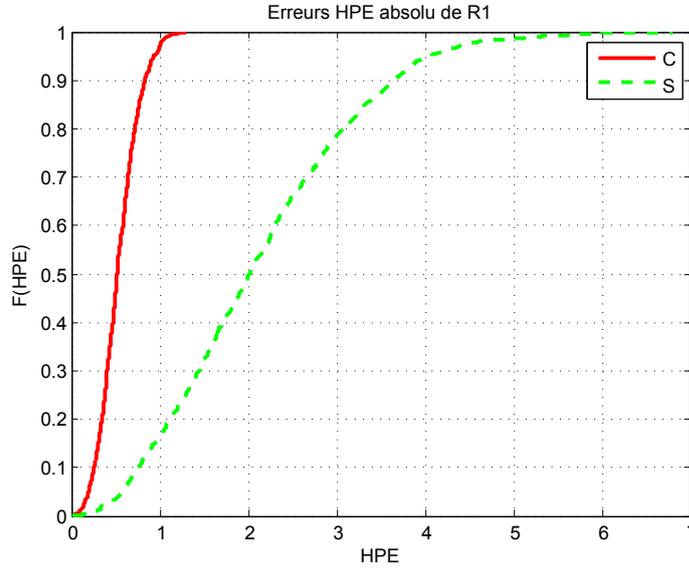


Figure 3.18 – Erreurs HPE absolue de R_1 .

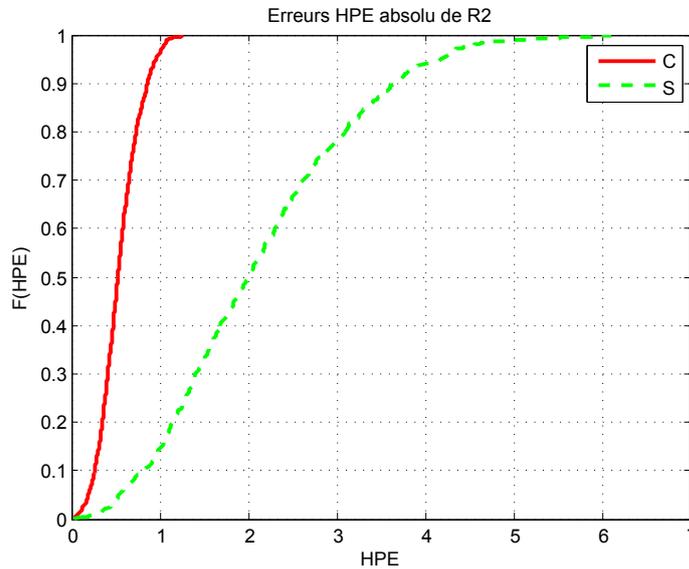


Figure 3.19 – Erreurs HPE absolue de R_2 .

Dans la figure 3.20, 95% des erreurs de la position relative sont inférieures à 0.79 m et 5.39 m en utilisant respectivement les méthodes (C) et (S). En corrigeant les

erreurs des mesures communes, une réduction d'incertitude de 85.34% est obtenu au niveau du positionnement relatif.

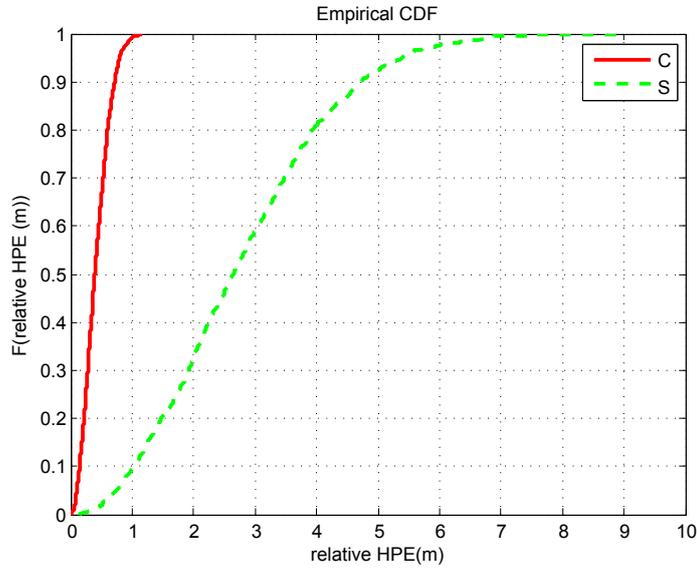


Figure 3.20 – Distance relative de véhicules.

Comme le montre la figure 3.21, 95% des domaines de confiance de R1 sont inférieurs à $14,27 m^2$ et $259 m^2$ en utilisant respectivement les méthodes (C) et (S). Dans la figure, 95% des domaines de confiance de R2, sont inférieurs à $14,14 m^2$ avec la méthode (C) et $247.6 m^2$ avec la méthode (S). Grâce à la coopération, les incertitudes de domaine de confiance de la position de deux véhicules sont significativement réduites (94.49% et 94.28%).

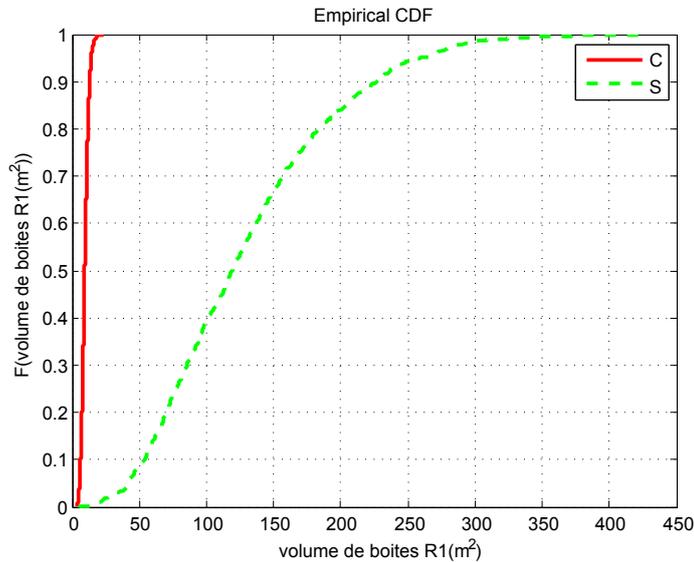


Figure 3.21 – Volume de domaine de confiance de R_1 .

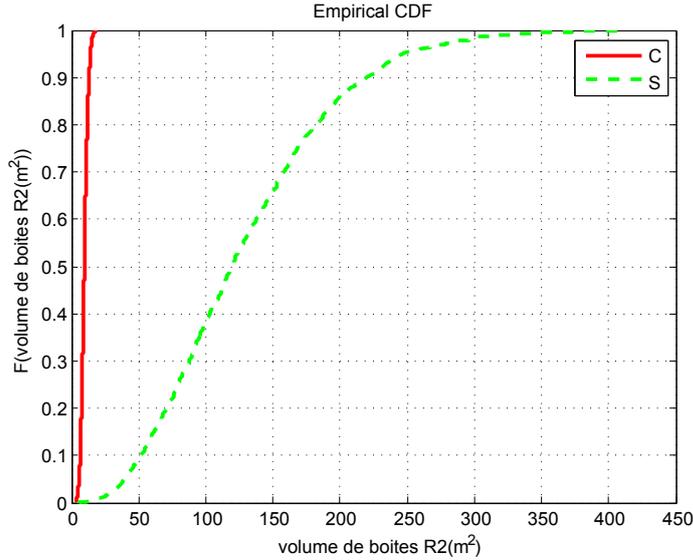


Figure 3.22 – Volume de domaine de confiance de R_2 .

D'autres performances statistiques des deux méthodes testées sont fournies dans le tableau 3.2. Par exemple, l'écart type des erreurs HPE est réduit d'environ de 80% pour les 2 véhicules en considérant la coopération.

		HPE R1	HPE R2	Distance Relative
écart type. (m)	C	0.22	0.22	0.21
	S	1.11	1.09	1.43
max (m)	C	1.29	1.27	1.16
	S	6.80	6.11	9.04
moyenne (m)	C	0.53	0.54	0.41
	S	2.13	2.13	2.79

		CDS R1	CDS R2
95ème percentile (m^2)	C	14.27	14.14
	S	259	247.6

Table 3.2 – Statistiques des erreurs.

Les biais sur chaque mesure de distances est initialisé à $[0, 5]$. Lors de la simulation on a imposé des biais positifs entre 0 et 5 sur les mesures de distance. La figure 3.23 présente les bornes des 10 premiers biais estimés dans le véhicule R1. Pour chaque sous-figure de la figure 3.23 l'axe des x exprime le temps en seconde et l'axe des y présente les bornes de l'intervalle de biais estimées $[b^j]$ en mètres où $j \in 1, \dots, n_b$. Il est à noter que les biais sont contractés. Les bornes estimées des biais contiennent la vraie valeur des erreurs de mesures de distances. Le filtre coopératif ensembliste garantit toutes les valeurs réelles de vecteur d'état de chaque véhicule.

Les 30 autres intervalles de biais estimés ont prouvé que la méthode est garantie et que le filtre converge vers les valeurs réelles. Cette convergence met en évidence l'utilité de l'analyse d'observabilité considérée dans le chapitre 3.

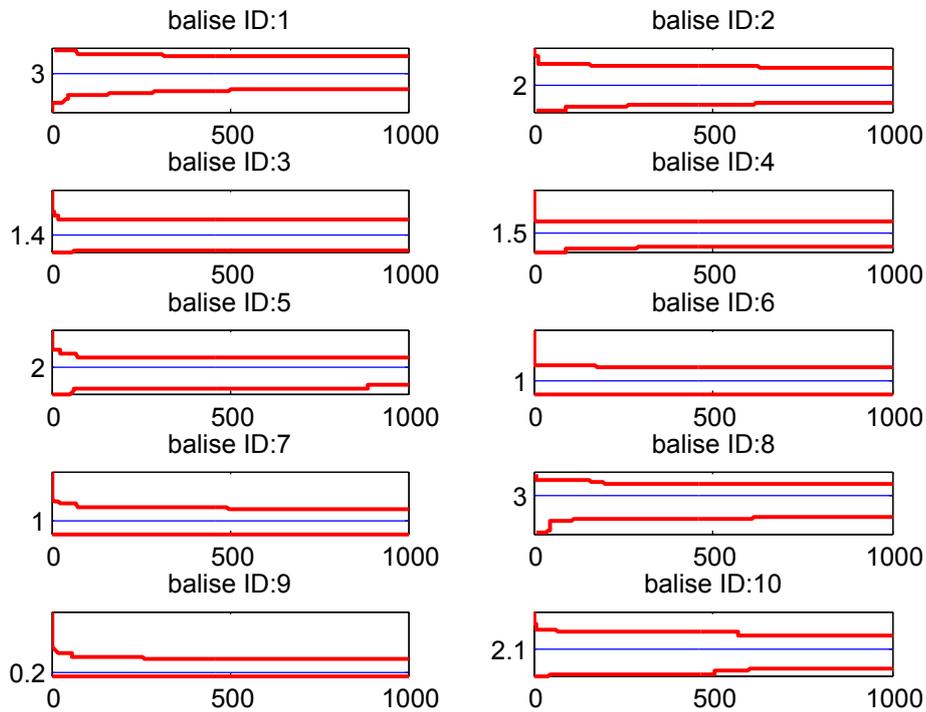


Figure 3.23 – Les 10 premiers biais de R_1 (unités des axes x et y sont secondes (s) et mètre (m)).

L'algorithme 3.5 proposé repose uniquement sur la propagation de contraintes comme solver ce qui semble être efficace quand des données simulées sont utilisées. Le solveur de propagation des contraintes (p) est très rapide, il peut être implémenté en temps réel. Il existe d'autres solveurs plus puissants comme SIVIAP (algorithme 3.3) qui peut être adapté au problème de l'estimation coopérative. Les résultats de la coopération avec deux véhicules sont plus précis que la méthode non coopérative, mais on se demande si la localisation de chaque véhicule ne peut pas être plus précise s'il coopère avec plus qu'un seul partenaire.

3.5.4 Amélioration de l'algorithme coopératif

Dans cette section, les améliorations de l'algorithme 3.5 proposé précédemment seront présentées. Ces améliorations consistent en l'utilisation du solveur SIVIAP dans le but d'avoir des meilleures performances du positionnement en utilisant d'abord deux robots ensuite dix robots. Les trois algorithmes utilisés seront notés : C_p pour l'algorithme 3.5 basé sur la propagation de contraintes, $C_{Sviap2R}$ pour l'algorithme

basé sur SIVIAP avec deux robots et $C_{Sviap10R}$ pour l'algorithme basé sur SIVIAP avec dix robots.

Les modifications de l'algorithme C_p sont d'abord présentées, ensuite on abordera la comparaison des performances de trois algorithmes en prenant comme exemple le véhicule 1.

3.5.4.1 Amélioration avec SIVIAP et augmentation du nombre de robots

L'algorithme SIVIAP est appliqué avec les modifications suivantes.

La solution envisagée est la boîte englobante de $\overline{\mathbb{X}}$, qui est l'union de sous pavages intérieurs et indéterminés $\underline{\mathbb{X}}$ et $\Delta\mathbb{X}$ comme il est indiqué dans la figure 3.5. Afin de réduire les bisections, nous limitons le temps de calcul à 1 s pour chaque mise à jour avec les mesures de distances des balises.

L'étape 15 de l'algorithme 3.5 est remplacée par le solveur SIVIAP. Les entrées de l'algorithme SIVIAP sont l'ensemble des contraintes CSP de l'équation (3.19), les mesures de distances $[{}^i\mathbf{y}]$, le vecteur d'état qui contient les domaines de variables à estimer $[{}^i\mathbf{x}]$ et la précision des boîtes à bissecter est $\varepsilon = 1 m$. La sortie fournie par SIVIAP est l'union des sous pavages intérieurs et indéterminés de la solution $[{}^i\mathbf{x}]$. Le nouvel algorithme est moins rapide que l'ancien mais il reste adapté en temps réel.

D'autres simulations de mesures sont effectuées pour tester le nouvel algorithme coopératif avec 10 véhicules. Le même scénario est testé où les 10 véhicules réalisent des trajectoires circulaires de différents rayons ($r_1 = 10m, r_2 = 20m, r_3 = 30m, r_4 = 40m, r_5 = 50m, r_6 = 60m, r_7 = 130m, r_8 = 80m, r_9 = 90m, r_{10} = 100m$). La figure 3.24 illustre le scénario de coopération et présente les boîtes de positions estimées de 10 véhicules.

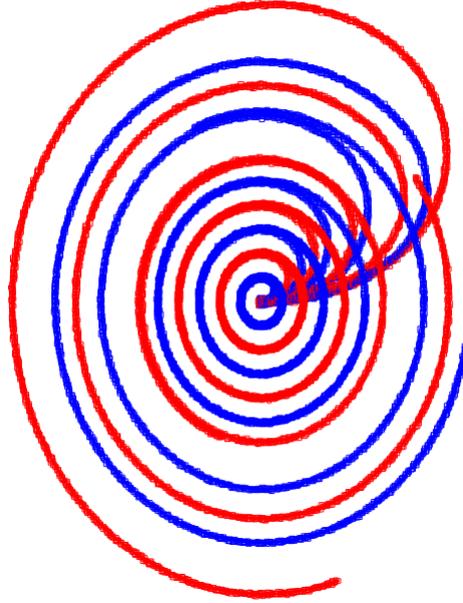


Figure 3.24 – Boîtes estimées de position des 10 véhicules.

3.5.4.2 Comparaison des algorithmes C_p , $C_{Sviap2R}$ et $C_{Sviap10R}$

La comparaison des résultats de l'algorithme modifié par SIVIAP en utilisant 2 et 10 véhicules sont présentés sur les figures 3.25 et 3.26.

D'abord, on peut noter que SIVIAP permet de réduire les erreurs HPE (figure 3.25) par rapport à la méthode n'utilisant que de la propagation de contrainte. L'estimation des biais est également meilleure (figure 3.26). L'utilisation de l'inversion ensembliste avec la propagation de contraintes améliore donc les résultats de l'algorithme basé uniquement sur la propagation de contraintes. Ainsi, la méthode SIVIAP sera utilisée dans la suite de cette thèse, en particulier pour la localisation avec des mesures de pseudodistances.

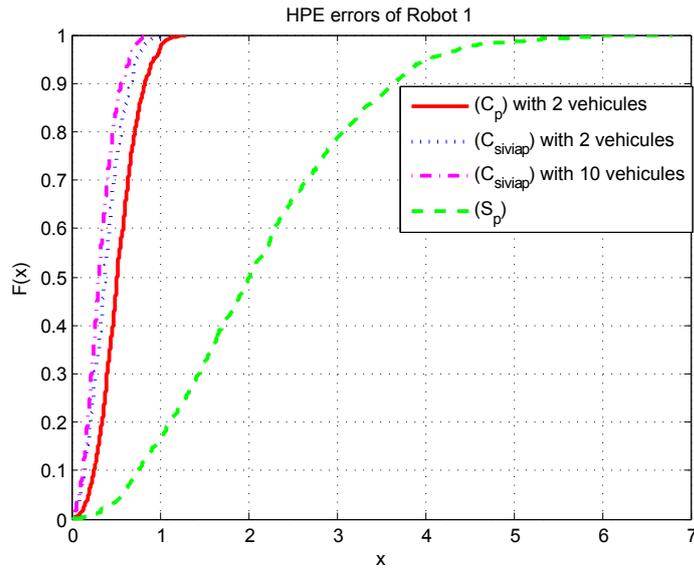


Figure 3.25 – Erreurs HPE absolues de R_1 des méthodes coopératives.

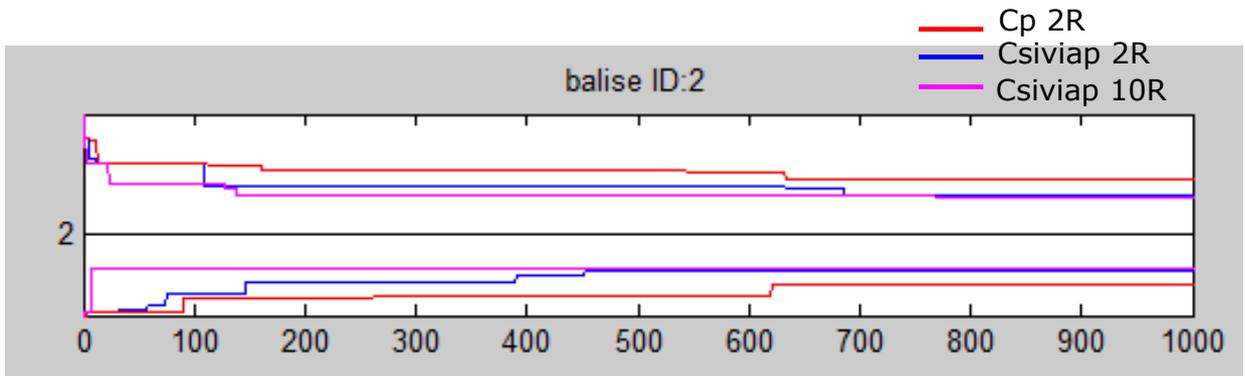


Figure 3.26 – Biais de R_1 sur la mesure de la balise 2. Comparaison des méthodes coopératives.

Intéressons-nous à voir maintenant l'effet de l'augmentation du nombre de robots.

Le tableau 3.3 donne les performances statistiques des trois tests.

		HPE R1	Distance Relative
écart type. (m)	$C_{SIV IAP2R}$	0.19	0.22
	$C_{SIV IAP10R}$	0.16	0.20
max (m)	$C_{SIV IAP2R}$	1.06	1.45
	$C_{SIV IAP10R}$	0.94	1.16
moyenne (m)	$C_{SIV IAP2R}$	0.39	0.40
	$C_{SIV IAP10R}$	0.33	0.39

		CDS R1
95ème percentile (m^2)	$C_{2SIV IAPR}$	10.39
	$C_{SIV IAP10R}$	8.55

Table 3.3 – Statistiques des erreurs des méthodes coopératives.

D’après ces résultats, on observe que les performances sont meilleures avec dix robots. De plus, la méthode $C_{Sviap10R}$ réduit le 95% percentile de HPE de la méthode $C_{Sviap2R}$ de 11% (de $0.73m$ à $0.65m$) (cf figure 3.25). Cet ordre de grandeur se retrouve dans les chiffres du tableau 3.3. En ce qui concerne la réduction d’incertitude, elle est de 17% avec 10 robots ce qui reste du même ordre de grandeur.

3.6 Localisation coopérative avec des mesures de pseudodistances

Dans cette partie, on s’intéresse au problème de localisation avec des mesures GNSS. Le principe de l’algorithme, reposant sur la coopération par correction des erreurs de mesure en commun, reste le même. Nous proposons une formulation de la coopération multi-véhicules qui repose sur les méthodes d’inversion ensembliste combinées avec des techniques de CSP sur des intervalles. Nous supposons que les véhicules sont en mesure de coopérer et d’échanger des informations telles que chaque véhicule peut calculer les positions des partenaires avec des domaines fiables. La contribution majeure est de montrer qu’il est possible d’améliorer la position absolue et mutuelle entre les véhicules en partageant des pseudodistances GNSS biaisées. Le scénario coopératif est illustré par la figure 3.27. La méthode développée permet la fusion de données des biais estimés de manière distribuée sans unité centrale de fusion et sans station de base. Une validation de l’algorithme avec deux véhicules expérimentaux est effectuée pour évaluer les performances en conditions réelles.

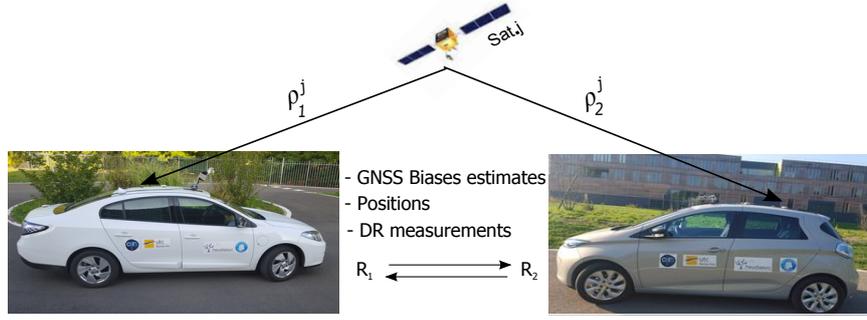


Figure 3.27 – Les véhicules expérimentaux et les données partagées.

La fusion de données de mesures GNSS avec mesures DR est souvent adoptée pour améliorer la précision de la localisation. Les modèles mathématiques des mesures sont décrits ci-après.

3.6.1 Modèles

Les mesures brutes de GNSS considérées ici sont les pseudodistances acquises sur la porteuse L1. La pseudodistance ρ_i^j [83] est le trajet entre chaque véhicule R_i ($i = \{1, \dots, n_r\}$) situé aux coordonnées $\mathbf{p}_i = [x_i, y_i, z_i]$ et chaque satellite j ($j = 1, \dots, n_s$) de position $\mathbf{p}^j = [x^j, y^j, z^j]$. L'erreur commune b^j est modélisée comme une erreur additive sur la mesure de pseudodistance. Le modèle de ρ_i^j est exprimée par l'Eq. (3.22).

$$\rho_i^j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} + b^j + d_i + \beta^j \quad (3.22)$$

où d_i représente le décalage d'horloge du récepteur et β^j désigne le bruit de mesure. soit ${}^i\mathbf{y} \in \mathbb{R}^{n_s}$ le vecteur de n_s mesures de pseudodistances et ${}^i\mathbf{x} \in \mathbb{R}^n$ le vecteur d'état du véhicule i . Le modèle d'observation à temps discret k est défini comme suit

$${}^i\mathbf{y}_k = \mathbf{g}({}^i\mathbf{x}_k) \quad (3.23)$$

Chaque véhicule est équipé de capteurs pour mesurer l'angle de cap et la vitesse linéaire. Ces mesures de DR sont considérées comme un vecteur d'entrée ${}^i\mathbf{u}(t) = [v_i \ \psi_i]^T$. Un modèle d'évolution classique de la position et un modèle auto-régressif (AR) de biais [106] sont décrits à l'instant t par la fonction continue \mathbf{f}

dans un repère local East North Up (ENU) :

$$f = \begin{cases} \dot{x}_i = v_i \cos(\psi_i) \\ \dot{y}_i = v_i \sin(\psi_i) \\ \dot{z}_i = 0 \\ \dot{d}_i = dr_i \\ \dot{d}r = 0 \\ \dot{b}^j = a b^j \end{cases} \quad (3.24)$$

où ψ_i est l'angle d'orientation doit être mesuré avec une bonne précision . v_i note la vitesse linéaire mesurée des véhicules. d_i et dr_i sont respectivement le décalage d'horloge du récepteur et sa dérive à estimer. $a = e^{-T_e/\tau}$, T_e étant la période d'échantillonnage et τ ($\tau = 6.6667$ min) indique la constante de temps du modèle auto-régressif du biais b^j ($a = 0.9995$). Les indices des véhicules et des satellites communs sont désignés respectivement par $i \in \{1 \dots n_r\}$ et $j \in \{1 \dots n_s\}$.

Pour une mise en œuvre pratique, le modèle d'évolution peut être écrit sous une forme vectorielle discrète comme suit (où \mathbf{f}_k est la fonction discrète de \mathbf{f} , présentée dans l'Eq. (3.26))

$${}^i \mathbf{x}_k = \mathbf{f}_k({}^i \mathbf{x}_{k-1}, {}^i \mathbf{u}_{k-1}) \quad (3.25)$$

$$f = \begin{cases} x_{i,k} = x_{i,k-1} + T_e v_{i,k-1} \cos \psi_{i,k-1} \\ y_{i,k} = y_{i,k-1} + T_e v_{i,k-1} \sin \psi_{i,k-1} \\ z_{i,k} = z_{i,k-1} \\ b_k^j = a b_{k-1}^j + \alpha_k^{b_j} \\ d_{i,k} = d_{i,k-1} + T_e \dot{d}_{i,k-1} + \alpha_k^d \\ \dot{d}_{i,k} = \dot{d}_{i,k-1} + \alpha_k^{\dot{d}_i} \end{cases} \quad (3.26)$$

3.6.2 Formulation du problème

L'objectif est de contracter tous les domaines des variables d'état des véhicules sans retirer aucune solution possible. Décrivons maintenant les variables, les domaines et les contraintes du CSP considérés.

Variables.

Dans chaque agent R_i , il y a un état propre (ego) à estimer et une position pistée à prédire qui concerne les autres véhicules R_o qui constituent le groupe, où $o \in \{1, \dots, n_r - 1\}$ et $o \neq 0$.

État ego :

soit

$$\nu = \{x_i, y_i, z_i, b^1, \dots, b^{n_s}, d, \dot{d}_i\}$$

les variables de l'état propre ${}^i\mathbf{x}_{ego}$ de chaque véhicule R_i de dimension $n = 5 + n_s$. On ne considère que ces variables dans le CSP.

On note l'état absolu par ${}^i\mathbf{q} = \{x_i, y_i, z_i\}$, les biais de tous les n_s satellites en vue par ${}^i\mathbf{b} = \{b^1, \dots, b^{n_s}\}$ et les variables intérieures de R_i par ${}^i\boldsymbol{\xi} = \{d_i, \dot{d}_i\}$ qui contiennent respectivement le décalage d'horloge d_i et sa dérive \dot{d}_i .

position pistée :

soit

$${}^i\mathbf{q}_o = \{{}^i\mathbf{q}_1, \dots, {}^i\mathbf{q}_{n_r-1}\}$$

la position des autres véhicules dans le groupe.

L'entrée de R_i est désignée par

$${}^i\mathbf{u}_{ego} = \{v_i, \psi_i\}$$

où v_i et ψ_i sont respectivement les mesures de la vitesse linéaire et le cap mesuré.

$${}^i\mathbf{u}_o = \{{}^i\mathbf{u}_1, \dots, {}^i\mathbf{u}_{n_r-1}\}$$

représente l'entrée de R_o constituée de ${}^i\mathbf{u}_{o=1\dots n_r-1} = \{v_{1\dots n_r-1}, \psi_{1\dots n_r-1}\}$.

Domaines.

Les domaines des variables sont des ensembles qui contiennent la valeur réelle des variables présentées par des intervalles pour la simplicité. Par exemple, les états des véhicules sont des vecteurs intervalles de \mathbb{R}^n présentés comme suit $[{}^i\mathbf{x}_{ego}] = \left[\begin{matrix} [{}^i\mathbf{q}]^T & [{}^i\mathbf{b}]^T & [{}^i\boldsymbol{\xi}]^T \end{matrix} \right]^T$ qui comporte les domaines des variables $[\nu] = [\nu_{true} - \delta_\nu, \nu_{true} + \delta_\nu]$, où ν_{true} représente la vraie valeur de ν et δ_ν indique la borne d'erreur.

Contraintes.

Détaillons maintenant les contraintes qui relient toutes les variables ν à chaque instant k comme étant (i)

$${}^i\mathbf{x}_{ego,k} = \mathbf{f}({}^i\mathbf{x}_{ego,k-1}, {}^i\mathbf{u}_{ego})$$

et (ii)

$${}^i\mathbf{y}_k = \mathbf{g}({}^i\mathbf{x}_{ego,k}).$$

La contrainte (i) contraint chaque véhicule à satisfaire les équations d'évolution (Eq. 3.25). La contrainte (ii) définit le CSP (Eq. 3.23) qui est utilisé dans SIVIAP considérant ses entrées \mathbf{g} comme la fonction d'inclusion, ${}^i\mathbf{y}_k$ l'ensemble de mesures à inverser et ${}^i\mathbf{x}_{ego,k}$ comme la boîte initiale de recherche.

Solveur.

SIVIAP approche le vecteur d'état ${}^i\mathbf{x}_{ego,k}$ tel que $\mathbf{g}({}^i\mathbf{x}_{ego,k}) \subseteq [{}^i\mathbf{y}_k]$ en utilisant les contracteurs de propagation et rétropropagation. L'ensemble de CSP est défini comme suit (Eq. 3.27)

$$\begin{aligned} CSP &= \{ {}^i\mathbf{x}_{ego,k} \subseteq [{}^i\mathbf{x}_{ego,k}] / \mathbf{g}({}^i\mathbf{x}_{ego,k}) \subseteq [{}^i\mathbf{y}_k] \} \\ &= \mathbf{g}^{-1}([{}^i\mathbf{y}_k]) \cap [{}^i\mathbf{x}_{ego,k}] \end{aligned} \quad (3.27)$$

3.6.3 Algorithme de localisation coopérative

L'algorithme coopératif proposé est présenté dans 3.6. L'agent R_i prédit son état ego ${}^i\mathbf{x}_{ego}$ à l'aide du modèle d'évolution de l'Eq. (3.25) et ses données DR (v_i, ψ_i) mesurés à chaque 0.01s (ligne 1). Ensuite, chaque véhicule i suit les autres véhicules (ligne 2) en utilisant les dernières mesures DR reçues et le modèle d'évolution de la position présenté dans l'Eq. (3.26).

Les lignes (4...14) de l'algorithme consiste à mettre à jour la prédiction de l'état ${}^i\mathbf{x}_{ego}$ par rapport aux mesures de GNSS qui sont disponibles chaque 0.2s. Afin d'écartier les valeurs aberrantes à chaque instant k où les mesures GNSS sont disponibles, un processus de validation des mesures de chaque satellite est réalisée. Pour chaque mesure de pseudodistance, nous vérifions si le SNR (Signal to Noise Ratio) du satellite est suffisamment élevé (e.g. 35 dB/Hz). Ensuite, on applique l'algorithme de SIVIAP présenté dans [107] avec les modifications suivantes.

La solution considérée est la boîte englobante de $\overline{\mathbb{X}}$, qui est l'union de sous pavages intérieur et indéterminé $\underline{\mathbb{X}}$ et $\Delta\mathbb{X}$ comme il est indiqué dans la figure 3.5. Afin d'arrêter les bisections, nous limitons le temps de calcul à 0.1 s pour chaque époque (temps de chaque mesure GNSS de pseudodistance).

Algorithme 3.6 une itération de l'algorithme coopératif dans iR

Coopération (in : $[{}^i\mathbf{x}_{ego}], [{}^i\mathbf{u}_{ego}], [\rho_1^1, \dots, \rho_1^{n_s}], [{}^o\mathbf{q}], [{}^o\mathbf{u}_{ego}], [{}^o\mathbf{b}], \mathbf{g}$; out : $[{}^i\mathbf{x}_{ego}], [{}^i\mathbf{q}_o]$)

- 1: $[{}^i\mathbf{u}_{ego}] = [v_i, \psi_i^m]^T = \mathbf{Get}(\text{DR measurements})$
- 2: **Prediction** (in : $[{}^i\mathbf{u}_{ego}]$; in out : $[{}^i\mathbf{x}_{ego}]$)
- 3: **Track** (in : $[{}^i\mathbf{q}_o], [{}^i\mathbf{u}_o]$; out : $[{}^i\mathbf{q}_o]$)
- 4: **if** New GNSS data is available **then**
- 5: $n_s =$ number of visibles satellites
- 6: $[\rho_1^1, \dots, \rho_1^{n_s}] = \mathbf{Get}(\text{GNSS measurements})$
- 7: Good_Pr = \emptyset
- 8: **for** $j = 1, \dots, n_s$ **do**
- 9: **if** (ρ^j is good) **then**
- 10: Add(ρ^j) to the Good_Pr list ($[{}^i\mathbf{y}_{good}]$)
- 11: **end if**
- 12: **end for**
- 13: **SiviaP** (in : $\mathcal{C}_{S_p}, [{}^i\mathbf{y}_{good}], \mathbf{g}, \varepsilon$; in out : $[{}^i\mathbf{x}_{ego}]$)
- 14: **end if**
- 15: **Communication**(in : dataS; out : dataR)
- 16: **Track_update**(in : $[{}^o\mathbf{q}]$; out : $[{}^i\mathbf{q}_o]$)
- 17: **Fusion**(in out : $[{}^i\mathbf{b}], [{}^o\mathbf{b}]$)

Le retard de la communication (ligne 15) n'est pas crucial puisque les véhicules ont synchronisé leurs horloges grâce aux récepteurs GNSS. Les données envoyées (dataS) et reçues (dataR) à chaque instant k par chacun sont l'égo position, la dernière estimation des biais, les mesures courantes de DR et l'identifiant id dans le groupe. La quantité de l'information transmise sur le réseau de communication est faible, étant donné que les véhicules n'échangent que les bornes inférieures et supérieures des boîtes.

Dans la ligne 16 de l'algorithme 3.6, chaque véhicule i met à jour la position pistée de R_o par la position estimée reçue $[{}^i\mathbf{q}_o] := [{}^o\mathbf{q}_{ego}]$. Enfin, R_i fusionne ses biais estimés avec ceux reçus des autres véhicules $R_o : [{}^i\mathbf{b}] = [{}^i\mathbf{b}] \cap [{}^o\mathbf{b}]$.

3.6.4 Résultats expérimentaux

Toujours pour quantifier le gain de la performance due à la coopération, les résultats de deux scénarios (Standalone (S) et coopératif (C)) sont présentés. L'approche proposée est testée sur deux véhicules expérimentaux (Fig. 3.27). Un récepteur de faible coût U-blox 4T fournissant des mesures de pseudodistances a été utilisé dans chaque véhicule. Un récepteur Polarcx Septentrio a été utilisé en mode RTK pour obtenir la vérité terrain et une mesure précise de cap. Une passerelle au bus CAN a permis de récupérer les vitesses linéaires v des véhicules. On considère donc, dans un premier temps, un cas simplifié où $\mathbf{u}_{ego} = \begin{bmatrix} v & \psi \end{bmatrix}^T$ est le vecteur de mesures

proprioceptives. 10 satellites étaient en vue pendant l'essai et 6 d'entre eux étaient au moins en commun ce qui satisfait la condition nécessaire de l'observabilité abordé dans le chapitre 2. La visibilité des satellites GPS a été parfois très limitée du fait que la zone d'essai était entourée par des bâtiments et des arbres. L'essai a duré 3 minutes. L'algorithme a été implémenté en C++ à l'aide de la librairie d'analyse par intervalle IBEX [105]. La figure 3.29 présente une vue de dessus des trajectoires et les boîtes de position estimées des deux véhicules utilisant la méthode (C). La représentation de la solution est la boîte englobante $\overline{\mathbb{X}}$ de l'union des sous pavages internes et indéterminés : $\underline{\mathbb{X}} \cup \Delta \mathbb{X}$.

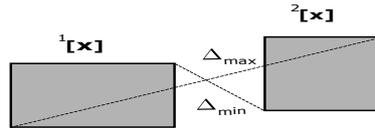


Figure 3.28 – Borne supérieure et inférieure relative ($\Delta_{min}, \Delta_{max}$). C'est un calcul simplifié de la distance relative.

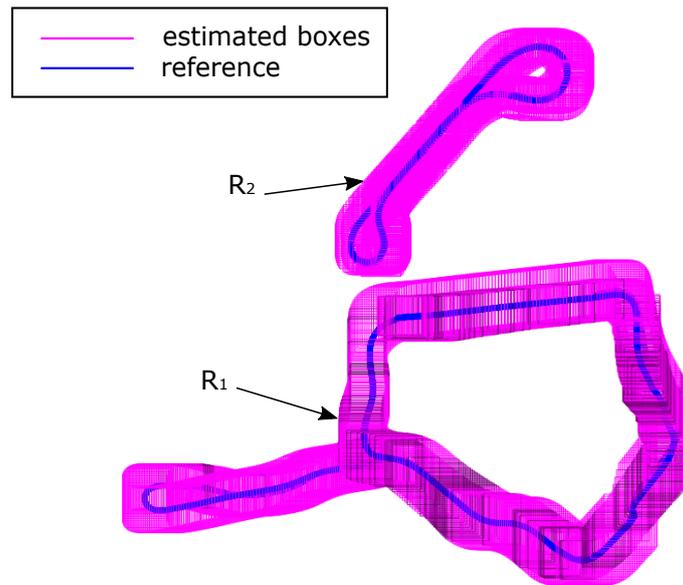


Figure 3.29 – Trajectoires et boîtes estimées de positions de véhicules.

Les figures 3.30 et 3.31 montrent les bornes des erreurs de position des véhicules respectivement selon les axes x et y avec les méthodes (C) et (S). Dès le premier regard, on peut vérifier l'intégrité des deux observateurs due au fait que les bornes contiennent toujours la valeur "0". Cela signifie que la position de référence RTK est toujours incluse dans les boîtes estimées (i.e. résultats consistants) indiquant un bon réglage des observateurs.

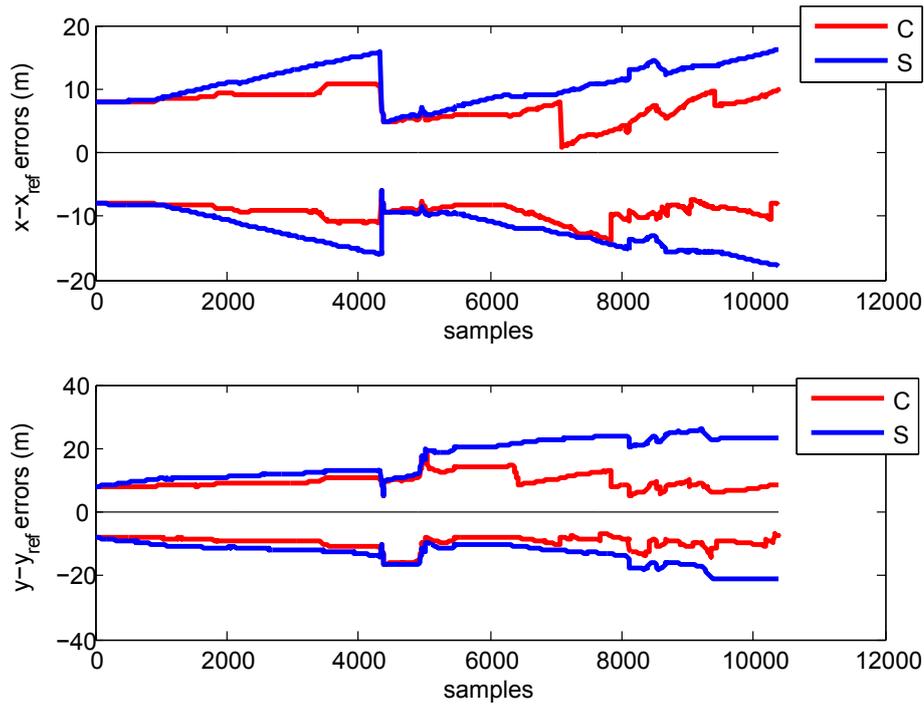


Figure 3.30 – Borne des erreurs de positions de R_1 .

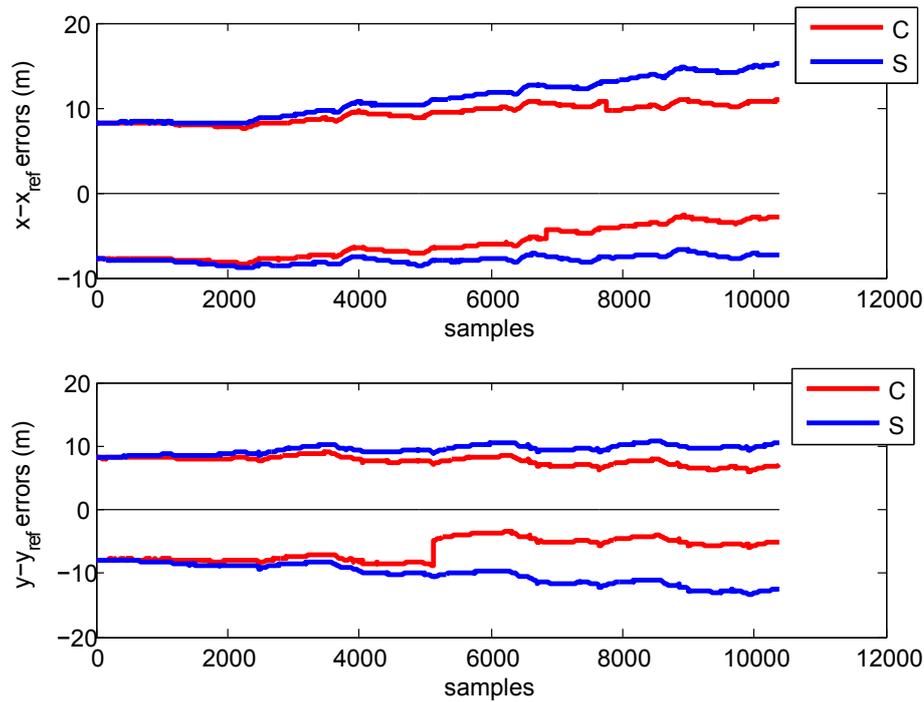


Figure 3.31 – Borne des erreurs de positions de R_2 .

La figure 3.32 présente les erreurs de distance relative (Relative Distance Error

RDE), les estimations ponctuelles présentent les centres des boîtes. Le calcul du centre d'une boîte est plus approprié en temps réel que celui du centre de gravité de sous pavage qui consomme plus de temps. Les erreurs RDE sont calculées de la façon suivante :

$$RDE = \sqrt{(\hat{d}_x - d_{x,ref})^2 + (\hat{d}_y - d_{y,ref})^2} \quad (3.28)$$

où $\hat{d}_x = (\hat{x}c_1 - \hat{x}c_2)$, $d_{x,ref} = x_{1,ref} - x_{2,ref}$, $\hat{d}_y = (\hat{y}c_1 - \hat{y}c_2)$ et $d_{y,ref} = y_{1,ref} - y_{2,ref}$. Grâce à la coopération, un gain de précision obtenu est de l'ordre de plusieurs mètres.

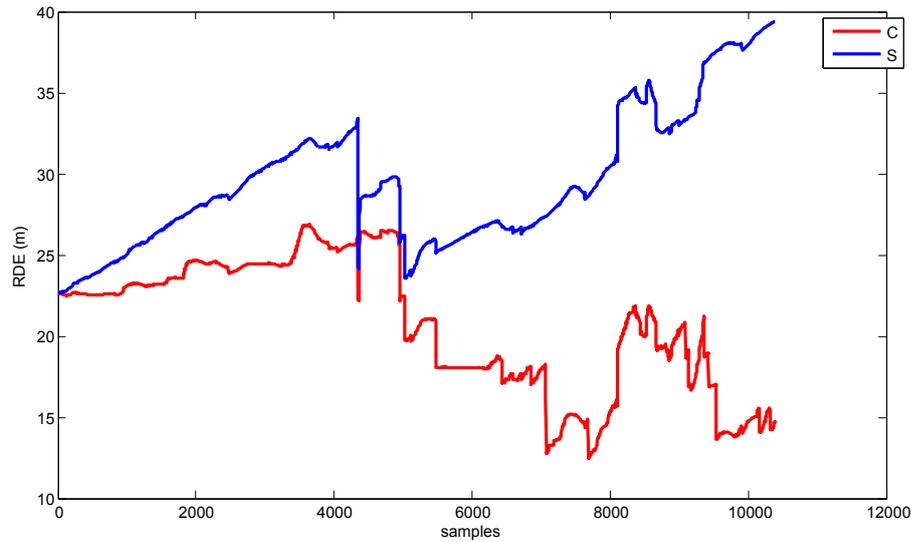


Figure 3.32 – Erreurs de distance relative RDE de deux véhicules.

La table 4.2 apporte quelques indicateurs de performance des deux méthodes en termes d'erreurs HPE absolues et relatives et les domaines de confiance (CDS) des boîtes résultantes. Les résultats de CDS sont présentés par la fonction de distribution cumulative (*Cumulative Distribution Function* : CDF) de volumes des boîtes de véhicules tout au long des trajectoires de la figure 3.33. Une amélioration nette est obtenue pour le véhicule R_1 en termes d'exactitude et d'incertitude. Par exemple, la moyenne des erreurs HPE est réduite de $2.41m$ à $1.67m$ et le CDS est 66% condensé grâce à la coopération puisque le 95ème percentile de CDS est inférieur à $466.6m^2$ comparé à $1390m^2$ avec la méthode S. En ce qui concerne le véhicule R_2 , l'amélioration des HPE n'est pas si pertinente que R_1 vu qu'on ne considère que le centre de boîte comme estimation et la contraction peut déformer la boîte et éloigner le centre de la référence. Le CDS cumulé est de 49% réduit puisque le 95ème percentile de CDS est inférieur à $256m^2$ comparé à $501.4m^2$ considérant la méthode S, ce qui signifie une amélioration substantielle de la méthode classique.

3.6 Localisation coopérative avec des mesures de pseudodistances

		HPE R1	HPE R2	Distance Relative
écart type (m^2)	C	1.69	1.41	2.46
	S	2.28	1.33	2.87
max (m)	C	6.02	4.30	8.87
	S	5.77	4.28	8.27
moyenne (m)	C	1.67	1.95	3.01
	S	2.41	1.76	3.66

Table 3.4 – Statistiques des HPE et CDS en utilisant la mesure ψ .

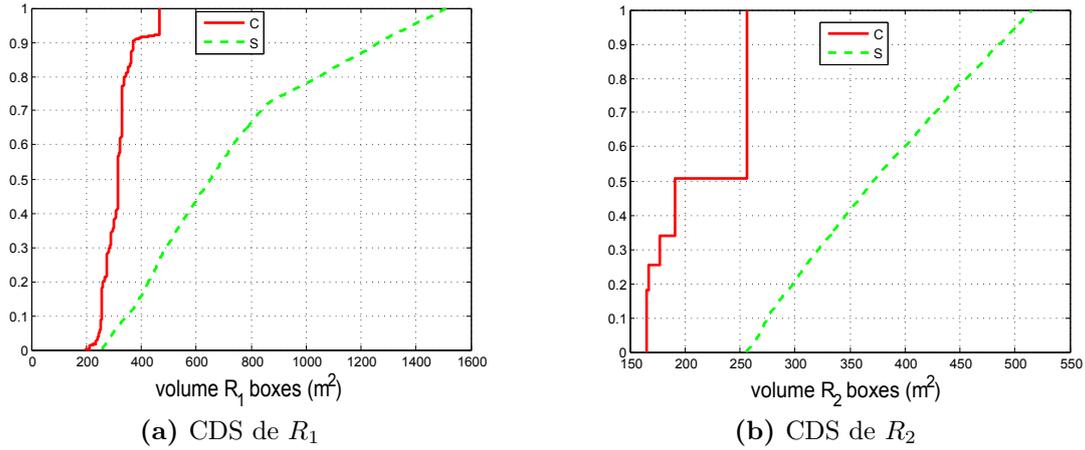


Figure 3.33 – CDF des CDS de véhicules en considérant la mesure de cap ψ .

En ce qui concerne les erreurs de RDE relatives, la méthode a permis d'améliorer l'exactitude de la localisation relative grâce à la fusion des biais estimés. Particulièrement, si on observe la moyenne et l'écart-type des erreurs, ils sont respectivement ramenés de $3.66m$ et $2.87m$ à $3.01m$ et $2.46m$.

Le biais sur chaque pseudodistance est initialisé avec l'intervalle $[-30, 30]$ sans aucune connaissance préalable. Pour chaque sous figure de 3.34, l'axe des x désigne le nombre d'échantillons durant le test, l'axe des y donne chaque centre et borne de l'intervalle de la boîte estimée de biais $[b^j]$ en mètres où $j \in 1, \dots, n_s$. Les sous figures sont tronquées afin d'observer la convergence illustrée par les asymptotes aux valeurs finales. Cette convergence met en évidence l'intérêt de l'analyse d'observabilité. Les valeurs finales obtenues des biais sont très fréquentes pour des récepteurs GNSS de faible coût.

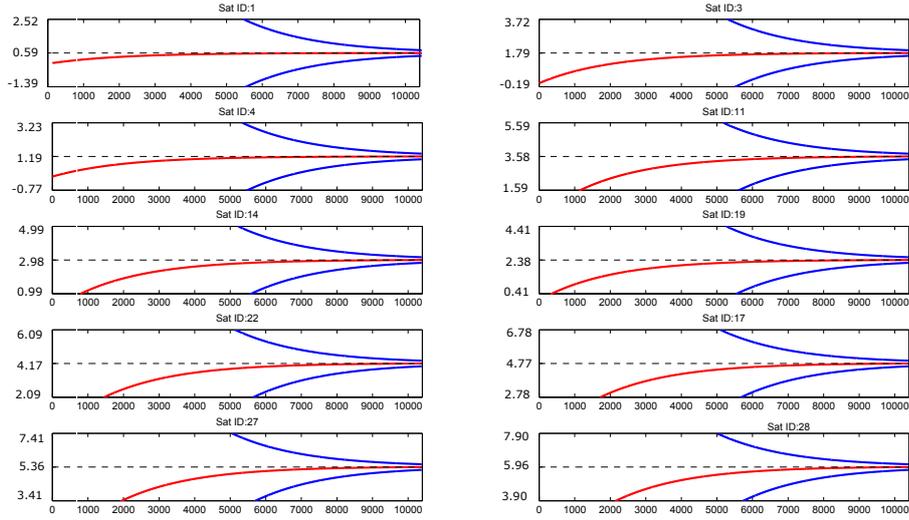


Figure 3.34 – Estimations et bornes de la correction de pseudodistances (les unités sont les échantillons et les mètres).

Une vidéo illustrant l’algorithme de coopération testé avec des données réelles en considérant v et ψ comme l’entrée du système est accessible à l’adresse : https://youtu.be/NIFj_e52mMo.

En outre, l’approche a été testée en remplaçant la mesure de cap ψ par la vitesse angulaire ω car en pratique les vitesses angulaires sont plus souvent mesurées. La mesure de vitesse angulaire est en général obtenue par un gyromètre de faible coût. Ce qui change dans ce cas est l’estimation d’état du système avec son cap et l’échange de la vitesse angulaire ω au lieu de ψ . Les variables d’état deviennent :

$$\nu = \{x_i, y_i, z_i, \psi_i, b^1, \dots, b^{n_s}, d, \dot{d}_i\}$$

avec $\dim \nu = 5 + n_s$. Le modèle d’évolution de l’équation 3.26 devient comme le suivant :

$$f = \begin{cases} x_{i,k} = x_{i,k-1} + T_e v_{i,k-1} \cos \psi_{i,k-1} \\ y_{i,k} = y_{i,k-1} + T_e v_{i,k-1} \sin \psi_{i,k-1} \\ z_{i,k} = z_{i,k-1} \\ \psi_{i,k} = \psi_{i,k-1} + T_e \omega_{i,k-1} \\ b_k^j = a b_{k-1}^j + \alpha_k^{b_j} \\ d_{i,k} = d_{i,k-1} + T_e \dot{d}_{i,k-1} + \alpha_k^d \\ \dot{d}_{i,k} = \dot{d}_{i,k-1} + \alpha_k^{\dot{d}} \end{cases} \quad (3.29)$$

L'entrée de robot R_i est :

$${}^i\mathbf{u}_{ego} = [v_i, \omega_i]^T$$

En confrontant les méthodes C et S, on aboutit aux mêmes différences de performance qui peuvent être observées dans le tableau 3.5 et les mêmes conclusions peuvent être tirées de la figure 3.35. En comparant les méthodes coopératives en utilisant les mesures de cap et la vitesse angulaire, on observe une amélioration des performances en termes d'exactitude et d'incertitude apportée par l'utilisation de la mesure de cap. En effet, l'utilisation de cap améliore la moyenne des HPE de 45.6% et 68.8% respectivement pour les véhicules R_1 et R_2 par rapport à l'utilisation de la vitesse angulaire. L'utilisation de cap apporte aussi un gain d'exactitude de distance relative de 57.84% et réduit les incertitudes des domaines de confiance de 42.47% et 30.04% respectivement pour les véhicules R_1 et R_2 par rapport à l'utilisation de la vitesse angulaire.

		HPE R1	HPE R2	Distance Relative
écart type (m^2)	C	1.42	3.97	5.08
	S	2.36	8.10	9.62
max (m)	C	7.05	18.39	19.49
	S	9.06	32.01	37.43
moyenne (m)	C	3.07	6.25	7.14
	S	4.34	9.74	10.18

		CDS R1	CDS R2
95ème percentile (m^2)	C	811.1	1987
	S	1239	10560

Table 3.5 – Statistiques des HPE et CDS en utilisant les mesures de la vitesse angulaire ω .

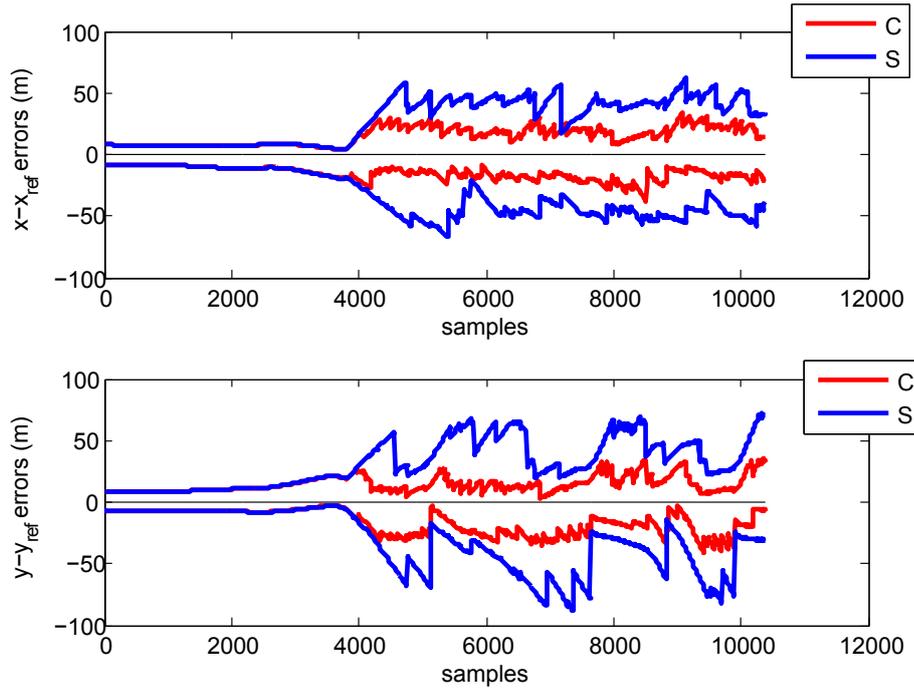


Figure 3.35 – Bornes d’erreurs de positions de R_2 en utilisant les mesures de la vitesse angulaire.

3.7 Conclusion

Dans ce chapitre, une étude de localisation coopérative de robots échangeant des erreurs communes de mesures de distances de balises ou de pseudodistances GNSS a été menée. Une nouvelle approche coopérative distribuée basée sur des méthodes ensemblistes a été proposée. Ces méthodes présentent l’avantage de garantir les positions réelles des véhicules en évitant le problème de consanguinité et en gérant rigoureusement les non-linéarités des équations. Cette nouvelle approche coopérative fournit une amélioration significative du positionnement absolu et relatif en termes d’exactitude et de domaines de confiance par rapport aux méthodes classiques non coopératives, ceci aussi bien en simulation qu’avec des données réelles. Les domaines de confiance obtenus sont très consistants. Différents problèmes ont été traités avec différents solveurs. Utiliser SIVIAP améliore sensiblement les résultats par rapport à une simple propagation de contrainte au prix d’un surcôt de calcul acceptable.

Chapitre 4

Comparaison avec une approche Bayésienne dans le cas des mesures de pseudodistance

Sommaire

4.1	Introduction	109
4.2	Méthodes Bayésiennes	110
4.3	Algorithmes de fusion distribuée	110
4.4	Algorithme de localisation coopérative	114
4.5	Résultats expérimentaux de méthode Bayésienne . . .	120
4.6	Comparaison entre la méthode Bayésienne CI et l’approche ensembliste	146
4.7	Conclusion	158

4.1 Introduction

Dans ce chapitre, nous allons effectuer une comparaison détaillée des performances de la méthode ensembliste avec des méthodes Bayésiennes en considérant certaines métriques de performance liées à l’exactitude et à l’incertitude. Pour cela, dans une première partie, une formulation Bayésienne du problème de localisation coopérative des véhicules est décrite. L’objectif est d’améliorer le positionnement en absolu et en relatif, en corrigeant les erreurs communes de mesures de pseudodistances GNSS (i.e. biais) par la fusion des estimations des biais. La fusion de ces erreurs estimées sera basée sur les opérateurs de Combinaison Convexe Simple (SCC) et d’Intersection de Covariance (CI).

Les approches classiques pour la localisation basées sur les méthodes Bayésiennes séquentielles reposent principalement sur le filtrage de Kalman pour les systèmes linéaires, le filtrage de Kalman étendu (EKF) ou le filtre de Kalman ”Unscented” (UKF) pour les systèmes non linéaires. Les méthodes basées sur l’EKF ont été utilisées pendant de nombreuses années et demeurent l’outil principal de l’estimation

non linéaire des problèmes allant du guidage de missiles au positionnement GNSS. L'efficacité de ces techniques à fournir des estimations précises et cohérentes dépend fortement de la fidélité des modèles des systèmes et des capteurs. Un filtre Bayésien coopératif basé sur l'EKF est d'abord présenté en utilisant deux types de fusion des biais (SCC et CI). Ensuite, les effets de ces fusions sur le positionnement absolu et relatif sont évalués et comparés. De cette comparaison, la méthode Bayésienne basée sur la fusion fournissant les meilleures performances est retenue pour une comparaison avec la méthode ensembliste coopérative présentée dans le chapitre 3.

4.2 Méthodes Bayésiennes

Pour qu'un système de localisation soit considéré Bayésien, il doit avoir les caractéristiques suivantes :

- **Distribution a priori.** Il doit y avoir une distribution a priori sur l'état des véhicules. Si les véhicules sont en mouvement, la distribution préalable doit inclure une description probabiliste des caractéristiques du mouvement des véhicules. Habituellement, le préalable est donné en termes d'un processus stochastique pour le mouvement des véhicules avec une connaissance initiale adaptée.
- **Fonctions de vraisemblance.** Les informations contenues dans les mesures des capteurs doivent être caractérisées par des fonctions de vraisemblance.
- **Distribution postérieure.** La sortie de base de la localisation Bayésienne est une distribution de probabilité a posteriori sur l'état des véhicules. Le postérieur à l'instant t est calculé en combinant le mouvement mis à jour avant l'instant t à la fonction de probabilité de l'observation(s) à l'instant t .

Le filtrage permet de poser le problème d'estimation d'état comme un problème d'estimation séquentielle qui minimise les erreurs quadratiques moyennes. Le filtrage de Kalman et le filtre de Kalman étendu (EKF) sont rappelés selon la présentation faite dans [48] dans l'Annexe B. Le filtre EKF est la méthode de filtrage utilisée au cours de cette thèse. Plus de détails sur le filtre UKF sont fournis dans [108][109].

4.3 Algorithmes de fusion distribuée

L'estimation moderne utilise souvent des capteurs multiples pour profiter de leur complémentarité. La fusion centralisée de mesures peut théoriquement fournir la meilleure performance d'estimation. Traiter toutes les mesures des capteurs à un emplacement fixe n'est parfois pas faisable en raison des contraintes de fiabilité ou de communication. Dans un système de fusion distribuée, chaque capteur traite ses mesures et communique les résultats avec d'autres capteurs ou d'autres processeurs. La fusion distribuée a l'avantage d'exiger des besoins de communication inférieurs et a une robustesse améliorée par rapport à la fusion centralisée. Cependant, les

algorithmes de fusion doivent traiter des problèmes qui ne sont pas présents dans la fusion centralisée. En particulier, les estimations de l'état à fusionner ne peuvent pas être traitées comme des mesures de capteurs et fusionnées avec un algorithme standard tel que le filtre de Kalman, puisque les erreurs des estimations de l'état sont généralement corrélées dans la fusion distribuée. La source de corrélation peut être le bruit de processus commun dans le cas de suivi de cibles lorsque les estimations ne sont pas fusionnées au même temps d'échantillonnage ou à chaque information préalable des estimations de la communication précédente. De nombreuses recherches ont été effectuées au cours des deux dernières décennies sur la fusion distribuée [110]. Il y a deux approches de base pour développer des algorithmes optimaux de fusion distribuée. La première est de trouver la "meilleure" combinaison linéaire des estimations pour optimiser certains critères, comme par exemple, la variance minimale [111, 112]. La seconde approche tente de reconstruire l'estimation centralisée optimale à partir des estimations [110, 113]. Dans ces travaux, des algorithmes de fusion de moyennes et covariances pour les systèmes linéaires et les distributions générales de probabilité ont été développés. Ces algorithmes de fusion sont basés sur la décorrélation de l'information préalable commune, et ne traitent pas la corrélation de bruit de mesure (ce qui peut entraîner du bruit de processus commun accumulé entre les instants de fusion). Récemment, le filtre d'intersection de covariance [114] a été proposé pour la fusion des estimations. Ce filtre ne nécessite aucune connaissance sur la corrélation entre les estimations à fusionner. Il est censé être plus robuste que l'algorithme de combinaison linéaire. Cependant, il est sous-optimal par rapport aux algorithmes qui exploitent la corrélation des estimations.

Il existe plusieurs opérateurs de fusion différents selon les hypothèses que les deux estimations à fusionner sont inter-corrélées ou non. On peut utiliser la Combinaison Convexe Simple (SCC), la Combinaison Bar-Shalom / Campo (BCC), la De-corrélation d'Information (ID), l'Estimateur Linéaire de Minimum de Variance (LMVE) et l'Intersection de Covariance (CI)[115, 116]. Ci-après, on définit les approches de la fusion utilisées dans ce travail et leurs hypothèses.

Supposons que l'état à estimer x est un vecteur aléatoire de dimension finie de moyenne \bar{x} et de covariance P . Supposons que n estimations \hat{x}_i doivent être fusionnées avec leur inter-covariance d'erreur donnée par $P_{ij} = E[(x - \hat{x}_i)(x - \hat{x}_j)^T]$ où $i \in \{1, \dots, n\}$ et $j \in \{1, \dots, n\}$. Pour simplifier la notation, P_{ii} sera représentée par P_i . Le problème de la fusion distribuée consiste à générer une estimation optimale \hat{x} à partir de n estimations \hat{x}_i .

4.3.1 Combinaison Convexe Simple

Lorsque la covariance entre deux estimations peut être ignorée, l'algorithme de la fusion est donnée par :

Cas de n estimations à fusionner

$$\left\{ \begin{array}{l} P_{SCC}^{-1} = \sum_{i=1}^n P_i^{-1} \\ \hat{x}_{SCC} = P \cdot \left(\sum_{i=1}^n P_i^{-1} \cdot x_i \right) \end{array} \right. \quad (4.1)$$

où \hat{x} est l'état estimé et P est la matrice de covariance.

Cas particulier de 2 estimations à fusionner

$$\left\{ \begin{array}{l} P_{SCC}^{-1} = P_1^{-1} + P_2^{-1} \\ \hat{x}_{SCC} = P \cdot (P_1^{-1} \cdot x_1 + P_2^{-1} \cdot x_2) \end{array} \right. \quad (4.2)$$

Cet opérateur de fusion a été utilisé largement pour sa simplicité de mise en œuvre. Il est optimal si $P_{ij} = 0$ et $R_{ij} = 0$ où $R_{ij} = E[\beta_i \beta_j^T]$ représente la matrice d'inter-covariance des bruits de mesures d'observation β . Sinon, il est sous optimal et dans ce cas P_{SCC} ne présente pas la covariance d'erreur réelle.

4.3.2 Intersection de Covariance

L'algorithme d'intersection de covariance est donné par :

Cas de n estimations à fusionner

$$\left\{ \begin{array}{l} P_{CI}^{-1} = \sum_{i=1}^n \omega_i P_i^{-1}, \sum_{i=1}^n \omega_i = 1 \\ \hat{x}_{CI} = P_{CI} \cdot \left(\sum_{i=1}^n \omega_i P_i^{-1} \cdot x_i \right) \end{array} \right. \quad (4.3)$$

où \hat{x} est l'état estimé et P^{-1} est la matrice de covariance.

Cas particulier de 2 estimations à fusionner

$$\left\{ \begin{array}{l} P_{CI}^{-1} = \omega P_1^{-1} + (1 - \omega) P_2^{-1} \\ \hat{x}_{CI} = P_{CI} \cdot (\omega P_1^{-1} \cdot x_1 + (1 - \omega) P_2^{-1} \cdot x_2) \end{array} \right. \quad (4.4)$$

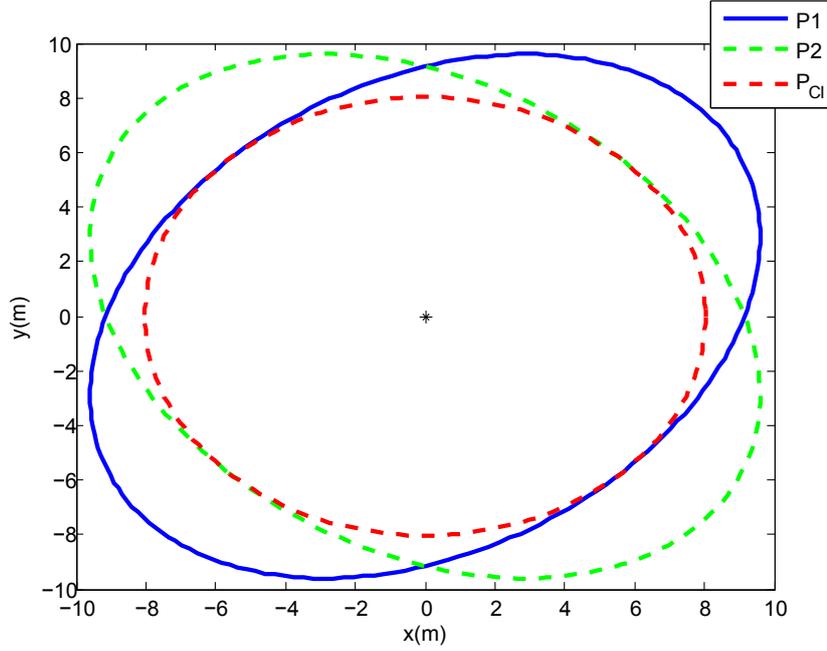


Figure 4.1 – Illustration de l'Intersection de Covariance (CI).

où ω est un nombre compris entre 0 et 1. L'objectif de l'algorithme d'intersection de covariance est de fournir une estimation conservatrice et robuste lorsque la corrélation entre \hat{x}_1 et \hat{x}_2 est inconnue comme dans le cas général d'un système de fusion distribuée. Dans ce cas, l'algorithme de fusion optimal peut être infaisable [117]. En utilisant ce filtre, la covariance d'erreur réelle est bornée par P_{CI} .

Le principe d'intersection des covariances est illustré par l'exemple de la figure 4.1, où P_{CI} représente le résultat de la fusion des matrices de covariance P_1 et P_2 .

L'algorithme de fusion de CI a la même forme que l'algorithme de SCC, avec la différence que les erreurs de covariance P_i sont pondérées. Plus précisément,

$$\begin{aligned} P_1 &\rightarrow \omega^{-1}P_1 \\ P_2 &\rightarrow (1 - \omega)^{-1}P_2 \end{aligned} \quad (4.5)$$

Ainsi, la covariance estimée par CI est conservatrice par rapport à SCC.

L'estimation fusionnée dépend du poids spécifique ω utilisé. Lorsque $\omega = 0.5$, l'estimation de CI est identique à l'estimation de SCC i.e.,

$$\hat{x}_{CI} = (P_1^{-1} + P_2^{-1})^{-1} \cdot (P_1^{-1} \cdot x_1 + P_2^{-1} \cdot x_2) = \hat{x}_{SCC} \quad (4.6)$$

Cependant, la covariance d'erreur de CI est le double de celle de SCC i.e.,

$$P_{CI} = 2(P_1^{-1} + P_2^{-1})^{-1} = 2P_{SCC} \quad (4.7)$$

En général, le poids ω est choisi afin de minimiser certains critères. Par exemple on peut choisir ω pour minimiser le déterminant ou la trace de P_{CI} .

Si $P_1 = P_2$, alors indépendamment de la valeur de ω on a :

$$P_{CI} = P_1 = P_2 \quad (4.8)$$

$$\hat{x}_{CI} = \omega \cdot x_1 + (1 - \omega) \cdot x_2 \quad (4.9)$$

Ainsi, l'incertitude n'est pas réduite par la fusion. Cela est vrai, quelque soit le nombre des estimations à fusionner même si généralement on espère une réduction de l'incertitude en utilisant plus de mesures.

4.4 Algorithme de localisation coopérative

Nous proposons une nouvelle formulation de la coopération multi-véhicules en utilisant une estimation Bayésienne séquentielle. La nouveauté de l'approche proposée réside dans l'amélioration de l'égo positionnement et de la localisation mutuelle entre les véhicules par le partage des biais de pseudodistances GNSS. Le processus développé permet la fusion des données de biais estimés de manière distribuée sans unité centrale. L'opérateur de fusion SCC est d'abord utilisé sous l'hypothèse que les erreurs de covariance entre les deux estimations peuvent être ignorées. Cependant, dans la réalité les biais de pseudodistances sont corrélés mais le niveau de corrélation est ignoré. Il est plus convenable donc que la fusion soit basée sur l'opérateur CI qui ne néglige pas les erreurs de covariance des estimations des biais grâce à une fusion conservatrice. Les deux opérateurs de fusion SCC et CI sont testés pour évaluer leur impact sur le positionnement des véhicules.

L'évaluation expérimentale utilisant deux véhicules a été réalisée pour faire face à des conditions réelles. Les deux méthodes, coopérative et standalone, sont comparées pour illustrer les améliorations apportées par la coopération.

L'algorithme coopératif a pour objectif d'améliorer le positionnement des véhicules en absolu et en relatif grâce à l'échange de données. Les véhicules disposent de capteurs DR, de récepteurs GNSS et peuvent communiquer entre eux via une communication véhicule à véhicule (V2V) (Fig. 4.2). Dans cette section, le processus

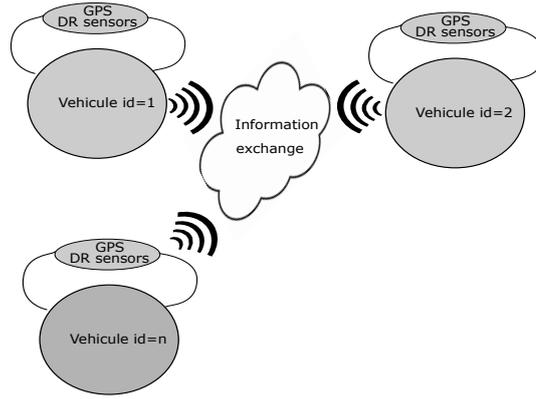


Figure 4.2 – Scénario de la coopération

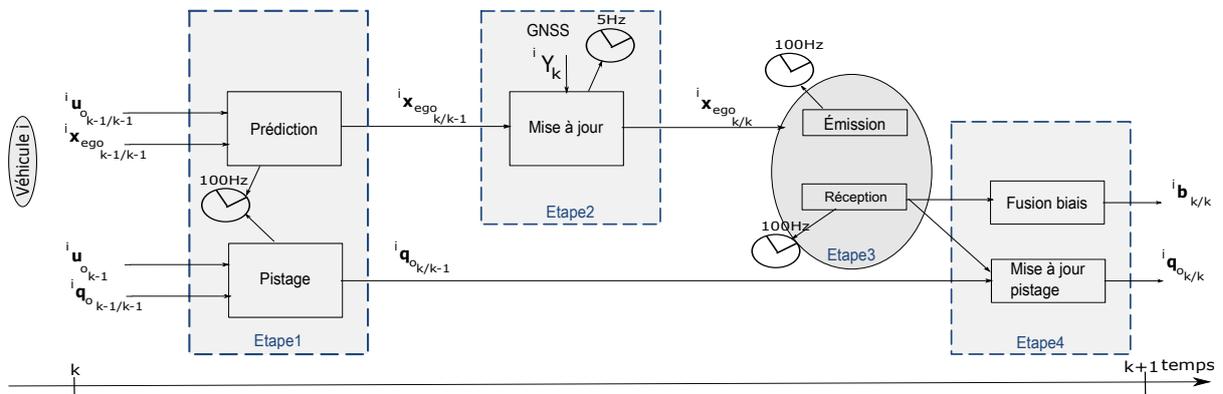


Figure 4.3 – Étapes principales du processus de l'estimation coopérative

de l'algorithme coopératif de la Fig.4.3 est décrit au sein du véhicule R_i . L'algorithme implique quatre étapes : prédiction / pistage, mise à jour, communication et fusion. D'abord une formulation du problème est présentée, ensuite les étapes de l'estimation coopérative sont détaillées.

Chaque véhicule R_i contient une partie égo identifiée par l'indice *ego* et une partie dédiée au pistage des autres agents dans le groupe identifiés par l'indice *o*. Soit R_i l'égo véhicule dans un groupe de n_r véhicules. L'indice des autres véhicules R_{o_j} est représenté par $j = \{1, \dots, M\}$ avec $j \neq i$ et $M = n_r - 1$ est le nombre des autres véhicules dans le groupe.

La première étape de l'algorithme de la figure 4.3 contient deux sous étapes. La première sous étape est dédiée à la prédiction de l'égo position du véhicule R_i en utilisant ses propres mesures de DR. Le rôle de la deuxième sous étape consiste à pister les partenaires en utilisant leurs mesures DR reçues : en cas de perte de communication les dernières mesures DR reçues seront utilisées. Dans les missions coopératives comme par exemple l'exploration d'une zone inconnue, la réalisation d'un rendez-vous ou encore la navigation en flotte, il est essentiel de connaître la

position relative des véhicules partenaires à chaque instant. Il est donc important de pister les autres véhicules entre les instants de communication. La deuxième étape de l'algorithme consiste à mettre à jour l'égo position prédite par les mesures GNSS. La troisième étape consiste à établir une communication V2V à la même fréquence de disponibilité des mesures DR. Mais généralement, la communication se fait à cadence plus faible $5Hz$ ou $1Hz$. La dernière étape contient la fusion des estimations des biais et la mise à jour de la position pistée.

Les variables estimées dans le véhicule R_i sont décrites pour les parties égo et pistage comme suit :

Variables de l'égo estimation

Soit ${}^i\mathbf{x}_{ego}$ l'égo état de R_i estimé dans le repère ENU tel que :

$${}^i\mathbf{x}_{ego} = \begin{bmatrix} {}^i\mathbf{q} & {}^i\mathbf{b} & {}^i\boldsymbol{\xi} \end{bmatrix}^T, \quad (4.10)$$

où

$${}^i\mathbf{q} = \begin{bmatrix} x_i & y_i & z_i & \psi_i \end{bmatrix}^T$$

représente la pose de R_i ,

$${}^i\mathbf{b} = [b^1, \dots, b^{n_s}]^T$$

sont les biais de n_s satellites en vue et

$${}^i\boldsymbol{\xi} = \begin{bmatrix} d_i & \dot{d}_i \end{bmatrix}^T$$

représente le vecteur contenant les variables internes de R_i qui sont respectivement le décalage d'horloge du récepteur d_i et sa dérive \dot{d}_i .

Soit ${}^iP_{ego}$ la matrice d'erreur de covariance de R_i

$${}^iP_{ego} = \begin{bmatrix} P_q & P_{q,b} & P_{q,\xi} \\ P_{q,b} & P_b & P_{b,\xi} \\ P_{q,\xi} & P_{b,\xi} & P_\xi \end{bmatrix}. \quad (4.11)$$

Le vecteur de mesure DR de R_i est noté par

$${}^i\mathbf{u}_{ego} = \begin{bmatrix} v_i & \omega_i \end{bmatrix}^T \quad (4.12)$$

où v_i et w_i sont respectivement les mesures de vitesses linéaire et angulaire.

Variables de pistages des partenaires

La pose des véhicules pistés par R_i est définie par

$${}^i\mathbf{q}_o = [{}^i\mathbf{q}_{o_1}^T, \dots, {}^i\mathbf{q}_{o_M}^T]^T \quad (4.13)$$

Le vecteur de mesures DR de tous les véhicules à pister R_o est représenté par

$${}^i\mathbf{u}_o = [{}^i\mathbf{u}_{o_1}^T, \dots, {}^i\mathbf{u}_{o_M}^T]^T \quad (4.14)$$

où chaque vecteur ${}^i\mathbf{u}_{o_j} = [v_{o_j} \ \omega_{o_j}]^T$ contient les mesures de vitesses linéaire et angulaire du véhicule R_{o_j} correspondant.

Les dernières variables nécessaires pour accomplir le pistage de partenaires sont les listes de sous-matrices dépendant respectivement de \mathbf{q} et \mathbf{b}^j de la matrice d'erreur de covariance ${}^iP_{o_j}$:

$$\begin{aligned} {}^iP_{\mathbf{q},o} &= \{{}^iP_{\mathbf{q},o_1}, \dots, {}^iP_{\mathbf{q},o_M}\} \\ {}^iP_{\mathbf{b},o} &= \{{}^iP_{\mathbf{b},o_1}, \dots, {}^iP_{\mathbf{b},o_M}\} \end{aligned} \quad (4.15)$$

Dans la suite, les étapes de l'algorithme coopératif sont détaillées.

Étape 1

Le véhicule R_i prédit son égo état en utilisant le modèle d'évolution donné par l'équation (4.16) et ses mesures DR. L'acquisition est effectuée à une fréquence élevée (typiquement de 100 Hz).

$$f = \begin{cases} x_{i,k} = x_{i,k-1} + T_e v_{i,k-1} \cos \psi_{i,k-1} + \alpha_k^x \\ y_{i,k} = y_{i,k-1} + T_e v_{i,k-1} \sin \psi_{i,k-1} + \alpha_k^y \\ z_{i,k} = z_{i,k-1} + \alpha_k^z \\ \psi_{i,k} = \psi_{i,k-1} + T_e \omega_{i,k-1} + \alpha_k^\psi \\ b_k^j = a b_{k-1}^j + \alpha_k^{b_j} \\ d_{i,k} = d_{i,k-1} + T_e \dot{d}_{i,k-1} + \alpha_k^d \\ \dot{d}_{i,k} = \dot{d}_{i,k-1} + \alpha_k^{\dot{d}} \end{cases} \quad (4.16)$$

où T_e présente la période d'échantillonnage, ψ_i est l'angle d'orientation du véhicule, $a \in [0, 1]$ est lié au temps de corrélation, α_k^x , α_k^y , α_k^z , α_k^ψ , α_k^d , $\alpha_k^{\dot{d}}$ et $\alpha_k^{b_j}$ représentent respectivement les bruits des variables d'état à chaque instant k .

En outre, R_i piste les autres véhicules en utilisant les dernières mesures DR reçues. L'évolution de l'état du groupe peut être modélisée par la fonction f_{track} selon l'Eq

Algorithm 4.1 Une itération de l'étape de prédiction pour R_i

Prediction(In out: ${}^i\hat{\mathbf{x}}_{ego}, {}^iP_{ego}$)

- 1: ${}^i\mathbf{u}_{ego} = [v_i, \omega_i]^T = \mathbf{Get}$ (individual DR measurements)
 - 2: ${}^i\hat{\mathbf{x}}_{ego} = f({}^i\hat{\mathbf{x}}_{ego}, {}^i\mathbf{u}_{ego})$
 - 3: ${}^iP_{ego} = A_k \cdot {}^iP_{ego} \cdot A_k^T + B_k \cdot N \cdot B_k^T + Q$
-

Algorithm 4.2 Une itération de l'étape de pistage pour R_i

Track(In: ${}^i\hat{\mathbf{q}}_o, {}^i\mathbf{u}_o$; out: ${}^i\hat{\mathbf{q}}_o$)

- 1: **for** $j = 1, \dots, M$ **do**
 - 2: ${}^i\hat{\mathbf{q}}_{o_j} = f_{track}({}^i\hat{\mathbf{q}}_{o_j}, {}^i\mathbf{u}_{o_j})$ // the speeds of the partners are supposed to be constant.
 - 3: **end for**
-

(4.17)

$${}^i\hat{\mathbf{q}}_{o_j} = f_{track}({}^i\hat{\mathbf{q}}_{o_j}, {}^i\mathbf{u}_{o_j}) = \begin{cases} x_{o_j,k} = x_{o_j,k-1} + T_e v_{o_j,k-1} \cos \psi_{o_j,k-1} + \alpha_k^x \\ y_{o_j,k} = y_{o_j,k-1} + T_e v_{o_j,k-1} \sin \psi_{o_j,k-1} + \alpha_k^y \\ z_{o_j,k} = z_{o_j,k-1} + \alpha_k^z \\ \psi_{o_j,k} = \psi_{o_j,k-1} + T_e \omega_{o_j,k-1} + \alpha_k^\psi \end{cases} \quad (4.17)$$

On note les jacobiennes comme suit :

$$A_k = \frac{\partial f({}^i\hat{\mathbf{x}}_{ego,k-1,k-1}, {}^i\mathbf{u}_{ego,k})}{\partial f_{track}({}^i\hat{\mathbf{q}}_{o_j}, {}^i\mathbf{u}_{o_j})^i\mathbf{x}_{ego}}$$

$$B_k = \frac{\partial f({}^i\hat{\mathbf{x}}_{ego,k-1,k-1}, {}^i\mathbf{u}_{ego,k})}{\partial {}^i\mathbf{u}_{ego}}$$

La première sous-étape du solveur EKF de l'étape 1 est décrite par l'algorithme 4.1 et le pistage par l'algorithme 4.2.

Étape 2

Cette étape consiste à mettre à jour l'état ${}^i\mathbf{x}_{ego}$ avec les mesures GNSS. L'acquisition des mesures GNSS est effectuée avec une fréquence inférieure à la fréquence des mesures de DR (par exemple 5 Hz en utilisant des récepteurs GPS classiques). Un processus de validation sur les mesures de chaque satellite est adopté afin de rejeter les mesures aberrantes à chaque instant k où les mesures GNSS sont disponibles. Soit j l'indice des mesures extéroceptives avec $j = \{1, \dots, n_m\}$. Pour chaque mesure de pseudodistance, on vérifie si le *SNR* (*Signal to Noise Ratio*) de chaque satellite est suffisamment élevé (e.g. 35 dB/Hz) et que la distance de Mahalanobis D_j qui dépend de l'innovation est inférieure à un seuil. Le solveur EKF à l'étape 2 est défini

Algorithm 4.3 Une itération de l'étape 2 pour R_i

Update(In out : ${}^i\hat{\mathbf{x}}_{ego}, {}^iP_{ego}$)

- 1: **if** New GNSS data is available **then**
- 2: n_s = number of visibles satellites
- 3: $[\rho^1, \dots, \rho^{n_s}] = \mathbf{Get}$ (GNSS measurements)
- 4: Good Pr = \emptyset
- 5: **for** $j = 1, \dots, n_s$ **do**
- 6: **if** (ρ^j is valid) & ($D_j < \text{Threshold}$) **then**
- 7: **Add**(ρ^j) to the Good_Pr list
- 8: **end if**
- 9: **end for**
- 10: $[{}^i\hat{\mathbf{x}}_{ego}, {}^iP_{ego}] = \mathbf{Update}$ (${}^i\hat{\mathbf{x}}_{ego}, {}^iP_{ego}, \text{Good_Pr}$)
- 11: **end if**

par l'algorithme 4.3. Le temps de la boucle principale du processus suit la fréquence des capteurs DR (100 Hz). Les latences faibles des mesures GNSS sont négligées.

Étape 3

Lorsqu'une communication est disponible, la coopération peut être effectuée en partageant les informations suivantes à chaque période d'échantillonnage k tel que est détaillé par l'algorithme 4.4 :

R_i transmet les données $dataS$ à ses partenaires

Le vecteur mesure DR de R_i à l'instant k : ${}^i\mathbf{u}_{ego}$

Les sous-vecteur et la sous-matrice correspondant à \mathbf{q} respectivement dans ${}^i\mathbf{x}_{ego}$ et ${}^iP_{ego}$: ${}^i\mathbf{q}_{ego}$ et ${}^iP_{q,ego}$

Les sous-vecteur et la sous-matrice correspondant à b^j dans ${}^i\mathbf{x}_{ego}$ et ${}^iP_{ego}$: ${}^i\mathbf{b}_{ego}$ et ${}^iP_{b,ego}$

R_i reçoit de chaque véhicule R_{o_j} les données $dataR$ suivantes

Le vecteur mesure DR de R_{o_j} à l'instant k : ${}^{o_j}\mathbf{u}_{ego}$

Les sous-vecteur et la sous-matrice correspondant à \mathbf{q} respectivement dans ${}^{o_j}\mathbf{x}_{ego}$ et ${}^{o_j}P_{ego}$: ${}^{o_j}\mathbf{q}_{ego}$ et ${}^{o_j}P_{q,ego}$

Les sous-vecteur et la sous-matrice correspondant à b^j dans ${}^{o_j}\mathbf{x}_{ego}$ et ${}^{o_j}P_{ego}$: ${}^{o_j}\mathbf{b}_{ego}$ et ${}^{o_j}P_{b,ego}$

L'identifiant des véhicules partenaires R_{o_j} : ${}^{o_j}id$

Algorithm 4.4 Une itération de l'étape 3 pour R_i

Communication(In : $dataS$; out : $dataR$)

```

1: for  $j = 1, \dots, M$  do
2:   Broadcast( ${}^i\mathbf{u}_{ego}, {}^i\mathbf{q}_{ego}, {}^iP_{q,ego}, {}^i\mathbf{b}_{ego}, {}^iP_{b,ego}, {}^i id$ )
3: end for
4: for  $j = 1, \dots, M$  do
5:    ${}^i\mathbf{u}_{j id} = \text{Receive}({}^{o j id}\mathbf{u}_{ego})$ 
6:    ${}^i\mathbf{q}_{j id} = \text{Receive}({}^{o j id}\mathbf{q}_{ego})$ 
7:    ${}^iP_{q, o j id} = \text{Receive}({}^{o j id}P_{q,ego})$ 
8:    ${}^i\mathbf{b}_{j id} = \text{Receive}({}^{o j id}\mathbf{b}_{ego})$ 
9:    ${}^iP_{b, o j id} = \text{Receive}({}^{o j id}P_{b,ego})$ 
10: end for

```

Algorithm 4.5 Une itération de l'étape de mise à jour du pistage de l'étape 4 pour R_i

Track_update(In: ${}^o\mathbf{q}_{ego}, {}^oP_{q,ego}$; out: ${}^i\mathbf{q}_o, {}^iP_{q,o}$)

```

1:  ${}^i\mathbf{q}_o = {}^o\mathbf{q}_{ego}$ 
2:  ${}^iP_{q,o} = {}^oP_{q,ego}$ 

```

Étape 4

La mise à jour du pistage consiste à remplacer respectivement ${}^i\mathbf{q}_o$ et ${}^iP_{q,o}$ par les estimations reçues ${}^o\mathbf{q}_{ego}$ et ${}^oP_{q,ego}$ à chaque période d'échantillonnage k (Algorithme 4.5). On ne fusionne pas les pistes prédites et les valeurs estimées reçues car la prédiction par suivi des autres est moins bonne que l'égo prédiction et peut être mauvaise s'il n'y a pas eu de communication depuis longtemps. Le rôle de pistage des autres véhicules dans cet algorithme est de connaître leur position au cas de perte de communication.

La mise à jour des biais correspond à la fusion des biais reçus ${}^o\mathbf{b}_{ego}$ estimés dans les véhicules partenaires avec les biais ${}^i\mathbf{b}_{ego}$ estimés dans R_i . La fusion des biais est effectuée par l'opérateur SCC en premier temps. Ensuite, l'opérateur CI sera utilisé. Par exemple, considérons le processus de fusion dans le véhicule R_i . En désignant par ${}^i\mathbf{b}_c$ le résultat de la fusion des biais, par ${}^iP_{b_c}$ le résultat de la fusion des erreurs de covariance des biais et n_M le nombre de véhicules partenaires R_{o_j} , on obtient l'algorithme de fusion 4.6. Nous pouvons maintenant écrire l'algorithme 4.7 coopératif principal.

4.5 Résultats expérimentaux de méthode Bayésienne

Nous présentons les résultats de deux scénarii, i.e. Standalone (S) et Coopératif (C), afin de quantifier le gain de performance grâce à la coopération. Lors de notre

Algorithm 4.6 Une itération de l'étape de la fusion SCC de l'étape 4 pour R_i

FusionSCC(In: ${}^i\mathbf{b}_{ego}, {}^iP_{\mathbf{b},ego}, {}^{o_j}P_{\mathbf{b},ego}, {}^{o_j}\mathbf{b}_{ego}$; out: ${}^iP_{\mathbf{b}_c}, {}^i\mathbf{b}_c$)

$$1: {}^iP_{\mathbf{b}_c} = {}^iP_{\mathbf{b},ego}^{-1} + \sum_{k=1}^{n_M} {}^{o_k}P_{\mathbf{b},ego}^{-1}$$

$$2: {}^i\mathbf{b}_c = P_{\mathbf{b}_c}^{-1}({}^iP_{\mathbf{b},ego} \cdot {}^i\mathbf{b}_{ego} + \sum_{k=1}^{n_M} {}^{o_k}P_{\mathbf{b},ego} \cdot {}^{o_k}\mathbf{b}_{ego})$$

Algorithm 4.7 Coopération de R_i avec R_o .

1: **for** $t = 1, \dots, t_{can}$ **do**

2: **Prediction**(In out : ${}^i\hat{\mathbf{x}}_{ego}, {}^iP_{ego}$)

3: **Track**(In : ${}^i\hat{\mathbf{q}}_o, {}^i\mathbf{u}_o$; out : ${}^i\hat{\mathbf{q}}_o$)

4: **Update**(In out : ${}^i\hat{\mathbf{x}}_{ego}, {}^iP_{ego}$) // when measurements are available

5: **Communication**(In : $dataS$; out : $dataR$)

6: **Track_update**(In : ${}^o\mathbf{q}_{ego}, {}^oP_{\mathbf{q},ego}$; out : ${}^i\mathbf{q}_o, {}^iP_{\mathbf{q},o}$)

7: **FusionSCC**(In : ${}^i\mathbf{b}_{ego}, {}^iP_{\mathbf{b},ego}, {}^{o_j}P_{\mathbf{b},ego}, {}^{o_j}\mathbf{b}_{ego}$; out : ${}^iP_{\mathbf{b}_c}, {}^i\mathbf{b}_c$)

8: **end for**

étude, nous avons testé notre approche uniquement avec deux véhicules. Deux véhicules expérimentaux automobiles ont effectué différentes trajectoires dans un milieu contraint entre des bâtiments et avec des arbres. Un récepteur GPS U-blox 4T de faible coût fournissant les mesures de pseudodistances à 5Hz a été utilisé. Chaque véhicule était équipé d'un récepteur Polarx Septentrio en mode RTK pour obtenir la vérité terrain et un bus CAN pour obtenir le vecteur mesure de DR : $\mathbf{u}_{ego} = \begin{bmatrix} v & w \end{bmatrix}^T$ à une fréquence de 100Hz. Les trajectoires estimées à l'aide de l'approche coopérative sont représentées sur la Fig. 4.4. Dix satellites étaient en vue au cours du test dont 5 étaient au moins communs aux véhicules ce qui satisfait la condition de l'étude de l'observabilité du chapitre 3. La présence des bâtiments et des arbres a limité par endroits la visibilité des satellites GPS. Les performances du positionnement sont étudiées en fonction des Erreurs du Positionnement Horizontal (HPE) définies dans l'Eq. 4.18 et des domaines de confiances des estimations (CDS : *Confidence Domain Size*) représentées par des ellipses comme l'exemple de la Fig.4.5.

$$HPE = \sqrt{e_x^2 + e_y^2} \quad (4.18)$$

où $e_x = \hat{x} - x_{ref}$ et $e_y = \hat{y} - y_{ref}$ avec (\hat{x}, \hat{y}) et (x_{ref}, y_{ref}) représentant respectivement l'estimation de la position en 2D et la vérité terrain. Ainsi les HPE indiquent l'exactitude de l'approche et les CDS ont pour objectif d'évaluer l'incertitude et la consistance de la méthode. Plus de détails sur le calcul des HPE et CDS sont fournis dans [106]. La consistance est vérifiée si l'ellipse représentant le CDS contient le



Figure 4.4 – Trajectoires de véhicules estimées par la méthode coopérative

HPE, i.e. $HPE < k\sigma_{HPE}$ où

$$\sigma_{HPE} = \sqrt{\frac{1}{\mathbf{u}_e^T P_{HPE}^{-1} \mathbf{u}_e}}, \text{ avec } \mathbf{u}_e = \begin{pmatrix} e_x \\ e_y \end{pmatrix} / \sqrt{e_x^2 + e_y^2} \quad (4.19)$$

\mathbf{u}_e est le vecteur unitaire supportant l'erreur HPE et P_{HPE} est la matrice d'erreur de covariance estimée en utilisant le filtre EKF. $k = 3.035$ est le risque de consistance choisi à 10^{-2} selon la loi du χ^2 .

L'erreur HPE absolue et son CDS correspondant sont présentés sur la figure 4.5. L'exemple de CDS présenté sur cette figure ne contient pas la référence. Dans ce cas, le filtre EKF n'est pas consistant.

Ci-après, on définit les erreurs de la distance relative (*Relative Distance Error* RDE) et les CDS relatifs entre R_1 et R_2 qui sont représentés sur la figure 4.6.

$$RDE = \sqrt{(\hat{d}_x - d_{x,ref})^2 + (\hat{d}_y - d_{y,ref})^2} \quad (4.20)$$

où $\hat{d}_x = \hat{x}_1 - \hat{x}_2$, $d_{x,ref} = \hat{x}_{1,ref} - \hat{x}_{2,ref}$ avec $\hat{d}_y = \hat{y}_1 - \hat{y}_2$ et $d_{y,ref} = y_{1,ref} - y_{2,ref}$.

Le CDS relatif est égal à $k\sigma_{RDE}$, où σ_{RDE} est présenté dans l'Eq.4.21.

$$\sigma_{RDE} = \sqrt{\frac{1}{\mathbf{u}_e^T P_{RDE}^{-1} \mathbf{u}_e}}, \text{ avec } \mathbf{u}_e = \begin{pmatrix} \hat{d}_x - d_{x,ref} \\ \hat{d}_y - d_{y,ref} \end{pmatrix} / \sqrt{(\hat{d}_x - d_{x,ref})^2 + (\hat{d}_y - d_{y,ref})^2} \quad (4.21)$$

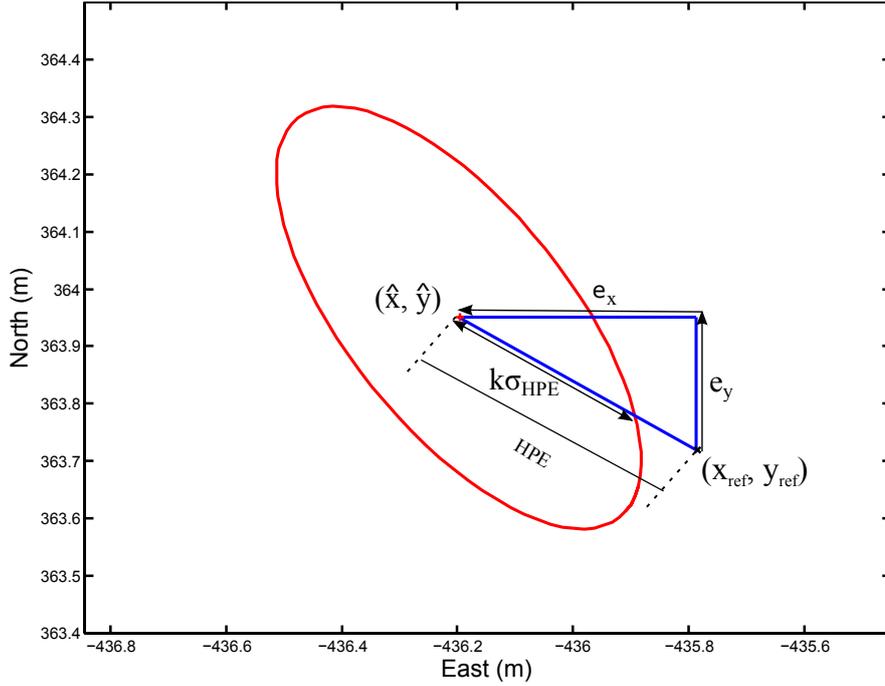


Figure 4.5 – CDS (i.e, $k \sigma_{HPE}$) et HPE d’une estimation absolue.

Variances des bruits de modèle	Variances des bruits de mesure	
R_1 and R_2	R_1	R_2
$Var(\alpha^d) = 5$	$Var(\beta) = 10$	$Var(\beta) = 10$
$Var(\alpha^d) = 10$	$Var(\gamma^v) = 10^{-3}$	$Var(\gamma^v) = 10^{-3}$
$Var(\alpha^b) = 5 \times 10^{-3}$	$Var(\gamma^\omega) = 2.5 \times 10^{-2}$	$Var(\gamma^\omega) = 5 \times 10^{-2}$

Table 4.1 – Paramètres du filtre EKF (International System Units)

ur_e est le vecteur unitaire supportant l’erreur de la distance relative RDE et P_{RDE} est la matrice d’erreur de covariance relative telle que

$$P_{RDE} = P_{HPE_1} + P_{HPE_2} \quad (4.22)$$

L’équation (4.22) est valable sous hypothèse d’erreurs d’estimation indépendantes.

4.5.1 Résultats de la fusion avec SCC

Les figures 4.7 et 4.8 représentent les erreurs de positionnement horizontales absolues (HPE) des véhicules R_1 et R_2 pour les méthodes Coopérative (C) et Standalone (S). Nous pouvons remarquer que la méthode C a réduit le domaine de confiance et les erreurs HPE, pour le véhicule R_1 en dépit du manque d’intégrité dans quelques instants par rapport à la méthode S.

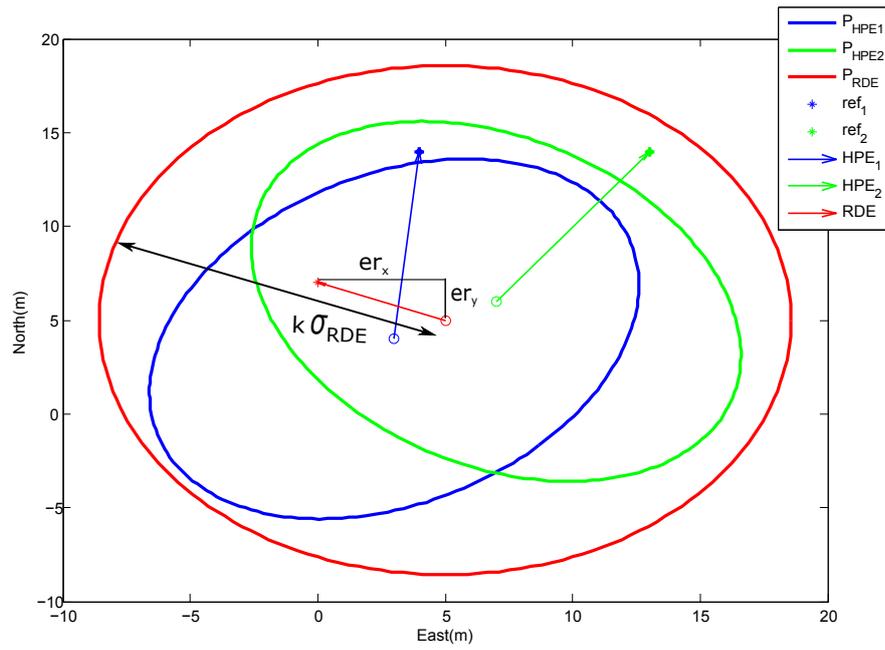


Figure 4.6 – CDS (i.e, $k \sigma_{RDE}$) et RDE d'une estimation relative.

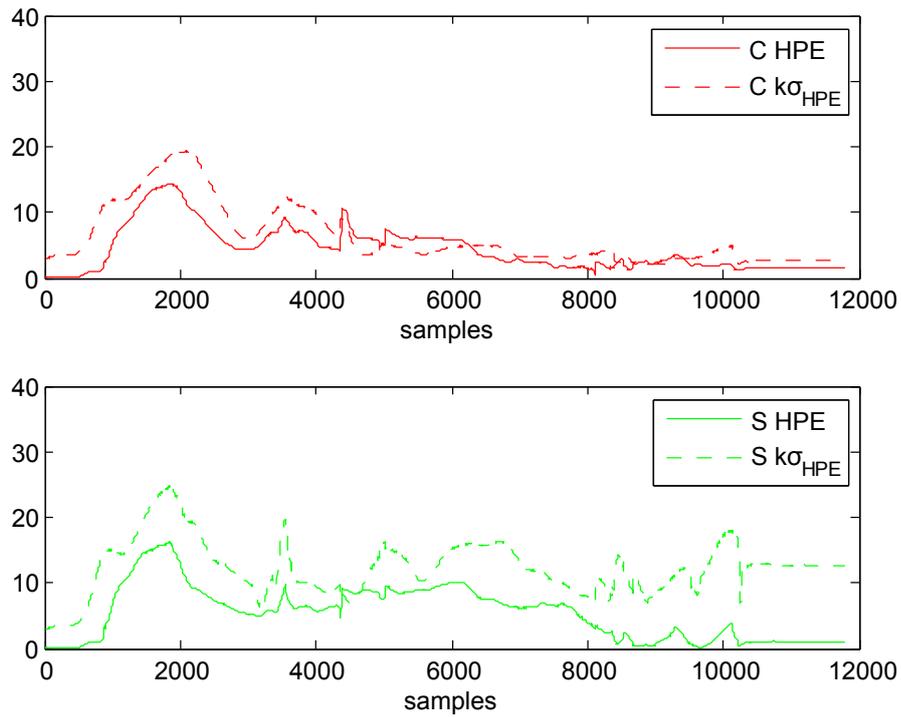


Figure 4.7 – Erreurs HPE absolues et domaines de confiance de R_1 . Fusion SCC.

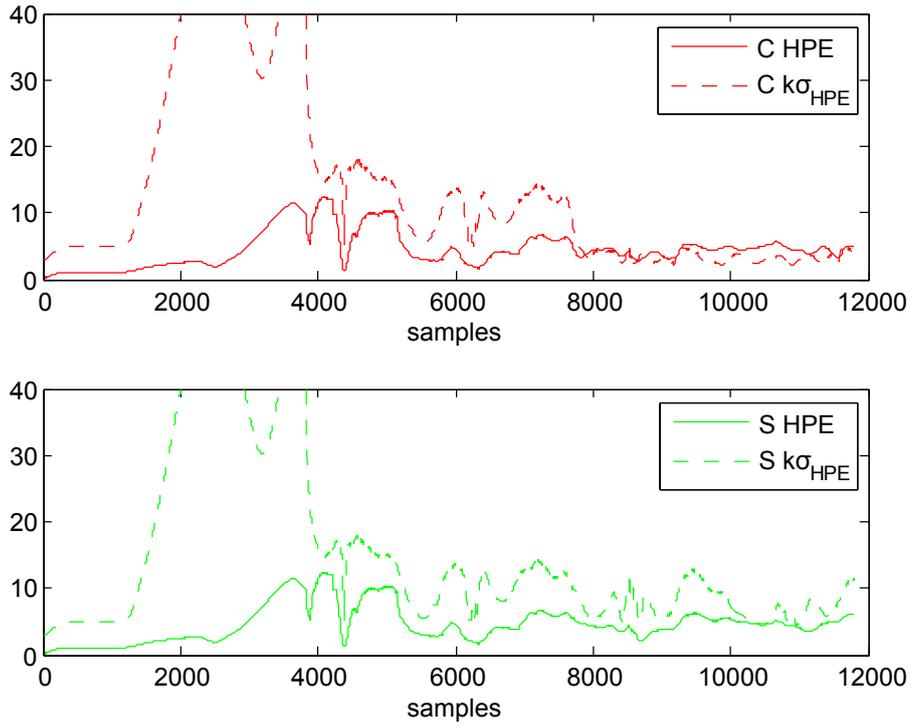


Figure 4.8 – Erreurs HPE absolues et domaines de confiance de R_2 . Fusion SCC.

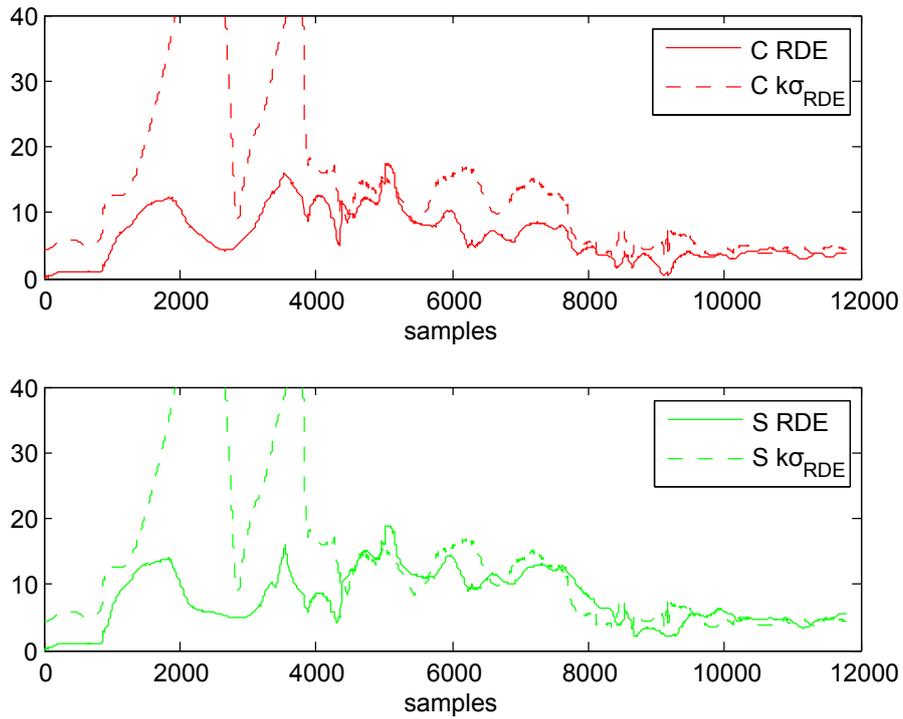


Figure 4.9 – Erreurs de distance relative RDE et leurs domaines de confiance. Fusion SCC.

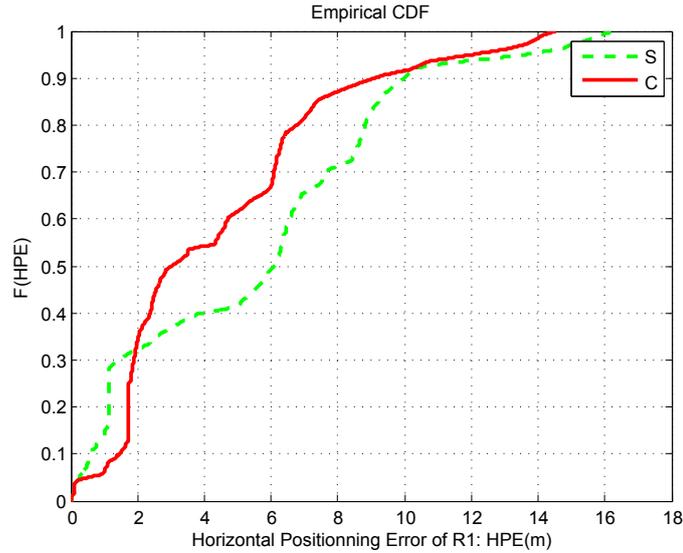


Figure 4.10 – Distribution cumulative des erreurs HPE de R_1 . Fusion SCC.

La figure 4.12 présente la distribution cumulative du domaine de confiance des erreurs HPE de R_1 en utilisant respectivement les méthodes S et C. Le 95^{ème} percentile des méthodes coopérative et standalone sont respectivement $17.16m$ et $20.02m$ considérant le véhicule R_1 . L’approche coopérative améliore de manière significative la méthode S en terme de réduction de l’incertitude.

La figure 4.9 montre les erreurs de la localisation relative pour les deux méthodes. Une réduction d’incertitude de l’ordre de plusieurs mètres est observée.

La figure 4.10 montre l’accumulation des erreurs HPE de R_1 tout au long l’essai par les méthodes C et S. On peut remarquer que la méthode coopérative par fusion SCC améliore l’exactitude de l’estimation par rapport à la méthode standalone puisque le 95^{ème} percentile des méthodes C et S sont respectivement de $17.17m$ et $20.04m$. Par contre, l’amélioration n’est pas nette (ou quasi nulle) en ce qui concerne le véhicule R_2 (cf. Fig. 4.11).

Le tableau 4.2 donne les indicateurs de performance pour les deux méthodes en termes d’erreurs HPE et de domaines de confiance. Une nette amélioration est obtenue pour le véhicule R_1 où l’écart type des HPE est réduit de $4.06m$ à $3.39m$ en utilisant notre approche par rapport à la méthode S. Concernant le véhicule R_2 , l’amélioration des HPE n’est pas autant notable que R_1 , tandis que la taille du domaine de confiance est améliorée par rapport à la méthode S au début de l’expérimentation comme l’indique la figure 4.13.

Le tableau 4.3 représente quelques performances des erreurs relatives de localisation entre les véhicules pour les deux méthodes. Notre approche permet d’améliorer l’exactitude de la localisation par rapport à la méthode classique grâce à la fusion des estimations de biais. En particulier, on observe la moyenne d’erreur et l’écart-type, qui sont réduits respectivement de $7.85m$ et $4.2m$ à $6.46m$ et $3.86m$ en utilisant

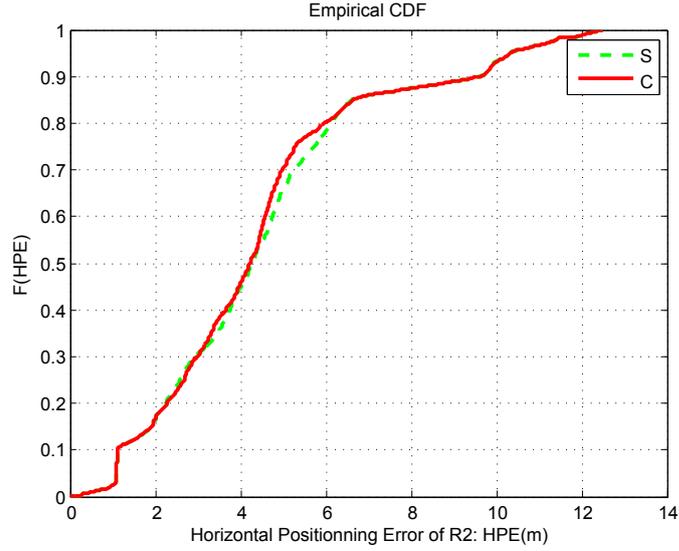


Figure 4.11 – Distribution cumulative des erreurs HPE de R_2 . Fusion SCC.

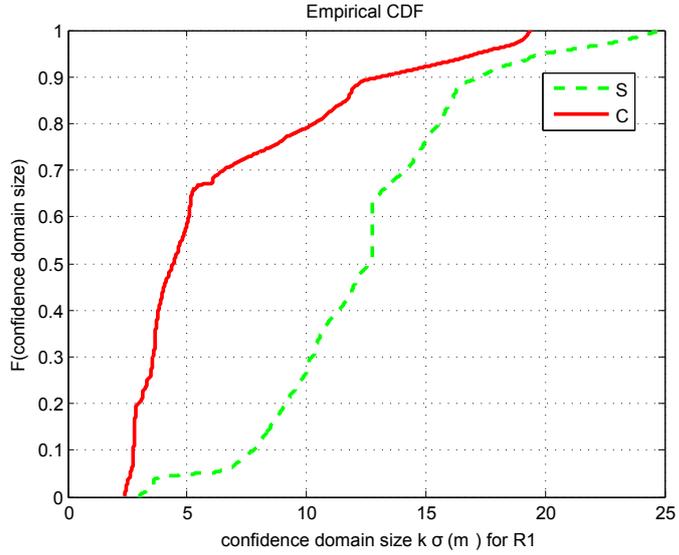


Figure 4.12 – Distribution cumulative de CDS de R_1 . Fusion SCC.

		R1	R2
écart type (m)	C	3.39	2.70
	S	4.06	2.71
Max (m)	C	14.55	12.50
	S	16.32	12.51
Moyenne (m)	C	4.36	4.51
	S	5.47	4.58

Table 4.2 – Statistiques des erreurs HPE de chaque véhicule. Fusion SCC.

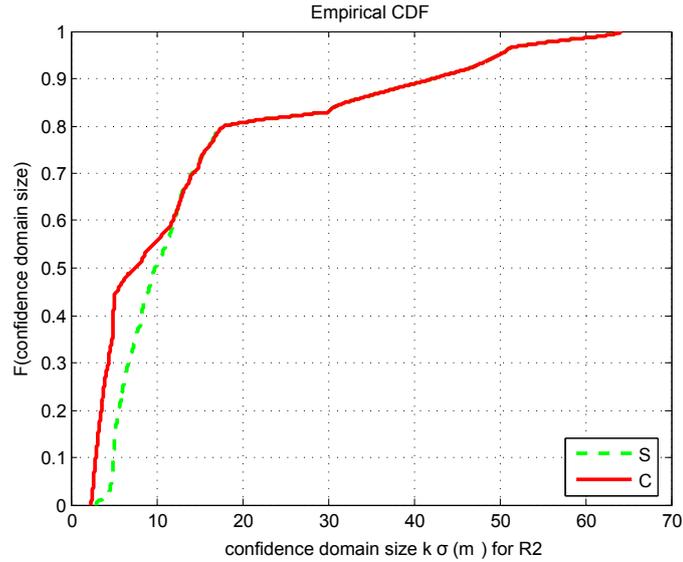


Figure 4.13 – Distribution cumulative de CDS de R_2 . Fusion SCC.

écart type (m)	C	3.86
	S	4.20
Max (m)	C	17.46
	S	19
Moyenne (m)	C	6.46
	S	7.85

Table 4.3 – Statistiques des erreurs relatives RDE entre les véhicules. Fusion SCC.

l'approche coopérative par fusion SCC.

Le biais de chaque pseudodistance est initialisé à zéro. Pour chaque courbe de la figure 4.14, l'axe des abscisses présente le temps en seconde, l'axe des ordonnées représente chaque biais estimé b^j en mètres ($j = 1, \dots, n_s$), n_s est le nombre de satellites en vue. Les mesures de satellites 17 et 28 ne sont disponibles qu'à partir des instants 76s et 116s. On observe que les biais estimés restent dans une plage assez petite de l'ordre du mètre.

La figure 4.16 montre les trajectoires de dix satellites en vue pendant l'essai. Les satellites en vue sont sélectionnés selon la qualité de SNR et chacun est validé par l'innovation lors du positionnement. La figure 4.15 représente le nombre de satellites utilisés à chaque instant de l'essai pour les deux véhicules et les satellites en commun. On peut remarquer que le véhicule R_1 a commencé la mise à jour par les mesures GNSS à partir de l'instant 17.24s, tandis que les mesures GNSS du véhicule R_2 n'étaient disponibles qu'à partir l'instant 76.52s.

Dans la Fig.4.15b, 5 satellites sont au moins en commun à chaque instant sachant que le nombre maximal de satellites en commun est égale à 8. Suite à la sélection des bonnes mesures, les véhicules R_1 et R_2 utilisent respectivement au moins 5 et

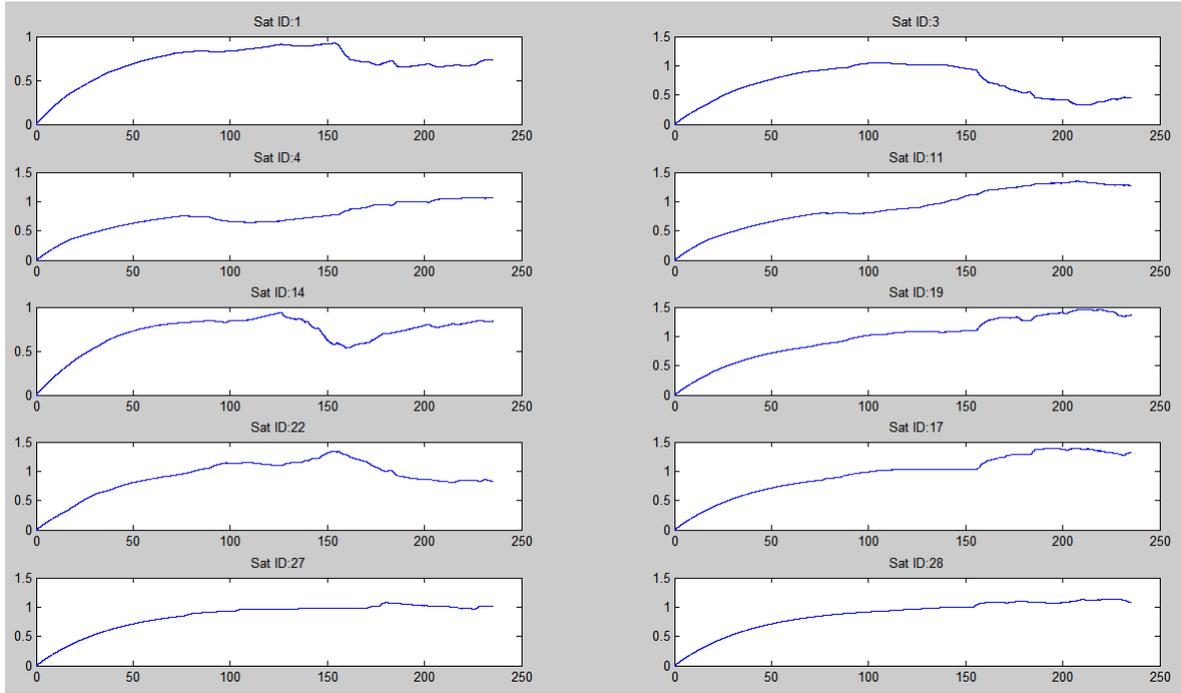
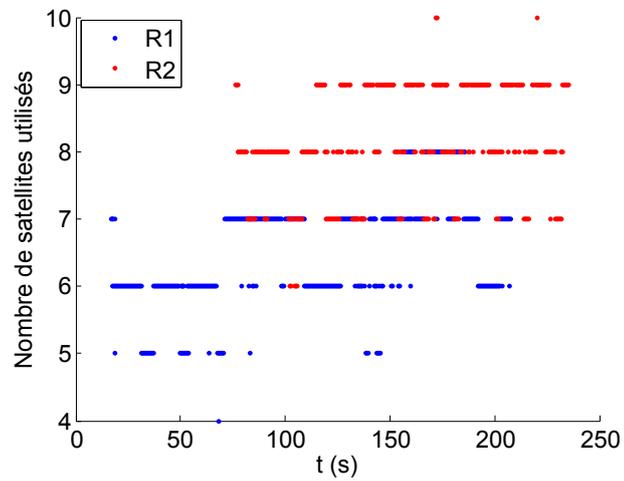


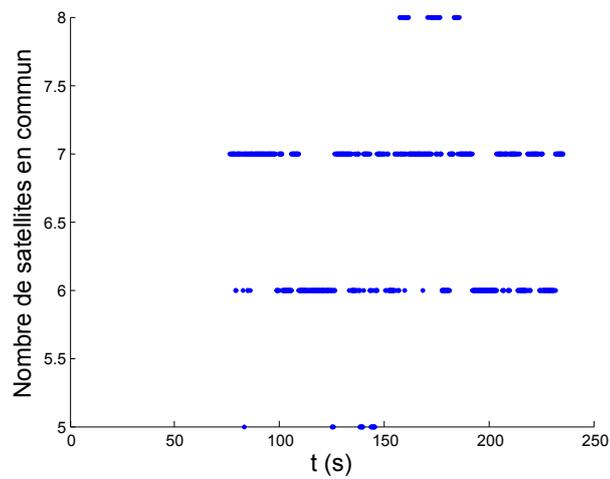
Figure 4.14 – Estimation des biais avec la fusion SCC (les unités sont secondes en abscisses et mètres en ordonnées).

6 satellites pour se localiser. Le nombre maximal de satellites utilisés pour l'étape de mise à jour par R_1 et R_2 sont respectivement égaux à 7 et 10, ce qui explique l'amélioration des performances de positionnement de R_1 par R_2 . Il faut souligner que R_2 utilise plus de mesures pour mettre à jour l'estimation des biais. Le nombre de satellites rejetés par R_1 est supérieur à celui de R_2 . Cela est dû au fait que le milieu d'évolution du véhicule R_1 est plus contraint, puisqu'il est entouré par au moins 3 bâtiments. On peut constater que les mesures des satellites 22, 17 et 27 ont plus de probabilité d'être rejetées puisque d'après la figure 4.16 leurs angles d'élévation sont au dessous de 20° . On peut remarquer aussi dans la figure 4.17 la variation du SNR du satellite 22 qui représente les SNR qui sont en dessous du seuil $35dB/Hz$ à plusieurs instants. Cependant, les satellites 4 et 11 sont les plus hauts satellites, ils ont des angles d'élévation supérieurs à 50° et sont suivis avec un meilleur SNR. La qualité du SNR du satellite 4 est toujours élevée (cf. Fig. 4.17a). Par contre, le signal du satellite 11 présente parfois des multiple-trajets quand le véhicule entre dans une zone contenant des obstacles (cf. Fig. 4.17b). Le satellite 11 étant vertical, il doit avoir des SNR élevés ce qui n'est pas toujours le cas et ce qui explique la présence des multitrajets dans quelques instants.

La figure 4.18 représente l'estimation de décalage d'horloge du récepteur pour chaque véhicule. On a estimé les valeurs initiales de décalage d'horloge du récepteur et sa dérive pour chaque véhicule en utilisant la méthode classique des moindres carrés en mode standalone. Ces estimations contiennent des sauts qui sont dus à l'initialisation



(a) Nombre de satellites utilisés par les véhicules durant l'essai.



(b) Nombre de satellites en commun durant l'essai.

Figure 4.15 – Nombre de satellites utilisés par les véhicules durant l'essai.

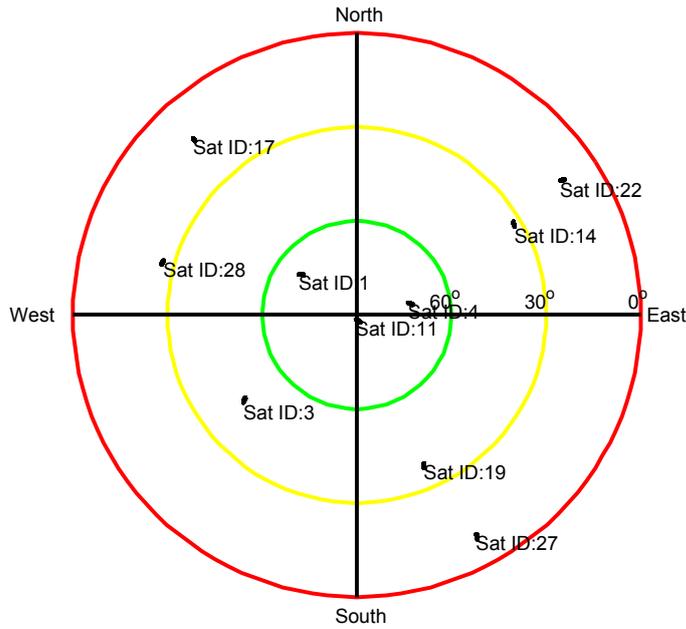


Figure 4.16 – Nombre de satellites en vue durant l'essai.

des valeurs de décalage d'horloge et ne présentent pas des sauts des horloges du récepteur Ublox. Les sauts des estimations de décalage d'horloge des véhicules R_1 et R_2 sont apparus respectivement aux instants $17.24s$ et $76.52s$ et correspondent aux moments de mise à jour avec les mesures GNSS.

4.5.2 Résultats de la fusion avec CI

Afin d'améliorer la consistance du filtre coopératif, la dépendance entre les estimations des biais à fusionner est désormais considérée. Le même algorithme coopératif 4.7 est utilisé en remplaçant l'étape de fusion avec l'opérateur SCC par l'opérateur CI. L'algorithme de la fusion avec CI présenté par l'algorithme 4.8 remplace l'algorithme 4.6. Le poids ω est ici le minimum du déterminant de la matrice de covariance des erreurs des biais fusionnés P_{b_c} . Il est connu que l'intersection de covariance assure la meilleure estimation possible quand les hypothèses d'indépendance ne sont pas réalistes, en tenant compte de la quantité d'information disponible. Dans le problème considéré, les erreurs des pseudodistances GNSS sont corrélées. Notons que la procédure ne fait pas d'hypothèse concernant l'indépendance, ni les distributions des estimations à fusionner.

Les figures 4.19 et 4.20 représentent respectivement les erreurs de positionnement horizontales absolues (HPE) des véhicules R_1 et R_2 pour les méthodes Coopérative (C) et Standalone (S). Nous remarquons que la méthode C a réduit les erreurs HPE par rapport à la méthode S et que les domaines de confiance bornent cette fois les erreurs HPE. Cela prouve la consistance du filtre coopératif basé sur la fusion CI.

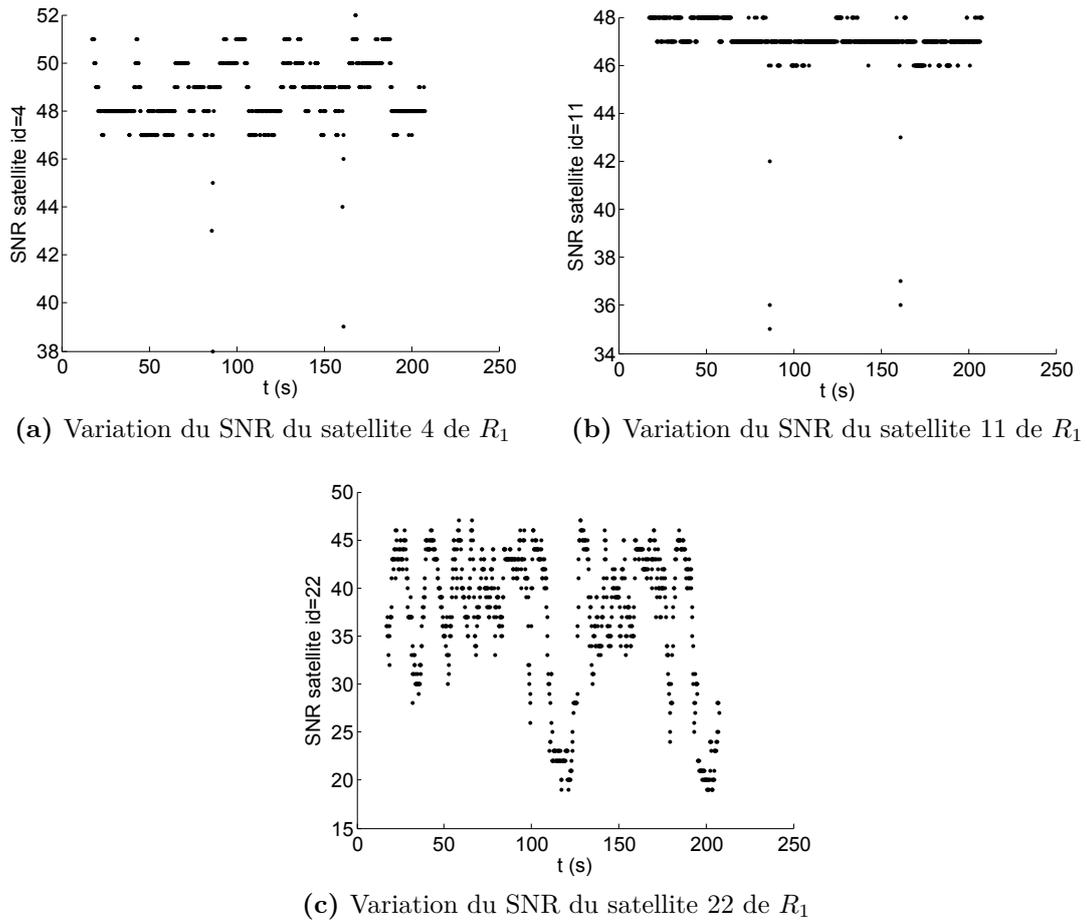


Figure 4.17 – Variation des SNR des satellites 4,11,22 pendant l’essai pour le véhicule R_1

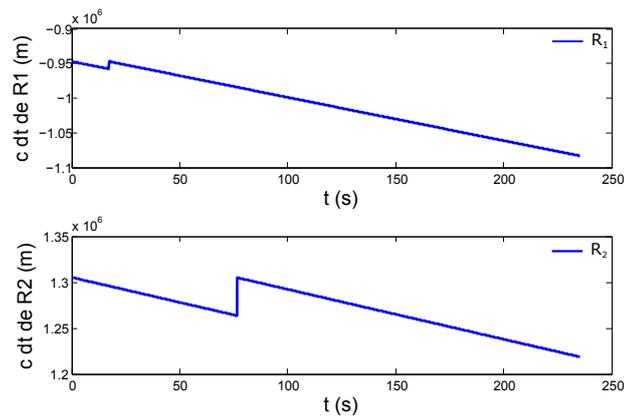


Figure 4.18 – Décalage d’horloge estimé (\times la vitesse de la lumière).

Algorithm 4.8 Une itération de la fusion CI à l'étape4 pour le véhicule R_i .

FusionCI(In: ${}^i\mathbf{b}_{ego}, {}^iP_{\mathbf{b},ego}, {}^{o_j}P_{\mathbf{b},ego}, {}^{o_j}\mathbf{b}_{ego}$; out: ${}^iP_{\mathbf{b}_c}, {}^i\mathbf{b}_c$)

$$1: P_{\mathbf{b}_c}^{-1} = \omega_i {}^iP_{\mathbf{b},ego}^{-1} + \sum_{k=1}^{n_M} \omega_k {}^{o_k}P_{\mathbf{b},ego}^{-1} \quad \omega_i + \sum_{k=1}^{n_M} \omega_k = 1$$

$$2: {}^i\mathbf{b}_c = P_{\mathbf{b}_c} (\omega_i {}^iP_{\mathbf{b},ego}^{-1} \cdot {}^i\mathbf{b}_{ego} + \sum_{k=1}^{n_M} \omega_k {}^{o_k}P_{\mathbf{b},ego}^{-1} \cdot {}^{o_k}\mathbf{b}_{ego})$$

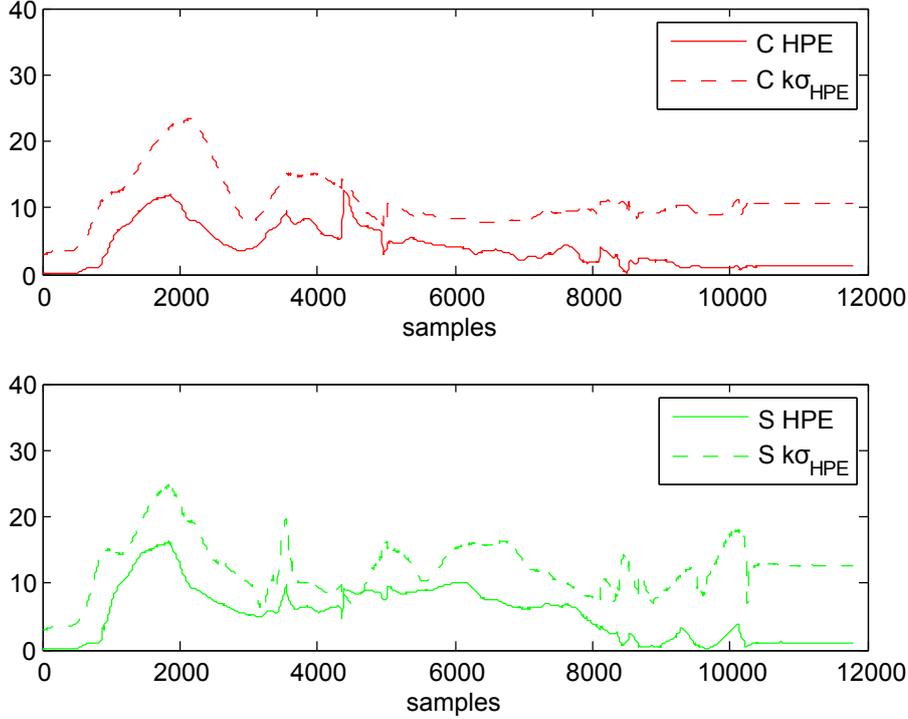


Figure 4.19 – Erreurs HPE absolues et domaines de confiance de R_1 . Fusion CI.

Cependant, l'incertitude de CDS est augmentée pour le véhicule R_2 par rapport à la méthode S. Cela peut être observé dans la figure 4.25.

Les mêmes observations sont aussi valables en considérant les erreurs relatives RDE. La méthode coopérative améliore l'exactitude du positionnement relatif et donne des domaines de confiance plus fiables par rapport à la méthode standalone.

Les figures 4.19 et 4.20 présentent la distribution cumulative des erreurs HPE de R_1 et R_2 en utilisant respectivement les méthodes S et C. 95% des HPE des méthodes coopérative et standalone sont respectivement inférieures à $7.23m$ et $9.87m$ considérant R_1 et inférieures à $7.63m$ et $10.31m$ quand on considère R_2 . L'approche coopérative avec la fusion CI améliore donc de manière significative la méthode S en terme d'exactitude et ce pour les deux robots (ce qui n'était pas le cas avec la fusion SCC).

Les figures 4.24 et 4.25 montrent la distribution cumulative des domaines de confiance

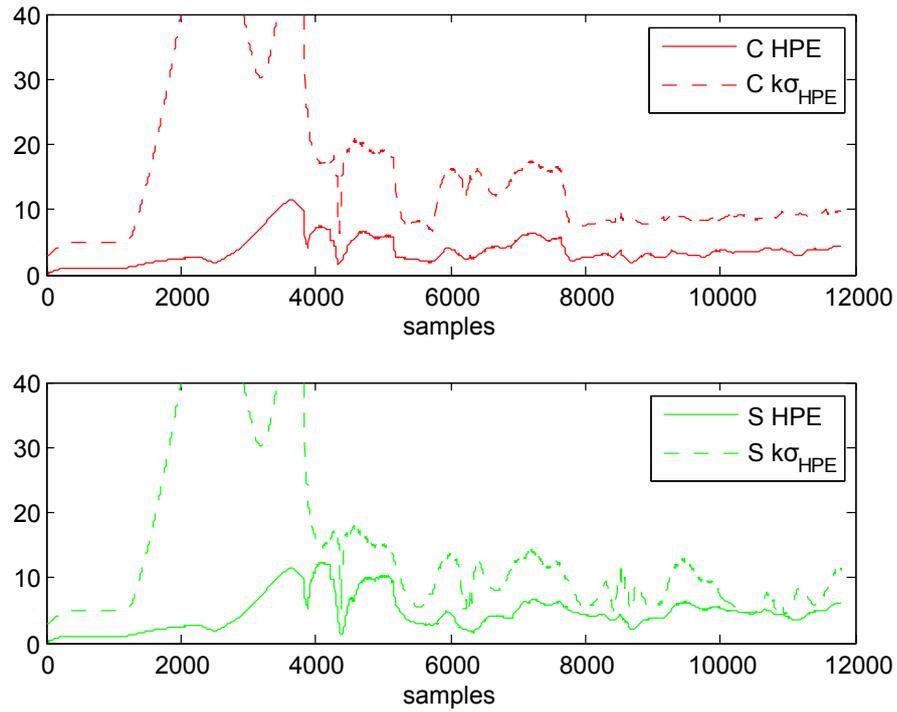


Figure 4.20 – Erreurs HPE absolues et domaines de confiance de R_2 . Fusion CI.

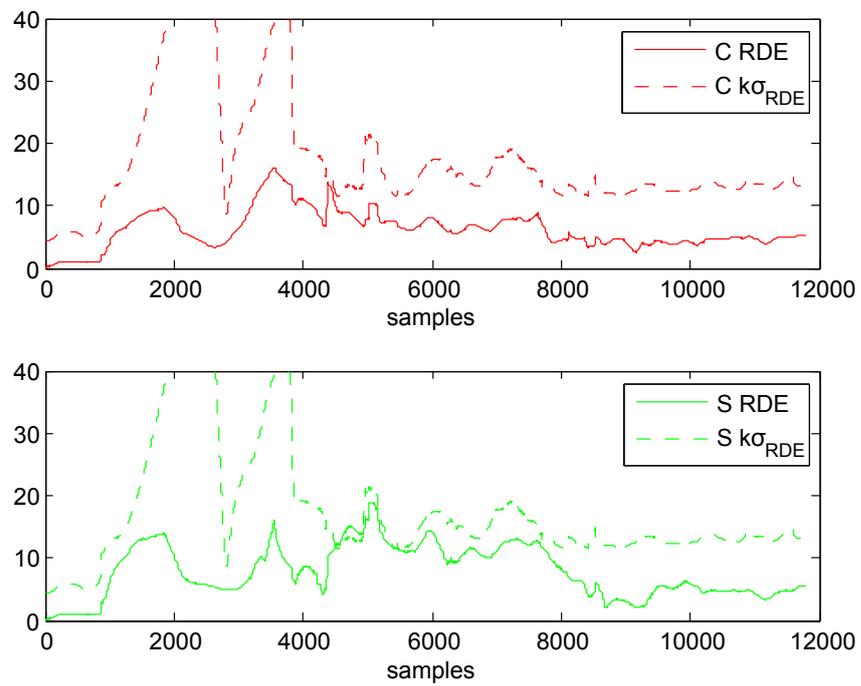


Figure 4.21 – Erreurs relatives RDE et leurs domaines de confiance. Fusion CI.

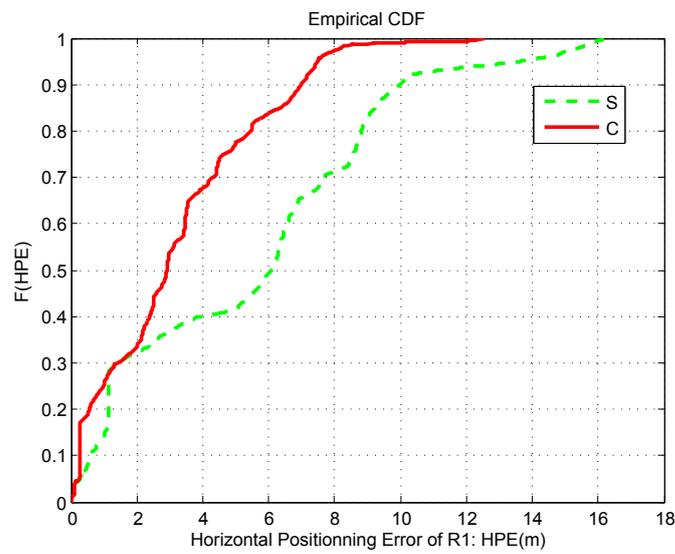


Figure 4.22 – Distribution cumulative des erreurs HPE de R_1 . Fusion CI.

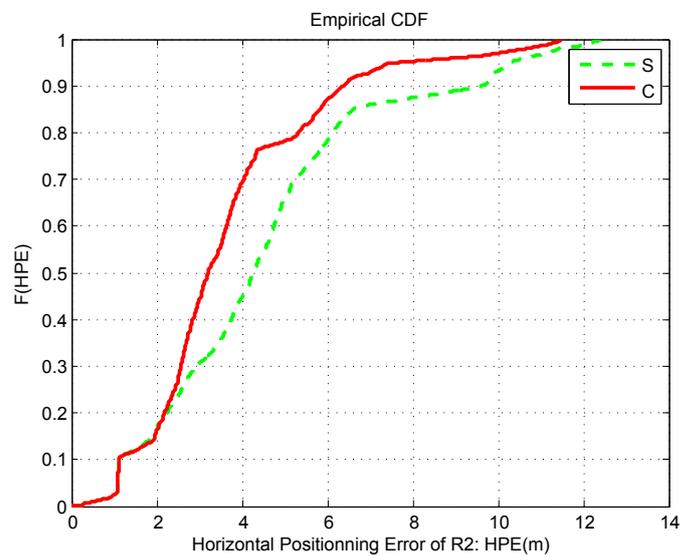


Figure 4.23 – Distribution cumulative des erreurs HPE de R_2 . Fusion CI.

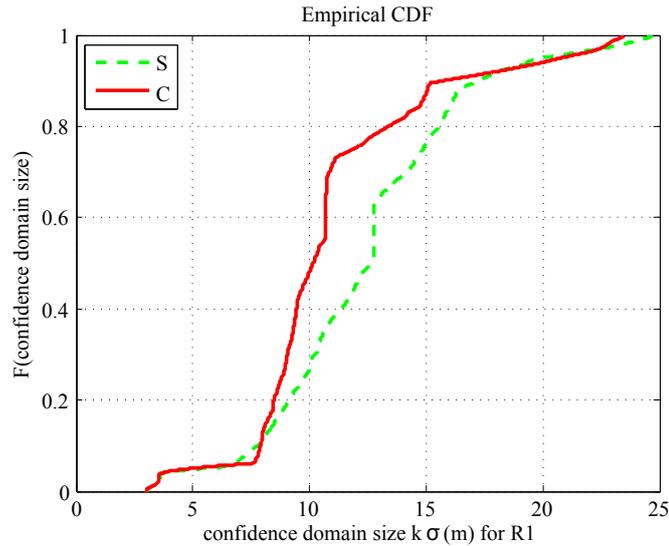


Figure 4.24 – Distribution cumulative de CDS de R_1 . Fusion CI.

cumulées de R_1 et R_2 tout au long l’essai par les méthodes C et S. On peut remarquer pour le véhicule R_1 que la méthode coopérative réduit parfois l’incertitude de l’estimation par rapport à la méthode standalone. Si on considère le 95% percentile, les CDS des méthodes C et S sont sensiblement les mêmes ($20.82m$ et $20.01m$). Concernant le véhicule R_2 , c’est plutôt l’inverse car la méthode C fournit des domaines de confiance plus larges. Les 95% percentiles des domaines de confiances des méthode standalone et coopératives sont encore sensiblement les mêmes. On peut donc conclure que globalement la méthode coopérative avec la fusion CI ne permet pas de réduire l’incertitude, ce qui est conforme avec l’opérateur de fusion utilisé.

La figure 4.26 montre la convergence des biais vers des valeurs assez différentes de la fusion SCC, mais ces erreurs de pseudodistance GNSS restent tout à fait dans des ordres de grandeurs de récepteurs de faibles coûts [118]. De plus, les estimations sont beaucoup moins lisses que celles fournies par la fusion SCC. Les valeurs des biais sont également plus grandes qu’avec la fusion SCC et on constate que les biais les plus importants (par exemple pour les satellites 22 et 17) correspondent à des satellites à angle d’élévation bas, ce qui est cohérent avec le fait qu’ils sont sujets à subir des allongements plus importants lors de la phase finale de la propagation du signal.

4.5.3 Comparaison entre les fusions SCC et CI

Dans cette section, nous allons comparer plus en détails l’impact des opérateurs de fusions SCC et CI sur le positionnement en absolu et en relatif des véhicules.

Les figures 4.27, 4.28 et 4.29 représentent respectivement les erreurs de positionnement horizontales absolues (HPE) et les erreurs de distance relative (RDE) des

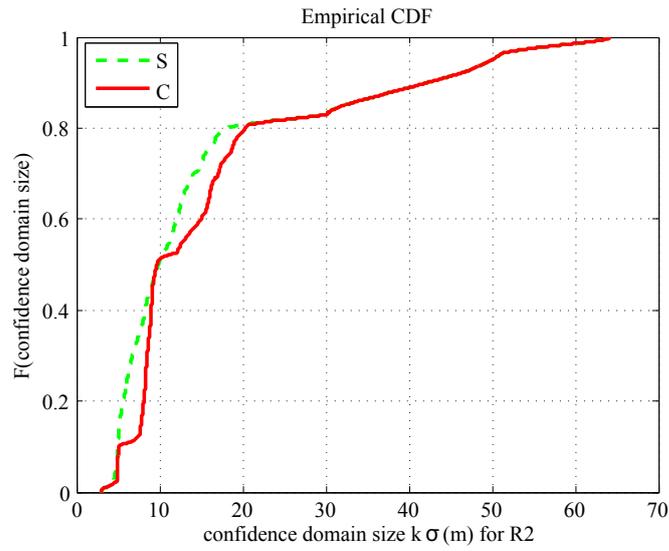


Figure 4.25 – Distribution cumulative de CDS de R_2 . Fusion CI.

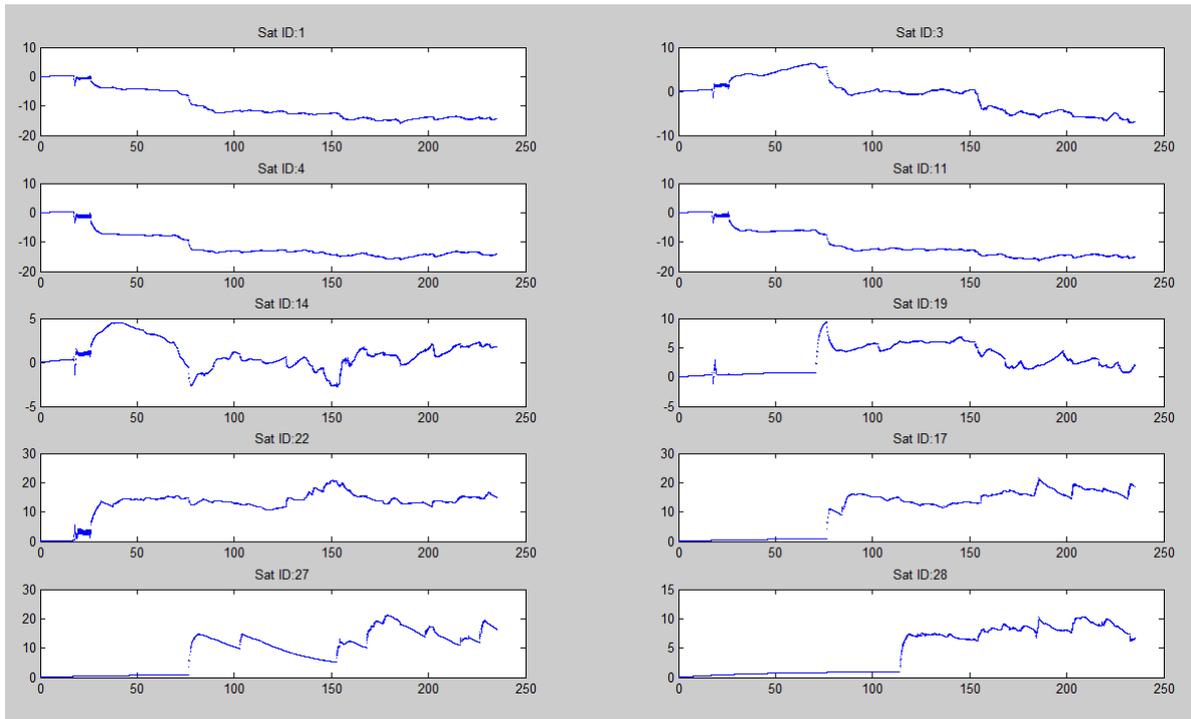


Figure 4.26 – Estimation des biais avec la fusion CI (les unités sont secondes et mètres).

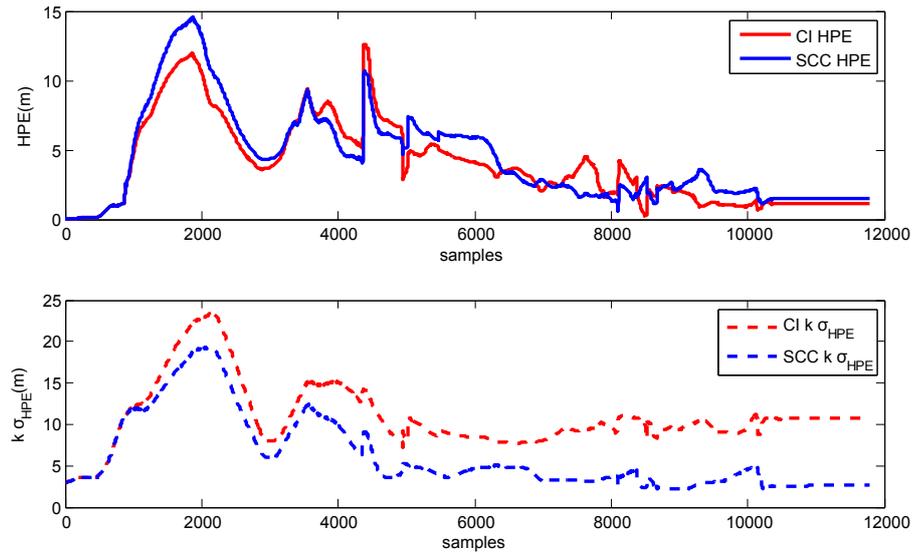


Figure 4.27 – Erreurs HPE absolues et domaines de confiance de R_1 avec les deux méthodes de fusion.

véhicules R_1 et R_2 et leurs domaines de confiance ($k \sigma_{HPE}$ et $k \sigma_{RDE}$) pour chaque type de fusion. On observe que les erreurs absolues HPE de la fusion CI sont inférieures aux erreurs absolues HPE de la fusion SCC en particulier pour le véhicule R_2 , ce qui confirme les observations déjà faites préalablement.

D'après les figures 4.30 et 4.28, 95% des HPE absolues de R_1 et R_2 sont respectivement inférieures à $10.45m$ et $7.64m$ en utilisant la fusion CI et elles sont respectivement inférieures à $12.09m$ et $10.32m$ dans le cas de la fusion basée sur SCC. La fusion CI améliore les erreurs HPE de R_1 de 13.56% et de 26% les erreurs HPE de R_2 par rapport à la fusion SCC. Les mêmes observations peuvent être faites concernant le positionnement en relatif (cf. 4.29). La fusion CI permet donc d'améliorer l'exactitude de façon assez significative.

Les tableaux 4.4 et 4.5 représentent les performances statistiques des fusions CI et SCC. On vérifie que la fusion CI améliore les performances de la localisation en absolu et en relatif quelque soit le critère considéré. Par exemple, la valeur moyenne et l'écart type des erreurs absolues HPE par la fusion CI sont réduits respectivement de 8.94% et 10.32% par rapport à la fusion SCC (cf. tableau 4.4). En positionnement relatif, l'amélioration des erreurs moyennes et l'écart type sont respectivement de 4.46% et 21% par rapport à la fusion SCC.

Intéressons nous maintenant à la taille des domaines de confiance. Les incertitudes de la fusion SCC sont plus petites que dans le cas de la fusion CI (cf. 4.27 et 4.28). D'après la figure 4.32, 95% des domaines de confiance de R_1 sont inférieurs à $17.16m$ en utilisant la fusion SCC et inférieurs à $20.82m$ en utilisant la fusion CI. Concernant le véhicule R_2 (Fig. 4.33), 95% des domaines de confiance sont égaux pour les deux fusions SCC et CI, puisqu'ils sont inférieurs à $50.05m$ dans les deux cas. Cependant

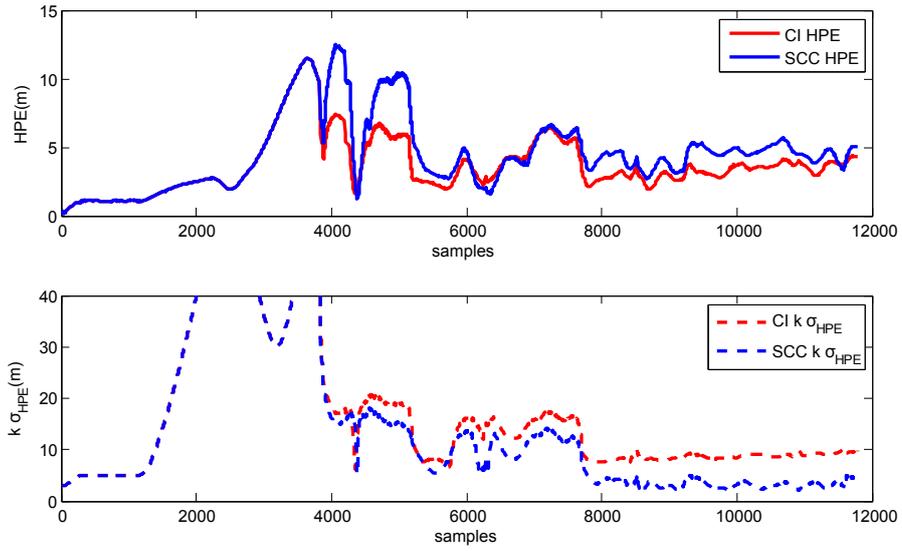


Figure 4.28 – Erreurs HPE absolues et domaines de confiance de R_2 avec les deux méthodes de fusion.

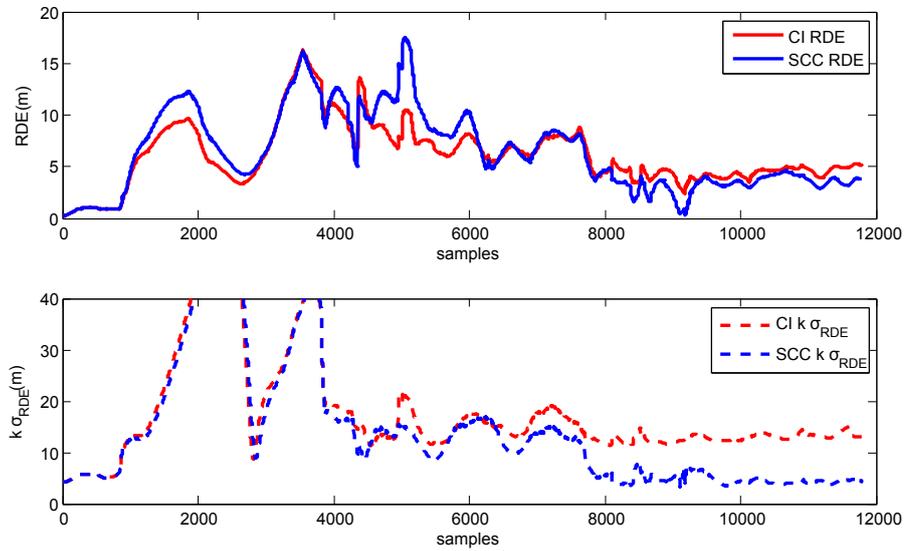


Figure 4.29 – Erreurs relatives RDE et leurs domaines de confiance relatifs avec les deux méthodes de fusion.

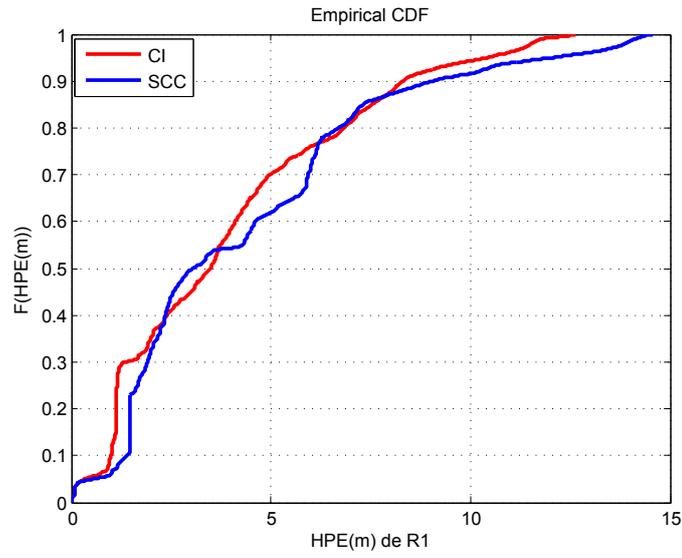


Figure 4.30 – Distribution cumulative des erreurs absolues HPE de R_1 avec les deux méthodes de fusion.

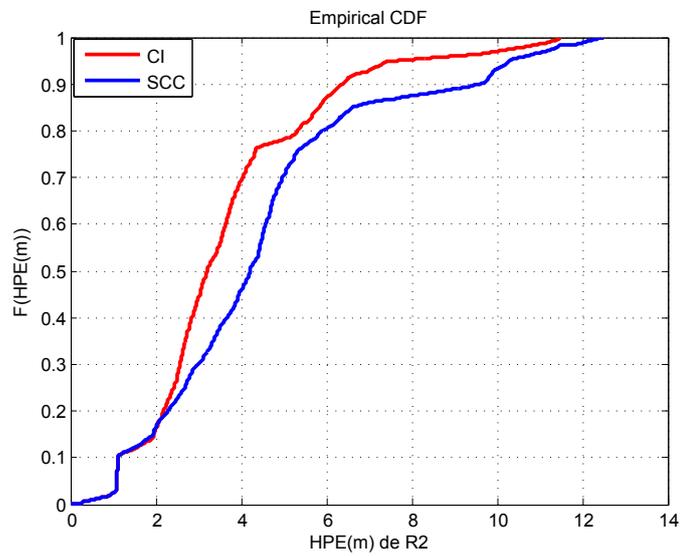


Figure 4.31 – Distribution cumulative des erreurs absolues HPE de R_2 avec les deux méthodes de fusion.

		R1	R2
écart type (m)	SCC	3.39	2.70
	CI	3.045	2.14
Max (m)	SCC	14.55	12.50
	CI	12.64	11.48
Moyenne (m)	SCC	4.36	4.51
	CI	3.97	3.70

Table 4.4 – Statistiques des erreurs HPE en considérant les fusions SCC et CI.

écart type (m)	SCC	3.86
	CI	3.05
Max (m)	SCC	17.46
	CI	16.25
Moyenne (m)	SCC	6.46
	CI	6.16

Table 4.5 – Statistiques des erreurs relatives RDE en considérant la fusion SCC et CI.

pour les percentiles qui sont inférieurs au 80^{ème} percentile, on peut remarquer que les incertitudes des domaines de confiance de la fusion SCC sont plus réduites.

La taille des domaines de confiance est à considérer par rapport à la consistance du filtre. La consistance est obtenue lorsque le domaine de confiance CDS borne correctement l’erreur HPE au risque choisi (i.e, $HPE < k\sigma_{HPE}$). La consistance des filtres selon les erreurs de position horizontale est fondamentale pour fournir un niveau de protection horizontale pertinent pour les véhicules coopératifs. Le niveau de protection vertical est principalement utilisé dans les applications aéronautiques et n’est pas considéré ici.

La consistance des filtres C-EKF-SCC et C-EKF-CI est évaluée par des histogrammes 2D dans la figure 4.34 pour les véhicules R_1 et R_2 dans les deux cas (en utilisant la fusion SCC et en utilisant la fusion CI). Les axes des abscisses et des ordonnées présentent respectivement les erreurs HPE et les CDS (i.e. $k\sigma_{HPE}$). Chaque pixel représente le nombre d’occurrences d’une paire particulière (HPE, CDS). L’échelle de la légende de couleur est logarithmique. Ces histogrammes sont des diagrammes de Stanford simplifiés puisqu’on ne s’intéresse qu’aux régions où on a respectivement $CDS > HPE$ et $HPE > CDS$. Les points où les CDS sont inférieurs aux HPE indiquent une perte de consistance.

D’après les figures 4.34a et 4.34c, les estimations du filtre C-EKF-SCC des véhicules R_1 et R_2 ont respectivement 21.55% et 30.39% des HPE en dehors des bornes de confiance CDS, ce qui indique une perte fréquente de consistance des filtres basés sur la fusion SCC. Cette perte de consistance peut être expliquée par la sensibilité de l’opérateur de fusion SCC au problème de consanguinité des données (*Data incest*).

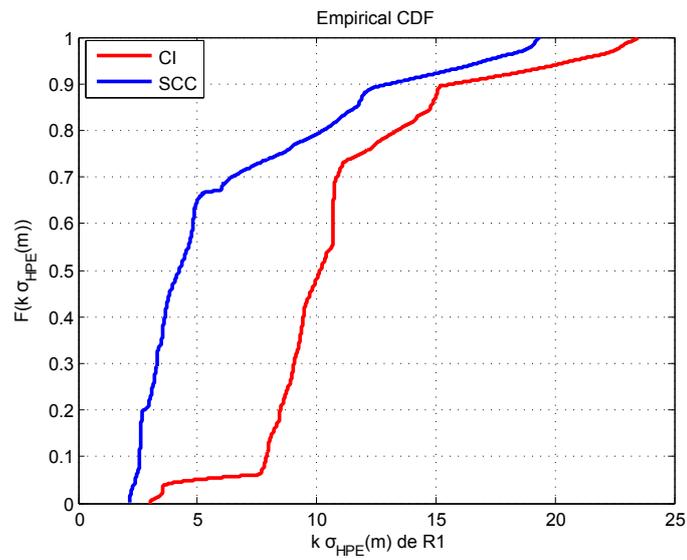


Figure 4.32 – Distribution cumulative des CDS de R_1

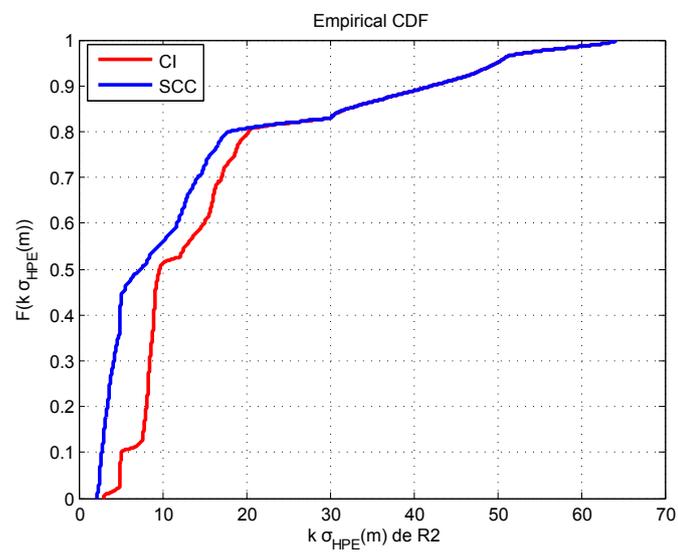


Figure 4.33 – Distribution cumulative des CDS de R_2

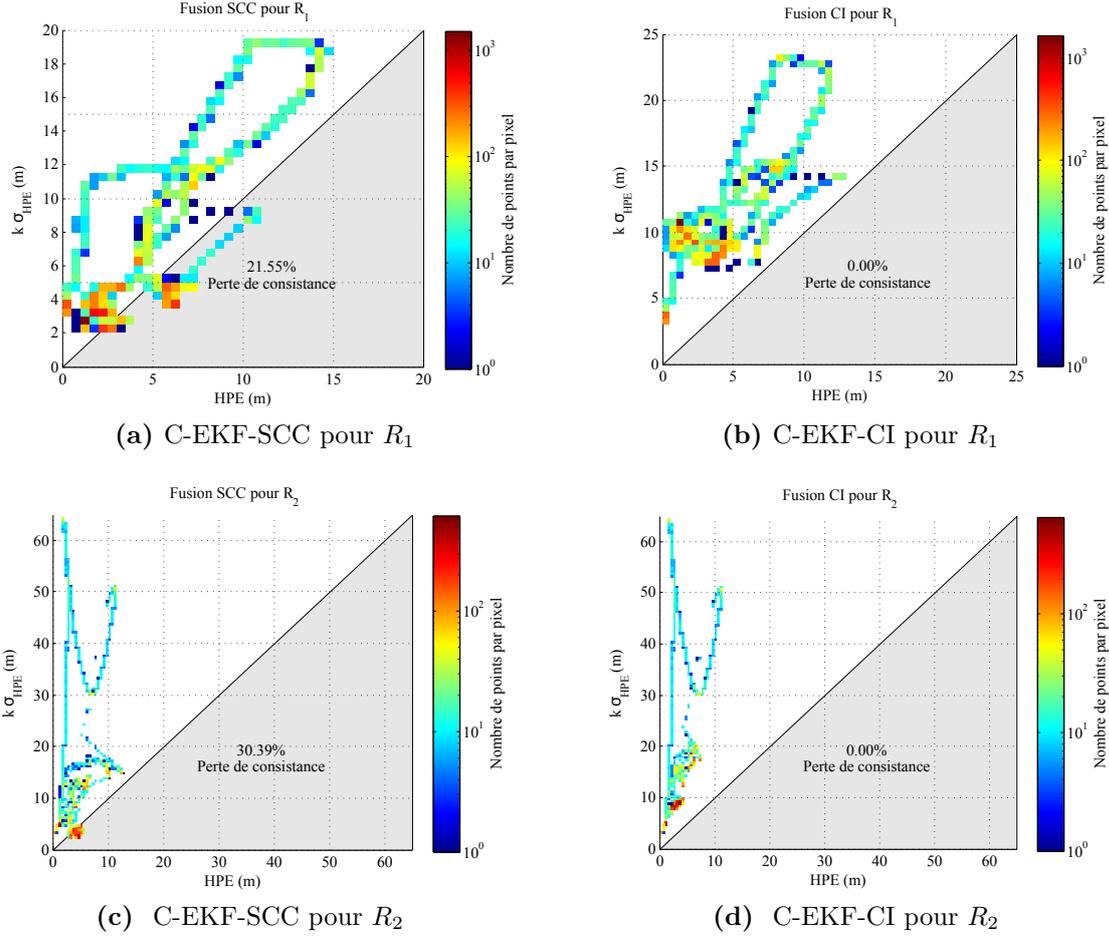


Figure 4.34 – Diagramme de Stanford simplifié : Consistance des filtres en utilisant les 2 fusions pour R_1 et R_2 . (a) C-EKF-SCC pour R_1 , (b) C-EKF-CI pour R_1 , (c) C-EKF-SCC pour R_2 , (d) C-EKF-CI pour R_2

Le problème de consanguinité des données provient de la réutilisation de la même information plusieurs fois dans la fusion. Le résultat de ce problème est illustré par la figure 4.35. Sur cette figure (4.35), deux densités de mesure m_1 (Mesure 1) et m_2 (Mesure 2) sont présentées avec leur fusion optimale de type SCC. Une répétition de la fusion de la densité optimale avec m_2 conduit à une sur-convergence sous forme d'une estimation sous-optimale due aux informations répétées. La fusion de cet exemple est effectuée par l'opérateur SCC de la façon suivante :

$$\begin{aligned}
 \text{Fusion optimale} &= f_{SCC}(m_1, m_2) \\
 \text{Sur-convergence} &= f_{SCC}(f_{SCC}(m_1, m_2), m_2)
 \end{aligned} \tag{4.23}$$

Dans le cas de la fusion SCC, le problème de consanguinité des données provient de

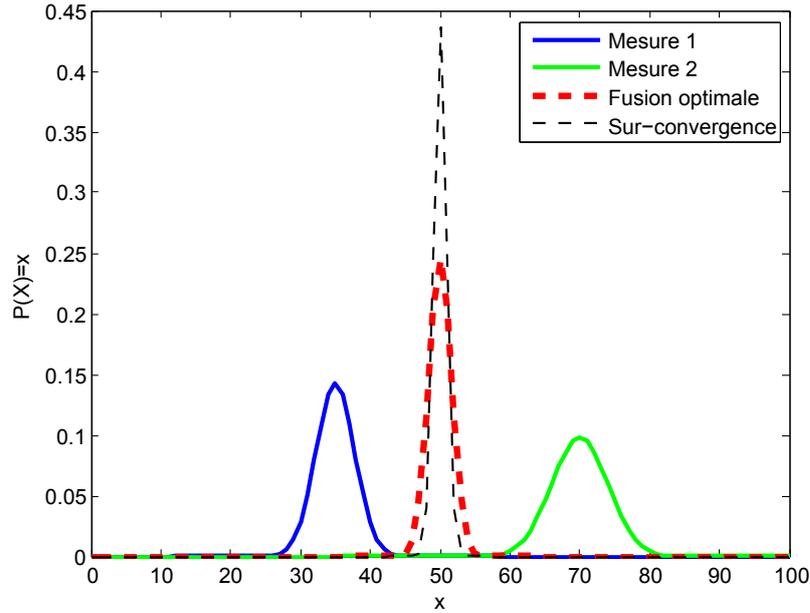


Figure 4.35 – Illustration du problème de sur-convergence dû à la consanguinité des données dans la fusion.

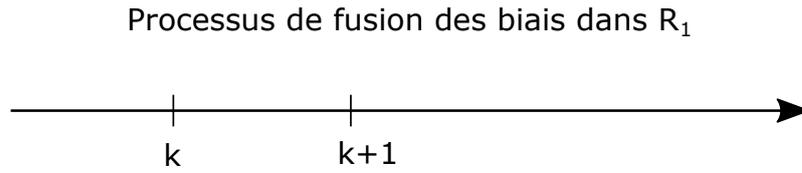


Figure 4.36 – Fusion dans R_1 .

la réutilisation des biais qui ont été déjà fusionnés de manière séquentielle. Le filtre basé sur la fusion SCC sur-estime parfois la confiance dans les biais ce qui engendre la perte de sa consistance. Ci-dessous, on illustre ce problème en se plaçant dans le véhicule R_1 . Les indices i et j seront utilisés respectivement pour R_1 et R_2 pour illustrer le cas général.

— À l’instant k :

Fusion de $\mathbf{b}_{i,k}$ et $\mathbf{b}_{j,k}$ après l’étape d’estimation :

$$\hat{\mathbf{b}}_{ij,k} = f_{SCC}(\mathbf{b}_{i,k}, \mathbf{b}_{j,k})$$

Prédiction de $\mathbf{b}_{i,k+1}$:

$$\begin{aligned}\mathbf{b}_{i,k+1} &= a \hat{\mathbf{b}}_{ij,k} \\ &= a f_{SCC}(\mathbf{b}_{i,k}, \mathbf{b}_{j,k})\end{aligned}$$

— À l'instant $k + 1$:

Fusion de $\mathbf{b}_{i,k+1}$ et $\mathbf{b}_{j,k+1}$ après l'étape de l'estimation :

$$\begin{aligned}\hat{\mathbf{b}}_{ij,k+1} &= f_{SCC}(\mathbf{b}_{i,k+1}, \mathbf{b}_{j,k+1}) \\ &= f_{SCC}(a f_{SCC}(\mathbf{b}_{i,k}, \mathbf{b}_{j,k}), a f_{SCC}(\mathbf{b}_{i,k}, \mathbf{b}_{j,k}))\end{aligned}$$

À ce niveau, le problème de consanguinité des données apparaît clairement car on fusionne $\mathbf{b}_{i,k}$ et $\mathbf{b}_{j,k}$ qui ont été déjà fusionnés à l'instant k . Ainsi de suite, les biais déjà fusionnés seront réutilisés de manière séquentielle.

Dans un système distribué, la même information peut être reçue et traitée plusieurs fois. En combinant l'information, il est donc utile d'utiliser une règle idempotente pour éviter de combiner la même information plusieurs fois comme si elle provenait de différentes sources indépendantes. L'opérateur de fusion SCC n'est pas idempotent puisqu'il ne gère pas les informations dépendantes d'où la nécessité d'utiliser l'opérateur CI.

En contrepartie, les figures 4.34b et 4.34d prouvent que les estimations des positions des véhicules en utilisant la fusion CI sont fournies avec des domaines de confiance fiables (dans l'essai rapporté la consistance est de 100% dans les deux robots), puisque le domaine de confiance borne à chaque instant l'erreur de positionnement horizontal. Les estimations des biais par la fusion CI sont plus fiables car la fusion CI ne néglige pas la dépendance des erreurs de mesures GNSS (i.e. biais). On peut conclure que la fusion CI est plus robuste au problème de consanguinité des données.

L'estimation garantie des positions des véhicules est essentielle dans cette thèse. Il est important de pouvoir positionner les véhicules dans des domaines de confiance fiables afin d'éviter tout risque de collision. Par conséquent, la localisation coopérative Bayésienne basée sur la fusion CI est plus adaptée à notre problème de positionnement garanti.

Dans la suite, la méthode Bayésienne basée sur la fusion CI est donc considérée pour effectuer une comparaison avec la méthode ensembliste basée sur les techniques de la propagation des contraintes et l'inversion ensembliste.

Variabiles		Écart types des bruits
$\rho(m)$	R1	$\sigma_\rho = \sqrt{90000.10^{-SNR/10}}$
	R2	$\sigma_\rho = 9$
$v(m.s^{-1})$	R1	$1e - 3$
	R2	$2e - 3$
$\psi(rad.s^{-1})$	R1	$2.5e - 3$
	R2	$5e - 3$

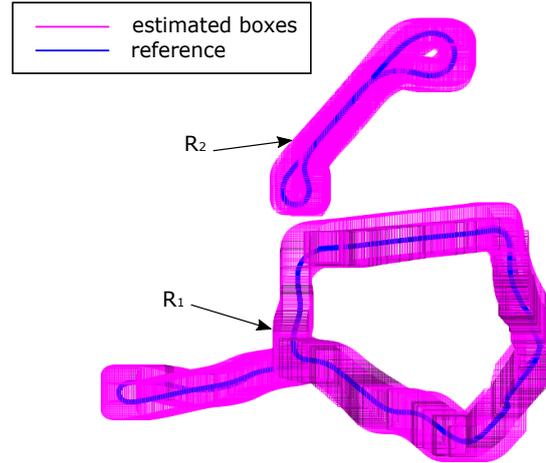
Table 4.6 – Paramètres des bruits des filtres C-EKF et C-SIVIAP.

4.6 Comparaison entre la méthode Bayésienne CI et l’approche ensembliste

Cette section est consacrée à une comparaison détaillée des performances du filtre Bayésien basé sur la fusion d’intersection de covariance (CI) étudié dans ce chapitre et le filtre ensembliste étudié dans le chapitre 3 en utilisant certaines métriques. Le bon choix des performances métriques est primordial pour l’évaluation d’un système et de son application. Pour les systèmes de localisation des véhicules, l’exactitude est une mesure importante qui se réfère généralement à des chiffres statistiques des erreurs de la position. Cependant, pour certaines applications coopératives, des intervalles de confiance fiables garantissant les positions des véhicules sont aussi très utiles pour éviter tout risque de collision.

Il est important de souligner que les deux approches sont très différentes. L’estimation ensembliste fournit des intervalles sur les estimations d’état si les modèles et les bornes des erreurs sont appropriés. Le filtrage de Kalman fournit des ellipsoïdes de confiance pour les estimations, sous l’hypothèse que les bruits sont centrés, blancs et de covariances connues. Afin de pouvoir comparer les deux méthodes, l’incertitude et la consistance des domaines de confiance et l’exactitude des erreurs du positionnement horizontal ont été choisies. Pour les domaines de confiance, les matrices de covariance du filtre de Kalman ont été choisies de telle sorte qu’il est possible d’établir une relation entre les boîtes englobantes définies pour le filtre ensembliste et les ellipses des bruits gaussiens de confiance à 99% de l’approche du filtre de Kalman. Pour les erreurs du positionnement horizontal, les centres des boîtes englobantes ont été adoptés comme approximation ponctuelle des estimations de la méthode ensembliste.

Les deux algorithmes ont été testés avec les mêmes véhicules expérimentaux et le même jeu de donnée a été utilisé après un post-traitement. La figure 4.37 montre une vue de dessus des trajectoires des deux véhicules en utilisant les deux méthodes coopératives. Les algorithmes ont été implémentés avec les mêmes valeurs de statistiques des bruits de mesures présentées dans le tableau 4.6 en utilisant comme entrée le vecteur de mesures DR $\mathbf{u} = \begin{bmatrix} v & \psi \end{bmatrix}^T$.



(a) Trajectoires estimées en utilisant la méthode C-SIVIAP



(b) Trajectoires estimées en utilisant la méthode C-EKF

Figure 4.37 – Trajectoires des véhicules. (a) Références et enveloppes de position en utilisant la méthode C-SIVIAP, (b) Trajectoires estimées en utilisant la méthode EKF. La vitesse moyenne des véhicules était d'environ 30 *Kmph*.

4.6.1 Travaux sur la comparaison des deux approches

Dans le cadre de localisation des robots mobiles, il existe quelques travaux dans la littérature qui ont comparé les deux filtres ensemblistes et Kalman étendu. Jaulin [62] a proposé un filtre basé sur l'approche de propagation des intervalles pour résoudre le problème du SLAM (*Simultaneous Localization And Mapping*) d'un robot sous marin. Dans ce travail, la confiance sur la trajectoire du robot a été comparée en utilisant les ellipses fournies par un filtre classique EKF. Gning et Bonnifait [119] ont présenté un filtre ensembliste basé sur les techniques de propagation des contraintes

	C-EKF	C-SIVIAP
1	résultats non garantis présentés par des valeurs statistiques	résultats garantis dans des enveloppes
2	avec linéarisation des équations	sans linéarisation des équations
	bruit suit une loi gaussienne	bruit inconnu mais borné
3	sur-estimation	ensemble vide
4	opérateur SCC ou CI	intersection des boîtes englobantes
5	vecteur mesure DR	vecteur mesure DR,
	matrice de covariance	estimation des biais et des positions GNSS
	estimation des biais et des positions GNSS	

Table 4.7 – Différences entre méthodes Bayésienne et Ensembliste

pour garantir la position d'un véhicule. Les bornes des erreurs sur la position en 2D ont été évaluées par les filtres BPF (*Box Particle Filter*) et EKF. Dans [120], les considérations théoriques et pratiques des méthodes Bayésiennes et ensemblistes de fusion des données de capteurs ont été comparées. Les performances des deux méthodes ont été examinées sur un système de localisation des robots. Dans le cadre de la navigation robotique visuelle, Di Marco [121] a estimé le temps du contact (i.e le temps nécessaire à l'agent pour atteindre un objet fixe sous l'hypothèse que la vitesse relative est constante) avec un filtre ensembliste. Il a comparé les bornes des bruits de l'estimation ensembliste avec la confiance de 99% ellipsoïdes des bruits gaussiens de l'approche du filtre Bayésien. Récemment, Zamora et Yu [122] ont développé un algorithme ensembliste basé sur les méthodes ellipsoïdes pour résoudre le problème du SLAM d'un robot mobile. L'objectif était de démontrer que les problèmes du SLAM à grande échelle peuvent être résolus en temps réel. Les performances de leur algorithme sont examinées avec le SLAM du filtre EKF. Dans ce chapitre, on note respectivement les méthodes d'erreurs bornées et bayésiennes coopératives par C-SIVIAP et C-EKF. Le tableau 4.7 montre des différences de concept des deux méthodes utilisées selon les critères suivants [62] :

1. Confiance.
2. Approximation.
3. Inconsistance (présence d'outliers ou modèle erroné).
4. Opérateur de fusion des biais.
5. Données partagées.

4.6.2 Critères utilisés pour la comparaison

La comparaison repose sur deux critères : l'exactitude des erreurs de positionnement horizontales (notées HPE) et l'incertitude des domaines de confiance (notés CDS). On redéfinit ci-dessous les HPE et les CDS des deux méthodes.

— **HPE**

4.6 Comparaison entre la méthode Bayésienne CI et l'approche ensembliste

Les HPE (i.e, la distance entre le centre de la boîte et la référence) indiquent l'exactitude de l'approche. Les HPE de deux méthodes sont définis comme suit :

$$HPE = \sqrt{e_x^2 + e_y^2} \quad (4.24)$$

où $e_x = \hat{x} - x_{ref}$, $e_y = \hat{y} - y_{ref}$. (\hat{x}, \hat{y}) et (x_{ref}, y_{ref}) représentent respectivement la position estimée en 2D et la référence.

Pour calculer les HPE de la méthode ensembliste C-SIVIAP, le centre des boîtes englobantes (x_{mid}, y_{mid}) estimées a été choisi comme une représentation ponctuelle. La représentation par le centre des boîtes est simple et bien appropriée au calcul en temps réel. Le centre de gravité des sous-pavages peut être aussi utilisé.

— CDS

Les CDS (i.e. $k\sigma_{HPE}$) ont pour objectif d'évaluer l'incertitude et la consistance de chaque méthode. Le risque de consistance k est égale à 1% pour les deux méthodes. Le σ_{HPE} de la méthode Bayésienne est donné par l'Eq. 4.19. La figure 4.5 représente les HPE ainsi que le CDS utilisés pour la méthode Bayésienne.

Pour la méthode ensembliste, le risque de consistance k à 1% est aussi pris en compte dans les bornes de bruits des mesures de pseudodistances comme suit :

$$[\rho_i^j] = [\rho_i^j - k\sigma_\rho, \rho_i^j + k\sigma_\rho] \quad (4.25)$$

où j est l'indice de satellite et σ_ρ présente l'écart type de mesure de pseudodistance indiqué dans le tableau 4.6. Afin de conserver l'analogie avec la méthode Bayésienne, le domaine de confiance de C-SIVIAP est défini par le segment $k\sigma_{HPE}$ dans la direction des HPE comme il est indiqué sur la figure 4.38. Le CDS de C-SIVIAP est donc l'intersection de la demi-droite HPE avec la boîte de position estimée, dans la direction de la vérité terrain.

4.6.3 Comparaison des méthodes C-SIVIAP et C-EKF-CI

Cette section aborde la comparaison des performances des filtres ensembliste et Bayésien avec la fusion d'intersection de covariance (CI) notée C-EKF-CI. D'après la section 4.5.3, la fusion CI apporte des améliorations en terme d'exactitude et de consistance par rapport à la fusion SCC.

4.6.3.1 Exactitude et incertitude

Les figures 4.39 et 4.41 montrent les HPE et les bornes de confiance de chaque méthode pour les 2 véhicules. Les instants t_1, t_2, t_3, t_4 et t_5 sur les figures 4.39 et 4.41 sont :

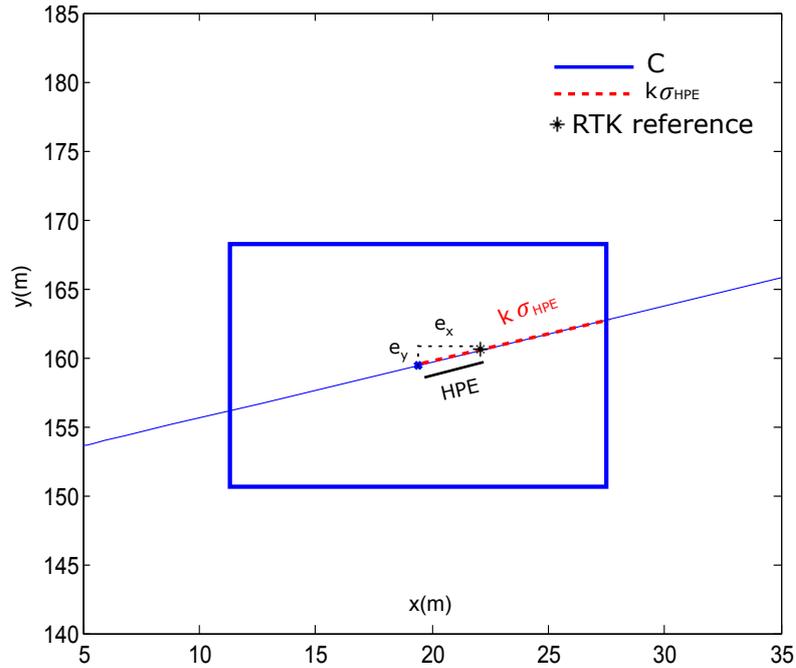


Figure 4.38 – HPE et $k\sigma_{HPE}$ de la méthode C-SIVIAP

$t_1 = 17.2s$: le début de la réception des mesures GNSS dans R_1 .

$t_2 = 76.5s$: le début de la réception des mesures GNSS dans R_2 .

$t_3 = 87.1s$: changement brutal des erreurs HPE dans R_1 dû à la référence.

$t_4 = 207.4s$: fin de la réception des mesures GNSS dans R_1 .

$t_5 = 225.1s$: changement brutal des erreurs HPE dans R_2 .

La fin de la réception des mesures GNSS dans R_2 est à la fin de l'essai.

Les filtres C-EKF-CI de R_1 et R_2 sont garantis puisque les bornes de confiance contiennent les erreurs HPE. Les HPE de la méthode Bayésienne présentent plus de fluctuations par rapport aux HPE de la méthode ensembliste. Les deux filtres de chaque véhicule sont initialisés par la vérité terrain pour les positions et par la solution d'une méthode standalone itérative de Newton-Raphson pour les décalages d'horloges et leur dérive. Ce qui explique des HPE proche de zéro avant l'instant t_1 .

Nous allons maintenant apporter des explications concernant les variations des estimations des méthodes pour les deux véhicules. Commençons par le véhicule R_1 (cf. 4.39). Dans l'intervalle $[t_1, t_2]$, on observe de grandes valeurs HPE pour R_1 avec la méthode Bayésienne. Cela est dû à une mauvaise visibilité des satellites qui correspond au passage du véhicule 1 entre deux bâtiments. Cependant, les HPE de la méthode C-SIVIAP de R_1 restent proche de zéro ce qui illustre la robustesse de la méthode ensembliste. Dans l'intervalle $[t_2, t_4]$, on remarque une réduction des HPE et leurs domaines CDS pour la méthode Bayésienne du véhicule R_1 grâce à la coopération avec le véhicule R_2 à l'exception de l'instant t_3 où on a une augmentation

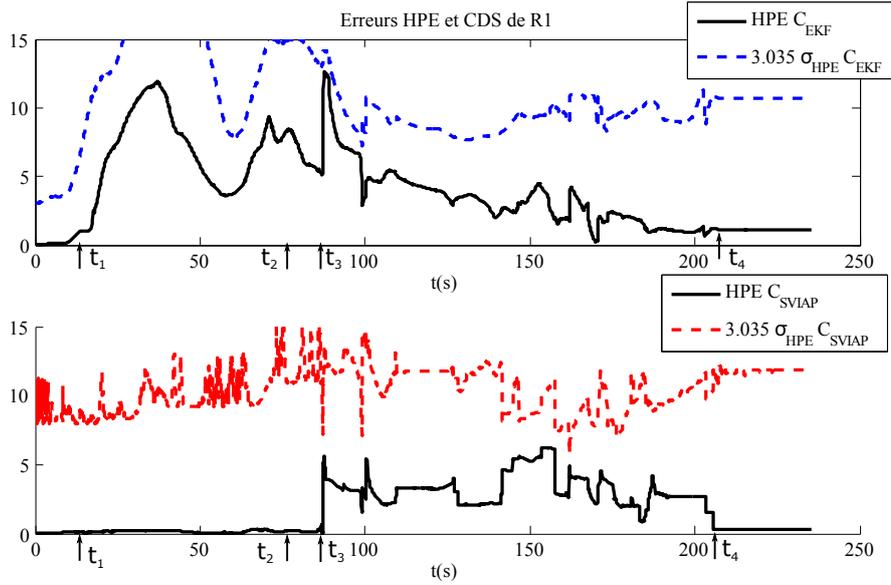


Figure 4.39 – HPE et CDS de la position absolue de R_1 par les méthodes C-EKF-CI et C-SIVIAP.

brutale des HPE. Ce changement brutal est dû à un saut de référence lors de l'acquisition des données comme il est montré sur la figure 4.40. Contrairement à la méthode C-EKF, on observe une augmentation des HPE de la méthode ensembliste avec une réduction d'incertitude des domaines CDS entre t_2 et t_4 . Une explication possible est que les boîtes estimées de positions sont mieux contractées qu'avant l'instant t_2 grâce à la coopération, ce qui provoque une déformation des boîtes ainsi que l'éloignement de leurs centres par rapport aux références. À partir de l'instant t_4 , les HPE et leurs CDS sont constants pour les deux méthodes car le véhicule R_1 ne reçoit plus de mesures GNSS. Il est en arrêt (position constante et vitesse nulle) mais il prédit encore sa position et coopère avec son partenaire R_2 qui continue de mettre à jour sa propre position.

Nous nous intéressons maintenant au véhicule R_2 (cf. figure 4.41). Dans l'intervalle $[t_1, t_2]$, le véhicule R_2 prédit sa position et coopère avec R_1 . Les HPE de la méthode Bayésienne augmentent à l'inverse de la méthode ensembliste qui garde des erreurs proches de zéro, comme juste après l'initialisation. Dans l'intervalle $[t_2, t_5]$, Les HPE de deux méthodes sont inférieurs à $6m$. Comme c'est le cas pour le véhicule R_1 (dans l'intervalle $[t_2, t_4]$), on observe une réduction des HPE de la méthode Bayésienne tandis que les HPE de la méthode ensembliste semblent dériver jusqu'à l'instant t_5 . Le changement brutal à l'instant t_5 est apparu juste après l'arrêt du véhicule R_1 ce qui prouve que R_1 dégrade parfois les performances de R_2 comme nous l'avons mentionné dans le chapitre 3.

Les figures 4.42 et 4.43 présentent les boîtes à moustache (i.e. box-plot) des erreurs HPE des méthodes Bayésienne et ensembliste pour les 2 véhicules. Ces graphiques

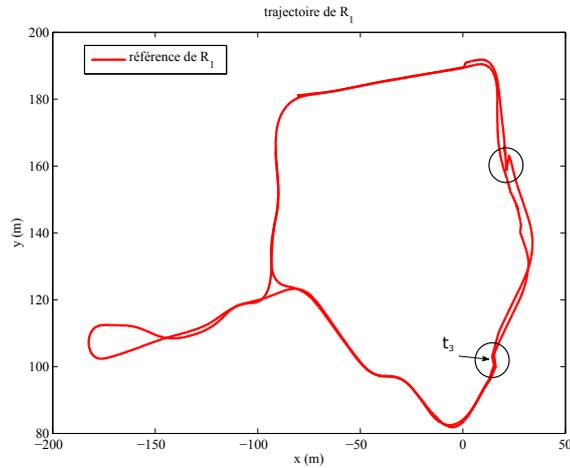


Figure 4.40 – Trajectoire de référence de R_1 . Les zones encadrées présentent des sauts de référence.

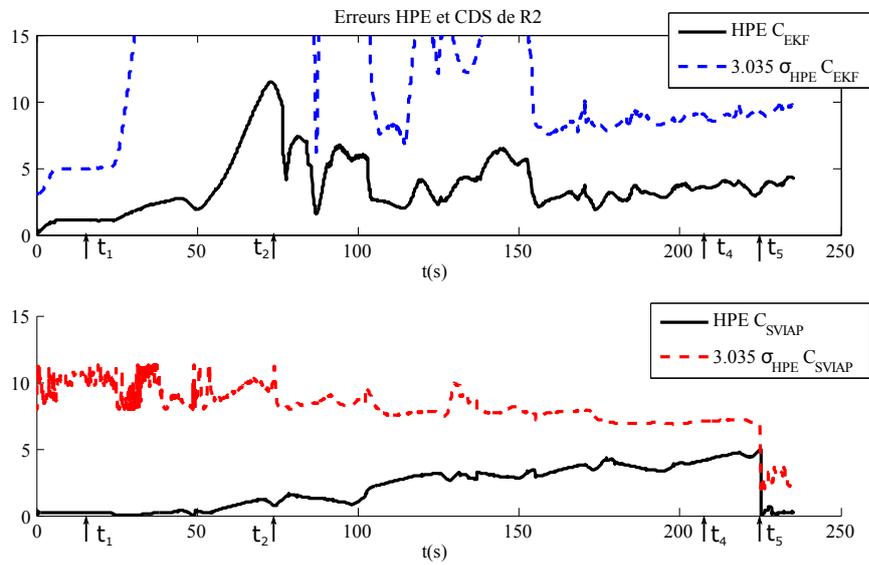


Figure 4.41 – HPE et CDS de la position absolue estimée en 2D de R_2 .

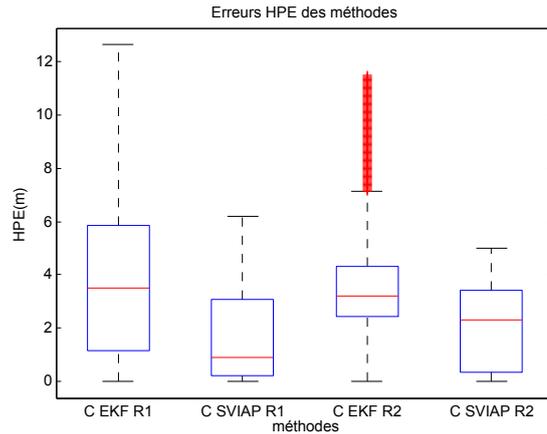


Figure 4.42 – Statistiques des HPE des véhicules R_1 et R_2 .

permettent de comprendre la répartition des erreurs et d'identifier les valeurs extrêmes qui se trouvent en dehors du rectangle, d'une façon visuelle. La valeur centrale du graphique est la médiane. Les bords du rectangle sont le premier et le troisième quartiles (i.e. $Q1$ et $Q3$). Les extrémités des moustaches sont respectivement : $\max(\min, Q1 - 1.5(Q3 - Q1))$ et $\min(\max, Q3 + 1.5(Q3 - Q1))$.

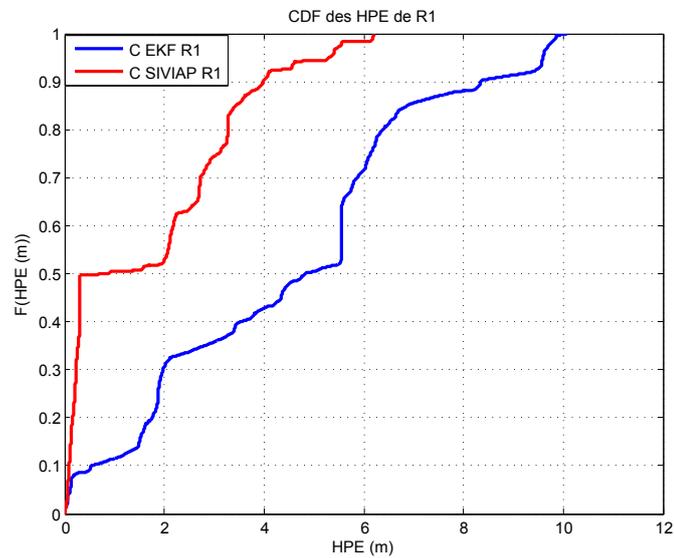
95% des HPE des méthodes C-EKF-CI et C-SIVIAP pour le véhicule R_1 sont respectivement inférieures à $10.46m$ et $5.32m$. 95% des HPE des méthodes C-EKF-CI et C-SIVIAP pour le véhicule R_2 sont respectivement inférieures à $7.64m$ et $4.41m$. Soit 49.13% et 42.27% de gain d'exactitude de C-SIVIAP par rapport à C-EKF-CI, respectivement pour R_1 et R_2 . Ces résultats prouvent que les erreurs du positionnement horizontale de la méthode ensembliste sont inférieures à la méthode Bayésienne.

Les figures 4.44 et 4.45 évaluent l'incertitude des domaines de confiance des méthodes C-EKF-CI et C-SIVIAP. D'après la Fig. 4.44, les médianes de CDS des deux méthodes sont proches considérant les deux véhicules.

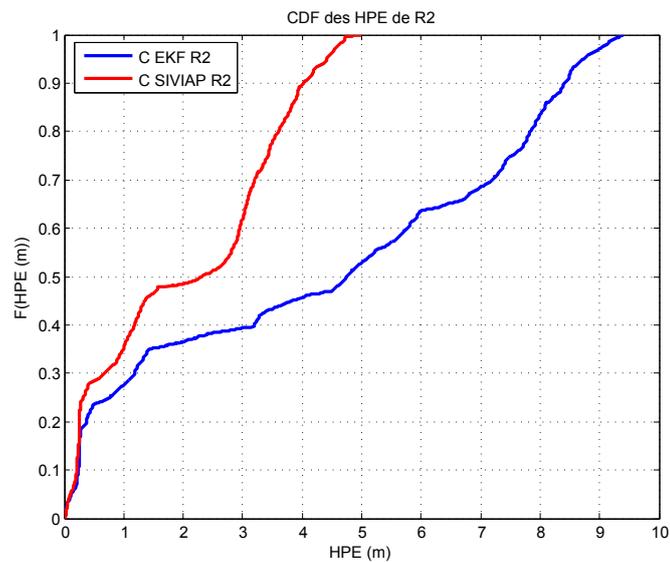
La taille des domaines de confiance de R_2 par le filtre ensembliste est nettement inférieure à la taille du filtre Bayésien : 95% des CDS des méthodes C-EKF-CI et C-SIVIAP sont respectivement inférieures à $50.02m$ et $10.46m$, soit 79.08% de différence (c.f, Fig. 4.45). Si on considère le véhicule R_1 la taille des CDS de la méthode C-SIVIAP est plus réduite qu'à la fin de l'essai, puisque les incertitudes de CDS à 95% sont réduites de 38.04% par C-SIVIAP par rapport à C-EKF-CI.

Les tableaux 4.8 et 4.9 résument les performances des deux méthodes C-SIVIAP et C-EKF-CI.

D'après le tableau 4.8, l'amélioration des erreurs absolues HPE apportée par le filtre ensembliste par rapport au filtre C-EKF-CI respectivement pour R_1 et R_2 sont de 52.64% et 34.96%.



(a) Accumulation des HPE de R_1



(b) Accumulation des HPE de R_2

Figure 4.43 – Accumulation des HPE pour les 2 véhicules

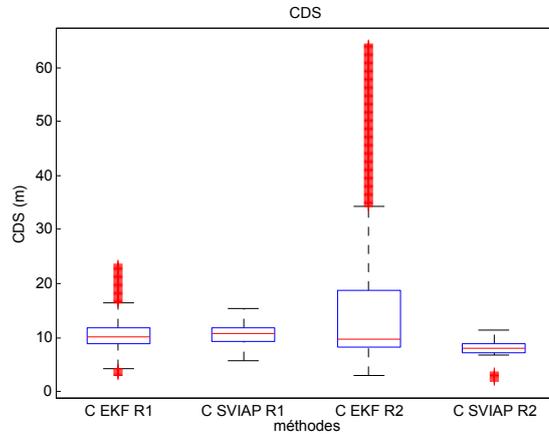
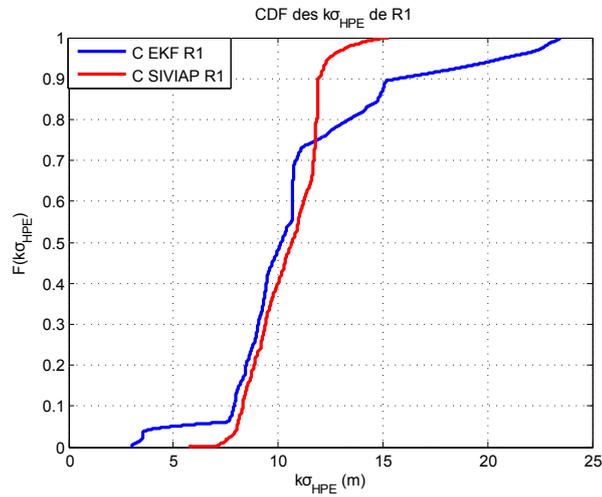


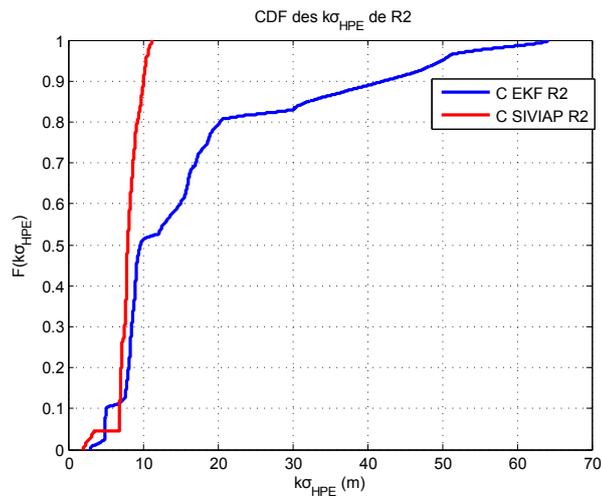
Figure 4.44 – CDS des véhicules R_1 et R_2 .

HPE(m)	C-SIVIAP		C-EKF-CI		C-SIVIAP% C-EKF-CI	
	R_1	R_2	R_1	R_2	GainR1(%)	GainR2(%)
moyenne	1.70	2.06	3.48	3.20	51.14	35.62
écart-type	1.71	1.53	3.04	2.14	43.75	28.50
médiane	0.89	2.31	3.48	3.20	74.42	27.81
75 ^{ème} percentile	3.08	3.42	5.84	4.30	47.26	20.46
95 ^{ème} percentile	5.32	4.41	10.46	7.64	49.13	42.24
max	6.21	4.99	12.64	11.48	50.16	56.53

Table 4.8 – Performances métriques des HPE des véhicules en considérant les deux méthodes.



(a) Distribution Cumulative des CDS de R1



(b) Distribution Cumulative des CDS de R2

Figure 4.45 – Distribution Cumulative des domaines de confiance des HPE pour les 2 véhicules

CDS(m)	C-SIVIAP		C-EKF-CI		C-SIVIAP% C-EKF-CI	
	R_1	R_2	R_1	R_2	GainR1(%)	GainR2(%)
moyenne	10.45	8.06	10.97	16.91	4.74	52.33
écart-type	1.55	1.60	4.02	13.97	61.44	88.54
médiane	10.66	8.00	10.16	9.68	-4.92	17.35
75 ^{ème} percentile	11.76	8.94	11.88	18.66	1.01	52.09
95 ^{ème} percentile	12.44	10.46	20.80	50.02	40.19	79.08
max	15.29	11.31	23.49	64.27	34.90	82.40

Table 4.9 – Performances métriques des CDS des véhicules en considérant les deux méthodes.

Le gain de réduction de la taille des domaines de confiance apporté par le filtre ensembliste par rapport au filtre C-EKF-CI respectivement pour R_1 et R_2 sont de 22.89% et 61.96% (cf. 4.9).

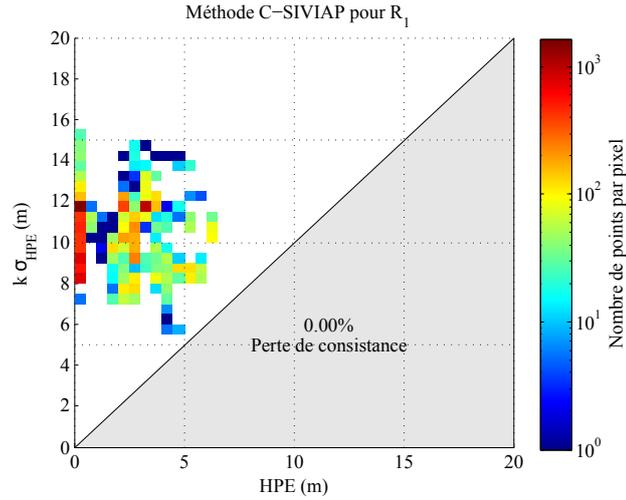
On peut donc conclure que l'approche ensembliste est plus exacte et fournit des domaines de confiance plus petit que l'approche Bayésienne.

4.6.3.2 Consistance

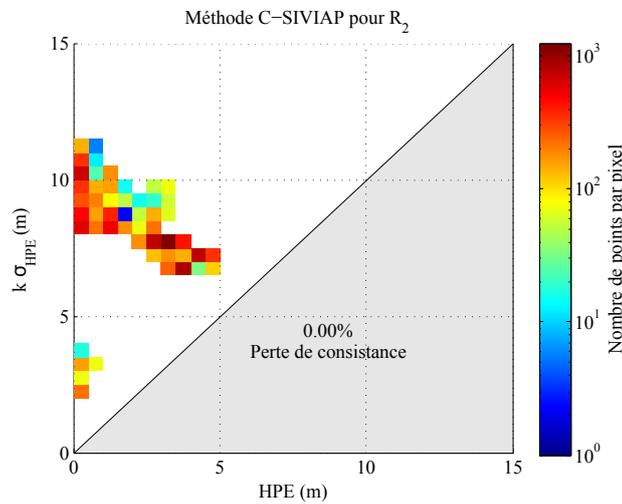
Dans nos essais, le filtre Bayésien est consistant à 100% avec la fusion par intersection de covariance des biais d'après les figures 4.34b et 4.34d de la section 4.6. Considérons maintenant la méthode C-SIVIAP. Le diagramme de Stanford simplifié de chaque véhicule sur la figure 4.46 prouve la consistance des filtres ensemblistes des véhicules R_1 et R_2 (puisque'il n'y a aucun point dans la zone grise). Il montre aussi que les incertitudes sont très diminuées par rapport à la méthode Bayésienne.

D'après la comparaison des résultats expérimentaux de deux méthodes considérées, on peut conclure que la méthode coopérative ensembliste présente des performances meilleures que la méthode Bayésienne, en particulier si on dispose d'un nombre élevé de satellites en vue comme c'est le cas pour le véhicule R_2 . L'approche ensembliste est non seulement plus exacte, mais elle fournit des domaines de confiance plus petits avec une consistance de 100%. Le gain est significatif selon ces deux critères.

Intéressons nous enfin aux performances du véhicule R_1 par rapport à R_2 . En se basant sur les 95^{ème} percentiles des erreurs HPE et du domaine de confiance en considérant les deux méthodes, les moins bonnes performances sont obtenues pour le véhicule R_1 . En observant les résultats des véhicules R_1 et R_2 en utilisant la méthode ensembliste, en particulier le 95^{ème} percentile des HPE et CDS, la dégradation des performances (en termes d'exactitude et d'incertitude) des estimations de R_1 par rapport R_2 peut être expliquée par le fait que R_1 possède des capteurs DR de qualité inférieures par rapport au véhicule R_2 . La coopération améliore surtout la localisation du véhicule qui possède les meilleurs capteurs de navigation à l'estime, ce qui tend à montrer que les capteur de DR jouent un rôle très important.



(a) C-SIVIAP pour R_1



(b) C-SIVIAP pour R_2

Figure 4.46 – Diagramme de Stanford simplifié. Consistance de filtre ensembliste de chaque véhicule.

4.7 Conclusion

Dans ce chapitre, nous avons développé une approche Bayésienne séquentielle coopérative pour le positionnement des véhicules coopératifs en absolu et en relatif afin d’avoir un élément de comparaison pour l’approche ensembliste. Les résultats fournis par cette approche coopérative Bayésienne montrent également une amélioration nette du positionnement absolu et mutuel. L’utilisation des opérateurs de fusion SCC et CI prouvent que la méthode coopérative donne de meilleurs résultats en terme d’exactitude par rapport à la méthode standalone. L’effet de l’opérateur de fusion sur les estimations et les domaines de confiance a également été étudié.

On a montré théoriquement que l'opérateur de fusion SCC est sensible au problème de consanguinité des données puisque des biais déjà fusionnés sont réutilisés de manière séquentielle dans la fusion. Expérimentalement, nous avons remarqué, qu'avec la fusion SCC, la méthode coopérative perd effectivement parfois sa consistance. La consistance du filtre est grandement améliorée par la fusion CI qui ne néglige pas la dépendance des erreurs de mesure GNSS. De façon plus étonnante, nous avons remarqué que la fusion CI améliore également l'exactitude de la méthode coopérative, ceci étant certainement dû à une meilleure estimation des biais.

Une fois le choix fait de la méthode Bayésienne, la localisation fournie par la méthode à erreurs bornées a été comparée avec la méthode Bayésienne utilisant l'intersection de covariance. Cette comparaison a montré que les deux méthodes ensembliste et Bayésienne sont garanties à 100% dans nos essais. La méthode coopérative ensembliste présente de meilleures performances en termes d'exactitude (erreurs horizontales plus faibles) et d'incertitude (domaines de confiance plus petits). Ce résultat montre que les méthodes ensemblistes peuvent être très performantes contrairement aux idées reçues qui généralement les qualifient de très pessimistes. Enfin, nous avons noté que si le véhicule dispose de bons capteurs de navigation à l'estime, comme c'est le cas pour le véhicule R_2 , la coopération lui apporte beaucoup. L'information clé à tirer de cette étude est que l'approche ensembliste est très adaptée aux applications nécessitant une haute intégrité pour la navigation robotique.

Chapitre 5

Conclusion et perspectives

5.1 Conclusion

Ces travaux de thèse ont abordé l'étude et la mise en œuvre de systèmes de localisation coopérative pour des systèmes robotiques mobiles en interaction. Les mesures directes entre agents n'ont pas été considérées dans cette thèse. Les technologies considérées pour la localisation utilisent des données de radionavigation sur des balises, comme le positionnement global par satellites (GNSS) et par balises radios (de type UWB). Ces systèmes offrent respectivement des mesures brutes de pseudodistances et de distances. Des telles mesures souffrent souvent d'erreurs systématiques (appelées biais) qui peuvent dégrader la qualité du positionnement.

Un objectif fréquent pour un système coopératif est que ses membres puissent naviguer ensemble et s'entraider mutuellement pour accomplir une mission coopérative. L'amélioration du positionnement en absolu et en relatif grâce à la coopération et l'échange de données est un élément important. La sécurité des robots, des objets ou des personnes aux alentours doit également être prise en considération. De ce fait, une estimation fiable de l'incertitude des positions calculées est nécessaire. Cette incertitude peut être représentée par une borne de l'erreur d'estimation sur la position, attachée à un indicateur de risque donné. De ce point de vue, le problème d'incertitude devient un problème de calcul de domaines de confiance associé aux positions, contenant les positions réelles des robots de la flotte avec une forte probabilité.

Dans cet objectif, nous avons proposé d'estimer les biais de mesures brutes pour réduire l'incertitude et son effet dégradant sur le positionnement. Pour cela des modèles de biais ont été introduits dans les observations. Nous avons focaliser au début les travaux sur des méthodes ensemblistes par intervalles car il est connu qu'elles sont capables d'estimer la position de chaque robot avec un domaine de confiance fiable.

Nous avons commencé cette recherche par étudier l'observabilité du système coopératif composé de n_r robots. C'est un moyen rassurant qui a montré la faisabilité de l'observation d'état du système coopératif augmenté avec les biais. Deux cas ont été considérés selon la nature des mesures brutes des observations. Dans le cas des mesures de distances sur balises, il a été démontré qu'il est possible d'observer l'état

du système dans le cas statique, s'il existe au moins $\frac{3n_r}{n_r-1}$ balises (en particulier, pour deux robots il faut au moins six balises). Cette observabilité est basée sur le critère du rang. Si la matrice d'observabilité est de rang plein alors le système est faiblement localement observable. Dans le cas des mesures de pseudodistances GNSS, il était connu que ce n'est pas possible d'observer les biais si les robots ne bougent pas. Les problèmes de non-observabilité sont dues aux corrélations de mesures GNSS et à l'insuffisance d'autres mesures indépendantes. Nous avons donc proposé d'étudier l'observabilité dans le cas dynamique en rajoutant de l'information sur le déplacement des robots et une connaissance de l'évolution des biais dans le temps. L'étude de l'observabilité est basée sur les techniques de dérivées de Lie ce qui conduit à une observabilité faible locale. Nous avons montré que, si les robots se déplacent, le système peut être observable sous la condition que le modèle des biais ne soit pas constant et que les robots possèdent au moins 4 satellites en commun. Ce résultat est indépendant du nombre de robots utilisés dans le système.

Une formulation du problème de localisation coopérative sous forme de contraintes liées aux mesures biaisées de distances et pseudodistances a été proposée. La linéarisation des biais sous forme d'erreurs additionnelles sur les distances ou les pseudodistances a été considérée. La linéarisation est fondée sur l'hypothèse que les lignes de vue, mesurées et réelles, entre les balises ou les satellites et les utilisateurs sont parallèles.

Les données échangées pour la fusion et la mise à jour de l'état du groupe sont les estimations de positions et de biais ainsi que les mesures de capteurs proprioceptifs. Ce type d'informations partagées entre les robots est considéré d'hybride avec des données d'un état mis à jour et des données de capteurs : ceci classe notre approche de localisation coopérative dans la catégorie des méthodes coopératives hybrides. Des problèmes de consanguinité de données peuvent être produits lors de la fusion des états mis à jour et des informations redondantes. Les méthodes ensemblistes sont considérées comme une solution à ce type de problème : les données redondantes peuvent être fusionnées autant de fois que l'on veut sans risque de sur-convergence des estimations (i.e, convergence vers des valeurs fausses). De plus, ces méthodes sont capables de gérer les problèmes non-linéaires. Des méthodes ensemblistes basées sur des techniques de propagation de contraintes ont d'abord été utilisées pour le problème du positionnement coopératif par balises. Le positionnement à partir des distances biaisées simulées a permis d'améliorer les performances de localisation en absolu et en relatif par rapport à une méthode classique non-coopérative. Les critères de comparaison sont l'exactitude et l'incertitude de chaque méthode selon les métriques des erreurs horizontales du positionnement et des domaines d'incertitude de positions bidimensionnelles. L'ajout de techniques d'inversion ensembliste permet d'améliorer des performances de localisation. L'augmentation du nombre de robots dans le groupe améliore également les performances de localisation en termes d'exactitude et d'incertitude. De ce fait, l'utilisation des méthodes de propagation de contraintes avec des méthodes d'inversion ensemblistes ont été adoptés pour le positionnement coopératif.

Notre approche ensembliste distribuée a ensuite été appliquée à la localisation coopérative de deux véhicules expérimentaux, cette fois, en utilisant des pseudodistances GNSS mesurées par des récepteurs de faible coût. Les résultats expérimentaux ont amélioré le positionnement standalone en donnant des positions de véhicules plus exactes et des domaines de confiance plus petits et fiables.

Par la suite, nous avons développé une approche Bayésienne séquentielle coopérative pour le positionnement des véhicules coopératifs afin de la comparer à l’approche ensembliste. Deux opérateurs de fusion par Combinaison Convexe Simple (SCC) et par Intersection de Covariance (CI) ont été utilisés. Les résultats fournis par cette approche Bayésienne avec les deux types de fusion ont aussi amélioré le positionnement absolu et mutuel en termes d’exactitude par rapport à la méthode classique. L’effet de la fusion sur l’estimation des domaines de confiance a également été étudié. Des pertes de consistance ont été observées avec la fusion SCC qui est sensible au problème de consanguinité des données, car des biais déjà fusionnés sont réutilisés de manière séquentielle dans la fusion. Cependant, la consistance du filtre a été améliorée par la fusion CI qui ne néglige pas la dépendance des erreurs de mesure GNSS en dépit d’une augmentation d’incertitude par rapport à la fusion SCC.

L’estimation de l’incertitude des positions des véhicules étant l’un des objectifs de cette thèse, il est donc important de pouvoir positionner les véhicules dans des domaines de confiances fiables afin d’éviter tout risque de collision. Par conséquent, la localisation garantie par des méthodes à erreurs bornées a été comparée avec la méthode Bayésienne basée sur l’intersection de covariance. De cette comparaison, on a pu voir que la méthode coopérative ensembliste présente des performances meilleures par rapport à la méthode Bayésienne en termes d’exactitude et de réduction de l’incertitude, en particulier si le véhicule dispose d’un nombre élevé de mesures GNSS et de bons capteurs de navigation à l’estime.

5.2 Contributions

L’étude de systèmes coopératifs avec des mesures biaisées est un sujet peu abordé dans la littérature, en particulier roboticienne. Nous avons proposé une étude originale de l’observabilité du système coopératif de n_r robots mobiles dans les cas des mesures biaisées de distances radios sur balises et de pseudodistances GNSS. A notre connaissance, l’observation de systèmes robotiques mobiles coopératifs avec uniquement des mesures GNSS et des modèles d’évolution de biais et de robots a été effectuée pour la première fois.

Nous avons développé un nouvel algorithme de localisation coopérative pour traiter le cas des mesures biaisées avec des méthodes ensemblistes basées sur la propagation de contraintes sur des intervalles et l’inversion ensembliste. La coopération a été réalisée par le partage des positions estimées, des biais estimés et des mesures proprioceptives. L’algorithme coopératif a été validé en simulation avec dix robots.

L'évaluation de l'algorithme proposé avec des essais expérimentaux sur deux véhicules est également une contribution de ce travail. Ceci a montré que la coopération apporte un gain réel par rapport à une méthode classique non coopérative.

Toujours en utilisant des données réelles, nous avons comparé les performances de la méthode ensembliste coopérative avec une méthode Bayésienne coopérative séquentielle basée sur la fusion par intersection de covariance qui constitue l'état de l'art du domaine. Nos résultats montrent que l'approche ensembliste est plus performante à la fois en termes d'exactitude et d'incertitude.

5.3 Perspectives

Les résultats obtenus par cette recherche sont encourageants. Cependant, ils soulèvent de nouveaux champs d'investigation. En effet, il reste encore quelques points qui méritent d'être étudiés. Ils sont cités ci-dessous.

Expériences à grande échelle avec une datation temps réel des données

L'hypothèse principale de ces travaux est que les différents véhicules comportent des erreurs de pseudodistances communes pour un satellite commun. Cela est vrai car les scénarii ont été effectués dans un environnement à ciel assez ouvert. Cependant, cette hypothèse ne tient plus en milieu urbain dense car les erreurs de pseudodistances peuvent être différentes pour les véhicules à cause de sources d'erreur comme les trajets multiples et les phénomènes NLoS¹ de propagation des signaux de satellites. De plus, la séquence de données de test que nous avons utilisée pour valider la méthode proposée n'est seulement que de trois minutes car il a été assez compliqué d'obtenir un bon jeu de données synchronisées. Cette durée est probablement suffisante pour les types de terrains considérés où la distance entre les deux véhicules ne dépasse pas 1km. D'autres essais de longues durées pourraient être réalisés afin d'observer l'effet de la coopération à plus long terme. D'autres tests doivent être effectués afin de vérifier l'efficacité de la méthode proposée dans différents scénarios, comme par exemple, planifier des scénarios dans des milieux à ciel ouvert, péri-urbains ou encore de type canyon urbain.

Nous avons rencontré aussi quelques problèmes techniques lors de l'acquisition des données de la vérité terrain avec le récepteur Septentrio. Ces imprécisions ont dégradé parfois l'analyse des résultats de la méthode coopérative. Une solution possible sera l'utilisation d'autres récepteurs plus précis comme par exemple le centrale inertielle SPAN CPT que le laboratoire vient d'acheter. Nous avons aussi passé beaucoup de temps pour la synchronisation des données du bus CAN avec les données brutes

1. Non Line Of Sight

du récepteur UBLOX 4T. Cette synchronisation a été réalisée en post-traitement en se basant sur le temps GPS codé comme le nombre de secondes de la semaine (*second of week*). L'utilisation de nouveaux mécanismes de synchronisation dans chaque véhicule en temps réel serait plus précis et plus facile à mettre en œuvre et donnerait certainement de meilleurs résultats. À titre d'exemple, la synchronisation peut être effectuée par le signal *pulse per seconde* (i.e, PPS). C'est un signal électrique de déclenchement de forme carré. Ce signal est fourni par certains récepteurs et il est très précis vu qu'il est basé sur l'horloge du récepteur. La synchronisation devient alors très simple.

Développement de méthodes coopératives ensemblistes robustes

L'identification des mesures aberrantes et des multitrajets n'est pas robuste dans cette thèse. Nous avons supposé que le rejet des mesures aberrantes est fiable en se basant sur le rapport signal à bruit (i.e, SNR) des satellites alors qu'il existe des algorithmes ensemblistes robustes à ce problème tels que la relaxation des mesures (q -relaxée) et l'algorithme robuste d'inversion ensembliste (RSIVIA). L'utilisation de l'inversion ensembliste q -relaxée permet l'identification et l'exclusion des mesures aberrantes jusqu'à un degré q . On peut de plus calculer un domaine de confiance de position sous hypothèse de mesures aberrantes non-détectées.

Il est donc possible d'améliorer la robustesse de l'algorithme de localisation coopérative en présence de mesures aberrantes à travers l'inversion ensembliste q -relaxée. De cette façon, l'ensemble solution sera alors d'avantage « garanti » de contenir la vraie position si le nombre de mesures aberrantes est inférieur ou égal à q (i.e, le nombre de mesures aberrantes). Le nombre de mesures aberrantes tolérées peut être fixé empiriquement ou à partir des statistiques d'occurrence des défauts dans le cas de disponibilité de plusieurs mesures simultanées. Cette stratégie peut être basée sur le seuillage du rapport signal à bruit des mesures utilisé pour rejeter des mesures aberrantes. Cependant, une surestimation du nombre q par rapport au nombre réel de mesures aberrantes peut engendrer des imprécisions sur les domaines de confiances estimés.

Une autre solution possible est l'utilisation d'une stratégie d'estimation du nombre de mesures aberrantes telle que GOMNE (*Guaranteed Outlier Minimal Number Estimator*) [123]. Cette stratégie permettrait de limiter le nombre de contraintes à relâcher. La solution pourrait alors être calculée en relâchant $q_{min} + 1$ contraintes, où q_{min} est le nombre minimal de mesures aberrantes calculé par GOMNE.

Prise en compte de la cadence et de la latence de communication

Dans notre étude, nous avons supposé que la communication est idéale par souci de simplification et qu'elle est continue (i.e, à chaque instant les véhicules échangent

des informations) vu la courte durée des tests réalisés.

Pour rendre notre algorithme plus réaliste, il faudrait étaler les instants de communication. Dans le cas d'expériences à grande échelle, il serait possible de chercher quelle fréquence serait bien adaptée pour le partage des données (i.e, à partir de quelle fréquence l'effet de la coopération ne serait le plus significatif).

L'algorithme coopératif proposé ne tient pas compte de délais de communication entre les véhicules alors que ce problème est souvent présent dans les applications réelles. En effet, les transmissions des données en temps réel avec un délais garanti ne sont pas toujours possibles à cause de la portée du réseau de communication utilisé ou bien à cause de problèmes de surcharge.

Dans ce cas, les filtres peuvent garder en mémoire historiques des données pour chaque véhicule dans une fenêtre temporelle glissante. En cas de perte de communication, avec la sauvegarde d'un registre des données échangées dans des buffers, il serait possible de faire la mise à jour des positions relatives dans le passé en la propageant jusqu'à l'instant actuel. C'est une mise à jour sur un horizon qui exploite des mesures précédentes sur une fenêtre temporelle (pour limiter le calcul) en cas d'absence d'une mesure actuelle.

Intégration des mesures relatives

Dans ces travaux, nous avons estimé les positions relatives sans utiliser de mesures d'inter-distances ou d'angle relatif. La coopération serait plus pertinente si plusieurs véhicules peuvent coopérer et s'échanger des informations absolues et relatives, notamment de perception ou de détection. Dans ce cas, les performances de localisation peuvent être améliorées. Nous avons observé que le positionnement global GNSS seul peut parfois ne pas être assez précis ou fiable. En revanche, les capacités de perception des véhicules (LIDAR, caméra, radar) assistées par des cartes permettent d'estimer de manière plus précise les positions relatives des véhicules voisins. Notre problème de la localisation coopérative peut être reformulé en considérant des véhicules communicants qui peuvent potentiellement se percevoir mutuellement par l'ajout des modèles d'observation des distances relatives par exemple. On pourrait aussi considérer des informations issues de cartes. Ces informations pourraient, initialement, être modélisées comme des contraintes spatiales imprécises. Ensuite, on pourrait considérer le cas où ces données peuvent elles-mêmes être entachées d'incertitudes spatiales et temporelles (mesures biaisées, cartes non à jour, délais de communication, etc.).

Application de l'algorithme coopératif avec des robots hétérogènes

En utilisant des robots hétérogènes, on peut bénéficier de la complémentarité de véhicules évoluant dans différents milieux. Cette complémentarité est liée aux com-

portements différents, autonomies énergétiques différentes ou encore aux capteurs différents. Par exemple, si on utilise des drones avec des véhicules terrestres pour assurer une mission d'exploration d'une zone, on peut compenser la « myopie » des robots terrestres par l'exploitation des robots aériens pour percevoir la zone. Dans notre problème de localisation coopérative, on pourrait intégrer des mesures de pseudodistances faites par un drone afin de surmonter les problèmes de multitrajets. En effet, un drone peut naviguer à une hauteur assez élevée ce qui permettrait à un agent du groupe d'éviter d'avoir des trajets multiples des signaux de satellites. Cela pourrait améliorer considérablement la qualité du positionnement.

Utilisation d'autres modèles de biais

Nous avons utilisé un modèle Auto-Regressif (AR) d'ordre 1 pour prédire l'évolution des biais. Ce choix est basé sur des travaux de la littérature et une expérience présente au laboratoire. On n'a pas étudié l'intérêt d'autres modèles de biais sur le positionnement coopératif. Le choix du modèle peut s'avérer déterminant et engendrer des améliorations importantes en termes d'exactitude et d'incertitude. Par exemple, d'autres modèles comme ARMA (modèles auto-régressifs et moyenne mobile) peuvent être testés.

Annexes

Annexe A

Éléments d'analyse par intervalles

L'analyse par intervalles introduit les intervalles comme un nouveau type de nombres. Elle a été introduite pour quantifier l'erreur sur les résultats de calculs numériques [124]. Usuellement dans les calculateurs, les nombres réels sont présentés par des flottants dont chacun est constitué d'une base et une précision finie. Cette représentation n'est pas exacte mais c'est une approximation entraînant des erreurs. La propagation de ces erreurs lors des opérations arithmétiques peut être dramatique dans certains cas et provoque même des résultats totalement incorrects. De nombreux exemples de calcul avec virgule flottante avec les erreurs d'approximation résultantes sont décrits en détails dans [125] et [126]. Dans ce travail de thèse, l'analyse par intervalles sert principalement à manipuler les incertitudes de mesures.

A.1 Définition des intervalles

Un intervalle est un sous-ensemble connexe fermé de \mathbb{R} . On note \mathbb{IR} l'ensemble des intervalles réels. Les grandeurs intervalles seront notées entre crochets afin de faciliter la lecture.

L'intervalle $[x]$ est défini comme l'ensemble des réels x compris entre sa *borne inférieure* \underline{x} et sa *borne supérieure* \bar{x} :

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

La borne inférieure $lb([x])$ de l'intervalle $[x]$, notée aussi \underline{x} , est définie par

$$\underline{x} = lb([x]) \triangleq \sup \{a \in \mathbb{R} \mid \forall x \in [x] a \leq x\}$$

La borne supérieure $ub([x])$ de l'intervalle $[x]$, notée aussi \bar{x} , est définie par

$$\bar{x} = ub([x]) \triangleq \inf \{b \in \mathbb{R} \mid \forall x \in [x] x \leq b\}$$

Le *centre* d'un intervalle non vide et borné est donné par

$$\text{mid}([x]) = \frac{\underline{x} + \bar{x}}{2}.$$

La *largeur* d'un intervalle non vide est définie par

$$w([x]) = \bar{x} - \underline{x}.$$

On note $\{x\} = [x, x]$ un intervalle ponctuel ou dégénéré. Par convention, la largeur d'un intervalle vide est l'infini négatif, et la largeur d'un intervalle dégénéré est nulle :

$$w([\emptyset]) = -\infty, \quad w(\{x\}) = 0$$

A.2 Arithmétique sur les intervalles

Soit α un réel et $[x]$ un intervalle non vide, alors l'intervalle

$$\alpha[x] \triangleq \{\alpha x \mid x \in [x]\} \tag{A.1}$$

est donné par

$$\alpha[x] = [\alpha \underline{x}, \alpha \bar{x}] \text{ si } \alpha \geq 0 \tag{A.2}$$

$$= [\alpha \bar{x}, \alpha \underline{x}] \text{ si } \alpha < 0. \tag{A.3}$$

L'intersection de deux intervalles fermés non vides $[x]$ et $[y]$, définie par

$$[x] \cap [y] = \begin{cases} [\max\{\underline{x}, \underline{y}\}, \min\{\bar{x}, \bar{y}\}] & \text{si } \max\{\underline{x}, \underline{y}\} \leq \min\{\bar{x}, \bar{y}\}, \\ \emptyset & \text{sinon.} \end{cases}, \tag{A.4}$$

est toujours un intervalle. Ce n'est pas le cas pour leur union qui est définie par

$$[x] \cup [y] = \{z \in \mathbb{R} \mid z \in [x] \text{ ou } z \in [y]\}.$$

Ce sera un intervalle si l'ensemble obtenu est convexe (intuitivement s'il n'y a pas de « trou »). Dans le cas d'une union de deux intervalles, il suffit que l'intersection de ces intervalles soit non vide pour que leur réunion soit convexe.

Dans le cas où l'ensemble obtenu est non convexe, la notion *hull intervalle* permet de donner un intervalle. Le *hull intervalle* de symbole \sqcup d'un sous-ensemble est défini comme étant le plus petit intervalle le contenant

$$\forall [x] \in \mathbb{IR}, \forall [y] \in \mathbb{IR}, [x] \sqcup [y] = [\min\{\underline{x}, \underline{y}\}, \max\{\bar{x}, \bar{y}\}]. \quad (\text{A.5})$$

L'union et le hull des intervalles vérifient la propriété suivante

$$[x] \cup [y] \subseteq [x] \sqcup [y]$$

Considérons deux intervalles $[x]$ et $[y]$ et un opérateur $\diamond \in \{+, -, *, /\}$, on note $[x] \diamond [y]$ le plus petit intervalle contenant toutes les valeurs de $x \diamond y$ quand $x \in [x]$ et $y \in [y]$, soit l'intervalle suivant [98] :

$$[x] \diamond [y] = [\{x \diamond y \in \mathbb{R} \mid x \in [x], y \in [y]\}].$$

Dans le cas d'intervalles fermés, les calculs s'effectuent de la façon suivante :

$$[x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}], \quad (\text{A.6})$$

$$[x] - [y] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}], \quad (\text{A.7})$$

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

On notera indifféremment le produit de deux intervalles $[x] * [y]$ ou $[x][y]$. Le calcul de l'inverse est donné par

$$1/[y] = \begin{cases} \emptyset & \text{si } [y] = [0, 0], \\ [1/\bar{y}, 1/\underline{y}] & \text{si } 0 \notin [y], \\ [1/\bar{y}, \infty[& \text{si } \underline{y} = 0 \text{ et } \bar{y} > 0, \\] - \infty, 1/\bar{y}] & \text{si } \underline{y} < 0 \text{ et } \bar{y} = 0, \\] - \infty, \infty[& \text{si } \underline{y} < 0 \text{ et } \bar{y} > 0, \end{cases} \quad (\text{A.9})$$

et la division par

$$[x]/[y] = [x] * (1/[y]). \quad (\text{A.10})$$

Ces règles se simplifient dans le cas d'intervalles ponctuels, et on retrouve alors les règles de l'arithmétique réelle. On peut ainsi considérer l'arithmétique intervalle comme une extension de l'arithmétique réelle.

A.3 Vecteurs et Matrices d'intervalles

On peut généraliser la notion d'intervalles à tout ensemble admettant une structure de treillis. Ainsi, outre les intervalles de réels, on peut considérer des intervalles de vecteurs, de booléens, de fonctions ou encore d'ensembles [63].

A.3.1 Vecteurs intervalles

Un intervalle vectoriel, aussi appelé pavé ou boîte $[\mathbf{x}]$ est un sous-ensemble de \mathbb{R}^n défini par le produit cartésien de n intervalles réels fermés, $n \in \mathbb{N}$

$$[\mathbf{x}] = [\underline{x}_1, \bar{x}_1] \times \cdots \times [\underline{x}_n, \bar{x}_n] = [x_1] \times \cdots \times [x_n].$$

C'est un vecteur dont les composantes sont des intervalles, noté

$$[\mathbf{x}] = ([x_1], \dots, [x_n])^T,$$

et la $i^{\text{ème}}$ composante intervalle $[x_i]$ est la projection de $[\mathbf{x}]$ sur le $i^{\text{ème}}$ axe. Ainsi, l'ensemble vide de \mathbb{R}^n doit être noté $\emptyset \times \cdots \times \emptyset$ car toutes ses composantes intervalles sont vides.

On note $\mathbb{I}\mathbb{R}^n$ l'ensemble des pavés de \mathbb{R}^n . Une boîte non vide peut être représentée par un parallélotope aligné sur les n axes (Figure A.1).

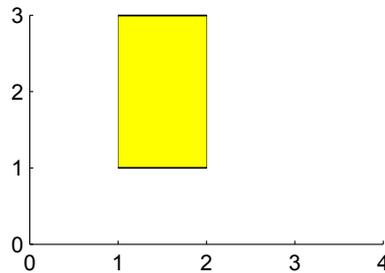


Figure A.1 – Représentation d'une boîte $[1, 2] \times [1, 3]$

La plupart des notions introduites pour les intervalles s'étendent facilement aux pavés. Ainsi, la borne inférieure et la borne supérieure d'un pavé sont définies par

$$\underline{\mathbf{x}} = (\underline{x}_1, \dots, \underline{x}_n)^T,$$

$$\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_n)^T.$$

La largeur d'un pavé est la largeur de son côté le plus long :

$$w([\mathbf{x}]) = \max_{1 \leq i \leq n} w([x_i]).$$

Les opérations classiques appliquées sur les vecteurs réels s'étendent facilement aux boîtes comme le produit cartésien, l'addition, etc [98].

A.3.2 Matrices intervalles

Les notions et les opérations appliquées sur les intervalles sont aussi étendues aux matrices, où chaque composante est un intervalle.

Les calculs classiques d'algèbre linéaire peuvent ainsi être « intervalisés », ce qui permet par exemple de tenir compte de l'erreur de linéarisation en définissant une matrice incertaine. L'utilisation de matrices intervalles permet également de tenir compte de la précision de la représentation numérique en machine et des calculs numériques, pour résoudre des problèmes mal conditionnés.

L'algèbre linéaire implique également des opérations plus sophistiquées telles que l'inversion de matrice et le calcul des valeurs et vecteurs propres [126].

A.4 Fonctions d'inclusion

L'image $f([x])$ d'un intervalle par une fonction f est

$$f([x]) = \{f(x) \mid x \in [x]\}.$$

Cette image n'est pas nécessairement un intervalle. En effet, si f n'est pas continue, $f([x])$ est une union d'intervalles.

L'extension intervalle d'une fonction est définie comme la fonction renvoyant l'enveloppe intervalle de $f([x])$:

$$[f]([x]) = [\{f(x) \mid x \in [x]\}].$$

L'extension intervalle des fonctions élémentaires peut être exprimée directement en fonction des bornes. Par exemple, pour un intervalle non vide $[x]$, la fonction exponentielle s'exprime

$$[\exp]([x]) = [\exp \underline{x}, \exp \bar{x}]. \quad (\text{A.11})$$

Dans le cas de fonctions non monotones, la situation est plus compliquée. Par exemple, $[\sin]([0, \pi]) = [0, 1]$ diffère de l'intervalle $[\sin(0), \sin(\pi)] = [0, 0]$. Des algorithmes spécifiques doivent donc être construits pour l'évaluation intervalle de ces fonctions (voir par exemple [98]).

L'image d'une boîte $[x]$ par une fonction f n'est pas forcément une boîte, et elle est en général un ensemble de forme quelconque de représentation difficile à manipuler. Les

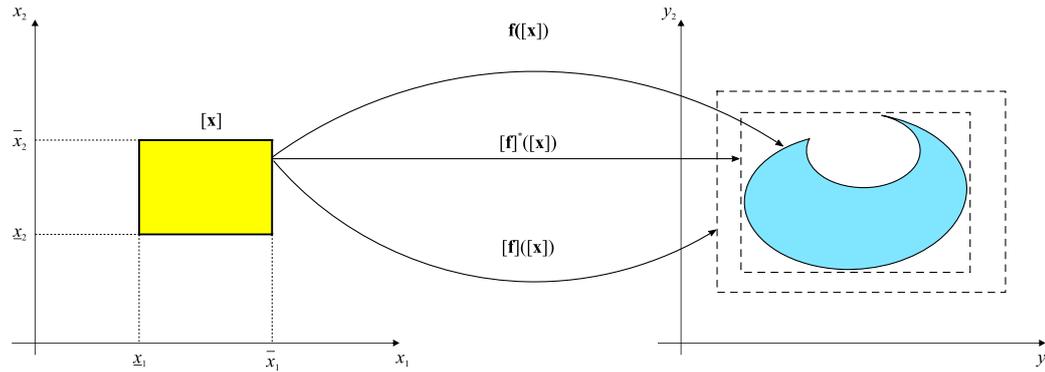


Figure A.2 – Images d'une boîte par des fonction.

fonctions intervalles, nommées aussi fonctions d'inclusion, proposent une solution en présentant les formes inconnues de $[f]([x])$ par une boîte englobante (Figure A.2).

La fonction intervalle $[f] : \mathbb{R}^n \rightarrow \mathbb{R}^m$ est une *fonction d'inclusion* de la fonction $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ si et seulement si

$$\forall [x] \in \mathbb{R}^n, f([x]) \subset [f]([x]).$$

Quelque soit la forme de l'ensemble image $f([x])$, une fonction d'inclusion $[f]$ de f permet de calculer un pavé $[f]([x])$ garanti de contenir l'image de $[x]$ par f .

A.4.1 Propriétés spécifiques de fonctions d'inclusion

Une fonction d'inclusion est dite *fine* si l'image d'un intervalle vectoriel ponctuel $\{x\}$ est ponctuelle, c'est-à-dire $[f]({x}) = \{f(x)\}$.

Une fonction d'inclusion $[f]$ est dite *convergente* si, pour toute suite de boîtes $[x](k)$,

$$\lim_{k \rightarrow \infty} w([x](k)) = 0 \Rightarrow \lim_{k \rightarrow \infty} w([f]([x](k))) = 0.$$

Cette propriété de convergence est en général requise pour la convergence des algorithmes basés sur l'analyse par intervalles.

On dit de $[f]$ qu'elle est *monotone pour l'inclusion* si

$$[x] \subset [y] \Rightarrow [f]([x]) \subset [f]([y]).$$

Une fonction d'inclusion $[f]$ est dite *minimale* si, pour tout $[x]$, $[f]([x])$ est la plus petite boîte contenant $f([x])$. La fonction d'inclusion minimale de f est unique et notée $[f]^*$ (Figure A.2). Cependant, une fonction d'inclusion dont l'image d'un intervalle n'est pas égale au plus petit intervalle contenant l'évaluation intervalle de la fonction réelle sur cet intervalle est dite *pessimiste*.

Soient m fonctions d'inclusion $[f_j] : \mathbb{IR}^n \rightarrow \mathbb{IR}$, $j = 1, \dots, m$ associées aux fonctions coordonnées f_j d'une fonction vectorielle $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Une fonction d'inclusion pour \mathbf{f} est donnée par

$$[\mathbf{f}]([\mathbf{x}]) = [f_1]([\mathbf{x}]) \times \cdots \times [f_m]([\mathbf{x}]).$$

$[\mathbf{f}]$ est convergente (respectivement fine, monotone pour l'inclusion, minimale) si toutes ses fonctions coordonnées $[f_i]$ sont convergentes (respectivement fines, monotones pour l'inclusion, minimales). La construction de fonctions d'inclusion efficaces d'une fonction vectorielle \mathbf{f} peut ainsi être ramenée à la construction de fonctions d'inclusion efficaces pour ses fonctions coordonnées. Nous allons donc nous intéresser plus particulièrement à l'élaboration de fonctions d'inclusion pour des fonctions à valeurs réelles de la forme $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ où n est la dimension de vecteur d'état et m est la dimension de vecteur de mesures.

A.4.2 Fonctions d'inclusion naturelles

Une fonction terme f est une fonction construite par une composition finie des fonctions élémentaires (telles que \sin , \cos , \exp , $\sqrt{\cdot}$, sqr , \max , \dots) et des opérateurs ($+$, $-$, $*$, $/$). Pour obtenir une fonction d'inclusion d'une fonction terme f il suffit de remplacer les variables scalaires (x, y, \dots) dans l'expression de f par leur contrepartie intervalle $([x], [y], \dots)$, et toutes les fonctions élémentaires réelles par leur extension intervalle. La fonction $[f]_n$ ainsi obtenue est la *fonction d'inclusion naturelle* de f . Elle est monotone pour l'inclusion et fine. De plus, si l'expression de f ne contient que des opérateurs et fonctions élémentaires continus, alors la fonction d'inclusion naturelle est convergente.

En général, $[f]_n$ n'est pas minimale, à cause de la dépendance entre les variables. Par contre, $[f]_n$ devient minimale dans le cas où chaque variable n'apparaît qu'une seule fois dans l'expression de f et que tous les fonctions élémentaires et les opérateurs utilisés sont continus.

Dans l'exemple de localisation à partir de trois balises, la fonction d'observation pour chaque robot est

$$\mathbf{g}_i : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \tag{A.12}$$

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} \mapsto \begin{pmatrix} \sqrt{(x_i - x^1)^2 + (y_i - y^1)^2} \\ \sqrt{(x_i - x^2)^2 + (y_i - y^2)^2} \\ \sqrt{(x_i - x^3)^2 + (y_i - y^3)^2} \end{pmatrix}.$$

La fonction d'inclusion naturelle $[\mathbf{g}_i]_n$ de \mathbf{g}_i est ainsi donnée par

$$[\mathbf{g}_i]_n : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \\ \begin{pmatrix} [x_i] \\ [y_i] \end{pmatrix} \mapsto \begin{pmatrix} \sqrt{([x_i] - x^1)^2 + ([y_i] - y^1)^2} \\ \sqrt{([x_i] - x^2)^2 + ([y_i] - y^2)^2} \\ \sqrt{([x_i] - x^3)^2 + ([y_i] - y^3)^2} \end{pmatrix}.$$

$[\mathbf{g}_i]_n$ est une fonction d'inclusion minimale de \mathbf{g}_i car on n'utilise que des opérateurs et fonctions continus et chaque variable n'apparaît qu'une seule fois dans chaque expression.

Annexe B

Éléments de filtrage Bayésien

B.1 Le filtrage optimal

Le problème du filtrage consiste à estimer l'état des variables d'un système dynamique (variables d'état) sujet à des perturbations et des observations partielles. La modélisation de l'évolution dans le temps des états $\mathbf{x}(t \geq 0)$ du système considéré s'écrit alors sous la forme d'une partie d'évolution déterministe et d'une partie stochastique :

$$\mathbf{x}_{t \geq 0} = \mathbf{x}_0 + \int_0^t f(\mathbf{x}_t, \boldsymbol{\alpha}_t, t) dt \quad (\text{B.1})$$

où \mathbf{x}_0 est l'état initial des variables d'état, f est une fonction déterministe décrivant la dynamique et $\boldsymbol{\alpha}(t)$ est un processus de Wiener standard modélisant les perturbations aléatoires de la dynamique du système. Le processus x_t est alors un processus Markovien. Dans le but de déterminer l'état d'un système discret, on est amené à construire une équation d'observation qui relie à des instants k l'observation \mathbf{y}_k à l'état courant \mathbf{x}_k sachant que les mesures sont entachées d'erreurs dues à l'imperfection des capteurs.

$$\mathbf{y}_k = h_k(\mathbf{x}_k) + \boldsymbol{\beta}_k \quad (\text{B.2})$$

où \mathbf{y}_k est une fonction d'observation connue et $\boldsymbol{\beta}_k$ un vecteur de variables aléatoires de statistique connue modélisant l'imperfection des observations. On cherche à déterminer l'état du système \mathbf{x}_k à partir des observations $(\mathbf{y}_1, \dots, \mathbf{y}_n)$. Selon la valeur de n on a un problème de :

- Lissage si $n > k$,
- Filtrage si $n = k$,
- Prédiction si $n < k$

Les hypothèses classiques du filtrage sont les suivantes :

- Les observations (\mathbf{y}_k) sont mutuellement indépendantes conditionnellement à $\mathbf{x}_{k \geq 0}$.
- le processus d'état $\mathbf{x}_{k \geq 1}$ est Markovien et le bruit de mesure \mathbf{v}_k est un ensemble de variables aléatoires indépendantes du processus \mathbf{x}_k .
- \mathbf{x}_0 l'état initial, présente une distribution de probabilité supposée connue et indépendante de β_k et de α_k .

Afin d'obtenir un filtrage optimal, on cherche classiquement à minimiser la variance de l'erreur de filtrage. Ainsi, si $\hat{\mathbf{x}}_k$ désigne un estimateur de l'état \mathbf{x}_k , celui-ci sera optimal s'il minimise l'erreur moyenne quadratique :

$$E[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T] \quad (\text{B.3})$$

Partant de l'hypothèse que les seules informations disponibles sur le système sont les mesures $(\mathbf{y}_1, \dots, \mathbf{y}_k)$, alors l'estimateur de variance minimale non biaisé est donné par :

$$\hat{\mathbf{x}}_k = E(\mathbf{x}_k | \mathbf{y}_1, \dots, \mathbf{y}_k) \quad (\text{B.4})$$

Dans le cas où la loi conditionnelle $p(\mathbf{x}_k | \mathbf{y}_1, \dots, \mathbf{y}_k)$ est multimodale, il est souhaitable de prendre l'estimateur du maximum a posteriori

$$\hat{\mathbf{x}}_k = \text{Arg Max } \mathbf{x}_k [p(\mathbf{x}_k | \mathbf{y}_1, \dots, \mathbf{y}_k)] \quad (\text{B.5})$$

Le filtrage optimal consiste à calculer la densité conditionnelle $p_{k|k}$ de l'état \mathbf{x}_k , connaissant les observations $\mathbf{y}_{1:k} = (\mathbf{y}_1, \dots, \mathbf{y}_k)$ jusqu'à l'instant courant k :

$$p_{k|k} = p(\mathbf{x}_k | \mathbf{y}_{1:k}) \quad (\text{B.6})$$

Dans la pratique, la taille du vecteur d'observation $\mathbf{y}_{1:k}$ augmentant à chaque instant k , il est souhaitable que l'on parvienne à contenir l'information ainsi accumulée dans un vecteur dont la dimension ne croisse pas de même. Il est alors intéressant d'établir une expression itérative de la densité conditionnelle $p_{k|k}$. Ainsi $p_{k|k}$ ne doit être exprimée qu'en fonction de la dernière observation \mathbf{y}_k et de la loi conditionnelle $p_{k-1|k-1}$ de l'instant précédent. Le passage de $p_{k-1|k-1}$ à $p_{k|k}$ fait intervenir la loi conditionnelle de $p_{k|k-1}$ de l'état $p_{k|k}$ sachant les observations $\mathbf{y}_{1:k-1} = (\mathbf{y}_1, \dots, \mathbf{y}_{k-1})$

$$p_{k|k-1} = p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) \quad (\text{B.7})$$

La suite des lois de probabilité $p_{k|k-1}$ est appelée filtre de prédiction et se décompose en deux étapes essentielles :

- Prédiction : l'étape de prédiction du filtre utilise la connaissance a priori de l'état du système et de son noyau de transition $p(\mathbf{x}_k, \mathbf{x}_{k-1})$ (défini par le modèle d'évolution du système), en réalisant la transition de $p_{k-1|k-1}$ à $p_{k|k-1}$ de la manière suivante,

$$p_{k|k-1} = \int_{\mathfrak{R}_d} p(x_k, x_{k-1}) p_{k-1|k-1} dx_{k-1} \quad (\text{B.8})$$

où \mathfrak{R}_d représente l'espace d'état.

- Correction : l'étape de correction utilise l'observation \mathbf{y}_k à travers la densité conditionnelle $p(\mathbf{y}_k | \mathbf{x}_k)$ (encore appelée densité de vraisemblance), et permet ainsi de corriger la densité prédite $p_{k|k-1}$, par application de la formule de Bayes.

$$p_{k|k} = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})} \quad (\text{B.9})$$

$$p_{k|k} = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p_{k-1|k-1}}{\int_{\mathfrak{R}_d} p(\mathbf{y}_k | \mathbf{x}_k) p_{k-1|k-1} dx_k} \quad (\text{B.10})$$

L'application de ces étapes donne la solution théorique du filtrage Bayésien optimal. Son calcul effectif exige le calcul d'intégrales multidimensionnelles. Malheureusement, dès que l'équation d'état et/ou l'équation d'observation est non-linéaire, ces intégrales ne se calculent pas analytiquement. On a alors recours à des approximations soit par des méthodes analytiques (par exemple le filtre de Kalman étendu qui considère le modèle d'état linéarisé), soit par des techniques numériques (le filtrage particulaire).

B.2 Le filtre de Kalman

Ce filtre a été développé par Kalman [127] pour le cas discret puis repris par Kalman et Bucy [128] pour le cas continu. Le modèle d'état est supposé linéaire, avec un bruit additif Gaussien (Eq. B.11). L'observation est une fonction linéaire de l'état entachée d'un bruit centré additif Gaussien (Eq. B.12).

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \boldsymbol{\alpha}_{k-1} \quad (\text{B.11})$$

où les vecteurs \mathbf{x}_k , \mathbf{x}_{k-1} sont les vecteurs d'état à l'instant k et $k-1$, \mathbf{u}_{k-1} est le vecteur de commande du système à l'instant $k-1$, $\boldsymbol{\alpha}_{k-1}$ est un vecteur de bruit additif

centré sur l'état. Les matrices \mathbf{A} et \mathbf{B} sont des matrices déterministes représentant la dynamique du système et l'interaction du vecteur de commande.

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \beta_k \quad (\text{B.12})$$

où \mathbf{y}_k est la mesure à l'instant k , \mathbf{C} est la matrice d'observation du système et β_k est le bruit de mesure.

De plus, les vecteurs de variables aléatoires α_{k-1} et β_k sont supposés mutuellement indépendants de l'état initial \mathbf{x}_0 du système, blancs et suivant une distribution de probabilité normale¹ centrée :

$$\begin{aligned} p(\alpha_{k-1}) &\sim N(0, \mathbf{Q}) \\ p(\beta_k) &\sim N(0, \mathbf{R}) \end{aligned} \quad (\text{B.13})$$

où les matrices \mathbf{Q} et \mathbf{R} sont les matrices de covariance associées respectivement au processus et aux mesures. Enfin, l'état initial \mathbf{x}_0 du filtre doit suivre la loi normale $N(\bar{\mathbf{x}}_0, P_0)$ avec :

$$\begin{aligned} \bar{\mathbf{x}}_0 &= E[\mathbf{x}_0] \\ \mathbf{P}_0 &= E[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T] \end{aligned} \quad (\text{B.14})$$

Vu la linéarité des équations d'état et de mesure et du caractère Gaussien de l'état et des observations, la loi de probabilité a posteriori gouvernant l'état du système est la loi normale $N(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$ qui peut être entièrement décrite par ses deux premiers moments. En tenant compte des équations de prédiction (B.8) et de correction (B.10), le filtre de Kalman peut se résumer aux trois étapes suivantes :

$$\begin{cases} p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) = N(\mathbf{x}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \\ p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = N(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ p(\mathbf{x}_k | \mathbf{y}_{1:k}) = N(\mathbf{x}_{k|k}, \mathbf{P}_{k|k}) \end{cases} \quad (\text{B.15})$$

où les deux premiers moments peuvent être à chaque instant donnés par les équations suivantes :

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_{k-1} \quad (\text{B.16})$$

1. L'hypothèse Gaussienne n'est pas obligatoire. Dans ce cas, l'estimation d'état se fait à l'ordre 2.

$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q} \quad (\text{B.17})$$

et

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}) \quad (\text{B.18})$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_{k|k-1} \quad (\text{B.19})$$

avec

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + \mathbf{R})^{-1} \quad (\text{B.20})$$

En général, on appelle l'innovation l'écart entre la mesure et sa prédiction. Elle est donnée par $\tilde{\mathbf{y}}_{k|k} = \mathbf{y}_{k|k} - \mathbf{y}_{k|k-1}$ en sachant que $\mathbf{y}_{k|k-1} = \mathbf{C}_k\hat{\mathbf{x}}_{k|k-1}$. Cet écart pondéré par le gain de Kalman \mathbf{K}_k représente la quantité d'information supplémentaire à apporter par la correction lors d'une nouvelle mesure. Le gain de Kalman joue un rôle très important dans l'étape de correction. Il permet de connaître la confiance que l'on peut apporter à la mesure par rapport à la prédiction. Il devient d'autant plus important que la covariance \mathbf{R} du bruit de mesure tend vers une matrice nulle ce qui privilégie la mesure devant la prédiction. À l'inverse, si la covariance $\mathbf{P}_{k|k-1}$ prédite tend vers 0 et donc lorsque la matrice \mathbf{Q} tend elle-même vers 0, le gain de Kalman tend aussi vers 0, alors le filtre privilégie la prédiction devant la mesure. Si l'hypothèse de bruits Gaussiens n'est plus vérifiée (en considérant le système dynamique toujours linéaire), alors on montre que le filtre de Kalman-Bucy fournit une estimation à variance minimale de l'état du système si l'on dispose de leurs deux premiers moments.

B.3 Le filtre de Kalman étendu EKF

Le filtre de Kalman étendu s'inscrit dans les problèmes de filtrage optimal dont le processus et/ou les fonctions de mesures sont linéaires. Malheureusement dans de nombreuses situations, ces contraintes ne sont pas vérifiées. Dans [129] l'auteur démontre que le filtre de Kalman étendu permet d'estimer l'état d'un système dont le processus et/ou les fonctions de mesures sont non linéaires en linéarisant les équations du filtre de Kalman autour d'un point de fonctionnement à chaque instant.

Pour réaliser ce changement nous devons dans un premier temps modifier les équations (B.11) et (B.12) de la manière suivante :

$$\begin{cases} \mathbf{x}_k = f_k(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\gamma}_{k-1}) + \boldsymbol{\alpha}_{k-1} \\ \mathbf{y}_k = g_k(\mathbf{x}_k) + \boldsymbol{\beta}_k \end{cases} \quad (\text{B.21})$$

où g_k et f_k sont des fonctions non linéaires. On considère aussi parfois que les entrées sont mesurées pour certains systèmes comme dans ce cas, le vecteur de commande \mathbf{u}_{k-1} est entaché d'un bruit blanc centré supposé gaussien de matrice de covariance \mathbf{N} (i.e. $\text{cov}(\boldsymbol{\gamma}_{k-1}) = \mathbf{N}$). Ce bruit est également indépendant des bruits de mesure et d'état ainsi que l'état initial du système.

Le filtre de Kalman étendu repose sur l'approximation Gaussienne de la loi a posteriori gouvernant l'état. Ainsi les équations (B.15) deviennent :

$$\begin{cases} p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) \simeq N(\mathbf{x}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \\ p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) \simeq N(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}) \\ p(\mathbf{x}_k | \mathbf{y}_{1:k}) \simeq N(\mathbf{x}_{k|k}, \mathbf{P}_{k|k}) \end{cases} \quad (\text{B.22})$$

En faisant l'approximation de Taylor au premier ordre des deux premiers moments, l'équation d'état peut être linéarisée autour de l'estimée $\hat{\mathbf{x}}_{k-1|k-1}$ et l'entrée mesurée \mathbf{u}_{k-1} pour devenir :

$$\mathbf{x}_k = f_k(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}, \boldsymbol{\gamma}_{k-1}) + \mathbf{A}_{k-1}[\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}] + \mathbf{B}_{k-1}\boldsymbol{\gamma}_{k-1} + \boldsymbol{\alpha}_k \quad (\text{B.23})$$

$$\mathbf{A}_{k-1} = \left. \frac{\partial f_k(\mathbf{x}, \mathbf{u}_{k-1}, \boldsymbol{\gamma}_{k-1})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}} \quad (\text{B.24})$$

$$\mathbf{B}_{k-1} = \left. \frac{\partial f_k(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}, \boldsymbol{\gamma})}{\partial \mathbf{u}} \right|_{\mathbf{u}=\mathbf{u}_{k-1}} \quad (\text{B.25})$$

De la même manière, l'équation d'observation peut être linéarisée autour de $\hat{\mathbf{x}}_{k|k-1}$ pour donner :

$$\mathbf{y}_k = g_k(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{C}_k[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}] + \boldsymbol{\beta}_k \quad (\text{B.26})$$

où

$$\mathbf{C}_k = \left. \frac{\partial g_k(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}} \quad (\text{B.27})$$

En appliquant le filtre de Kalman à ces équations linéarisées on obtient alors les équations du filtre de Kalman étendu :

$$\hat{\mathbf{x}}_{k|k-1} = f_k(\mathbf{x}_{k-1|k-1}, \mathbf{u}_{k-1}, \gamma_{k-1}) \quad (\text{B.28})$$

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{A}_{k-1}^T + \mathbf{B}_{k-1}^T \mathbf{N} \mathbf{B}_{k-1}^T + \mathbf{Q} \quad (\text{B.29})$$

et

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - g_k(\hat{\mathbf{x}}_{k|k-1})) \quad (\text{B.30})$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_{k|k-1} \quad (\text{B.31})$$

où

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R})^{-1} \quad (\text{B.32})$$

Il y a deux sources d'inexactitude qui peuvent influencer l'estimation du filtre de Kalman étendu. La première est la linéarisation des modèles d'état et d'observation à chaque instant k . La seconde est de faire l'hypothèse que les estimations a priori et a posteriori de l'état restent toujours Gaussiennes. En effet, malgré le théorème de la limite centrale, les non-linéarités dans les modèles conduisent souvent les distributions de probabilité a priori et a posteriori vers des distributions de probabilité non Gaussiennes. Ainsi les deux premiers moments estimés par le filtre ne sont plus que des approximations de la réalité. Intuitivement, plus la linéarisation est fine (développement de Taylor à des ordres supérieurs), plus le filtre doit donner de meilleurs résultats. D'ailleurs, l'exactitude de la linéarisation dépend des quantités exprimées dans l' Eq (B.33) :

$$\begin{aligned} & E[(\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_{k|k-1})(\hat{\mathbf{x}}_{k|k-1} - \mathbf{x}_{k|k-1})^T] \\ & E[(\hat{\mathbf{x}}_{k-1|k-1} - \mathbf{x}_{k-1|k-1})(\hat{\mathbf{x}}_{k-1|k-1} - \mathbf{x}_{k-1|k-1})^T] \end{aligned} \quad (\text{B.33})$$

ainsi que du degré de non-linéarité des équations d'état et d'observation f et g . Plus ces quantités sont petites plus l'approximation est meilleure. Une approximation de ces quantités peut être fournie par les traces de $\mathbf{P}_{k-1|k-1}$ et de $\mathbf{P}_{k|k-1}$. Dans la pratique, si le système comporte des non-linéarités trop importantes, le système linéarisé est trop éloigné du système réel, par conséquent le filtre de Kalman, accordant trop de confiance au système linéarisé, a tendance à diverger. Pour remédier à ce problème plusieurs approches ont été développées comme l'UKF [130], les filtres de Différences Divisés d'ordre 1 et 2 (DD1, DD2) [131, 132] et les techniques de Multi-Modèles Interactifs (IMM) [133]. Des approches reposant sur des techniques de Monté Carlo appelées filtres particulières ont aussi été développées.

Bibliographie

- [1] Official U.S. Government information about the Global Positioning System and related topics. Space segment.
- [2] Vincent Drevelle. *Study of robust set estimation methods for a high integrity multi-sensor localization. Application to navigation in urban areas.* Theses, Université de Technologie de Compiègne, December 2011.
- [3] R. Beckers, O.E. Holland., and J.L. Deneubourg. From local actions to global tasks : Stigmergy and collective robotics biblio. In *International Workshop on the Synthesis and Simulation of Living Systems.*, pages 181–189, Cambridge, USA, 1994.
- [4] Arab Ali Cherif. Collective behavior for a micro-colony of robots. In *European Conference on Artificial Life (ECAL)*, Brussels, Belgium, 1993.
- [5] Arab Ali Cherif. A constitution and an economic model for the organisation and emergence of collective behaviour in a colony of robots. In *From Perception to Action conference*, pages 334–337, Lausanne, Switzerland, Sept 1994.
- [6] G. Dudek, M. R. M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4) :375–397, 1996.
- [7] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics : Antecedents and directions. *Autonomous Robots*, 4(1) :7–27, 1997.
- [8] T. Arai, E. Pagello, and L. E. Parker. Editorial : Advances in multi-robot systems. *IEEE Transactions on Robotics and Automation*, 18(5) :655–661, 2002.
- [9] A. Farinelli, L. Iocchi, and D. Nardi. Multirobot systems : a classification focused on coordination. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5) :2015–2028, Oct 2004.
- [10] L. E. Parker. *Multi-Robot Systems From Swarms to Intelligent Automata*, chapter The effect of heterogeneity in teams of 100+ mobile robots, pages 205–215. Kluwer 2003, 2003.
- [11] Barnes D. P. and Gray J. O. Behaviour synthesis for cooperant mobile robot control. In *International Conference on Control*, pages 1135–1140, Edinburgh, UK, 1991.
- [12] Parker L. E. *Heterogeneous Multi-Robot Cooperation. PhD thesis, Massachusetts Institute of Technology.* PhD thesis, 1994.

- [13] F. Arrichiello. *Coordination Control of Multiple Mobile Robots*. PhD thesis, University degli Studi di Cassino, 2006.
- [14] B. P. Gerkey and M. J. Mataric. Sold! : auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5) :758–768, Oct 2002.
- [15] R. T. Vaughan, K. Stoy, G. S. Sukhatme, and M. J. Mataric. Lost : localization-space trails for robot teams. *IEEE Transactions on Robotics and Automation*, 18(5) :796–812, Oct 2002.
- [16] R. Zlot, A. Stentz, M. B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 3016–3023, 2002.
- [17] J. L. Baxter, E. K. Burke, J. M. Garibaldi, and Mark Norman. Shared potential fields and their place in a multi-robot coordination taxonomy. *Robotics and Autonomous Systems*, 57(10) :1048–1055, Oct 2009.
- [18] T. S. Dahl, M. J. Mataric, and G. S. Sukhatme. Multi-robot task allocation through vacancy chain scheduling. *Robotics and Autonomous Systems*, 57(6-7) :674–687, June 2009.
- [19] J. Wawerla and R. T. Vaughan. A fast and frugal method for team-task allocation in a multi-robot transportation system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1432–1437, May 2010.
- [20] C. Secchi, A. Franchi, H. H. Bulthoff, and P. R. Giordano. Bilateral teleoperation of a group of uavs with communication delays and switching topology. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4307–4314, May 2012.
- [21] L. E. Parker. Alliance : an architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In *IEEE/RSJ/GI International Conference on the Intelligent Robots and Systems (IROS). 'Advanced Robotic Systems and the Real World'*, volume 2, pages 776–783 vol.2, Sep 1994.
- [22] H. Asama, A. Matsumoto, and Y. Ishida. Design of an autonomous and distributed robot system : Actress. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS)*, pages 283–290, Sept 1989.
- [23] R. Madhavan, K. Fregene, and L. E. Parker. Distributed heterogeneous outdoor multi-robot localization. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 374–381 vol.1, 2002.
- [24] B. P. Gerkey and M. J. Mataric. Murdoch : Publish/subscribe task allocation for heterogeneous agents. In *International conference on Autonomous agents*, pages 203–204, Barcelona, Spain, June 2000.
- [25] S. I. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5) :781–795, Oct 2002.
- [26] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2797–2802, April 2005.

- [27] H. Kruppa D. Fox, W. Burgard and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3) :325–344, 2000.
- [28] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 476–481, 2000.
- [29] I. Rekleitis, V. Lee-Shue, Ai Peng New, and H. Choset. Limited communication, multi-robot team based coverage. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3462–3468 Vol.4, April 2004.
- [30] I.M. Rekleitis, G. Dudek, and E.E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1340–1345, 1997.
- [31] L. Wu, D. P. M. A. Garcia, and A. Sole. Voronoi-based space partitioning for coordinated multi-robot exploration. *Journal of Physical Agents*, 1(1) :37–44, Sept 2007.
- [32] A. Cowley, C. J. Taylor, and B. Southall. Rapid multi-robot exploration with topometric maps. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1044–1049, May 2011.
- [33] J. S. Jennings, G. Whelan, and W. F. Evans. Cooperative search and rescue with a team of mobile robots. In *International Conference on Advanced Robotics (ICAR)*, pages 193–200, Jul 1997.
- [34] R. Zlot and A. Stentz. Complex task allocation for multiple robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1515–1522, April 2005.
- [35] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang, and V. A. Ziparo. Towards heterogeneous robot teams for disaster mitigation : Results and performance metrics from robocup rescue. *Journal of Field Robotics*, 24(11-12) :943–967, Nov 2007.
- [36] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert. Multi-robot cooperation in the martha project. *IEEE Robotics Automation Magazine*, 5(1) :36–47, Mar 1998.
- [37] Z. Yan, N. Jouandeau, and A. Ali-Cherif. Multi-robot heuristic goods transportation. In *IEEE International Conference on Intelligent Systems*, pages 409–414, Sept 2012.
- [38] J. Wawerla and R. T. Vaughan. A fast and frugal method for team-task allocation in a multi-robot transportation system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1432–1437, May 2010.
- [39] I. J. Cox. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2) :193–204, Apr 1991.

- [40] H. R. Everett. *Sensors for Mobile Robots*. AK Peters, 1995.
- [41] J. Borenstein and Liqiang Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, 12(6) :869–880, Dec 1996.
- [42] J. Borenstein and L. Feng. Gyrodometry : a new method for combining data from gyros and odometry in mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 423–428 vol.1, Apr 1996.
- [43] C.F. Olson and L.H. Matthies. Maximum likelihood rover localization by matching range maps. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 272–277, May 1998.
- [44] S.I. Roumeliotis and G. A. Bekey. Bayesian estimation and kalman filtering : a unified framework for mobile robot localization. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2985–2992 vol.3, 2000.
- [45] M.S. Fontan and M.J. Mataric. Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 14(5) :815–822, Oct 1998.
- [46] L.E. Parker. Alliance : an architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2) :220–240, Apr 1998.
- [47] R. Kurazume, S. Nagata, and S. Hirose. Cooperative positioning with multiple robots. In *IEEE International Conference on Robotics and Automation*, pages 1250–1257 vol.2, May 1994.
- [48] Nadir Karam. *Agrégation de données décentralisées pour la localisation multi-véhicules*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2009.
- [49] S. I. Roumeliotis and G. A. Bekey. Collective localization : a distributed kalman filter approach to localization of groups of mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2958–2965 vol.3, 2000.
- [50] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2797–2802, April 2005.
- [51] S. I. Roumeliotis and I. M. Rekleitis. Analysis of multirobot localization uncertainty propagation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 1763–1770 vol.2, Oct 2003.
- [52] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1402–1409, May 2009.
- [53] A. Howard, M. J. Matark, and G. S. Sukhatme. Localization for mobile robot teams using maximum likelihood estimation. In *IEEE/RSJ International*

- Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 434–439 vol.1, 2002.
- [54] L. Montesano, J. Gaspar, J. Santos-Victor, and L. Montano. Cooperative localization by fusing vision-based bearing measurements and motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2333–2338, Aug 2005.
- [55] Y. Lin, P. Vernaza, J. Ham, and D. D. Lee. Cooperative relative robot localization with audible acoustic sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3764–3769, Aug 2005.
- [56] S. Worrall and E. Nebot. Using non-parametric filters and sparse observations to localise a fleet of mining vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 509–516, April 2007.
- [57] H. Mu, T. Bailey, P. Thompson, and H. Durrant-Whyte. Decentralised solutions to the cooperative multi-platform navigation problem. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2) :1433–1449, April 2011.
- [58] A. Howard, M. J. Mataric, and G. S. Sukhatme. Putting the 'i' in 'team' : an ego-centric approach to cooperative localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 868–874 vol.1, Sept 2003.
- [59] N. Karam, F. Chausse, R. Aufreere, and R. Chapuis. Localization of a group of communicating vehicles by state exchange. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 519–524, Oct 2006.
- [60] M. Kieffer. *Estimation ensembliste par analyse par intervalle : Application à la localisation de véhicules*. PhD thesis, Université de Paris Sud, 1999.
- [61] E. Seigneur, M. Kieffer, A. Lambert, E. Walter, and T. Maurin. Experimental vehicle localization by bounded-error state estimation using interval analysis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1084–1089, Aug 2005.
- [62] L. Jaulin. A nonlinear set membership approach for the localization and map building of underwater robots. *IEEE Trans. on Robotics*, 25(1) :88–98, Feb 2009.
- [63] L. Jaulin. Range-only SLAM with occupancy maps; a set-membership approach. *IEEE T-RO*, 27(5) :1004–1010, 2011.
- [64] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, pages 1053–1064, 1993.
- [65] V. Drevelle and Ph. Bonnifait. Localization Confidence Domains via Set Inversion on Short-Term Trajectory. *IEEE Trans on Robotics*, 29(5) :1244–1256, 2013.
- [66] D. Meizel, O. Leveque, L. Jaulin, and E. Walter. Initial localization by set inversion. *IEEE Trans on Robotics and Automation*, 18(6) :966–971, 2002.

- [67] L. Jaulin. Interval constraint propagation with application to bounded-error estimation. *Automatica*, 36(10) :1547–1552, 2000.
- [68] V. Drevelle, L. Jaulin, and B. Zerr. Guaranteed characterization of the explored space of a mobile robot by using subpavings. In *IFAC Symposium on Nonlinear Control Systems*, pages 44–49, Sept 2013.
- [69] A. Bethencourt. *Interval Analysis for swarm localization. Application to underwater robotics*. PhD thesis, University of Bretagne, Sept 2014.
- [70] Kyoung-Hwan Jo and Jihong Lee. Cooperative localization of multiple robots with constraint propagation technique. In *IEEE Int Conf on Intelligent Robots and Systems*, pages 3477–3482, Sept 2008.
- [71] T. W. Barrett. Technical features, history of ultra wideband communications and radar : part i, uwb communications. *Journal of Microwave*, 44(1) :22–56, 2001.
- [72] Bouna Fall. *Evaluation des performances d’un système de localisation de véhicules de transports guidées fondée sur l’association d’une technique radio ULB et d’une technique de retournement temporel*. PhD thesis, Université de VALENCIENNES ET DU HAINAUT-CAMBRESIS, 2013.
- [73] FCC. First report and order, revision of part 15 of the commission’s rules regarding ultra-wideband transmission systems. Technical Report 98-153, FCC, avril 2002.
- [74] J. D. Taylor. *Introduction to Ultra-Wideband Radar Systems*. CRC Press, 1995.
- [75] R. Scholtz. Multiple access with time-hopping impulse modulation. In *IEEE Military Communications Conference*, volume 2, pages 447–450 vol.2, Oct 1993.
- [76] P. Pagani, F. Tchoffo Talom, and P. Pajusco. *Communication ultra large bande : le canal de propagation radioélectrique*. Collection technique et scientifique des télécommunications. Hermès science publications / Lavoisier, Paris, 2007.
- [77] R. J. M. Cramer, R. A. Scholtz, and M. Z. Win. Evaluation of an ultra-wideband propagation channel. *IEEE Transactions on Antennas and Propagation*, 50(5) :561–570, May 2002.
- [78] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6) :1067–1080, Nov 2007.
- [79] A. Ledergerber, M. Hamer, and R. D’Andrea. A robot self-localization system using one-way ultra-wideband communication. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3131–3137, Sept 2015.
- [80] Y. Gao, X. Meng, C. M. Hancock, and S. Stephenson. Uwb/gnss-based cooperative positioning method for v2x applications. In *International Technical*

- Meeting of The Satellite Division of the Institute of Navigation*, Tampa, Florida, 2014.
- [81] *Global Positioning System Standard Positioning Service Performance Standard*. Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, 2001.
- [82] A. Leick. *GPS satellite surveying, Third edition*. Wiley, 2004.
- [83] E. D. Kaplan and C. Hegarty. *Understanding GPS : Principles and Applications*. Second Edition, Artech House, 2005.
- [84] F. Duquenne, S. Botton, F. Peyret, D. Bétaille, and P. Willis. *GPS : localisation et navigation par satellites*. HermÈs - Lavoisier, 2005.
- [85] Stéphane Durand. *Amélioration de la précision de la localisation différentielle temps réel par mesure de phase des systÈmes GNSS*. PhD thesis, IGN / LCPC Nantes - Observatoire de Paris, 2003.
- [86] Brian Tolman, R. Benjamin Harris, Tom Gaussiran, David Munton, Jon Little, Richard Mach, Scot Nelsen, and Brent Renfro. The GPS Toolkit : Open Source GPS Software. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, California, September 2004.
- [87] P. S. Maybeck. *Stochastic models, estimation and control. Volume I*. 1979.
- [88] G. Antonelli, F. Arrichiello, S. Chiaverini, and G. Sukhatme. Observability analysis of relative localization for auvs based on ranging and depth measurements. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4276–4281, May 2010.
- [89] X. Xiao J. Rife. Estimation of spatially correlated errors in vehicular collaborative navigation with shared gnss and road-boundary measurements. In *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation*, pages 1667 – 1677, September 21 - 24 2010.
- [90] J. Rife. Collaborative vision-integrated pseudorange error removal : Team-estimated differential gnss corrections with no stationary reference receiver. *IEEE Transactions on Intelligent Transportation Systems*, 13(1) :15–24, March 2012.
- [91] R. Hermann and Arthur J. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5) :728–740, Oct 1977.
- [92] G.L. Mariottini, G. Pappas, D. Prattichizzo, and K. Daniilidis. Vision-based localization of leader-follower formations. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 635–640, Dec 2005.
- [93] A. Martinelli and R. Siegwart. Observability analysis for mobile robot localization. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1471–1476, Aug 2005.

- [94] X.S. Zhou and S.I. Roumeliotis. Robot-to-robot relative pose estimation from range measurements. *IEEE Transactions on Robotics*, 24(6) :1379–1393, Dec 2008.
- [95] J.L. Casti. *Recent Developments and Future Perspectives in Nonlinear System Theory*. Research report (International Institute for Applied Systems Analysis). International Institute for Applied Systems Analysis, 1982.
- [96] R. Christof and M. Marcel. Indoor location tracking in non-line-of-sight environments using a IEEE 802.15.4a wireless network. In *Conference Proceedings of IROS09. St. Louis*, pages 552–557, 2009.
- [97] David L. Waltz. *Generating Semantic Description from Drawings of Scenes with Shadows*. PhD thesis, Massachusetts Institute of Technology, 1972.
- [98] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.
- [99] G. Chabert and L. Jaulin. Contractor Programming. *Artificial Intelligence*, 173 :1079–1100, 2009.
- [100] F. Benhamou, F. Goualard, L. Granvilliers, and J.F. Puget. Revising hull and box consistency. In *International Conference on Logic Programming*, pages 230–244. MIT press, 1999.
- [101] L. Jaulin. *Le calcul ensembliste par analyse par intervalles et ses applications*. PhD thesis, Habilitation à diriger des recherches, 2000.
- [102] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4) :1053–1064, 1993.
- [103] Peter Corke. *Robotics, Vision and Control : Fundamental Algorithms in MATLAB*. 73. Springer-Verlag Berlin Heidelberg, 2011.
- [104] R. Aufrere, N. Karam, F. Chausse, and R. Chapuis. A state exchange approach in real conditions for multi-robot cooperative localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4346–4351, Oct 2010.
- [105] G. Chabert. IBEX(<http://www.ibex-lib.org>), 2007.
- [106] Zui Tao. *Autonomous Road Vehicles Localization Using Satellites, Lane marking and Vision*. PhD thesis, University of Technology of Compiègne, 2016.
- [107] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied interval analysis with examples in parameter and state estimation, robust control and robotics*. Springer, 2001.
- [108] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. pages 182–193, 1997.
- [109] Eric A. Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. pages 153–158, 2000.
- [110] C. Y. Chong. Distributed fusion architectures and algorithms. In *International Conf. on Multisource-Multisensor Data Fusion*, 1998.

- [111] Y. Zhu and X. R. Li. Best linear unbiased estimation fusion. In *Int. Conf. On Information Fusion*, Sunnyvale, CA, USA, 1999.
- [112] X. R. Li and K. Zhang. Optimality and efficiency of optimal distributed fusion. In *International Conf. On Information Fusion*, Montreal, Canada, 2001.
- [113] C. Y. Chong, S. Mori, and K. C. Chang. Information fusion in distributed sensor networks. In *American Control Conf*, pages 830–835, 1985.
- [114] Julier S and J. Uhlmann. *General Decentralized Data Fusion with Covariance Intersection (CI)*. Handbook of Multisensor Data Fusion, 2001.
- [115] CY. Chong and S. Mori. Convex combination and covariance intersection algorithms in distributed fusion. In *4th International Conference on Information Fusion*, Canada, 2001.
- [116] Y. Bar-Shalom and L. Campo. The effect of the common process noise on the two-sensor fused-track covariance. *IEEE Transactions on Aerospace and Electronic Systems (AES)*, 22(6) :803–805, 1986.
- [117] U. W. Utete. *Network Management in Decentralized Sensing Systems*. PhD thesis, Department of Engineering Science, University of Oxford, 1995.
- [118] Olivier Le Marchand. *Approche autonome pour la localisation et la surveillance de l'intégrité d'un véhicule automobile en environnement complexe*. PhD thesis, Université de Technologie de Compiègne, 2010.
- [119] A. Gning and P. Bonnifait. Constraint propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, pages 1167–1175, July 2006.
- [120] G. D. Hager, S. P. Engelson, and S. Atiya. On comparing statistical and set-based methods in sensor data fusion. In *IEEE International Conference on Robotics and Automation*, pages 352–358 vol.2, May 1993.
- [121] Mauro Di Marco. *Set-membership estimation techniques for mobile robotics applications*. PhD thesis, Bologna University, Partner Institutions : Firenze, Padova, Siena, 2001.
- [122] E. Zamora and W. Yu. Ellipsoid method for simultaneous localization and mapping of mobile robot. In *IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 5334–5339, Dec 2014.
- [123] Luc Jaulin, Éric Walter, and Olivier Didrit. Guaranteed robust nonlinear parameter bounding. In *CESA'96 IMACS Multiconference (Symposium on Modelling, Analysis and Simulation)*, volume 2, pages 1156–1161, Lille, 1996.
- [124] R.E. Moore. *Interval analysis*. Prentice Hall, 1966.
- [125] D. A. Patterson and J. L. Hennessy. *computer Organization and Design : The Hardware/Software Interface, Third Edition*. The Morgan Kaufmann Series in Computer Architecture and Design. Elsevier Science, 2005.
- [126] Arnold Neumaier. *Introduction to numerical analysis*. Cambridge University Press, 2001.

Bibliographie

- [127] R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 1960.
- [128] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME, Ser. D, J. Basic Eng*, page 109, 1961.
- [129] P.S. Maybeck. *Stochastics models, estimation and control*. Academic Press, New York, USA, 1979.
- [130] S. J. Julier and J. K. Uhlmann. Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. volume 2, pages 887–892, 2002.
- [131] D. T. Magill. Optimal adaptive estimation of sampled stochastic processes. *CONTROL*, AC-10 :434–439, 1965.
- [132] E. Mazor, A. Averbuch, Y. Bar Shalom, and J. Dayan. Interacting multiple model methods in target tracking : A survey. *IEEE Trans. in Aerospace and Electronic Systems*, 34(1) :103–123.
- [133] M. Nørgaard, N. K. Poulsen, and O. Ravn. Advances in derivative-free state estimation for nonlinear systems. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 1998. Revised edition, Oct 2004.

Liste des symboles

AUV	Autonomous Underwater Vehicle
BDS	BeiDou navigation Satellite system
CDS	Confidence Domain Size
CSP	Constraint Satisfaction Problem, Problème de Satisfaction de Contraintes
CSP	Constraint Satisfaction Problem
DoD	Departement of Defense
DR	Dead Reckoning
DR	Dead Reckoning
ESA	European Space Agency
Galileo	Europe's own global navigation satellite system
GLONASS	Globalanaya Navigatsionnaya Sputnikovaya Sistema
GNSS	Global Navigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HPE	Horizontal Positioning Errors
MAS	Multi-Agent Systems
MCS	Master control station
MMRS	Multiple Mobile Robot Systems
MRS	Multi-Robot Systems
NAVSTAR	NAVigation System by Timing And Ranging
NLOS	Non Line Of Sight
SA	Selective Availability

Bibliographie

SIVIA	Set Inversion Via Interval Analysis
SIVIA	Set Inversion via Interval Analysis
UWB	Ultra Wide Band
V2V	Vehicle to Vehicle
WGS 84	World Geodetic System 1984

Résumé

On étudie dans cette thèse des méthodes de localisation coopérative de robots mobiles sans utilisation de mesures extéroceptives relatives, comme des angles ou des distances entre robots. Les systèmes de localisation considérés sont basés sur des mesures de radionavigation sur des balises fixes ou des satellites. Pour ces systèmes, on observe en général un écart entre la position observée et la position réelle. Cet écart systématique (appelé biais) peut être dû à une mauvaise position de la balise ou à une différence entre la propagation réelles des ondes électromagnétiques par rapport aux conditions standard utilisées pour établir les modèles d'observation. L'influence de ce biais sur la localisation des robots est non négligeable. La coopération et l'échange de données entre les robots (estimations des biais, estimations des positions et données proprioceptives) est une approche qui permet de corriger ces erreurs systématiques.

La localisation coopérative par échange des estimations est sujette aux problèmes de consanguinité des données qui peuvent engendrer des résultats erronés, en particulier trop confiants. Lorsque les estimations sont utilisées pour la navigation autonome à l'approche, on doit éviter tout risque de collision qui peut mettre en jeu la sécurité des robots et des personnes aux alentours. On doit donc avoir recours à un mécanisme d'intégrité vérifiant que l'erreur commise reste inférieure à une erreur maximale tolérable pour la mission. Dans un tel contexte, il est nécessaire de caractériser des domaines de confiance fiables contenant les positions des robots mobiles avec une forte probabilité. L'utilisation des méthodes ensemblistes à erreurs bornées est considérée alors comme une solution efficace. En effet, ce type d'approche résout naturellement le problème de consanguinité des données et fournit des domaines de confiance fiables. De surcroît, l'utilisation de modèles non-linéaires ne pose aucun problème de linéarisation.

Après avoir modélisé un système coopératif de n_r robots avec des mesures biaisées sur des balises, une étude d'observabilité est conduite. Deux cas sont considérés selon la nature des mesures brutes des observations. En outre, des conditions d'observabilité sont démontrées. Un algorithme ensembliste de localisation coopérative est ensuite présenté. Les méthodes considérées sont basées sur la propagation de contraintes sur des intervalles et l'inversion ensembliste. La coopération est effectuée grâce au partage des positions estimées, des biais estimés et des mesures proprioceptives. L'échange des estimations de biais permet de réduire les incertitudes sur les positions des robots.

Dans un cadre d'étude simple, la faisabilité de l'algorithme est évaluée grâce à des simulations de mesures de distances sur balises en utilisant plusieurs robots. La

Bibliographie

coopération est comparée aux méthodes non coopératives. L'algorithme coopératif ensembliste est ensuite testé sur des données réelles en utilisant deux véhicules. Les performances de la méthode ensembliste coopérative sont enfin comparées avec deux méthodes Bayésiennes séquentielles, notamment une avec fusion par intersection de covariance. La comparaison est conduite en termes d'exactitude et d'incertitude.

Summary

In this work, we study some cooperative localization issues for mobile robotic systems that interact with each other without using relative measurements (e.g. bearing and relative distances). The considered localization technologies are based on beacons or satellites that provide radio-navigation measurements. Such systems often lead to offsets between real and observed positions. These systematic offsets (i.e, biases) are often due to inaccurate beacon positions, or differences between the real electromagnetic waves propagation and the observation models. The impact of these biases on robots localization should not be neglected. Cooperation and data exchange (estimates of biases, estimates of positions and proprioceptive measurements) reduce significantly systematic errors. However, cooperative localization based on sharing estimates is subject to data incest problems (i.e, reuse of identical information in the fusion process) that often lead to over-convergence problems.

When position information is used in a safety-critical context (e.g. close navigation of autonomous robots), one should check the consistency of the localization estimates. In this context, we aim at characterizing reliable confidence domains that contain robots positions with high reliability. Hence, set-membership methods are considered as efficient solutions. This kind of approach enables merging adequately the information even when it is reused several time. It also provides reliable domains. Moreover, the use of non-linear models does not require any linearization.

The modeling of a cooperative system of n_r robots with biased beacons measurements is firstly presented. Then, we perform an observability study. Two cases regarding the localization technology are considered. Observability conditions are identified and demonstrated. We then propose a set-membership method for cooperative localization.

Cooperation is performed by sharing estimated positions, estimated biases and proprioceptive measurements. Sharing biases estimates allows to reduce the estimation error and the uncertainty of the robots positions.

The algorithm feasibility is validated through simulation when the observations are beacons distance measurements with several robots. The cooperation provides better performance compared to a non-cooperative method. Afterwards, the cooperative algorithm based on set-membership method is tested using real data with two experimental vehicles. Finally, we compare the interval method performance with a sequential Bayesian approach based on covariance intersection. Experimental results indicate that the interval approach provides more accurate positions of the vehicles

Bibliographie

with smaller confidence domains that remain reliable. Indeed, the comparison is performed in terms of accuracy and uncertainty.

