



HAL
open science

Localisation coopérative robuste de robots mobiles par mesure d'inter-distance

Cyrille Pierre

► **To cite this version:**

Cyrille Pierre. Localisation coopérative robuste de robots mobiles par mesure d'inter-distance. Automatique / Robotique. Université Clermont Auvergne [2017-2020], 2020. Français. NNT : 2020CLFAC045 . tel-03158978

HAL Id: tel-03158978

<https://theses.hal.science/tel-03158978>

Submitted on 4 Mar 2021

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.

UNIVERSITÉ CLERMONT-AUVERGNE
École doctorale des Sciences Pour l'Ingénieur

Thèse présentée par
Cyrille PIERRE

en vue de l'obtention du grade de
Docteur d'université
Spécialité : Électronique et Systèmes

**Localisation coopérative robuste de robots mobiles
par mesure d'inter-distance**

Thèse soutenue le 16/12/2020.

Travaux dirigés par **Christophe DEBAIN**
et co-encadrés par **Romuald AUFRÈRE** et **Roland CHAPUIS**

Composition du jury :

Ouidad LABBANI-IGBIDA	Professeur des universités	Université de Limoge	Rapporteur
Fawzi NASHASHIBI	Directeur de Recherches	INRIA Paris	Rapporteur
Romuald AUFRÈRE	Maitre de Conférences	Université Clermont Auvergne	Examineur
Roland CHAPUIS	Professeur des universités	Université Clermont Auvergne	Examineur
Michel DEVY	Directeur de recherche	LAAS-CNRS	Examineur
Christophe DEBAIN	Chargé de Recherches	INRAE Clermont-Ferrand	Directeur de thèse

Résumé

Il existe de plus en plus d'applications en robotique mobile faisant intervenir plusieurs robots capables de communiquer entre eux et naviguer en coopération. L'objectif de ces travaux est d'exploiter la communication et la détection des robots avoisinants afin de réaliser de la localisation coopérative. Le moyen de détection des robots utilisé ici se base sur la technologie *ultra-wideband*, permettant d'effectuer des mesures précises de distance entre deux capteurs. L'approche que nous avons développée privilégie la robustesse et l'intégrité des estimations. Elle permet de prendre en compte des scénarios dans lesquels le peu d'information disponible rend la tâche de localisation difficile à gérer pour les robots. Ainsi, notre solution gère deux problématiques importantes de la localisation coopérative par mesure de distance : la corrélation des données échangées entre les robots et la non-linéarité du modèle d'observation des mesures.

Pour résoudre ces problématiques, nous avons fait le choix de développer une approche décentralisée dans laquelle l'aspect coopératif est pris en compte par une modélisation particulière des observations de robots. Une observation correspond ici à une mesure de distance avec une balise (représentant un robot ou un objet statique) dont la position est représentée par une distribution normale. Après plusieurs observations successives d'une même balise, la corrélation devient forte entre l'état du robot et la position de la balise observée. Notre approche s'appuie alors sur la méthode de fusion d'un filtre à intersection de covariance partitionné afin d'éviter les problèmes de sur-convergence dus à la corrélation. De plus, les estimations des états des robots sont modélisées par des mixtures de gaussiennes permettant ainsi de représenter au mieux les formes réelles des distributions obtenues après la fusion d'une mesure de distance. Notre algorithme de localisation coopérative a également la particularité de faire évoluer le nombre de particules des mixtures de gaussiennes afin de se ramener à une modélisation équivalente à un filtre gaussien lorsque les conditions le permettent.

Le fonctionnement de notre approche de localisation coopérative est présenté grâce à l'étude de quelques situations basiques mettant en évidence différentes caractéristiques de l'algorithme. Le manuscrit se termine par la présentation de trois scénarios de localisation coopérative faisant intervenir plusieurs robots et objets statiques. Les deux premiers exploitent un simulateur réaliste capable de simuler la physique des robots. Le troisième est une expérimentation réelle utilisant une plateforme urbaine dont dispose le laboratoire. L'objectif de ces expérimentations est de montrer que notre approche reste intègre dans les situations les plus difficiles.

Mots clés. localisation coopérative, robots mobiles, mesure de distance, ultra-wideband, filtre à intersection de covariance, mixture de gaussiennes

Abstract

There is an increasing number of applications in mobile robotics involving several robots able to communicate with each other to navigate cooperatively. The aim of this work is to exploit the communication and the detection of robots in order to achieve cooperative localization. The perception tool used here rely on ultra-wideband technology, which allows to perform precise range measurements between two sensors. The approach we have developed focuses on the robustness and consistency of robot state estimation. It enables to take into account scenarios where the localization task is difficult to handle due to limited data available. In that respect, our solution of cooperative localization by range measurements addresses two important problematics: the correlation of data exchanged between robots and the non-linearity of the observation model.

To solve these issues, we have choosed to develop a decentralized approach in which the cooperative aspect is taken into account by a specific robot observation model. In this context, an observation corresponds to a range measurement with a beacon (that is, a robot or a static object) where the position is represented by a normal distribution. After several observations of the same beacon, the correlation between the robot state and the beacon position increases. Our approach is based on the fusion method of the Split Covariance Intersection Filter in order to avoid the problem of over-convergence induced by data correlation. In addition, the robot state estimates are modeled by Gaussian mixtures allowing best representation of the distributions obtained after merging a range measurement. Our localization algorithm is also able to dynamically adjust the number of Gaussians of mixture models and can be reduced to a simple Gaussian filter when conditions are favorable.

Our cooperative localization approach is studied using basic situations, highlighting important characteristics of the algorithm. The manuscript ends with the presentation of three scenarios of cooperative localization implying several robots and static objects. The first two take advantage of a realistic simulator able to simulate the physics of robots. The third is a real world experimentation using a platform for urban experimentation with vehicles. The aim of these scenarios is to show that our approach stay consistent in difficult situations.

Keywords. cooperative localization, mobile robots, range measurement, ultra-wideband, covariance intersection filter, Gaussian mixture model

Remerciements

La réalisation de mon travail de thèse ne s'est pas fait seul mais grâce au soutien d'une équipe de chercheurs et d'ingénieurs. C'est pourquoi je souhaite mettre en avant l'ensemble des personnes qui ont été impliquées pendant ces quelques années.

Je remercie mon encadrant Romuald AUFRÈRE, sans qui, je n'aurais sans doute pas fait cette thèse. Je l'ai d'abord côtoyé en tant qu'élève à l'ISIMA pendant mon cursus d'étudiant ingénieur en informatique mais également en tant qu'étudiant au master de robotique. Il m'a guidé dans la plupart de mes choix et m'a apporté beaucoup d'aide sur les aspects scientifiques, mais également sur tous les aspects qui tourne autour de la vie d'un chercheur à l'Institut Pascal. Je remercie mon encadrant Roland CHAPUIS, pour son important apport scientifique à mes travaux. Mes premiers échanges avec lui ont commencé pendant mes cours de master en robotique puis se sont accentués pendant mes travaux de thèse. Le chemin que j'ai suivi durant ces dernières années a été tracé par les discussions et parfois débats que j'ai eu le plaisir d'avoir avec lui. Je remercie mon directeur de thèse Christophe DEBAIN d'avoir accepté de me prendre pour doctorant et de me guider sur la réalisation de ce travail. La plupart de mes actions en tant que chercheur ont été guidé par ses précieux conseils. Je le remercie aussi pour avoir toujours été là pour m'aider sur les points que je maîtrise le moins.

Je remercie aussi toute l'équipe d'ingénieurs de l'Institut Pascal pour l'aide qu'ils m'ont apporté pour la réalisation des expérimentations. Je remercie Laurent MALATERRE pour le développement de beaucoup de modules ROS que j'ai utilisé dans mes travaux. Je le remercie également d'avoir toujours été disponible pour m'aider sur la réalisation d'expérimentations. Je remercie Ted MORELL, avec qui, j'ai passé beaucoup de temps sur Pavin pour mettre au point une expérimentation importante mais qui n'a malheureusement jamais été présentée. Je remercie Serge ALIZON pour les informations et les compétences qu'il m'a apporté concernant l'utilisation des outils et véhicules de l'institut Pascal.

Je souhaite également remercier mes collègues de bureau Abderrahim KASMI, Johann LACONTE, Marwa LADHARI, Jérémy TAQUET et Elie RANDRIAMIARINTSOA pour leur soutien. Il m'arrivait occasionnellement de prendre un peu de leur temps pour discuter de certains points de mes travaux et ils m'ont toujours apportés des réponses intéressantes et de nouvelles idées. J'appré-

cie également tous les bons moments que j'ai pu avoir avec eux en dehors de mes travaux de recherche.

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du programme Investissements d'Avenir dans le cadre du Laboratoire d'Excellence IMobS3 (ANR-10-LABX-0016), de l'Initiative d'Excellence IDEX-ISITE CAP 20-25 (ANR-16-IDEX-0001) et de l'Équipement d'Excellence RobotEx (ANR-10-EQPX-0044).

Mes derniers remerciements vont à ma famille pour m'avoir soutenu tout au long de ma thèse.

Table des matières

1	Introduction	15
1.1	Généralités sur la robotique mobile	15
1.2	Collaboration en robotique mobile	18
1.3	Problématique	19
1.4	Contribution de la thèse	21
1.5	Présentation du manuscrit	21
2	Outils fondamentaux pour l'estimation d'états	23
2.1	Introduction	23
2.2	Estimation bayésienne	24
2.3	Filtres gaussiens	26
2.3.1	Filtre de Kalman	26
2.3.2	Filtre de Kalman étendu	27
2.3.3	Filtre de Kalman sans parfum	28
2.3.4	Filtre à intersection de covariance	31
2.3.5	Filtre à intersection de covariance partitionné	33
2.4	Filtre particulaire	35
2.5	Mixture de gaussiennes	36
2.5.1	Filtre à somme de gaussiennes	37
2.5.2	Filtre particulaire à somme de gaussiennes	38
2.6	Méthodes ensemblistes	40
2.6.1	Modélisation des estimations	41
2.6.2	Analyse par intervalles	42
2.6.3	Estimation d'états par méthode ensembliste	44
2.7	Outils de qualification des estimations	46
2.7.1	Représentation 2D d'une densité de probabilité Gaussienne	46
2.7.2	Réduction de mixture standard	47
2.7.3	Union de covariance	48
2.7.4	Union de covariance généralisée	49
2.7.5	Distance de Mahalanobis	51

2.7.6	Divergence de Bhattacharyya	52
2.7.7	Divergence de Kullback-Leibler	53
2.7.8	Vocabulaire de l'estimation	54
3	État de l'art de la localisation coopérative	57
3.1	Introduction	57
3.2	Capteurs	57
3.2.1	GNSS	58
3.2.2	Caméra	59
3.2.3	Lidar	60
3.2.4	Radar	60
3.2.5	UWB	62
3.3	Présentation des approches de localisation coopérative	63
3.3.1	Approches centralisées	63
3.3.2	Approche décentralisée	65
3.4	Gestion de la corrélation des données	67
3.4.1	Corrélation des états des robots	68
3.4.2	Illustration du problème	69
3.4.3	Solutions existantes	73
3.5	Gestion de la non-linéarité des observations	77
3.5.1	Problème des mesures de distance	77
3.5.2	Solutions au problème	79
3.6	Bilan	82
4	Approche de localisation coopérative proposée	85
4.1	Introduction	85
4.2	Modèle utilisé	86
4.2.1	Modélisation de la flotte de robots	87
4.2.2	Modèle d'observation des mesures de distance	89
4.3	Processus de fusion coopératif	92
4.3.1	Présentation du filtre	92
4.3.2	Les étapes de l'algorithme	93
4.4	Calcul de l'évolution	95
4.4.1	Partitionnement à l'étape d'évolution	96
4.4.2	Étape d'évolution d'une particule	100
4.5	Calcul de la mise à jour	101
4.5.1	Test de linéarité	101
4.5.2	Mise à jour d'une particule Gaussienne	103

4.5.3	Partitionnement à l'étape d'observation	105
4.5.4	Réduction de la mixture de gaussiennes	108
5	Expérimentations et résultats	111
5.1	Introduction	111
5.2	Illustration des cas canoniques	111
5.2.1	Corrélation des informations échangées entre les robots	112
5.2.2	Prise en compte de la non-linéarité du processus de fusion basé sur des mesures de distance	114
5.3	Outils pour les expérimentations	118
5.3.1	Le simulateur 4D-Virtualiz	118
5.3.2	Véhicule	119
5.3.3	Modèle d'observation par GNSS	121
5.3.4	Les balises UWB	122
5.3.5	Gestion temporelle du filtre	122
5.4	Simulations réalistes et expérimentations réelles	124
5.4.1	Localisation dans un environnement connu	124
5.4.2	Groupe de robots mobiles	127
5.4.3	Localisation de balises statiques (expérimentation réelle)	130
5.5	Bilan	134
6	Conclusion et perspectives	135
6.1	Conclusion	135
6.2	Contributions	137
6.3	Perspectives	138

Table des figures

1.1	Robots mobiles manipulateurs (Robotnik et Omron)	16
1.2	Exemples d'applications de la robotique mobile.	17
1.3	Illustration de scénarios impliqués dans les projets COVEL et MuCCA	19
1.4	Situations pouvant faire intervenir une collaboration de robots mobiles	20
2.1	Réseau bayésien : interaction entre les estimations d'états et les mesures	25
2.2	Comparaison entre la fusion des algorithmes CC et CI	32
2.3	La fusion CI pour différentes valeurs de ω	32
2.4	Exemple de pavé $[x]$ recouvrant un ensemble non convexe \mathbb{X} de dimension 2	42
2.5	Recouvrement d'un anneau par 3 différentes résolutions de pavage	42
2.6	Arbre dont les nœuds représentent des opérations élémentaires	43
2.7	Représentation par une ellipse d'une loi normale de dimension 2	46
2.8	Ellipses représentant une loi normale à 1 sigma et 99 %	47
2.9	Fusion de deux gaussiennes par la SMR	48
2.10	Fusion de deux gaussiennes en utilisant l'union de covariance.	49
2.11	Mesures de distance de Mahalanobis avec une loi Normale de dimension 2	52
2.12	Exemple d'utilisation du vocabulaire pour qualifier des estimations	55
3.1	Géolocalisation d'un récepteur : principe de triangulation	59
3.2	Exemple de caméra de la marque Baumer.	60
3.3	Le lidar Sick LMS150 et un exemple de mesure obtenue	61
3.4	Exemple d'observation obtenue à partir d'un radar 360°	61
3.5	Le radar Pelican développé par l'Inrae.	62
3.6	Le capteur UWB développé par l'Inrae et l'Institut Pascal.	62
3.7	Communication entre les robots et la machine dédiée au calcul de la localisation centralisée	63
3.8	Communication entre les robots à la portée les uns des autres, utilisée dans la localisation décentralisée	66
3.9	Simulation de deux robots mobiles (odométrie uniquement)	70
3.10	Trajectoires obtenues avec l'EKF centralisé	71

3.11	Trajectoires obtenues avec l'EKF décentralisé	73
3.12	Les différents cas d'estimation de la position d'un robot après la prise en compte de mesures de distance.	78
3.13	Illustration du problème de linéarisation par mesure de distance	79
3.14	Approximation des estimations d'état avec paramétrisation en polaire	80
3.15	Évolution d'une mixture de gaussiennes estimant l'état d'une balise statique . .	82
4.1	Exemples de situation de localisation coopérative par mesures de distance . . .	88
4.2	Représentation du modèle géométrique utilisé pour la mesure de distance . . .	90
4.3	Densités de probabilité d'une mesure distance robot / balise (espace 2D)	92
4.4	Étapes impliquées dans le calcul d'une nouvelle estimation d'état	94
4.5	Représentation de la non-linéarité du modèle d'évolution	97
4.6	Partitionnement du support d'une gaussienne pour la création de particules . . .	98
4.7	Approximation d'une gaussienne par une mixture de 5 particules	99
4.8	Évolution d'un robot sur un court trajet et particules générées	100
4.9	Représentation géométrique du critère de partitionnement	102
4.10	Génération des particules pour représenter la mesure de distance	107
4.11	Exemple d'estimation obtenu après partitionnement de la mesure de distance .	108
5.1	Rappel des simulations de localisation coopérative de deux robots	113
5.2	Trajectoires de la simulation testant la corrélation	113
5.3	Simulation de la localisation d'un robot par mesure de distance (1 balise) . . .	115
5.4	Simulation d'un robot bien orienté mais avec une position inconnue	117
5.5	Exemples d'images obtenues dans le simulateur de 4D-Virtualiz	119
5.6	Robots utilisés pour les différentes expérimentations	120
5.7	Représentation d'un véhicule de type Ackermann.	121
5.8	Capteur de distance UWB développé par l'institut Pascal et l'INRAE	122
5.9	Liste d'observation avec recalage temporel des mesures	123
5.10	Représentation de l'initialisation de la simulation	125
5.11	Résultats obtenus pour la localisation des robots 2 et 3 (sur 50 simulations) . .	127
5.12	Aperçu de l'environnement et des robots simulés ainsi que leurs trajectoires . .	128
5.13	Résultat obtenu pour la localisation des robots 2 et 3 (sur 50 simulations) . . .	130
5.14	Aperçu de l'environnement d'expérimentation et du placement initial des éléments dans la carte	131
5.15	Résultat obtenu pour l'expérimentation réelle avec 2 robots et 5 balises statiques	133

Chapitre 1

Introduction

1.1 Généralités sur la robotique mobile

La robotique mobile s'inscrit dans la quête du progrès que mène l'homme depuis ses origines et en particulier dans les évolutions récentes qui ont suivi la révolution industrielle. En proposant toujours plus de mécanisation puis d'automatisation, elles ont permis d'accroître la productivité tout en diminuant la pénibilité des tâches de production. C'est particulièrement vrai pour le déplacement des objets ou des personnes au sein d'un même bâtiment, sur un site industriel tout entier ou plus généralement sur l'espace public. Depuis le tapis roulant purement mécanique jusqu'au robot intelligent capable de sélectionner l'objet et sa destination pour le déplacer de manière adéquate la complexité n'a cessé de s'accroître. Les systèmes développés intègrent de plus en plus d'innovations technologiques.

Cette évolution se rencontre dans de nombreux domaines que ce soit dans l'industrie manufacturière, dans le domaine militaire, dans celui des transports, dans l'agriculture et même depuis quelques années dans nos environnements domestiques. Aussi la robotique mobile prend différentes formes en fonction de l'objectif à atteindre que nous proposons de regrouper en deux familles :

- les robots qui ont une action de transformation sur leur environnement ou sur un objet de leur environnement (soudure, nettoyage, peinture, désherbage)
- les robots qui ont principalement une action de déplacement pour transporter un objet ou pour faire de l'observation.

Les deux familles ne sont pas complètement disjointes. Il est possible de retrouver des solutions hybrides intégrant un bras robotisé sur une plateforme mobile (figure 1.1). Ces solutions ont été développées pour le transport d'objet nécessitant un ou plusieurs robots. Leurs applications ont la particularité de faire intervenir la plupart des domaines impliqués en robotique.

Dans ces travaux, nous nous intéresserons essentiellement aux robots de la deuxième famille sachant que nos résultats peuvent facilement s'appliquer à l'hybridation évoquée.

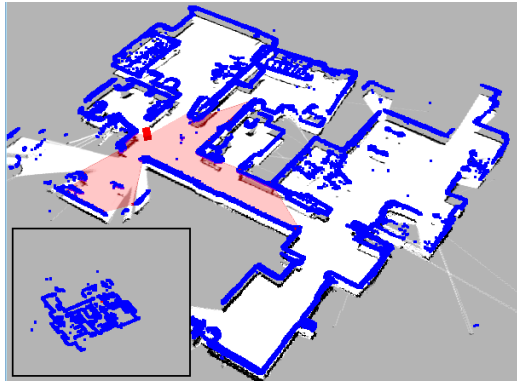


FIGURE 1.1 – Exemples de robots mobiles manipulateurs développés par les industriels Robotnik et Omron

Généralement, l'objectif d'un robot mobile est de se rendre d'un point A à un point B avec une trajectoire plus ou moins complexe dépendant de sa mission. Ces actions comprennent :

- les missions d'exploration, où l'objectif sera d'acquérir des informations sur l'environnement. Il peut s'agir, par exemple, de reconstruction archéologique, d'évaluation de rendements agricoles, de recherche d'indices particuliers, etc. La figure 1.2a montre un exemple de cartographie 2D d'un bâtiment obtenu grâce aux mesures d'un lidar installé sur un robot mobile se déplaçant dans le bâtiment.
- les missions de transport, où l'objectif sera de déplacer un objet le plus rapidement possible et en toute sécurité. Ici, la trajectoire du point A au point B sera la plus directe. Ce type de mission est très présent dans les entrepôts (figure 1.2b).

Ces deux types de missions impliquent une planification de la trajectoire à réaliser ainsi qu'un ensemble d'actions permettant au robot de se déplacer le long de la trajectoire calculée. La planification est une étape qui est généralement réalisée avant le commencement du déplacement du véhicule. Il existe cependant des scénarios plus complexes faisant intervenir de la replanification au cours de la mission. Dans un second temps, la trajectoire sera suivie par le véhicule commandé automatiquement à partir des informations perçues via ses capteurs ou données par des éléments de son environnement. C'est donc un ensemble d'actions qui va



(a) Cartographie d'un bâtiment.
(source : RUIZ-SARMIENTO et al. 2017)



(b) Transport de marchandise.
(entreprise GreyOrange)

FIGURE 1.2 – Exemples d'applications de la robotique mobile.

permettre au robot d'agir, de se localiser et d'interpréter son environnement pour effectuer sa mission. Parmi ces actions, nous pouvons citer :

- des lois de commande généralement plus complexes que celles implémentées dans des automates industriels, car permettant d'asservir le robot simultanément en position et en orientation comme proposer par exemple dans (THUILOT et al. 2001).
- des algorithmes de localisation permettant au robot d'évaluer sa position par rapport à sa trajectoire ou plus globalement par rapport aux contraintes de sa mission.
- une perception des éléments de son environnement pour assurer sa sécurité et satisfaire les contraintes qui lui sont imposées (vitesse, stabilité, ...)

Les environnements considérés sont très variés impliquant différents types de robots :

- les véhicules à guidage automatique (*automatic guided vehicle*, AGV) qui sont généralement équipés de roues ou de chenilles pour se déplacer sur le domaine terrestre.
- les aéronefs sans pilote (*unmanned aerial vehicle*, UAV) se déplaçant dans le domaine aérien.
- les robots sous-marins autonomes (*autonomous underwater vehicle*, AUV) se déplaçant dans l'eau.

Si mécaniquement l'architecture des robots est adaptée à leur environnement, les capteurs embarqués sont, de la même manière, souvent spécifiques. Ainsi, les UAV utiliseront principalement des AHRS (*Attitude and Heading Reference System*), des radars et des capteurs de type système de positionnement par satellite (GNSS), alors que les robots sous-marins sont généralement pourvus de sonars et de systèmes inertiels très performants. Ces spécificités sont rendues nécessaires du fait de la grande dépendance des robots envers la perception de leur environnement. En effet, celui-ci, en plus de sa nature, peut varier fortement au cours du temps du fait de la présence potentielle d'autres acteurs qui peuvent être statiques ou mobiles. Par exemple, un entrepôt qui a été préalablement cartographié dans des conditions vierges de tout autre acteur

peut présenter des difficultés du fait des évolutions nécessaires à son fonctionnement. Ainsi, des éléments tel que des objets statiques (cartons, palettes) peuvent apparaître, mais aussi d'autres acteurs mobiles et en particulier d'autres robots mobiles peuvent être présents. Ces changements peuvent poser des difficultés du fait des masquages engendrés, mais peuvent également être source d'informations complémentaires pour peu qu'ils soient capables de les communiquer. C'est ce que propose le concept de robotique collaborative que nous allons évoquer par la suite.

1.2 Collaboration en robotique mobile

De la même façon que différentes personnes collaborent pour un même un objectif (porter une charge lourde, explorer un territoire, se repérer dans l'espace), des robots peuvent développer le même type de stratégie pour une plus grande efficacité, ou pour des actions nécessitant un nombre important d'entités (ex : entrepôt de logistique, transport publics, travaux agricoles). Cette collaboration introduit de nouvelles problématiques qui sont encore aujourd'hui largement ouvertes :

- la communication entre les robots incluant différentes technologies mais également le choix et la pertinence des informations échangées. Il existe des solutions de communication développées spécialement pour la collaboration en robotique mobile ; par exemple, la solution VANET (*Vehicular Ad Hoc Network*), dont l'objectif est de fournir une architecture réseau décentralisée avec des méthodes de routage spécifiques à l'utilisation de véhicules (GAIKWAD et ZAVERI 2011).
- la perception et l'identification des robots avoisinants permettant de connaître l'origine des informations échangées (position relative et type d'information)

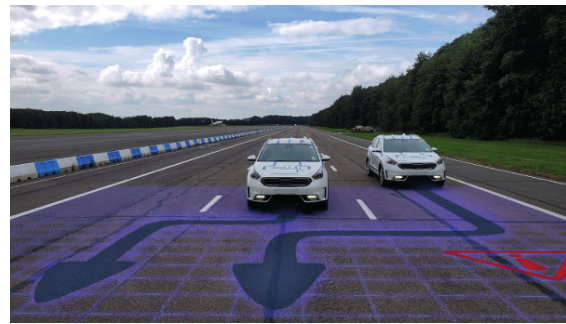
Ces problématiques adressent plusieurs types d'application. Par exemple, un véhicule autonome se déplaçant dans une agglomération et arrivant à un carrefour pourra communiquer avec d'autres véhicules proches du même carrefour mais dans une autre voie de circulation afin d'évaluer les dangers et de réduire ou pas sa vitesse. Un autre exemple concerne le dépassement d'un bus à l'arrêt qui peut masquer un piéton parfaitement visible par un autre véhicule arrivant en sens inverse. **Dans nos travaux, nous nous intéresserons à la localisation de robots hétérogènes mais coopérant au sein d'un même environnement. L'objectif étant d'améliorer la précision et la robustesse de chaque entité en bénéficiant de leur potentiel respectif.**

1.3 Problématique

De nombreux projets, en particulier dans le domaine automobile, s'intéressent à la coopération de robots pour améliorer la sécurité, pour fluidifier le trafic ou pour de la gestion de flotte. Citons par exemple le projet COVEL (*Cooperative Vehicle Localization for Safe and Sustainable Mobility*), impliquant le constructeur Fiat, dont l'objectif est de partager des informations de cartographie entre les véhicules pour améliorer la localisation dans la carte (figure 1.3a). Un autre projet nommé MuCCA (*Multi-Car Collision Avoidance*), utilise la coopération de véhicules pour synchroniser leurs déplacements lors de l'évitement d'un obstacle détecté par un ou plusieurs véhicules de la flotte (figure 1.3b). Les véhicules impliqués dans cette expérimentation ont la connaissance de la position et de la vitesse des autres véhicules. Cela permet de définir une stratégie d'évitement synchronisée mais spécifique à chaque véhicule.



(a) Projet COVEL



(b) Projet MuCCA

FIGURE 1.3 – Illustration de scénarios impliqués dans les projets COVEL et MuCCA

Parmi les différents domaines scientifiques adressés par la coopération de robots, la localisation inter-véhicule est une des problématiques qui peut bénéficier le plus de l'échange d'informations entre les différentes entités. Il est ainsi possible d'apporter des informations utiles à des robots qui n'ont pas forcément la capacité de les percevoir par eux-mêmes du fait de leur équipement ou de la situation dans laquelle ils se trouvent. C'est le cas des véhicules agricoles évoluant en convois équipés différemment avec par exemple un des véhicules embarquant un GPS RTK précis au centimètre, les autres utilisant des systèmes moins onéreux mais également moins précis (figure 1.4a). L'intérêt d'utiliser une approche coopérative dans la localisation des véhicules permettrait à l'ensemble de la flotte de bénéficier d'une précision proche de celle du robot équipé du GPS RTK. Cela nécessite néanmoins de mettre en place des moyens de communication et de détection entre les véhicules leur permettant d'échanger en temps réel des informations et de connaître leur position relative. C'est également le cas pour les véhicules urbains de types navettes autonomes, acteurs prochains de l'inter-modalité des transports dans les grandes agglomérations (figure 1.4b). Ce type de véhicules peut être présent en quantité importante et peut fonctionner seul ou en convoi. Ils évoluent dans des environnements et dans des conditions où leur système de localisation peut être mis en défaut. Ils sont cependant rarement

seuls et les infrastructures qui, dans certains cas, peuvent être équipées pour leur fournir de l'information. L'idée, là aussi, réside dans l'échange d'information entre les différentes navettes ou avec d'autres acteurs fixes ou mobiles. Cependant, l'échange d'informations entre robots, en particulier pour améliorer la précision et la robustesse de leur localisation pose plusieurs grandes questions :

- Tous les robots doivent pouvoir communiquer en temps réel. L'aspect coopératif de la localisation s'appuie sur le fait que des informations datées sont échangées entre les robots qui se trouvent à proximité.
- La perte de communication avec un des robots ne doit pas bloquer le bon fonctionnement de l'algorithme de localisation des autres robots. La méthode doit être indépendante du nombre de robots, ce qui nous conduit à privilégier des approches décentralisées.
- Pour assurer la qualité de la communication, il est préférable de limiter la quantité d'informations échangées et donc de privilégier des informations d'états par rapport aux informations capteurs. Cela implique une observation entre les robots permettant d'établir un lien unique nécessitant d'identifier parfaitement les deux acteurs.
- La qualité d'une information, d'autant plus si elle est échangée, doit être garantie. Aussi, les estimations des états des robots qui sont communiquées doivent être suffisamment fiables pour que la pose réelle d'un robot soit comprise dans la zone de son estimation.
- Il faut prendre en compte le degré de corrélation des informations échangées entre les robots : ceci correspond au phénomène de rumeur qui peut entacher la fiabilité des données.
- Pour être suffisamment générique, la solution ne doit pas impliquer de contraintes sur l'environnement. Il peut s'agir d'un environnement connu ou inconnu.



(a) Activité agricole



(b) Navettes en ville

FIGURE 1.4 – Exemples de situations pouvant faire intervenir une collaboration de robots mobiles

1.4 Contribution de la thèse

Ce travail s'inscrit dans le cadre de l'action IRMA AVN ("*Integrated Risk Management Architecture for Autonomous Vehicle Navigation*") du Labex Imobs³ ("*Innovative Mobility: Smart and Sustainable Solutions*"). L'objectif principal est de proposer une solution de localisation coopérative robuste pour une flotte de robots mobiles évoluant dans un environnement quelconque. Cet objectif doit intégrer les différentes contraintes permettant aux robots d'évoluer avec le maximum de sécurité et dans les scénarios les plus variés. L'approche proposée est une approche décentralisée incluant une indépendance de tous les robots, ce qui permet une gestion dynamique de la flotte. Chaque robot ne calcule que son propre état en étant capable d'intégrer les observations issues des autres entités fixes ou mobiles. Cette stratégie et la gestion de la consanguinité des informations qu'elle induit est la première contribution de ces travaux.

Pour répondre à la problématique de l'observation des robots, nous avons choisi d'utiliser la technologie radio Ultra-Wideband (UWB) intégrées dans des balises conçues et réalisées au sein du Labex. Celle-ci fournit une mesure de distance entre deux balises communicantes installées sur deux robots. Du fait de la large bande de fréquence utilisée, le calcul de distance fourni est relativement précis. Cette technologie permet de répondre aux problèmes de l'observation inter-robot tout en les identifiant précisément. Cependant, l'utilisation d'une unique information de distance introduit une forte non-linéarité dans le calcul de l'état dont la prise en compte est la seconde contribution de ces travaux.

Enfin, ces travaux ont donné lieu à de nombreuses expérimentations en simulation et en conditions réelles, dont les résultats ont permis de valider la robustesse de l'approche proposée.

1.5 Présentation du manuscrit

Ce manuscrit se divise en 6 chapitres. Le premier chapitre nous a permis d'introduire le sujet avec ses problématiques ainsi que les principales contributions apportées. Le second chapitre présente les principaux outils utilisés pour réaliser l'estimation d'états dynamiques. Cela correspond aux algorithmes de filtrage et leurs outils de caractérisation. Le troisième chapitre se focalise sur la localisation coopérative en présentant les différents travaux qui ont été réalisés dans ce domaine. Les points comme la corrélation et la non-linéarité des observations seront détaillés. Le quatrième chapitre se consacre à la présentation de l'approche que nous avons développée. Le cinquième chapitre présente quelques expérimentations illustrant les différents points évoqués dans la construction de l'algorithme. Pour finir, le dernier chapitre résume l'ensemble des contributions et des résultats. Quelques perspectives seront également proposées.

Chapitre 2

Outils fondamentaux pour l'estimation d'états

2.1 Introduction

L'une des sciences fondamentales impliquées en robotique est l'automatique. Elle correspond à l'étude de la commande de systèmes dynamiques. La commande d'un système consiste à l'amener d'un état à un autre en un temps fini. Par exemple, cela peut correspondre au déplacement de l'effecteur d'un bras robotique d'un point A à un point B . Dans ce cas, l'état du système peut correspondre à la position et à l'orientation de l'effecteur. Pour qu'une commande soit efficace, il est nécessaire de connaître l'état du système au départ, mais également au fur et à mesure du déplacement, afin de corriger d'éventuelles erreurs de commande.

Malheureusement, l'état d'un système est une valeur abstraite qui est souvent impossible à obtenir. Pour les systèmes les plus simples, il est possible de la mesurer. Si l'on prend l'exemple d'un four, son état peut correspondre à sa température. L'état peut donc être mesuré avec un thermomètre. Cependant, les thermomètres ne sont pas des machines parfaites. L'information de température obtenue comportera obligatoirement une part d'erreur. Pour d'autres systèmes plus complexes, il est souvent difficile de mesurer directement son état. Il s'agit alors d'une estimation obtenue par calcul à partir d'autres informations, qui elles, sont mesurables. C'est le cas notamment de la localisation en robotique mobile. Les robots mobiles sont souvent placés dans des environnements où il n'existe pas de solution technologique permettant de mesurer leur position et leur orientation.

Puisque l'estimation d'un état s'appuie également sur des mesures, elle ne peut donc que s'approcher de la valeur vraie. L'erreur d'une estimation est plus importante qu'une simple erreur de mesure puisqu'elle fait également intervenir l'erreur de modélisation. En effet, une

estimation s'obtient par résolution d'un modèle mathématique construit spécifiquement pour le système à étudier. Si ce modèle représente mal le système réel, l'erreur de l'estimation sera grande. Dans la pratique, les modèles utilisés font beaucoup d'approximations afin que des systèmes informatiques soient capables de donner des résultats en un temps fini.

L'objectif de ce chapitre est donc de présenter quelques outils mathématiques couramment utilisés pour l'estimation d'états en robotique mais également dans d'autres domaines. L'explication commence par une présentation du modèle d'état et de son estimation dans le cas général. En appliquant certaines contraintes sur les bruits de mesures, cette forme générale trouve alors une solution analytique, utilisée dans la famille des filtres gaussiens. Nous nous intéresserons ensuite aux filtres particuliers, pour lesquels les contraintes sont moins fortes mais les coûts des calculs sont plus élevés. Nous aborderons alors les mixtures de gaussiennes, dont l'intérêt est de relâcher ces mêmes contraintes mais en restant assez proches des approches gaussiennes. Nous verrons également les approches ensemblistes, qui utilise un point de vue très différent du problème de l'estimation. Nous finirons par présenter quelques outils permettant de qualifier la qualité de ces estimations.

2.2 Estimation bayésienne

L'estimation de l'état d'un système dynamique évolue au cours du temps. Cet état est représenté par un vecteur \mathbf{x}_k à un instant k . Il est calculé à partir d'un modèle d'évolution f et d'un modèle d'observation h :

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \quad (2.1)$$

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \quad (2.2)$$

Le modèle d'évolution f fait intervenir l'estimation précédente \mathbf{x}_{k-1} , un vecteur de mesure \mathbf{u}_k et un vecteur de bruit \mathbf{w}_k servant à modéliser les erreurs de mesure et de modélisation. Cette fonction est utilisée pour essayer de prédire la valeur de l'état à l'instant k . Le modèle d'observation h établit un lien entre l'état \mathbf{x}_k et un vecteur de mesure \mathbf{z}_k représentant une observation de l'environnement du système. Ce modèle sert à corriger la prédiction de l'état en lui apportant des informations extérieures. Puisque la modélisation et les mesures ne sont jamais parfaites, celui-ci fait également intervenir un vecteur de bruit \mathbf{v}_k .

Pour correctement prendre en compte l'imprécision des mesures et des modélisations, les approches probabilistes modélisent l'estimation de l'état par une loi de distribution. L'état \mathbf{x}_k , ainsi que les vecteurs de bruits \mathbf{w}_k et \mathbf{v}_k sont considérés comme des variables aléatoires, et sont donc associés à des densités de probabilité. Le calcul d'une estimation consiste alors à déterminer une nouvelle loi de probabilité à un instant k à partir des mesures et des états précédents.

Les interactions entre les mesures et les états peuvent être modélisées par un réseau bayésien (figure 2.1). Les transitions de ce schéma représentent l'existence d'une dépendance entre les variables présentées.

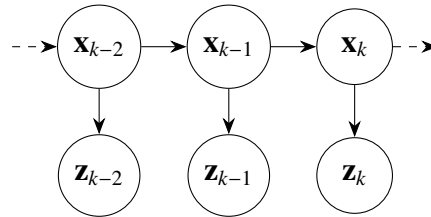


FIGURE 2.1 – Réseau bayésien représentant l'interaction entre les estimations de l'état du robot \mathbf{x} et les mesures \mathbf{z} à différents instants k .

Notre objectif est de connaître l'état du robot à un instant k en prenant en compte toutes les mesures jusqu'à cet instant. Cela revient à chercher la densité de probabilité conditionnelle $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ avec $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$. Dans cette modélisation, le calcul de cette estimation se fait en deux étapes. La première étape correspond à la prédiction de l'état à l'instant k . L'expression de sa densité est obtenue grâce à l'équation de Chapman-Kolomgorov :

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}. \quad (2.3)$$

La densité $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ est calculée à partir de la fonction d'évolution de l'équation 2.1. La seconde étape correspond à la mise à jour de l'état après la prise en compte de l'observation à l'instant k . Sa densité de probabilité est calculée grâce à la formule de Bayes :

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \quad (2.4)$$

La densité $p(\mathbf{z}_k|\mathbf{x}_k)$, appelée *vraisemblance*, est calculée à partir de la fonction d'observation donnée à l'équation 2.2.

Ces expressions correspondent à la forme générale pour l'estimation bayésienne de l'état d'un robot. En pratique, ces formules ne peuvent être utilisées directement puisqu'il s'agit de densité de probabilité continue et quelconque. Ce genre de probabilité n'est pas directement manipulable en informatique.

2.3 Filtres gaussiens

2.3.1 Filtre de Kalman

Le filtre de Kalman (KF) est un algorithme permettant d'estimer l'état d'un système en utilisant le modèle d'état des équations (2.1) et (2.2). Cependant, cet algorithme ne s'applique qu'à un cas particulier où les variables aléatoires impliquées dans les modèles d'évolution et d'observation sont fortement contraintes. Les vecteurs de bruit \mathbf{w}_k et \mathbf{v}_k sont considérés comme blancs, gaussiens, centrés et indépendants :

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \quad (2.5)$$

$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k). \quad (2.6)$$

Les matrices \mathbf{Q}_k et \mathbf{R}_k correspondent aux matrices de covariance de ces bruits. Elles sont choisies de façon à ce que la modélisation du bruit corresponde au mieux à l'imprécision des fonctions d'évolution et d'observation et des capteurs utilisés. Le filtre de Kalman requiert également que les fonctions d'évolution et d'observation soient linéaires. Elles s'écrivent donc sous la forme

$$f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) = \mathbf{F} \mathbf{x}_{k-1} + \mathbf{B} \mathbf{u}_k + \mathbf{w}_k \quad (2.7)$$

$$g(\mathbf{x}_k, \mathbf{v}_k) = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k. \quad (2.8)$$

Les matrices \mathbf{F} , \mathbf{B} et \mathbf{H} sont des matrices propres aux modèles utilisés. En utilisant des modèles linéaires et des lois normales pour les bruits, l'estimation de l'état va aussi correspondre à une loi normale. Nous utilisons ici la syntaxe $\mathbf{x}_{i|j}$ pour désigner l'espérance de l'état \mathbf{x} à un instant i en prenant en compte les mesures jusqu'à un instant j . Les densités de probabilité de l'état *a priori* $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ et *a posteriori* $p(\mathbf{x}_k | \mathbf{z}_{1:k})$ prennent donc la forme :

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1}) \quad (2.9)$$

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_{k|k}, \mathbf{P}_{k|k}) \quad (2.10)$$

Le filtre de Kalman permet d'estimer un état en ne manipulant que l'espérance et la matrice de covariance de sa loi normale. Dans l'étape de prédiction, le couple $(\mathbf{x}_{k|k-1}, \mathbf{P}_{k|k-1})$ est obtenu grâce aux équations

$$\mathbf{x}_{k|k-1} = \mathbf{F} \mathbf{x}_{k-1} + \mathbf{B} \mathbf{u}_k \quad (2.11)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F} \mathbf{P}_{k-1|k-1} \mathbf{F}^\top + \mathbf{Q}_k. \quad (2.12)$$

Pour l'étape de mise à jour, le couple $(\mathbf{x}_{k|k}, \mathbf{P}_{k|k})$ se calcule de la façon suivante :

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}^\top (\mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^\top + \mathbf{R}_k)^{-1} \quad (2.13)$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \mathbf{x}_{k|k-1}) \quad (2.14)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H} \mathbf{P}_{k|k-1} \quad (2.15)$$

Les équations 2.11 à 2.15 sont l'estimation de l'état à l'instant k . Ces calculs sont répétés périodiquement depuis l'instant 0 pour avoir la progression de la localisation du robot. L'état \mathbf{x}_0 doit cependant être initialisé manuellement en fonction des conditions de départ, en faisant en sorte d'être dans de bonnes conditions pour cette première estimation.

Le filtre de Kalman donne une solution optimale si le système respecte les contraintes présentées. Dans la pratique, il est rarement possible de modéliser les fonctions d'évolution et d'observation de façon linéaire. C'est justement le problème de la plupart des modèles utilisés en robotique mobile. Pour le résoudre, des variations de l'algorithme ont été développées.

2.3.2 Filtre de Kalman étendu

Le filtre de Kalman étendu (EKF) est une version du KF permettant de relâcher la contrainte de linéarité des modèles d'évolution et d'observation. L'algorithme autorise l'utilisation de fonctions différentiables pour ces modèles. La stratégie adoptée dans cet algorithme est de linéariser les fonctions au niveau de la dernière estimation. L'étape de prédiction prend alors la forme :

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{0}) \quad (2.16)$$

$$\mathbf{P}_{k|k-1} = \mathbf{J}_{fx} \mathbf{P}_{k-1|k-1} \mathbf{J}_{fx}^\top + \mathbf{J}_{fw} \mathbf{Q}_k \mathbf{J}_{fw}^\top \quad (2.17)$$

avec

$$\mathbf{J}_{fx} = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}_{k-1|k-1}, \mathbf{u}_k, \mathbf{0}), \quad \mathbf{J}_{fw} = \frac{\partial f}{\partial \mathbf{w}}(\mathbf{x}_{k-1|k-1}, \mathbf{u}_k, \mathbf{0}). \quad (2.18)$$

L'étape de mise à jour devient :

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{J}_{hx}^\top (\mathbf{J}_{hx} \mathbf{P}_{k|k-1} \mathbf{J}_{hx}^\top + \mathbf{J}_{hv} \mathbf{R}_k \mathbf{J}_{hv}^\top)^{-1} \quad (2.19)$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\mathbf{x}_{k|k-1}, \mathbf{0})) \quad (2.20)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{J}_{hx} \mathbf{P}_{k|k-1} \quad (2.21)$$

avec

$$\mathbf{J}_{hx} = \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}_{k|k-1}, \mathbf{0}), \quad \mathbf{J}_{hv} = \frac{\partial h}{\partial \mathbf{v}}(\mathbf{x}_{k|k-1}, \mathbf{0}). \quad (2.22)$$

L'EKF permet d'utiliser les modèles d'évolution et d'observation que l'on rencontre classiquement en robotique mobile, mais il faut bien garder à l'esprit qu'il effectue une linéarisation de ces modèles. La solution n'est donc pas optimale et peut parfois s'éloigner fortement de la réalité lorsque les modèles sont trop non-linéaires. Il est donc nécessaire de s'assurer que la non-linéarité des modèles ne soit pas trop importante pour que l'algorithme soit efficace.

2.3.3 Filtre de Kalman sans parfum

Le filtre de Kalman sans parfum, ou *uncented Kalman filter* (UKF) en anglais, est une variante du filtre de Kalman utilisé dans le cas où les non-linéarités sont trop importantes pour qu'un EKF puisse les gérer correctement. Cet algorithme a donc le même objectif que le filtre de Kalman étendu : obtenir une estimation gaussienne de l'état à un instant k à partir de l'estimation à l'instant précédent, en utilisant des modèles d'évolution f et d'observation h non linéaires. Le problème est que l'application d'une fonction non linéaire à une variable aléatoire gaussienne ne va pas donner une variable aléatoire également gaussienne. Pour le résoudre, l'EKF se base sur une linéarisation de la transformation en utilisant les matrices jacobiniennes des fonctions f et h . Cette linéarisation a tendance à donner des résultats assez éloignés de la réalité lorsque les non-linéarités sont fortes. L'UKF remplace cette opération par une transformation donnant de meilleurs résultats, tout en ayant des performances de calcul similaires.

Uncented transformation

La transformation utilisée par l'UKF est appelée *uncented transformation* (UT). Elle se base sur la création d'échantillons appelés *sigma points*. Ces échantillons sont choisis dans l'espace d'état afin de représenter au mieux l'espérance et la covariance de cet état. Ils sont ensuite transformés par l'application d'une fonction f pour obtenir un ensemble d'échantillons représentant un nouvel état. De cet ensemble est alors déduit, l'espérance et la covariance du nouvel état.

Soient \mathbf{x} et \mathbf{P}_x , l'espérance et la covariance d'une variable aléatoire de dimension n . L'UT nécessite la génération de $2n + 1$ *sigma points* \mathcal{X}_i , pour lesquels sont calculées, les images $\mathcal{Y}_i = f(\mathcal{X}_i)$. L'espérance et la covariance de la gaussienne obtenues après cette transformation

ont pour expression :

$$\mathbf{y} = \sum_{i=0}^{2n} w_i^{(m)} \mathcal{Y}_i, \quad \mathbf{P}_y = \sum_{i=0}^{2n} w_i^{(c)} (\mathcal{Y}_i - \mathbf{y})(\mathcal{Y}_i - \mathbf{y})^\top \quad (2.23)$$

avec $w_i^{(m)}$ et $w_i^{(c)}$, les poids des *sigma points* utilisés dans les calculs de l'espérance et la covariance. Ces poids sont calculés à partir de 3 paramètres :

- $\alpha \in [0; 1]$ sert à configurer la dispersion des échantillons
- κ sert également à contrôler la dispersion, mais il est souvent considéré comme nul
- β permet d'introduire une connaissance *a priori* sur la distribution de probabilité du vecteur d'état (pour une loi normale, $\beta = 2$).

Les *sigma points* et leurs poids sont obtenus par les expressions suivantes :

$$\mathcal{X}_0 = \mathbf{x} \quad (2.24)$$

$$\mathcal{X}_i = \mathbf{x} + \left(\sqrt{(n + \lambda) \mathbf{P}_x} \right)_i \quad \forall i \in \llbracket 1; n \rrbracket \quad (2.25)$$

$$\mathcal{X}_i = \mathbf{x} - \left(\sqrt{(n + \lambda) \mathbf{P}_x} \right)_i \quad \forall i \in \llbracket n+1; 2n \rrbracket \quad (2.26)$$

$$w_0^{(m)} = \frac{\lambda}{n + \lambda} \quad (2.27)$$

$$w_0^{(c)} = \frac{\lambda}{n + \lambda} + 1 - \alpha^2 + \beta \quad (2.28)$$

$$w_i^{(m)} = w_i^{(c)} = \frac{1}{2(n + \lambda)} \quad \forall i \in \llbracket 1; 2n \rrbracket \quad (2.29)$$

avec $\lambda = \alpha^2 (n + \kappa) - n$. Le vecteur $\left(\sqrt{(n + \lambda) \mathbf{P}_x} \right)_i$ correspond à la $i^{\text{ième}}$ colonne de la décomposition de Cholesky.

Estimation d'état avec l'UKF

L'objectif est d'obtenir le couple $(\mathbf{x}_{k|k}, \mathbf{P}_{k|k})$ correspondant à l'estimation de l'espérance et la covariance de l'état à l'instant k . On considère que l'on connaît l'estimation $(\mathbf{x}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})$ obtenue à l'instant $k - 1$.

La construction des *sigma points* se fait en utilisant les équations (2.24) à (2.29). Cependant, l'état utilisé ici est une version étendue de l'état à estimer dont la taille prend en compte également les dimensions des bruits d'évolution \mathbf{w}_k et d'observation \mathbf{v}_k . Cet état est composé

d'un vecteur \mathbf{x}_{k-1}^a et d'une matrice \mathbf{P}_{k-1}^a dont les expressions sont les suivantes :

$$\begin{aligned}\mathbf{x}_{k-1}^a &= \begin{bmatrix} \mathbf{x}_{k-1|k-1}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix}^\top \\ \mathbf{P}_{k-1}^a &= \begin{bmatrix} \mathbf{P}_{k-1|k-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_k \end{bmatrix}\end{aligned}\quad (2.30)$$

avec \mathbf{Q}_k et \mathbf{R}_k , les matrices de covariance des bruits d'évolution et d'observation. Si l'on appelle n , la dimension de \mathbf{x}_{k-1}^a alors le nombre de particules à générer vaut $2n$. Les *sigma points* obtenus peuvent donc être découpés en 3 parties

$$\mathcal{X}_{k-1}^a = \left[(\mathcal{X}_{k-1}^x)^\top \quad (\mathcal{X}_{k-1}^w)^\top \quad (\mathcal{X}_{k-1}^v)^\top \right]^\top \quad (2.31)$$

correspondant au trois vecteurs utilisés pour la construction de \mathbf{x}_{k-1}^a . L'étape de prédiction correspond aux calculs suivants :

$$\mathcal{X}_k^x = f(\mathcal{X}_{k-1|k-1}^x, \mathbf{u}_k, \mathcal{X}_{k-1}^w) \quad (2.32)$$

$$\mathbf{x}_{k|k-1} = \sum_{i=0}^{2n} w_i^{(m)} \mathcal{X}_k^x \quad (2.33)$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} w_i^{(c)} (\mathcal{X}_k^x - \mathbf{x}_{k|k-1})(\mathcal{X}_k^x - \mathbf{x}_{k|k-1})^\top \quad (2.34)$$

avec \mathbf{u}_k , le vecteur de mesure utilisé dans le modèle d'évolution. L'étape de mise à jour correspond aux calculs suivants :

$$\mathcal{Y}_k = h(\mathcal{X}_k^x, \mathcal{X}_{k-1}^v) \quad (2.35)$$

$$\mathbf{y}_k = \sum_{i=0}^{2n} w_i^{(m)} \mathcal{Y}_k \quad (2.36)$$

$$\mathbf{P}_{yy,k} = \sum_{i=0}^{2n} w_i^{(c)} (\mathcal{Y}_k - \mathbf{y}_k)(\mathcal{Y}_k - \mathbf{y}_k)^\top \quad (2.37)$$

$$\mathbf{P}_{xy,k} = \sum_{i=0}^{2n} w_i^{(c)} (\mathcal{X}_k^x - \mathbf{x}_{k|k-1})(\mathcal{Y}_k - \mathbf{y}_k)^\top \quad (2.38)$$

$$\mathbf{K}_k = \mathbf{P}_{xy,k} \mathbf{P}_{yy,k}^\top \quad (2.39)$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{y}_k) \quad (2.40)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} + \mathbf{K}_k \mathbf{P}_{yy,k} \mathbf{K}_k^\top \quad (2.41)$$

avec \mathbf{z}_k , le vecteur de mesure de l'observation.

Avec un UKF, les problèmes de non-linéarités sont mieux gérés dans les calculs, mais l'entrée et la sortie de chaque étape d'évolution ou de mise à jour restent toujours des approximations gaussiennes. Cela signifie que l'effet de la non-linéarité sur les résultats est amoindri mais est toujours perceptible sur le long terme. En effet, bien que l'on considère que l'estimation soit gaussienne à l'état initial, la densité de probabilité réelle de l'état risque d'être très déformée après une longue période. Dans le pire des cas, cette densité peut même devenir multi-modale. Ce genre de filtre se limite donc à des applications où l'état reste facilement modélisable par des lois normales.

2.3.4 Filtre à intersection de covariance

Le filtre à intersection de covariance (*covariance intersection filter*, CIF), introduit par (S. J. JULIER et J. K. UHLMANN 1997), est une variante du filtre de Kalman qui a la particularité d'éviter la sur-convergence lorsque les données en entrée du filtre sont corrélées. Dans le filtre de Kalman, nous considérons que les mesures utilisées lors de la mise à jour du vecteur d'état sont totalement indépendantes de ce dernier. Pour le filtre à Intersection de Covariance, nous considérons cette fois-ci que les mesures sont corrélées au vecteur d'état. Ce filtre peut être vu comme une version *pessimiste* du filtre de Kalman.

Pour illustrer les différences entre ces deux algorithmes, prenons un exemple simple. Soient \mathbf{x} et \mathbf{P} , l'espérance et la covariance d'une estimation d'un état. Soient \mathbf{y} et \mathbf{R} , l'espérance et la covariance d'une estimation d'une mesure. On nomme $(\mathbf{x}_{\mathcal{K}}, \mathbf{P}_{\mathcal{K}})$ la fusion de (\mathbf{x}, \mathbf{P}) et de (\mathbf{y}, \mathbf{R}) en utilisant la méthode classique du filtre de Kalman. Le résultat de cette opération est obtenu par les équations

$$\mathbf{P}_{\mathcal{K}}^{-1} = \mathbf{P}^{-1} + \mathbf{R}^{-1} \quad (2.42)$$

$$\mathbf{x}_{\mathcal{K}} = \mathbf{P}_{\mathcal{K}}(\mathbf{P}^{-1}\mathbf{x} + \mathbf{R}^{-1}\mathbf{y}). \quad (2.43)$$

Cette méthode de fusion, présentée par (CHONG et MORI 2001), est parfois appelée combinaison convexe (*Convex Combination*, CC). Dans la version du filtre de Kalman habituellement utilisée, elle n'apparaît pas directement sous cette forme puisque qu'elle prend en compte également les transformations appliquées par la fonction d'observation. Pour l'intersection de covariance (*covariance intersection*, CI), c'est pratiquement la même formule. La somme des inverses est juste pondérée par un scalaire $\omega \in [0; 1]$

$$\mathbf{P}_C^{-1} = \omega\mathbf{P}^{-1} + (1 - \omega)\mathbf{R}^{-1} \quad (2.44)$$

$$\mathbf{x}_C = \mathbf{P}_C(\omega\mathbf{P}^{-1}\mathbf{x} + (1 - \omega)\mathbf{R}^{-1}\mathbf{y}) \quad (2.45)$$

Le coefficient ω peut être choisi de manière à optimiser un critère comme la minimisation de la

trace ou le déterminant de \mathbf{P}_C .

La figure 2.2 illustre la différence entre les résultats de la fusion CC et CI. Dans le cas de la CC, la fusion de \mathbf{P} et \mathbf{R} donne un résultat plus précis, car ces deux estimations sont considérées comme indépendantes. Le résultat de leur fusion maximise la quantité d'information apportée. À l'inverse, la CI considère que ces estimations sont entièrement corrélées entre elles. Cela revient à considérer que les deux estimations correspondent à une même estimation mais avec un point de vue différent.

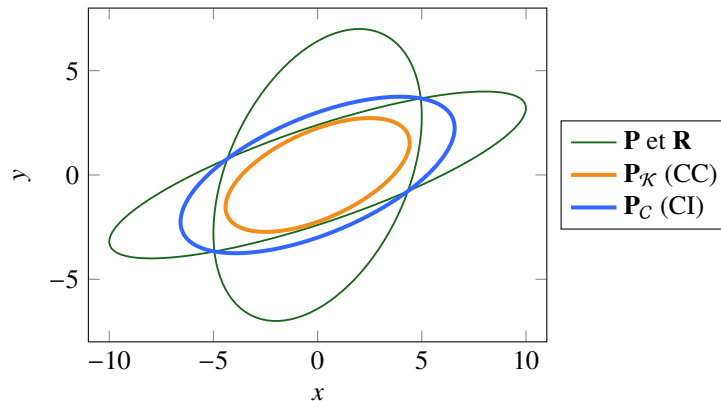


FIGURE 2.2 – Comparaison entre la fusion des algorithmes CC et CI. La fusion CI minimise le déterminant de \mathbf{P}_C .

(S. J. JULIER et J. K. UHLMANN 1997) ont montré que le résultat reste intègre pour n'importe quel choix de $\omega \in [0; 1]$. Cette variable représente un coefficient d'interpolation entre les deux estimations. Nous pouvons en effet remarquer que si $\omega = 1$ alors le résultat de la CI correspond à \mathbf{R} . Si $\omega = 0$ alors la matrice \mathbf{P}_C est équivalente à \mathbf{P} . La figure 2.3 montre l'ellipse obtenue pour différentes valeurs possibles de ω . Puisque tous ces résultats restent intègres, il suffit de prendre celui ayant les meilleurs caractéristiques. Le choix de la minimisation du déterminant à l'avantage de correspondre graphiquement à l'ellipse avec la plus petite aire.

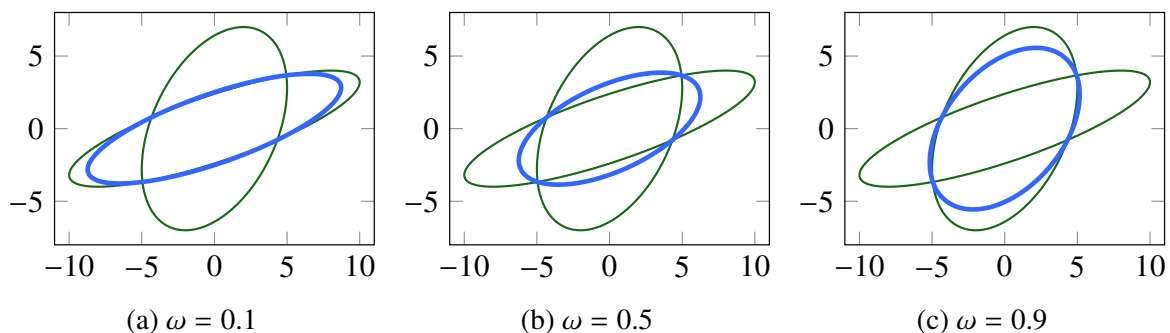


FIGURE 2.3 – La fusion CI pour différentes valeurs de ω .

La méthode de fusion que nous venons de voir peut être intégrée dans un filtre de Kalman en lieu et place de sa méthode de fusion CC. Puisque la méthode de fusion n'interfère pas avec la linéarisation, l'intersection de covariance peut être utilisée également dans un EKF. Ce

filtre est alors appelé filtre à intersection de covariance (*covariance intersection filter*, CIF). Les équations de l'étape de prédiction sont identiques à celles présentées en (2.16) et (2.17). L'étape de mise à jour s'écrit de la façon suivante (en utilisant les mêmes variables que le §2.3.1) :

$$\mathbf{P}_1 = \omega^{-1} \mathbf{P}_{k|k-1} \quad (2.46)$$

$$\mathbf{P}_2 = (1 - \omega)^{-1} \mathbf{R}_k \quad (2.47)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{J}_{hx}^T (\mathbf{J}_{hx} \mathbf{P}_1 \mathbf{J}_{hx}^T + \mathbf{J}_{hv} \mathbf{P}_2 \mathbf{J}_{hv}^T)^{-1} \quad (2.48)$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\mathbf{x}_{k|k-1}, \mathbf{0})) \quad (2.49)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_1 - \mathbf{K}_k \mathbf{J}_{hx} \mathbf{P}_1 \quad (2.50)$$

avec

$$\mathbf{J}_{hx} = \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}_{k|k-1}, \mathbf{0}), \quad \mathbf{J}_{hv} = \frac{\partial h}{\partial \mathbf{w}}(\mathbf{x}_{k|k-1}, \mathbf{0}). \quad (2.51)$$

Les matrices \mathbf{P}_1 et \mathbf{P}_2 sont des variables intermédiaires correspondant aux matrices $\mathbf{P}_{k|k-1}$ et \mathbf{R}_k et prenant en compte le coefficient ω obtenu après minimisation du déterminant du résultat.

En l'absence d'information sur la corrélation éventuelle entre l'état et une mesure, le CIF a l'avantage de donner un résultat dont l'information apportée par la mesure ne génère pas de sur-convergence. Contrairement au KF, appliquer plusieurs fois la même mesure n'a pas de conséquence sur le résultat. Cependant, il arrive parfois que l'on ait des informations sur la corrélation entre l'état et la mesure. Par exemple, nous pouvons savoir qu'un paramètre du vecteur de mesure est totalement indépendant de l'état. Pour ce genre de situation, il existe une alternative au CIF donnant de meilleurs résultats.

2.3.5 Filtre à intersection de covariance partitionné

Le filtre à intersection de covariance partitionné (*split covariance intersection filter*, SCIF) est une variante du CIF permettant de gérer plus précisément la corrélation des mesures. L'algorithme a été introduit par (S. JULIER et J. UHLMANN 2001). Le KF et le CIF ne permettent de manipuler les données que dans les deux cas extrêmes de la corrélation : lorsque les données à fusionner sont totalement indépendantes ou totalement corrélées. Le SCIF permet de régler plus finement la corrélation en fonction des connaissances que l'on peut avoir sur les informations à fusionner. Par exemple, si une partie du vecteur de mesure correspond à une information indépendante de l'état, le SCIF fusionnera cette mesure comme un EKF pour cette partie du vecteur et comme un CIF pour le reste.

Cet algorithme mélange les méthodes employées par le KF et le CIF. Ces deux approches modélisent l'estimation de l'état par un couple (\mathbf{x}, \mathbf{P}) correspondant à l'espérance et la cova-

riance d'une loi normale. L'estimation consiste à estimer un triplet $(\mathbf{x}, \mathbf{P}_i, \mathbf{P}_d)$ avec \mathbf{P}_i correspondant à la matrice de covariance indépendante et \mathbf{P}_d , la matrice de covariance corrélée. La matrice de covariance utilisée habituellement n'est autre que la somme

$$\mathbf{P} = \mathbf{P}_i + \mathbf{P}_d \quad (2.52)$$

La matrice \mathbf{P}_i est utilisée pour définir l'incertitude pour laquelle nous savons qu'il n'y a pas de corrélation avec l'état. La matrice \mathbf{P}_d contient l'incertitude sur laquelle nous n'avons pas d'information sur sa corrélation avec l'état.

Soit $(\mathbf{x}_{k-1|k-1}, \mathbf{P}_{i,k-1|k-1}, \mathbf{P}_{d,k-1|k-1})$, l'estimation de l'état obtenue à une étape précédente du filtre. Soit $(\mathbf{u}_k, \mathbf{Q}_{i,k}, \mathbf{Q}_{d,k})$, une mesure utilisée pour l'évolution de l'état. L'étape de prédiction du SCIF correspond aux expressions

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1|k-1}, \mathbf{u}_k, \mathbf{0}) \quad (2.53)$$

$$\mathbf{P}_{k|k-1} = \mathbf{J}_{fx} \mathbf{P}_{k-1|k-1} \mathbf{J}_{fx}^T + \mathbf{J}_{fw} \mathbf{Q}_k \mathbf{J}_{fw}^T \quad (2.54)$$

$$\mathbf{P}_{i,k|k-1} = \mathbf{J}_{fx} \mathbf{P}_{i,k-1|k-1} \mathbf{J}_{fx}^T + \mathbf{J}_{fw} \mathbf{Q}_{i,k} \mathbf{J}_{fw}^T \quad (2.55)$$

$$\mathbf{P}_{d,k|k-1} = \mathbf{P}_{k|k-1} - \mathbf{P}_{i,k|k-1} \quad (2.56)$$

avec

$$\mathbf{J}_{fx} = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}_{k-1|k-1}, \mathbf{u}_k, \mathbf{0}) \quad \mathbf{J}_{fw} = \frac{\partial f}{\partial \mathbf{w}}(\mathbf{x}_{k-1|k-1}, \mathbf{u}_k, \mathbf{0}). \quad (2.57)$$

Pour une observation $(\mathbf{z}_k, \mathbf{R}_{i,k}, \mathbf{R}_{d,k})$, l'étape de mise à jour correspond aux expressions

$$\mathbf{P}_1 = \omega^{-1} \mathbf{P}_{d,k|k-1} + \mathbf{P}_{i,k|k-1} \quad (2.58)$$

$$\mathbf{P}_2 = (1 - \omega)^{-1} \mathbf{R}_{d,k} + \mathbf{R}_{i,k} \quad (2.59)$$

$$\mathbf{K} = \mathbf{P}_1 \mathbf{J}_{hx}^T (\mathbf{J}_{hx} \mathbf{P}_1 \mathbf{J}_{hx}^T + \mathbf{J}_{hv} \mathbf{P}_2 \mathbf{J}_{hv}^T)^{-1} \quad (2.60)$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{h}(\mathbf{x}_{k|k-1}, \mathbf{0})) \quad (2.61)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K} \mathbf{J}_{hx}) \mathbf{P}_1 \quad (2.62)$$

$$\begin{aligned} \mathbf{P}_{i,k|k} &= (\mathbf{I} - \mathbf{K} \mathbf{J}_{hx}) \mathbf{P}_{i,k|k-1} (\mathbf{I} - \mathbf{K} \mathbf{J}_{hx})^T \\ &\quad + (\mathbf{K} \mathbf{J}_{hv}) \mathbf{R}_{i,k} (\mathbf{K} \mathbf{J}_{hv})^T \end{aligned} \quad (2.63)$$

$$\mathbf{P}_{d,k|k} = \mathbf{P}_{k|k} - \mathbf{P}_{i,k|k} \quad (2.64)$$

avec

$$\mathbf{J}_{hx} = \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}_{k|k-1}, \mathbf{0}) \quad \mathbf{J}_{hv} = \frac{\partial h}{\partial \mathbf{u}}(\mathbf{x}_{k|k-1}, \mathbf{0}). \quad (2.65)$$

Les matrices \mathbf{P}_1 et \mathbf{P}_2 font intervenir le coefficient ω mais uniquement sur la partie corrélée des

matrices de covariance. Ce filtre nécessite donc également une optimisation de ce paramètre afin de minimiser un critère à choisir. La partie indépendante de ces covariances est prise en compte comme dans un EKF classique.

2.4 Filtre particulaire

Les approches vues précédemment se basent sur le filtre de Kalman et modélisent chaque estimation par une loi normale. Ces modélisations, facile à manipuler, peuvent parfois ne pas suffir pour obtenir des estimations correctes. Le filtre particulaire (*particle filter*, PF), appelé aussi méthode de Monte-Carlo, est un outil permettant de calculer des estimations bayésiennes en utilisant un échantillonnage afin de modéliser au mieux la vraie densité de probabilité de chaque estimation. L'algorithme se base sur la génération d'un nombre N de particules $\{(\mathbf{x}_k^i, w_k^i), i \in \llbracket 0; N \rrbracket\}$ à l'instant k , possédant un vecteur de mesure \mathbf{x}_k^i et un poids w_k^i . Chacune de ces particules représente un échantillon ponctuel de densité de probabilité. L'objectif est alors d'obtenir une estimation de l'état à l'instant k en ayant connaissance de l'ensemble des observations $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$ effectuées jusqu'à cet instant. Une approximation de l'estimation *a posteriori* de l'état peut être obtenue en utilisant les différentes particules :

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=0}^N w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (2.66)$$

avec $\delta(\cdot)$, la fonction de Dirac. Cette approximation converge vers la vraie densité lorsque les particules sont bien réparties sur l'ensemble de définition et que le nombre N de particules est grand. Les différents filtres particuliers décrits dans la littérature, proposent des méthodes d'échantillonnage afin d'obtenir une bonne approximation en utilisant peu de ressources.

L'une des méthodes de filtrage particulaire les plus simples est l'échantillonnage séquentiel par importance (*sequential importance sampling*, SIS). La génération des particules à l'instant k se fait en utilisant une densité d'importance $q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ à définir. Cette fonction se base sur les particules de l'estimation précédente ainsi que sur la dernière observation obtenue. Les poids des nouvelles particules se calculent à partir de ceux des particules à l'instant précédent :

$$w_k^i = w_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k^i) p(\mathbf{x}_{k-1}^i | \mathbf{z}_{1:k})}{q(\mathbf{x}_k^i | \mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})}. \quad (2.67)$$

Les poids w_k^i obtenus doivent ensuite être normalisés pour respecter la condition $\sum_i w_k^i = 1$. Le choix optimal pour la densité d'importance q est lorsque tous les poids w_k^i ont la même valeur. Cela signifierait que les particules seraient très bien réparties. Pour avoir ce genre de répartition il faudrait échantillonner avec la densité $p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$, ce qui n'est pas envisageable puisqu'il

s'agit justement de la densité que l'on cherche à obtenir. Les approches de PF les plus simples utilisent la densité d'importance basée uniquement sur le modèle d'évolution :

$$q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k | \mathbf{x}_{k-1}^i). \quad (2.68)$$

Dans la pratique, le SIS est rarement utilisé car la solution diverge assez rapidement. Les particules ont tendance à se disperser et la majorité du poids de la densité se retrouve à être définie par une seule particule. Pour améliorer la répartition des particules, une étape de ré-échantillonnage peut être effectuée après le calcul des poids de l'instant k . Ces procédures sont appelées *sequential importance sampling and resampling* (SISR ou SIR). Le principe est de ré-échantillonner en utilisant l'estimation que l'on obtient à la fin d'une étape du SIS pour que les poids deviennent uniformes. Cette solution n'est cependant pas parfaite car elle s'appuie sur le choix de la densité d'importance. Si cette dernière génère une mauvaise approximation de la densité, le ré-échantillonnage risque de ne pas couvrir correctement l'espace d'état. Par exemple, si la majorité du poids est sur une seule particule, le SIR va manquer de diversité et va finir par diverger.

Le SIR est la base des approches modernes de filtrage particulaire. La principale problématique que ces nouvelles approches tentent de résoudre, est d'éviter la divergence de l'estimation. Les principales variations vont se faire sur le choix de la densité d'importance. Les solutions existantes se regroupent en deux familles :

- essayer de s'approcher le plus possible de l'estimation *a posteriori* correcte
- couvrir au mieux les zones importantes de l'espace d'état pour éviter la dégénérescence de l'estimation.

(WANG et al. 2017) présente les avancements récents du filtrage particulaire.

2.5 Mixture de gaussiennes

L'objectif des filtres basés sur une modélisation de l'état par une mixture de gaussienne est de pouvoir relâcher la contrainte des modèles gaussiens simples basées sur le filtre de Kalman. Cela permet par exemple de pouvoir modéliser l'état avec une loi multimodale. Ce genre de situation peut arriver lorsque les informations apportées ne permettent pas de résoudre une ambiguïté sur l'estimation. Les filtres particuliers permettent de répondre également à ce genre de problématique, mais les mixtures de gaussiennes ont l'avantage d'être moins coûteuses en temps de calcul.

Soit $p(\mathbf{x}_k | \mathbf{z}_{1:k})$, la densité de probabilité de l'estimation *a posteriori* de l'état à l'instant k . Soit $\{(w_k^i, \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i), i \in \llbracket 1; N \rrbracket\}$, un ensemble de N particules gaussiennes représentant cette esti-

mation. Les paramètres w_k^i , $\mathbf{x}_{k|k}^i$, $\mathbf{P}_{k|k}^i$ représentent le poids, l'espérance et la covariance de chaque échantillon gaussien. En utilisant cette modélisation, l'expression de la densité de l'estimation *a posteriori* peut être approximée par

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^i \mathcal{N}(\mathbf{x}_k; \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i). \quad (2.69)$$

Les poids doivent respecter la contrainte $\sum_i w_k^i = 1$. La modélisation de cette estimation est assez similaire à celle que l'on peut avoir dans un filtre particulaire. Néanmoins celle-ci a l'avantage d'avoir des éléments ayant une solution analytique.

2.5.1 Filtre à somme de gaussiennes

Le filtre à somme de gaussiennes (*Gaussian sum filter, GSF*) est présenté par (ANDERSON et MOORE 1979, p. 211, §8.4). Cette approche possède les mêmes contraintes qu'un EKF concernant les modèles d'évolution et d'observation, mais l'estimation de l'état à chaque instant est approximée par une mixture de gaussiennes. L'étape de prédiction consiste en l'application du modèle d'évolution $f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k)$ sur chaque échantillon gaussien représentant l'estimation précédente. Ce modèle utilise un vecteur de mesure \mathbf{u}_k ainsi qu'un vecteur de bruit \mathbf{w}_k supposé gaussien, centré, blanc et indépendant. La matrice \mathbf{Q}_k correspond à la covariance de ce bruit. Les échantillons gaussiens de l'estimation *a priori* obtenus ont pour expression :

$$\begin{aligned} \mathbf{x}_{k|k-1}^i &= f(\mathbf{x}_{k-1|k-1}^i, \mathbf{u}_k, \mathbf{0}) \\ \mathbf{P}_{k|k-1}^i &= \mathbf{J}_{f_x} \mathbf{P}_{k-1|k-1}^i \mathbf{J}_{f_x}^\top + \mathbf{J}_{f_w} \mathbf{Q}_k \mathbf{J}_{f_w}^\top \end{aligned} \quad (2.70)$$

avec

$$\mathbf{J}_{f_x} = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}_{k-1|k-1}^i, \mathbf{u}_k, \mathbf{0}), \quad \mathbf{J}_{f_w} = \frac{\partial f}{\partial \mathbf{w}}(\mathbf{x}_{k-1|k-1}^i, \mathbf{u}_k, \mathbf{0}). \quad (2.71)$$

L'étape de mise à jour utilise le modèle d'observation $h(\mathbf{x}_k, \mathbf{v}_k)$ et une observation \mathbf{z}_k . Comme pour le modèle d'évolution, le vecteur de bruit \mathbf{v}_k de mesure est considéré comme gaussien, centré, blanc et indépendant, et dont la covariance est représentée par \mathbf{R}_k . Chaque échantillon gaussien $(w_k^i, \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i)$ est calculé de la façon suivante :

$$\begin{aligned} \mathbf{Y}_k &= \mathbf{J}_{h_x} \mathbf{P}_{k|k-1}^i \mathbf{J}_{h_x}^\top + \mathbf{J}_{h_v} \mathbf{R}_k \mathbf{J}_{h_v}^\top \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}^i \mathbf{J}_{h_x}^\top \mathbf{Y}_k^{-1} \\ \mathbf{x}_{k|k}^i &= \mathbf{x}_{k|k-1}^i + \mathbf{K}_k (\mathbf{z}_k - h(\mathbf{x}_{k|k-1}^i, \mathbf{0})) \\ \mathbf{P}_{k|k}^i &= \mathbf{P}_{k|k-1}^i - \mathbf{K}_k \mathbf{J}_{h_x} \mathbf{P}_{k|k-1}^i \\ w_k^i &= c^{-1} w_{k-1}^i \mathcal{N}(\mathbf{z}_k; h(\mathbf{x}_{k|k-1}^i, \mathbf{0}), \mathbf{Y}_k) \end{aligned} \quad (2.72)$$

avec

$$\mathbf{J}_{hx} = \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}_{k|k-1}^i, \mathbf{0}), \quad \mathbf{J}_{hv} = \frac{\partial h}{\partial \mathbf{w}}(\mathbf{x}_{k|k-1}^i, \mathbf{0}), \quad c = \sum_{i=0}^N w_{k-1}^i \mathcal{N}(\mathbf{z}_k; h(\mathbf{x}_{k|k-1}^i, \mathbf{0}), \mathbf{Y}_k). \quad (2.73)$$

Bien que cet algorithme puisse approximer une estimation quelconque de l'état, il reste cependant limité à des modèles d'évolution et d'observation facilement linéarisables. En effet, les échantillons gaussiens sont soumis aux mêmes problématiques qu'un EKF. La non-linéarité peut provoquer une divergence des estimations et la mixture de gaussiennes risque de s'éloigner de la densité réelle. Puisque le nombre d'échantillons n'évolue pas, cet algorithme n'est également pas adapté aux systèmes dont la distribution de l'estimation se complexifie avec le temps, mais ce genre de situation se produit essentiellement lorsque les modèles sont non-linéaires.

2.5.2 Filtre particulière à somme de gaussiennes

Les contraintes de l'algorithme présenté précédemment restent importantes. (KOTECHA et DJURIC 2003) proposent des variations de cet algorithme pour enlever les contraintes limitant les systèmes fortement non-linéaires. Leur version 2 du filtre particulière à somme de gaussiennes (*gaussian sum particle filter*, GSPF-II) utilise une approche de filtre particulière pour l'estimation de la mixture de gaussiennes représentant l'état. Chaque échantillon gaussien est calculé en utilisant un filtre particulière gaussien. Les densités de probabilité des modèles d'évolution et d'observation sont également considérées comme quelconques et sont approximées par des mixtures de gaussiennes.

Pour l'étape de prédiction, l'objectif est d'obtenir une approximation de $p(\mathbf{x}_k | \mathbf{z}_{1:k-1})$ sous forme d'une mixture de gaussiennes. Pour cela, on considère que l'on connaît l'estimation de l'état à l'instant $k - 1$:

$$p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) = \sum_{i=1}^N w_{k-1}^i \mathcal{N}(\mathbf{x}_{k-1}; \mathbf{x}_{k-1|k-1}^i, \mathbf{P}_{k-1|k-1}^i). \quad (2.74)$$

Le bruit du modèle d'évolution est également considéré comme non-gaussien et modélisé comme une mixture de gaussiennes composée de J échantillons gaussiens :

$$p(\mathbf{w}_k) = \sum_{j=1}^J \alpha_k^j \mathcal{N}(\mathbf{w}_k; \mathbf{w}_k^j, \mathbf{Q}_k^j). \quad (2.75)$$

Ces deux mixtures sont combinées afin d'obtenir l'estimation *a priori* à l'instant k sous forme d'une mixture de $N \times J$ échantillons. Le calcul de ces échantillons se fait donc à partir des

couples d'échantillons

$$\left\{ \left((w_{k-1}^i, \mathbf{x}_{k-1|k-1}^i, \mathbf{P}_{k-1|k-1}^i), (\alpha_k^j, \mathbf{w}_k^j, \mathbf{Q}_k^j) \right), (i, j) \in \llbracket 1; N \rrbracket \times \llbracket 1; J \rrbracket \right\} \quad (2.76)$$

Dans une approche basée sur l'EKF, les échantillons de l'estimation *a priori* sont obtenus en calculant l'équation (2.70) avec chacun de ces couples. L'algorithme complet est présenté sous le nom de GSF-II par (KOTECHA et DJURIC 2003).

Dans le cas du GSPF-II, l'estimation *a priori* est obtenue par une méthode de Monte-Carlo. Un ensemble $\{\mathbf{x}_{k-1}^{i,m}, m \in \llbracket 1; M \rrbracket\}$ contenant M particules est généré en suivant la distribution $\mathcal{N}(\mathbf{x}_{k-1|k-1}^i, \mathbf{P}_{k-1|k-1}^i)$ de chaque échantillon gaussien de l'estimation *a posteriori* de l'état précédent. Les nouveaux ensembles de particules $\{\mathbf{x}_k^{r,m}, m \in \llbracket 1; M \rrbracket\}$ sont ensuite générés grâce à la fonction d'évolution $f(\mathbf{x}_{k-1}^{i,m}, \mathbf{u}_k, \mathbf{w}_k^j)$ pour chaque couple $(i, j) \in \llbracket 1; N \rrbracket \times \llbracket 1; J \rrbracket$, avec $r \in \llbracket 1; NJ \rrbracket$ et $r = i + (j - 1)J$. Chaque échantillon r parmi les $N \times J$ gaussiennes de l'estimation *a priori* a alors pour expression :

$$\begin{aligned} \mathbf{x}_{k|k-1}^r &= \frac{1}{M} \sum_{m=1}^M \mathbf{x}_k^{r,m} \\ \mathbf{P}_{k|k-1}^r &= \frac{1}{M} \sum_{m=1}^M (\mathbf{x}_k^{r,m} - \mathbf{x}_{k|k-1}^r) (\mathbf{x}_k^{r,m} - \mathbf{x}_{k|k-1}^r)^\top \\ w_{k|k-1}^r &= w_{k-1}^i \alpha_k^j \left(\sum_{i=1}^N \sum_{j=1}^J w_{k-1}^i \alpha_k^j \right)^{-1}. \end{aligned} \quad (2.77)$$

L'étape de mise à jour se base sur l'utilisation d'une densité d'importance $q(\mathbf{x}_k | \mathbf{z}_{1:k})$ à définir en fonction de l'application. Cette fonction est calculée pour chaque échantillon $r \in \llbracket 1; NJ \rrbracket$ de l'estimation *a priori*, afin de générer des ensembles $\{(w_k^{r,m}, \mathbf{x}_k^{r,m}), m \in \llbracket 1; M \rrbracket\}$ de particules pondérées. Chaque poids est calculé de la façon suivante :

$$w_k^{r,m} = \frac{p(\mathbf{z}_k | \mathbf{x}_k^{r,m}) \mathcal{N}(\mathbf{x}_k^{r,m}; \mathbf{x}_{k|k-1}^r, \mathbf{P}_{k|k-1}^r)}{q(\mathbf{x}_k^{r,m} | \mathbf{z}_{1:k})} \quad (2.78)$$

Le calcul de l'échantillon $r \in \llbracket 1; NJ \rrbracket$ de l'estimation *a posteriori* de l'état correspond aux

expressions

$$\begin{aligned}
c &= \sum_{m=1}^M w_k^{r,m} \\
\mathbf{x}_{k|k}^r &= c^{-1} \sum_{m=1}^M w_k^{r,m} \mathbf{x}_k^{r,m} \\
\mathbf{P}_{k|k}^r &= c^{-1} \sum_{m=1}^M w_k^{r,m} (\mathbf{x}_k^{r,m} - \mathbf{x}_{k|k}^r) (\mathbf{x}_k^{r,m} - \mathbf{x}_{k|k}^r)^\top \\
w_{k|k}^r &\propto c w_{k|k-1}^r \left(\sum_{i=1}^N \sum_{j=1}^J w_k^{i,j} \right)^{-1}.
\end{aligned} \tag{2.79}$$

Les poids $w_{k|k}^r$ doivent être normalisés pour respecter la contrainte $\sum_r w_{k|k}^r = 1$. À cette étape de l'algorithme, le nombre d'échantillons gaussiens vaut $N \times J$. Pour éviter d'augmenter ce nombre en permanence et ralentir l'exécution, il est nécessaire d'effectuer un ré-échantillonnage afin de se limiter à N échantillons. La méthode la plus simple consiste en la conservation des N échantillons ayant les poids les plus importants parmi les $N \times J$ de l'estimation *a posteriori*.

Le GSPF-II est un algorithme générique comme peut l'être le SISR dans la catégorie des filtres particuliers. Les choix de la densité d'importance et des méthodes d'échantillonnages des particules utilisées dans l'étape de prédiction et de mise à jour, sont laissés libres à l'utilisateur. Puisque la plupart des distributions est facilement modélisable par des mixtures de gaussiennes, ce filtre peut être utilisé pour des problématiques équivalentes à celle que l'on peut avoir lorsque l'on utilise un filtre particulier classique. Cet algorithme a alors l'avantage de pouvoir effectuer ses opérations en utilisant moins de particules, et par conséquent, en utilisant moins de ressources machines.

2.6 Méthodes ensemblistes

L'estimation de l'état d'un système en utilisant une approche ensembliste est assez éloignée des méthodes que nous avons vu jusqu'à maintenant. Ce genre d'approche se base uniquement sur le support des densités de probabilité. Pour une densité $p(\mathbf{x})$, le support correspond à l'ensemble $\{\mathbf{x} \in \mathbb{R}^n, p(\mathbf{x}) \neq 0\}$. Dans le cas d'une loi normale, le support est infini, mais la majorité du poids de la densité se trouve autour de son espérance. Pour qu'une estimation basée sur un filtre gaussien soit valide, il faut que la valeur vraie que l'on cherche à estimer reste à proximité de l'espérance. Dans une approche ensembliste, les supports des densités de probabilité utilisées sont finis. C'est le cas par exemple des lois uniformes, dont la définition prend directement en paramètre le support. L'utilisation de lois uniformes n'est cependant pas une bonne solution pour l'estimation de l'état, car une combinaison de variables aléatoires uniformes donne rarement une solution uniforme. Par exemple, si A et B sont des variables uniformes, la densité

d'une variable aléatoire $C = A + B$ correspondrait au produit de convolution $p(C) = p(A) * p(B)$. L'addition de variables aléatoires uniformes génère donc une densité non uniforme. Le support de $p(C)$ est cependant fini et calculable dans cet exemple.

L'objectif des approches ensemblistes est donc de pouvoir donner une approximation du support de l'état en garantissant que la valeur vraie soit toujours incluse. Les modèles d'évolution ou d'observation n'ont pas de contrainte de linéarité et les mesures utilisées ne nécessitent aucune information sur leur dépendance avec l'état. Ces algorithmes ont donc l'avantage de pouvoir obtenir des estimations dont la validité est prouvée mathématiquement. Une erreur sur l'estimation de l'état signifie alors une erreur de modélisation ou de mesure.

2.6.1 Modélisation des estimations

Une estimation d'une variable aléatoire suivant une loi quelconque \mathbf{x} est représentée par un ensemble \mathbb{X} également quelconque. L'une des premières difficultés de manipuler des ensembles est de trouver une représentation utilisable algorithmiquement. La solution utilisée habituellement dans les approches ensemblistes est la représentation par *recouvrement*. Cela consiste en l'utilisation d'une classe de sous-ensemble de \mathbb{R}^n que l'on sait manipuler aisément. Les éléments de cette classe sont appelés des *réceptifs*. Il existe par exemple les ellipsoïdes, polytopes, ou encore zonotopes. La classe présentée ici est celle des pavés. Des explications plus détaillées sur le calcul ensembliste en utilisant des pavés sont présentées par (JAULIN 2000).

Un pavé $[\mathbf{x}]$ de dimension n se construit simplement en calculant le produit cartésien d'intervalles sur chacune de ses dimensions :

$$[\mathbf{x}] = [x_1; \bar{x}_1] \times \cdots \times [x_n; \bar{x}_n]. \quad (2.80)$$

La figure 2.4 présente un exemple de pavé $[\mathbf{x}]$ de dimension 2 recouvrant un ensemble \mathbb{X} . Ce pavé $[\mathbf{x}]$ est un recouvrement de \mathbb{X} si $\mathbb{X} \subset [\mathbf{x}]$. L'ensemble \mathbb{X} peut également être représenté par un *encadrement* de deux réceptifs \mathbb{X}^- et \mathbb{X}^+ s'ils respectent la contrainte $\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+$. Cela permet d'obtenir facilement des informations utiles sur \mathbb{X} (non vide, non connexe, ...). La longueur $w([\mathbf{x}])$ d'un pavé correspond à la longueur du plus grand de ses côtés. Nous pouvons noter que ce pavé n'est pas minimal. Il est en effet possible de construire un pavé $[\mathbf{y}] \neq [\mathbf{x}]$ tel que $\mathbb{X} \subset [\mathbf{y}] \subset [\mathbf{x}]$.

Pour améliorer la précision de l'estimation d'un pavé $[\mathbf{x}]$, il est possible de le découper en *sous-pavage* correspondant à une union de sous-pavés de $[\mathbf{x}]$. Si un sous-pavage couvre entièrement $[\mathbf{x}]$, on l'appelle alors *pavage* de $[\mathbf{x}]$. Les algorithmes permettant de construire un pavage se base sur l'opération de bisection. Une bisection de $[\mathbf{x}]$ correspond à la création d'un couple de pavés $(L[\mathbf{x}], R[\mathbf{x}])$ en coupant $[\mathbf{x}]$ à la moitié de sa longueur. En répétant des

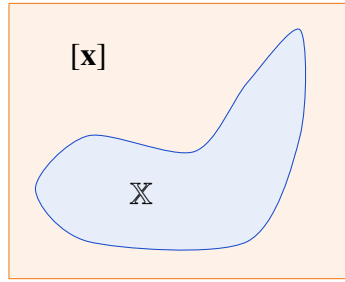


FIGURE 2.4 – Exemple de pavé $[x]$ recouvrant un ensemble non convexe X de dimension 2

bissections successives et en ne conservant que les sous-pavés dont l'intersection avec X est non nul, il est alors possible d'obtenir un pavage dont l'approximation est aussi proche que l'on veut de X . En effet, la limitation de la précision sera définie par le choix de la longueur minimale d'un pavé pouvant être bissecté. La figure 2.5 représente trois pavages avec différents niveaux de précision, dont l'ensemble à estimer est l'anneau $\{(x, y), x^2 + y^2 \in [1; 2]\}$.

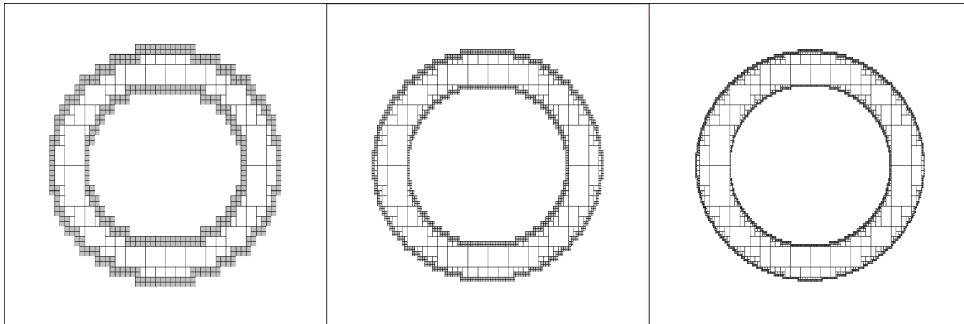


FIGURE 2.5 – Recouvrement d'un anneau par 3 différentes résolutions de pavage. Les boîtes grisées sont partiellement recouvertes par l'anneau. Source : (JAULIN 2000, fig. 2.3)

2.6.2 Analyse par intervalles

L'un des objectifs du calcul ensembliste est le calcul de transformations d'ensembles par des fonctions. En utilisant une représentation des ensembles par des pavés le domaine mathématique impliqué est l'analyse par intervalles. Soit $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ une fonction. Il est possible de définir le calcul de l'image Y de l'ensemble X par f de façon directe ou indirecte :

$$\begin{aligned} \mathbf{f}(X) &\triangleq \{\mathbf{f}(\mathbf{x}), \mathbf{x} \in X\} \\ \mathbf{f}^{-1}(Y) &\triangleq \{\mathbf{x} \in \mathbb{R}^n, \exists \mathbf{y} \in Y, \mathbf{f}(\mathbf{x}) = \mathbf{y}\}. \end{aligned} \quad (2.81)$$

Il faut noter que la fonction réciproque $\mathbf{f}^{-1}(\mathbf{y})$ n'est pas impliquée dans le calcul. La fonction \mathbf{f} n'a donc pas besoin d'être bijective sur ces ensembles. Le calcul ensembliste se limite cependant à une combinaison d'opérations élémentaires (+, ×, sin, exp, sqrt, inv, max, ...). La division et la soustraction ne sont pas prises en compte mais peuvent être obtenues en effectuant des opérations avec les fonctions élémentaires inv et neg. Cela permet d'éviter de devoir gérer les

opérateurs non commutatifs.

Bien que l'on sache représenter un pavé $[\mathbf{x}]$, l'image de celui-ci $\mathbb{Y} = \mathbf{f}([\mathbf{x}])$ n'est pas représentable puisqu'il ne correspond presque jamais à un pavé. Il est cependant possible de définir ce que l'on appelle une fonction d'inclusion $[\mathbf{f}]([\mathbf{x}])$ dont le résultat correspond à un pavé. Cette fonction respecte la contrainte

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \quad \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}]) \quad (2.82)$$

avec \mathbb{IR}^n , l'ensemble des pavés de \mathbb{R}^n . Pour chaque fonction élémentaire $\mathbf{e}([\mathbf{x}])$, les fonctions d'inclusions $[\mathbf{e}]([\mathbf{x}])$ et $[\mathbf{e}^{-1}]([\mathbf{x}])$ sont connues. Il en est de même pour les opérations d'addition $[+]$ et de multiplication $[\times]$. Les fonctions d'inclusions de fonctions plus complexes peuvent donc facilement être obtenues en découpant leur expression en opérations élémentaires. Prenons l'exemple de la fonction

$$f(\mathbf{x}) = \exp(x_1) + x_2 \sin(x_3) \quad (2.83)$$

avec $\mathbf{x} = (x_1, x_2, x_3)^\top$. Cette fonction peut se décomposer en opérations élémentaires comme présenté dans la figure 2.6. Il est alors possible de calculer des pavés intermédiaires en utilisant des fonctions d'inclusions connues. En reprenant l'exemple, le calcul de $[y] = [f]([\mathbf{x}])$ peut s'écrire sous la forme du système

$$\begin{aligned} [y] &= [s_1] [+] [s_2] \\ [s_1] &= [\exp]([x_1]) \\ [s_2] &= [x_2] [\times] [s_3] \\ [s_3] &= [\sin]([x_3]) \end{aligned} \quad (2.84)$$

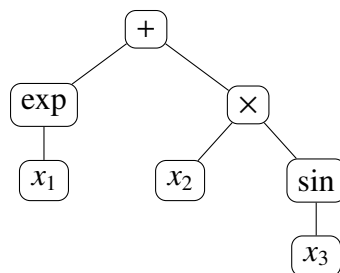


FIGURE 2.6 – Arbre dont les nœuds représentent les opérations élémentaires d'une fonction $f(\mathbf{x}) = \exp(x_1) + x_2 \sin(x_3)$.

Le calcul des images directes et indirectes d'un pavé par une fonction peut être utilisé pour appliquer des contraintes sur un ensemble. Le contracteur par propagation-rétropropagation $C_{\downarrow\uparrow}(\mathbf{f}, [\mathbf{x}])$ permet d'obtenir un pavé $[\mathbf{x}'] \subset [\mathbf{x}]$ qui minimise l'ensemble $\{\mathbf{f}(\mathbf{x}) = \mathbf{0}, \forall \mathbf{x} \in [\mathbf{x}]\}$. L'application de contraintes sur un ensemble ne peut donc que réduire sa taille. L'algorithme du contracteur est le suivant :

1. $[\mathbf{y}] := \mathbb{R}^n$
2. mettre à jour $[\mathbf{y}] := [\mathbf{f}](\mathbf{x}) \cap [\mathbf{y}]$ en utilisant les fonctions d'inclusion des opérations élémentaires
3. mettre à jour $[\mathbf{x}] := [\mathbf{f}^{-1}](\mathbf{y}) \cap [\mathbf{x}]$ en utilisant les fonctions d'inclusion inverses des opérations élémentaires
4. si la réduction de $[\mathbf{x}]$ n'est pas négligeable, revenir à l'étape 2
5. retourner $[\mathbf{x}]$

L'application de la réduction est faite en boucle dans l'algorithme, car le calcul des sous-contraintes représentées par les opérations élémentaires ne donne pas directement le résultat optimal.

2.6.3 Estimation d'états par méthode ensembliste

L'objectif de l'estimation de l'état d'un système en utilisant une méthode ensembliste est d'obtenir un ensemble représentant au mieux les valeurs possibles de cet état à un instant donné. Pour que l'approximation soit précise, l'état est représenté par un sous-pavage \mathcal{X} recouvrant l'ensemble support de cet état. Dans un filtre bayésien, l'évolution et la mise à jour du système utilise les équations 2.1 et 2.2 que l'on rappelle ici :

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \quad (2.85)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{v}_k) \quad (2.86)$$

afin de calculer une approximation de \mathbf{x}_k à l'instant k . En utilisant l'approche ensembliste, l'obtention de cet état consiste à calculer l'image de l'état précédent par la fonction d'évolution \mathbf{f} et ensuite de lui appliquer la contrainte représentée par l'observation \mathbf{z}_k . Cette observation est modélisée ici par un sous-pavage \mathcal{Y} . Le calcul de l'état à l'instant k correspond à l'expression suivante :

$$\mathcal{X}_k = \mathbf{f}(\mathcal{X}_{k-1}, \mathcal{U}_k) \cap \mathbf{h}^{-1}(\mathcal{Z}_k) \quad (2.87)$$

avec \mathcal{U}_k , un sous-pavage représentant le support du modèle d'évolution et \mathcal{Z}_k , un sous-pavage représentant le support d'une observation. Ce calcul nécessite d'être capable d'obtenir une approximation de l'image directe ou indirecte d'un sous-pavage par une fonction.

Le calcul de l'inversion ensembliste $\mathbf{h}^{-1}(\mathcal{Z}_k)$ peut être obtenu en utilisant l'algorithme SIVIA (*set inversion via interval analysis*). L'objectif de l'algorithme SIVIA(\mathbf{f} , $[\mathbf{x}]$, \mathbb{Y} , ϵ) est d'obtenir un sous-pavage \mathcal{X} tel que $\mathbf{f}^{-1}(\mathbb{Y}) \subset \mathcal{X}$. Le paramètre $[\mathbf{x}]$ doit être un pavé suffisamment grand pour qu'il englobe l'ensemble à obtenir. Il est utilisé pour créer le sous-pavage \mathcal{X} par bisections successives. Le scalaire ϵ permet de définir la longueur minimale des pavés contenus dans \mathcal{X} . Voici l'algorithme SIVIA(\mathbf{f} , $[\mathbf{x}]$, \mathbb{Y} , ϵ) :

1. si $[\mathbf{f}]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset$, retourner \emptyset
2. si $[\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y}$ ou $w([\mathbf{x}]) < \epsilon$, retourner $[\mathbf{x}]$
3. bissecter $[\mathbf{x}]$ en $L[\mathbf{x}]$ et $R[\mathbf{x}]$
4. retourner $\text{SIVIA}(\mathbf{f}, L[\mathbf{x}], \mathbb{Y}, \epsilon) \cup \text{SIVIA}(\mathbf{f}, R[\mathbf{x}], \mathbb{Y}, \epsilon)$

L'algorithme ayant une double récursion, il peut parfois prendre beaucoup de temps de calcul lorsque les ensembles à estimer ont une forme complexe et dont la taille est grande par rapport à ϵ . C'est pourquoi il existe des variantes ajoutant une étape de contraction de $[\mathbf{x}]$ en début d'algorithme ou après l'étape 3.

Le calcul de l'image directe correspond au calcul d'un sous-pavage \mathcal{Y} tel que $\mathbf{f}(\mathcal{X}) \subset \mathcal{Y}$ avec \mathcal{X} , un sous-pavage connu. Cette opération est plus compliquée que le calcul de l'image réciproque puisqu'elle nécessite d'effectuer le test de la présence d'un élément $\mathbf{y} \in \mathbf{f}(\mathbb{X})$, or l'évaluation de \mathbf{f} est justement l'objectif que nous souhaitons obtenir. L'algorithme $\text{ImageSP}(\mathbf{f}, \mathcal{X}, \epsilon)$ permet d'obtenir le sous-pavage \mathcal{Y} en trois étapes : le hachage, l'évaluation et la régularisation.

1. Le hachage consiste à découper \mathcal{X} en un sous-pavage \mathcal{X}_ϵ dont les pavés ont tous une longueur inférieure à ϵ .
2. L'évaluation correspond à la construction du sous-pavage \mathcal{Y}_ϵ pour lequel, chacun de ses pavés $[\mathbf{y}_i] = [\mathbf{f}]([\mathbf{x}_i])$ avec $[\mathbf{x}_i]$ un pavé de \mathcal{X}_ϵ .
3. La régularisation permet d'obtenir le sous-pavage \mathcal{Y} en reconstruisant des pavés de tailles raisonnables. Celui-ci est obtenu en utilisant l'algorithme SIVIA avec la fonction identité $\text{Id}(\cdot)$ afin de respecter la condition $\mathbf{f}(\mathcal{X}) \subset \text{Id}^{-1}(\mathcal{Y})$.

Nous avons donc $\mathcal{Y} = \text{SIVIA}(\text{Id}, [\mathbf{f}]([\mathbf{x}]), \mathcal{Y}_\epsilon, \epsilon)$.

L'étape de hachage permet de régler la précision que nous souhaitons obtenir en réglant la valeur de ϵ . La précision est meilleure lorsque ϵ est petit mais le nombre de calculs augmente.

En utilisant ces deux algorithmes, l'équation (2.87), correspondant au calcul de l'état à l'instant k par une méthode ensembliste, devient :

$$\mathcal{X}_k = \text{ImageSP}(\mathbf{f}, \mathcal{X}_{k-1} \times \mathcal{U}_k, \epsilon) \cap \text{SIVIA}(\mathbf{h}, \mathcal{Z}_k, \epsilon). \quad (2.88)$$

L'algorithme ImageSP est l'étape la plus lente du calcul, mais il est cependant possible de l'optimiser sous certaines conditions. En effet, si la fonction \mathbf{f} est bijective, il est alors possible de remplacer le calcul d'une image ensembliste par le calcul d'une inversion ensembliste moins coûteux. La prédiction de l'état peut alors être obtenue en utilisant SIVIA avec la fonction \mathbf{f}^{-1} . Sous cette condition, le calcul global peut donc être réécrit de la façon suivante :

$$\mathcal{X}_k = \text{SIVIA}(\mathbf{f}^{-1}, \mathcal{X}_{k-1} \times \mathcal{U}_k, \epsilon) \cap \text{SIVIA}(\mathbf{h}, \mathcal{Z}_k, \epsilon). \quad (2.89)$$

2.7 Outils de qualification des estimations

2.7.1 Représentation 2D d'une densité de probabilité Gaussienne

Lorsque l'on souhaite représenter l'incertitude d'une estimation gaussienne d'une variable à 1 dimension, il est possible d'utiliser un intervalle autour de l'espérance de cette loi normale. Il n'est pas possible d'inclure la totalité de la densité de probabilité dans l'intervalle car une loi normale s'étend à l'infini. Dans la pratique, cet intervalle est souvent défini autour de l'espérance avec une distance correspondant à un multiple de l'écart-type. Par exemple, il est possible de définir l'intervalle à 1 écart-type, ou encore l'intervalle contenant 99 % de la densité.

Dans le cas d'une estimation par une loi normale de dimension 2, cet intervalle prend la forme d'une ellipse. Cette ellipse représente l'ensemble des points ayant une même valeur de densité de probabilité. Cela revient à prendre l'intersection entre la courbe gaussienne et un plan parallèle à l'origine. La figure 2.7 montre un exemple de construction d'une ellipse pour les points distants d'un écart-type (ou 1 sigma) de l'espérance. L'espérance se trouve toujours au centre de l'ellipse. Dans le cas d'une loi normale de dimension 3, la représentation de cette loi dans l'espace prend la forme d'un ellipsoïde.

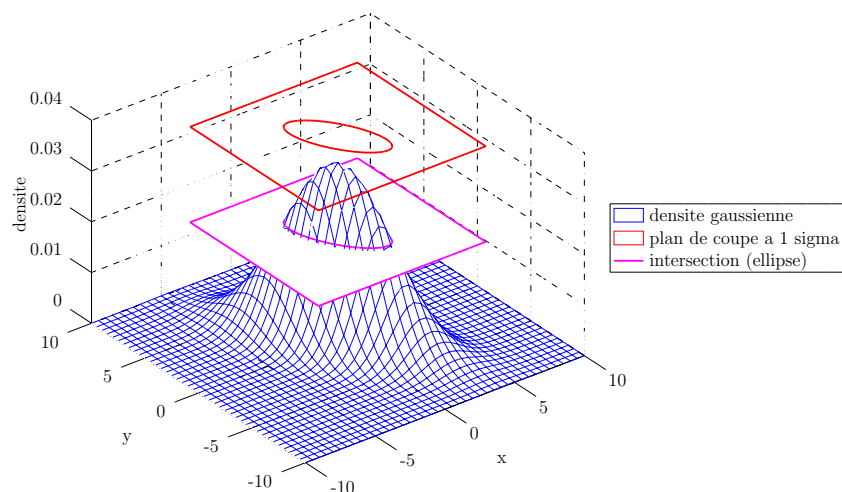


FIGURE 2.7 – Représentation d'une densité de probabilités d'une loi normale de dimension 2 par une ellipse. Cette ellipse représente une coupe à 1 écart-type de la densité.

Le choix de la taille de l'ellipse (ou ellipsoïde) utilisée pour les représentations des lois normales est laissée libre à l'utilisateur. Il est par exemple possible de dessiner une ellipse à 1 sigma ou un autre multiple de l'écart-type, mais il est également possible de dessiner l'ellipse contenant un certain pourcentage de la densité de probabilité. La fonction de répartition de la loi du χ^2 permet de calculer le pourcentage de la densité que contient l'ellipse à un certain multiple de l'écart-type. La réciproque permet de faire l'opération inverse. Par exemple, l'ellipse à 1 sigma contient 39 % de la densité de probabilité, et l'ellipse à 99 % correspond à l'ellipse à 3.03

sigma. La figure 2.8 montre un exemple de représentation de ces deux ellipses.

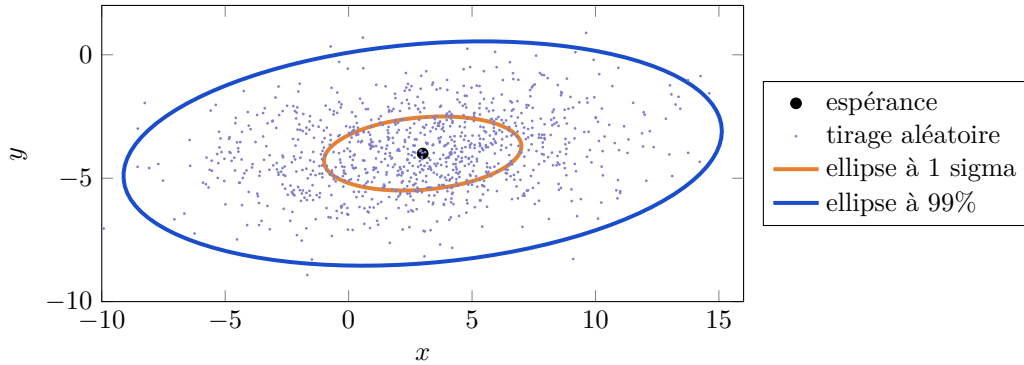


FIGURE 2.8 – Exemple d’ellipses représentant une même loi normale à 1 sigma et 99 % de la densité. Les points sont des tirages aléatoires suivant cette loi.

2.7.2 Réduction de mixture standard

Les algorithmes de réduction de mixture de gaussiennes servent à limiter le nombre de particules tout en essayant de conserver une bonne approximation de la densité de probabilité de l’estimation. Ces algorithmes se basent sur la fusion d’un ensemble de gaussiennes en une unique gaussienne ayant une densité de probabilité similaire. La méthode de fusion qui est la plus utilisée est celle que l’on peut retrouver dans l’algorithme de réduction de (RUNNALLS 2007). Ce papier décrit la réduction de mixture standard, ou *standard mixture reduction* (SMR) en anglais, dans le cas où l’ensemble des particules à fusionner est composé de deux gaussiennes

$$\{(w_1, \mathbf{x}_1, \mathbf{P}_1), (w_2, \mathbf{x}_2, \mathbf{P}_2)\} \quad (2.90)$$

avec w_1 le poids de la particule, \mathbf{x}_1 l’espérance et \mathbf{P}_1 la covariance de la gaussienne. Le résultat de cette fusion permet d’obtenir une unique particule $(w_m, \mathbf{x}_m, \mathbf{P}_m)$ préservant les moments d’ordre 0 et 1 de la mixture représentant cet ensemble de particules. Les équations permettant de calculer cette nouvelle particule sont les suivantes :

$$w_m = w_1 + w_2 \quad (2.91)$$

$$\mathbf{x}_m = w_1 \mathbf{x}_1 + w_2 \mathbf{x}_2 \quad (2.92)$$

$$\mathbf{P}_m = w_1 \mathbf{P}_1 + w_2 \mathbf{P}_2 + \frac{w_1 w_2}{w_1 + w_2} (\mathbf{x}_1 - \mathbf{x}_2)(\mathbf{x}_1 - \mathbf{x}_2)^\top. \quad (2.93)$$

La figure 2.9 représente le résultat de la SMR dans le cas de particules de dimension 1 et 2.

La SMR ne se limite pas uniquement aux ensembles de deux particules. Il est en effet possible de fusionner N particules :

$$\{(w_i, \mathbf{x}_i, \mathbf{P}_i), i \in \llbracket 1; N \rrbracket\} \quad (2.94)$$

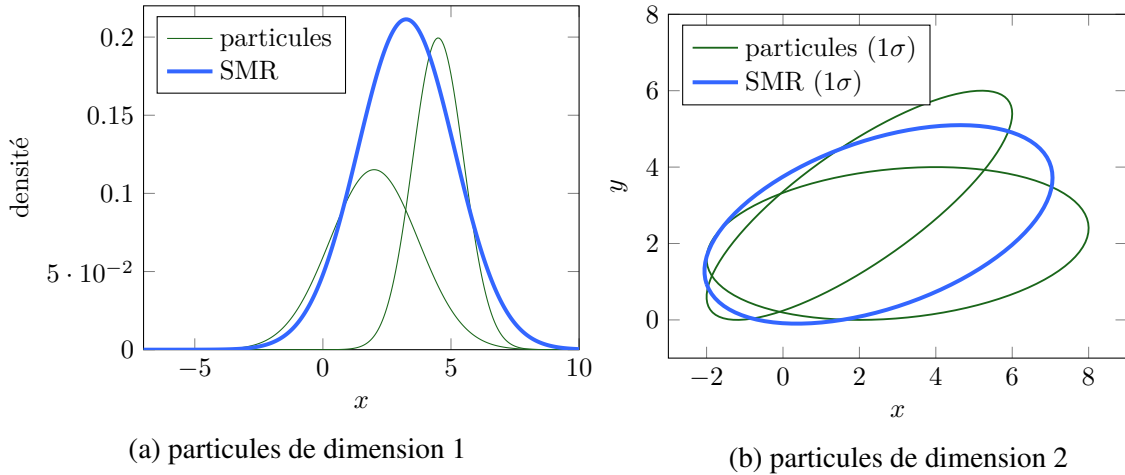


FIGURE 2.9 – Illustration des gaussiennes obtenues après la fusion de deux gaussiennes par la SMR dans le cas de particules de dimension 1 et 2.

La formule revient alors à calculer la gaussienne $(w_m, \mathbf{x}_m, \mathbf{P}_m)$ correspondant au poids, à l'espérance et à la covariance de la mixture représentant ces N particules. Ces formules prennent la forme suivante :

$$w_m = \sum_{i=1}^N w_i \quad (2.95)$$

$$\mathbf{x}_m = \frac{1}{w_m} \sum_{i=1}^N w_i \mathbf{x}_i \quad (2.96)$$

$$\mathbf{P}_m = \frac{1}{w_m} \sum_{i=1}^N w_i (\mathbf{x}_i - \mathbf{x}_m)(\mathbf{x}_i - \mathbf{x}_m)^T. \quad (2.97)$$

2.7.3 Union de covariance

Comme la SMR, l'union de covariance ou *covariance union* (CU) en anglais, est une méthode de fusion d'hypothèses gaussiennes permettant la réduction de la taille de la mixture. Cette approche, présentée par (S. J. JULIER, J. K. UHLMANN et NICHOLSON 2004), est utilisée pour construire une particule gaussienne représentant un ensemble de deux ou plusieurs gaussiennes. Dans la CU, chaque hypothèse gaussienne à fusionner est considérée comme une alternative possible pour l'estimation de l'état. Sans information complémentaire permettant de déterminer quelle modélisation de l'état est la bonne, la CU construit une gaussienne permettant de conserver l'intégrité de l'estimation. Cela signifie que le résultat de la CU permet d'obtenir une unique hypothèse gaussienne valide sans avoir à résoudre le problème du choix de la bonne hypothèse dans le groupe à fusionner.

Supposons que l'on souhaite fusionner un ensemble de deux particules $\{(\mathbf{x}_1, \mathbf{P}_1), (\mathbf{x}_2, \mathbf{P}_2)\}$. L'information de pondération des particules est considérée comme inconnue car chaque parti-

cule correspond à une hypothèse sur l'état du robot pour laquelle sa corrélation avec les autres hypothèses n'est pas prise en compte. La CU permet d'obtenir une particule $(\mathbf{x}_u, \mathbf{P}_u)$ qui respecte les contraintes :

$$\mathbf{P}_u \geq \mathbf{U}_1, \quad \mathbf{P}_u \geq \mathbf{U}_2 \quad (2.98)$$

avec

$$\mathbf{U}_1 = \mathbf{P}_1 + (\mathbf{x}_u - \mathbf{x}_1)(\mathbf{x}_u - \mathbf{x}_1)^\top \quad (2.99)$$

$$\mathbf{U}_2 = \mathbf{P}_2 + (\mathbf{x}_u - \mathbf{x}_2)(\mathbf{x}_u - \mathbf{x}_2)^\top. \quad (2.100)$$

pour toute particule i . (S. J. JULIER, J. K. UHLMANN et NICHOLSON 2004) propose une solution permettant de calculer \mathbf{P}_u respectant ces contraintes. Soit $\mathbf{S} = \text{chol}(\mathbf{U}_2)$, le triangle supérieur de la décomposition de Cholesky. Soient \mathbf{V} et \mathbf{D} , les matrices des vecteurs propres et des valeurs propres de $\mathbf{S}^{-\top} \mathbf{U}_1 \mathbf{S}^{-1}$. La matrice de covariance \mathbf{P}_u est obtenue par :

$$\mathbf{P}_u = \mathbf{S}^\top \mathbf{V} \max(\mathbf{D}, \mathbf{I}) \mathbf{V}^\top \mathbf{S} \quad (2.101)$$

avec $\max(\mathbf{D}, \mathbf{I})$, la matrice contenant les valeurs maximales de chaque élément. Le seul paramètre indéterminé ici est l'espérance \mathbf{x}_u . Celui-ci doit être résolu par une méthode d'optimisation cherchant à minimiser un paramètre comme le déterminant de la matrice de covariance \mathbf{P}_u obtenue. La figure 2.10 montre un exemple de fusion utilisant la CU pour des particules de dimension 1 et 2.

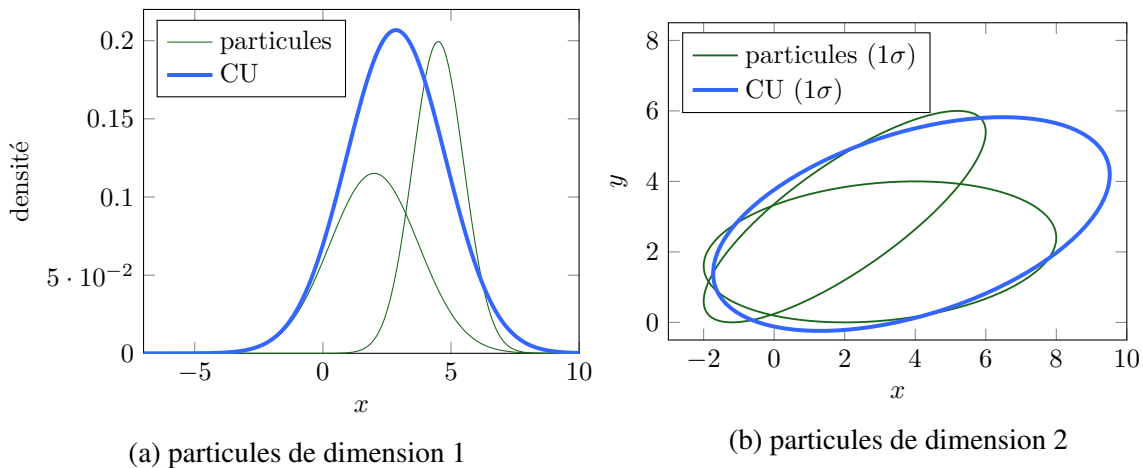


FIGURE 2.10 – Exemple de fusion de deux particules gaussiennes de dimension 1 et 2 en utilisant l'union de covariance.

2.7.4 Union de covariance généralisée

L'union de covariance généralisée ou *generalised covariance union* (GCU) en anglais, est une généralisation des fusions SMR et CU. L'approche a été introduite par (REECE et ROBERTS

2010). De la même façon que la CU, cette formule permet de calculer une fusion d'un ensemble d'hypothèses gaussiennes en respectant l'intégrité de l'estimation. La différence entre la CU et cette approche est que l'information de probabilité de chacune des hypothèses peut être utilisée. Il est en effet possible de définir une borne supérieure et inférieure pour chaque poids des particules à fusionner. Cette solution est donc équivalente à une CU lorsque l'intervalle de chaque poids vaut $[0; 1]$, et équivalente à une SMR lorsque chaque borne est fixée au poids de la particule.

Supposons que l'on souhaite fusionner un ensemble de deux particules :

$$\{(w_1, \mathbf{x}_1, \mathbf{P}_1), (w_2, \mathbf{x}_2, \mathbf{P}_2)\} \quad (2.102)$$

afin d'obtenir la particule $(\mathbf{x}_g, \mathbf{P}_g)$ par l'approche GCU. Les poids w_1 et w_2 des particules ne sont pas connus mais bornés par les intervalles :

$$w_1 \in [l_1; u_1], \quad w_2 \in [l_2; u_2] \quad (2.103)$$

dont les valeurs sont connues. De la même façon que pour la CU, on définit les matrices :

$$\mathbf{U}_1 = \mathbf{P}_1 + (\mathbf{x}_g - \mathbf{x}_1)(\mathbf{x}_g - \mathbf{x}_1)^\top \quad (2.104)$$

$$\mathbf{U}_2 = \mathbf{P}_2 + (\mathbf{x}_g - \mathbf{x}_2)(\mathbf{x}_g - \mathbf{x}_2)^\top. \quad (2.105)$$

La contrainte que la matrice de covariance \mathbf{U}_g doit respecter est la suivante :

$$\mathbf{U}_g \geq \frac{w_1 \mathbf{U}_1 + w_2 \mathbf{U}_2}{w_1 + w_2} \quad \forall w_1 \in [l_1; u_1], w_2 \in [l_2; u_2] \quad (2.106)$$

La résolution de \mathbf{U}_g se fait d'une façon similaire à la CU. Soit $\mathbf{S} = \text{chol}(\mathbf{U}_2)$, le triangle supérieur de la décomposition de Cholesky. Soient \mathbf{V} et \mathbf{D} , les matrices des vecteurs propres et des valeurs propres de $\mathbf{S}^{-\top} \mathbf{U}_1 \mathbf{S}^{-1}$. La matrice de covariance \mathbf{P}_g est obtenue par l'expression :

$$\mathbf{P}_g = \mathbf{S}^\top \mathbf{V} \mathbf{A} \mathbf{V}^\top \mathbf{S} \quad (2.107)$$

avec \mathbf{A} une matrice dont les éléments A_{ij} valent :

$$A_{ii} = 1 + \max \left\{ \frac{u_1}{u_1 + l_2} (D_{ii} - 1), \frac{l_1}{l_1 + u_2} (D_{ii} - 1) \right\} \quad (2.108)$$

$$A_{ij} = 0 \quad \forall i \neq j.$$

Comme pour la CU, l'espérance \mathbf{x}_g doit être optimisée pour obtenir une matrice de covariance \mathbf{U}_g ayant le plus petit déterminant possible. Le poids w_g de la particule obtenue après fusion par

GCU peut être estimé à partir des intervalles des particules utilisées :

$$w_g \in [l_1 + l_2; u_1 + u_2]. \quad (2.109)$$

Cet intervalle est cependant optimisable en ayant connaissance des intervalles des autres particules représentant une mixture. Si l'on considère que la mixture contient d'autres particules non-fusionnées, dont les indices sont contenus dans l'ensemble \mathcal{S} , alors la borne inférieure l_g et la borne supérieure u_g de la particule après GCU peuvent être resserrées pour respecter les inégalités

$$l_g \geq \sum_{i \in \mathcal{S}} l_i, \quad u_g \leq \sum_{i \in \mathcal{S}} u_i. \quad (2.110)$$

L'union de covariance généralisée a l'avantage de pouvoir tendre vers les résultats d'une SMR ou d'une CU en fonction de la connaissance que l'on peut avoir sur les paramètres des particules à fusionner. En effet, si l'on a aucune information sur le poids des particules, les intervalles sont tous définis à $[0; 1]$. On se retrouve alors dans une configuration similaire à la CU. À l'inverse, si les poids des particules sont connus, les bornes des intervalles sont donc confondues avec le poids des particules. La GCU donne alors des résultats équivalents à la SMR.

2.7.5 Distance de Mahalanobis

La distance de Mahalanobis permet de calculer la distance entre une variable aléatoire multidimensionnelle et une valeur particulière. Contrairement à une distance euclidienne, cette distance prend en compte la corrélation qui existe entre les composantes du vecteur aléatoire. Dans le cas de l'estimation bayésienne, la distance de Mahalanobis est un bon critère pour mesurer la justesse d'une estimation en la comparant à la valeur réelle.

Supposons que nous ayons une variable aléatoire de dimension n suivant une loi de probabilité quelconque. On note \mathbf{P} sa matrice de covariance et $\boldsymbol{\mu}$ son vecteur d'espérance. La distance de Mahalanobis d'une observation \mathbf{x} à cette variable aléatoire est définie de la façon suivante :

$$d(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^\top \mathbf{P}^{-1} (\mathbf{x} - \boldsymbol{\mu})}. \quad (2.111)$$

Lorsque la variable aléatoire suit une loi multi-normale, il existe une relation liant de façon unique, la densité de probabilité d'une observation et sa distance de Mahalanobis. Cette relation est basée sur la loi du χ^2 :

$$d^2(\mathbf{x}) \sim \chi_n^2 \quad (2.112)$$

Cela signifie que l'ensemble des observations ayant une même distance de Mahalanobis corres-

pond à l'ensemble des observations ayant une même valeur de densité. Cet ensemble d'observation représente un hyper-ellipsoïde. La figure 2.11 montre les ellipses obtenues pour différentes valeurs de distance à une loi gaussienne de dimension 2. La distance de Mahalanobis est donc une façon alternative de définir les ellipses représentant les densités de probabilité gaussiennes. L'ellipse à 1σ correspond à l'ensemble des points ayant pour distance $d(\mathbf{x}) = 1$.

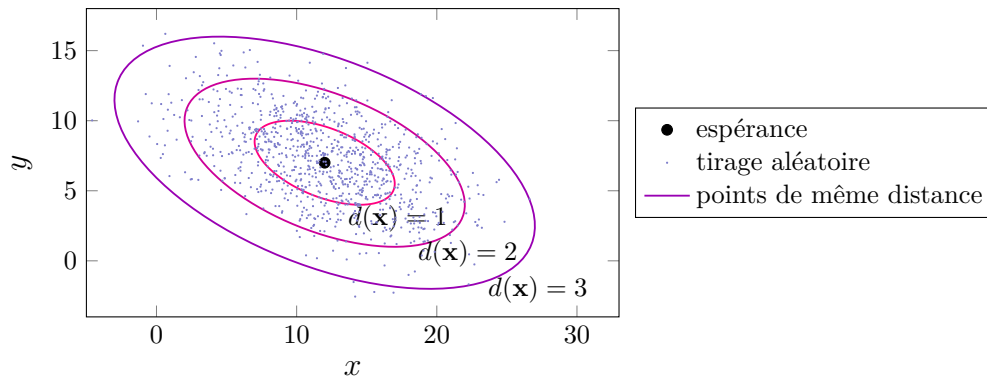


FIGURE 2.11 – Exemple de mesures de distance de Mahalanobis avec une loi Normale de dimension 2. Les ellipses représentent l'ensemble des observations ayant une même valeur de distance. Les points correspondent à des tirages aléatoires suivant la loi mesurée.

2.7.6 Divergence de Bhattacharyya

La divergence de Bhattacharyya (BHATTACHARYYA 1943) est une mesure de similarité entre deux densités de probabilité. Elle permet de mesurer la quantité de chevauchement entre les deux densités. Cette divergence est surtout utilisée en statistique pour déterminer si des classes sont séparables.

Soient p et q , des densités de probabilité discrètes définies sur l'ensemble \mathcal{E} . La divergence de Bhattacharyya discrète est définie par :

$$d_B(p, q) = -\log \sum_{\mathbf{x} \in \mathcal{E}} \sqrt{p[\mathbf{x}] q[\mathbf{x}]}. \quad (2.113)$$

Soient p et q , des densités de probabilité continues définies sur l'ensemble \mathcal{E} . La divergence de Bhattacharyya continue est définie par :

$$d_B(p, q) = -\log \int_{\mathcal{E}} \sqrt{p(\mathbf{x}) q(\mathbf{x})} d\mathbf{x}. \quad (2.114)$$

Lorsque les expressions des densités ne sont pas connues ou que l'expression analytique de cette divergence n'est pas résoluble, il est nécessaire d'utiliser une méthode de calcul numérique pour en obtenir une approximation. Dans le cas où p et q sont les densités des lois

normales $\mathcal{N}(\boldsymbol{\mu}_1, \mathbf{P}_1)$ et $\mathcal{N}(\boldsymbol{\mu}_2, \mathbf{P}_2)$, il existe une solution analytique à cette divergence :

$$d_B(p, q) = \frac{1}{8}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \mathbf{C}^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \frac{1}{2} \log \frac{\det \mathbf{C}}{\sqrt{\det \mathbf{P}_1 \det \mathbf{P}_2}} \quad (2.115)$$

avec

$$\mathbf{C} = \frac{\mathbf{P}_1 + \mathbf{P}_2}{2} \quad (2.116)$$

Nous pouvons remarquer que lorsque $\mathbf{P}_1 = \mathbf{P}_2$, la divergence de Bhattacharyya devient proportionnelle au carré de la distance de Mahalanobis.

Voici une courte liste des propriétés que possède la divergence de Bhattacharyya :

- $d_B(p, q) \geq 0$
- $p = q$ si et seulement si $d_B(p, q) = 0$
- d_B est symétrique : $d_B(p, q) = d_B(q, p)$.
- d_B ne valide pas l'inégalité triangulaire.

2.7.7 Divergence de Kullback-Leibler

La divergence de Kullback-Leibler (KULLBACK et LEIBLER 1951) est une mesure de dissimilarité entre deux densités de probabilité, basée sur la notion d'entropie relative. Elle est souvent utilisée pour comparer une densité de probabilité théorique à une approximation plus facile à manipuler. Soient p et q , des densités de probabilité discrètes définies sur l'ensemble \mathcal{E} . La divergence de Kullback-Leibler discrète est définie par :

$$d_{KL}(p, q) = \sum_{\mathbf{x} \in \mathcal{E}} p[\mathbf{x}] \log \frac{p[\mathbf{x}]}{q[\mathbf{x}]} \quad (2.117)$$

Soient p et q , des densités de probabilité continues définies sur l'ensemble \mathcal{E} . La divergence de Kullback-Leibler continue est définie par :

$$d_{KL}(p, q) = \int_{\mathcal{E}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \quad (2.118)$$

Dans le cas où p et q sont les densités des lois normales $\mathcal{N}(\boldsymbol{\mu}_1, \mathbf{P}_1)$ et $\mathcal{N}(\boldsymbol{\mu}_2, \mathbf{P}_2)$, il existe une solution analytique à cette divergence :

$$d_{KL}(p, q) = \text{trace} \left(\mathbf{P}_2^{-1} (\mathbf{P}_1 - \mathbf{P}_2 + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top) \right) + \log \frac{\det \mathbf{P}_2}{\det \mathbf{P}_1} \quad (2.119)$$

Voici une courte liste des propriétés que possède la divergence de Kullback-Leibler :

- $d_{KL}(p, q) \geq 0$

- $p = q$ si et seulement si $d_{KL}(p, q) = 0$
- d_{KL} n'est pas symétrique : $d_{KL}(p, q) \neq d_{KL}(q, p)$.
- d_{KL} ne valide pas l'inégalité triangulaire.

Bien que cette divergence ne soit pas symétrique, il est cependant possible de construire une divergence respectant cette propriété de la façon suivante :

$$\begin{aligned} d_J(p, q) &= d_{KL}(p, q) + d_{KL}(q, p). \\ &= \int_{\mathcal{E}} (p(\mathbf{x}) - q(\mathbf{x})) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \end{aligned} \quad (2.120)$$

La divergence d_J est souvent nommée divergence de Jeffreys.

2.7.8 Vocabulaire de l'estimation

Puisque ce manuscrit s'intéresse surtout à la localisation, et plus généralement à l'estimation d'états probabilistes, il est nécessaire de s'accorder sur le vocabulaire utilisé pour la caractérisation de la qualité des estimations. Les définitions de certains termes comme *précis*, *intègre* ou encore *consistant*, peuvent varier en fonction des auteurs. Pour éviter d'éventuelles confusions sur leur sens, cette partie présente les définitions utilisées dans ce manuscrit. Ces définitions s'appuient sur le vocabulaire utilisé en métrologie qui a l'avantage d'être standardisé.

Précision (anglais : *precision*) Le terme est utilisé ici dans le même sens que *fidélité* en métrologie. La précision correspond à l'étroitesse de l'estimation. Elle est quantifiable en utilisant des outils mathématiques tels que l'écart-type pour les états de dimension 1, la trace ou le déterminant de la matrice de covariance dans le cas où l'état est vectoriel. Géométriquement, cela correspond à la taille d'une ellipse ou ellipsoïde représentant l'incertitude d'une estimation gaussienne 2D ou 3D.

Justesse (anglais : *trueness*) Cela correspond au fait que la valeur vraie se trouve dans une zone de l'estimation où la probabilité est forte. La valeur vraie se trouve alors dans le voisinage d'un maximum local de la densité de probabilité. Dans le cas d'une loi normale, cela signifie que cette valeur est à proximité de l'espérance. Contrairement à la précision, il n'est pas possible de mesurer directement la justesse pendant une expérimentation. Il est en effet nécessaire d'utiliser une vérité de terrain qui servira alors d'approximation de la valeur vraie. La justesse est quantifiable en utilisant par exemple la distance de Mahalanobis entre la valeur vraie et la densité de l'estimation. Géométriquement, cela correspond à mesurer la distance de la valeur vraie à l'ellipse de covariance.

Exactitude (anglais : *accuracy*) Une estimation est exacte lorsqu'elle est à la fois précise et juste. Cela signifie que la densité de probabilité de l'estimation est étroite et à proximité de la valeur vraie. La figure 2.12 montre l'utilisation de ce vocabulaire pour différentes qualités d'estimations.

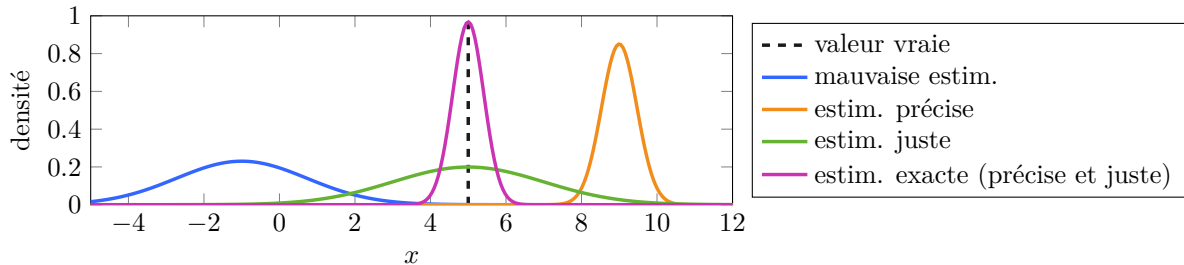


FIGURE 2.12 – Exemple d'utilisation du vocabulaire pour qualifier plusieurs estimations de dimension 1, dont la valeur à estimer vaut 5.

Intégrité (anglais : *consistency*) Cela correspond au fait qu'une estimation reste juste pendant une longue période d'une expérimentation. Généralement, on parle de perte d'intégrité lorsque qu'il y a une accumulation d'erreurs dans l'estimation provoquant un biais constant ou incrémental. L'estimation a alors tendance à diverger, et la justesse, à se dégrader. Contrairement à la justesse, l'intégrité n'est pas quantifiable, mais il est cependant possible de définir une méthode permettant de considérer si une estimation est intègre ou non. Les auteurs de (BAR-SHALOM et al. 2004, §5.4) présentent une de ces méthodes.

Soient $\mathbf{x}_{k|k}$ et $\mathbf{P}_{k|k}$, l'espérance et la covariance de chaque estimation d'un état de dimension n , obtenues par un filtre. Soit \mathbf{x}_k la valeur vraie que le filtre cherche à estimer. Le test d'intégrité se base sur le calcul de l'erreur quadratique de l'estimation normalisée ou *normalized estimation error squared* (NEES) en anglais :

$$\epsilon_k = (\mathbf{x}_k - \mathbf{x}_{k|k})^\top \mathbf{P}_{k|k}^{-1} (\mathbf{x}_k - \mathbf{x}_{k|k}). \quad (2.121)$$

La NEES est donc équivalent au calcul du carré de la distance de Mahalanobis entre l'estimation et la valeur vraie. Cette erreur suit une loi du χ^2 à n degrés de liberté. Si l'on souhaite tester l'intégrité à 95 %, il suffit alors de s'assurer que 95 % des estimations soient inférieures à la borne $\chi_n^2(0.95)$. Par exemple, pour $n = 2$, cette limite vaut

$$\chi_2^2(0.95) = 5.99. \quad (2.122)$$

Si l'expérimentation compte 100 points, il faut donc qu'il n'y ait pas plus de 5 points dont la NEES dépasse le seuil de 5.99 pour que l'estimation soit considérée comme intègre. Nous pouvons cependant noter ici que le choix de 95 % est arbitraire.

La méthode peut être améliorée dans le cas des simulations de Monte-Carlo. Une même

simulation est exécutée N fois en générant des bruits de mesures aléatoires. Dans ce cas-ci, les erreurs $\bar{\epsilon}_k$ utilisées pour le test d'intégrité sont les moyennes des NEES $\epsilon_{i,k}$ de chaque simulation à chaque instant k :

$$\bar{\epsilon}_k = \frac{1}{N} \sum_{i=1}^N \epsilon_{i,k}. \quad (2.123)$$

Le test d'intégrité est alors effectué par rapport à un seuil obtenu avec une loi du χ^2 à $n \times N$ degrés de liberté. Ce seuil est divisé par le nombre de simulations avant de pouvoir être comparé aux erreurs moyennes $\bar{\epsilon}_k$. Par exemple, pour 50 simulations d'une estimation d'un état de dimension 2, le seuil à 95 % est de

$$\frac{\chi_{100}^2(0.95)}{50} = 2.49. \quad (2.124)$$

Chapitre 3

État de l'art de la localisation coopérative

3.1 Introduction

Pendant de nombreuses années, la recherche en robotique mobile s'est surtout intéressée à la perception, la localisation et le contrôle de robots mobiles dans des situations où les robots travaillent seuls dans leur environnement. Avec l'avancement des travaux dans différents domaines de recherche, les applications de la robotique mobile se sont complexifiées et les premières problématiques d'interactions entre les robots ont commencé à apparaître. (PREMVUTI et YUTA 1990) est l'un des premiers articles à évoquer le sujet. Les auteurs soulignent l'intérêt grandissant de l'utilisation de plusieurs robots mobiles et présentent une stratégie de décision pour l'interaction entre les robots. Dans le domaine de la localisation coopérative, (KURAZUME et al. 1994) présente une première approche de localisation basée uniquement sur le positionnement relatif d'un groupe de robots par rapport à un autre. Cette approche calcule également l'incertitude des positions des robots estimées sous la forme d'informations de variance et covariance pour chaque robot. La solution est assez générique puisqu'elle travaille dans un espace de dimension 3 et considère les mesures relatives entre les robots en coordonnées sphériques. Cela reste cependant très théorique puisqu'elle ne traite pas de la façon dont les mesures entre les robots sont obtenues.

3.2 Capteurs

Les capteurs dont les mesures sont exploitées par les algorithmes de localisation en robotique mobile peuvent se regrouper en deux catégories : proprioceptifs ou extéroceptifs. Les capteurs proprioceptifs permettent d'obtenir des informations sur les actions d'un robot. Les mesures obtenues sont donc liées à l'état interne d'un robot. Dans le cas d'un robot mobile à

roue, il peut par exemple s'agir d'odomètres mesurant l'angle de rotation des roues, ou encore l'angle de braquage dans certains modèles de robots. Les capteurs extéroceptifs correspondent aux capteurs capables d'obtenir des informations sur l'environnement du robot. Dans le domaine de la localisation, les capteurs proprioceptifs sont utilisés pour établir un modèle d'évolution de l'état du robot et les capteurs extéroceptifs donnent des informations externes à l'état qui pourront être fusionnées pour améliorer l'estimation.

3.2.1 GNSS

Le sigle GNSS (global navigation satellite system) est le nom générique des systèmes de navigation satellitaires fournissant une couverture globale de géopositionnement (Duquenne et al., 2005). L'objectif de ces systèmes est de donner la géolocalisation d'un mobile ainsi que sa vitesse en tout lieu du globe et dans un référentiel global. Les premiers créateurs d'un tel système sont les militaires américains (GPS : global positioning system) suivis de très près par les russes (GLONASS : GLObal nAvigation satellite system). Depuis 2000, le système GPS est devenu accessible aux populations civiles avec une précision métrique, ce qui a permis à la géolocalisation GPS de se démocratiser. À l'heure actuelle les services utilisant ces systèmes sont très nombreux et variés avec parmi les exemples les plus connus la géonavigation automobile et la randonnée. Cette technologie est également largement utilisée dans les dispositifs expérimentaux à différents niveaux, nécessitant diverses échelles de précision et devant répondre à différentes contraintes :

- géolocalisation d'habitats, d'espèces patrimoniales, d'essences de fruitiers (précision métrique);
- définition du contour d'un dispositif expérimental (précision centimétrique);
- géonavigation appliquée à l'agriculture de précision... (précision centimétrique d'un objet en mouvement).

La méthode la plus couramment utilisée pour déterminer la position d'un point à la surface du globe repose sur le principe de la triangulation. La triangulation permet par exemple de localiser un objet en connaissant la distance qui le sépare de 3 points de référence (figure 3.1). En 3D il est nécessaire de rajouter une quatrième distance reliant l'objet à un autre point de référence pour connaître l'altitude, on parle alors de multilatération. Extrapolée à la géolocalisation d'un récepteur à la surface du globe, il est possible de définir les coordonnées X, Y et Z d'un point si l'on connaît les distances qui le séparent de quatre satellites. Les coordonnées du point d'intersection des sphères représentant les signaux émis par les différents satellites seront d'autant plus précises que le nombre de satellites captés sera important.

Le principe du calcul des distances séparant le récepteur d'au moins quatre satellites repose sur le temps que mettent les signaux pour parcourir la distance séparant l'opérateur des



FIGURE 3.1 – Illustration du principe de triangulation pour géolocaliser un récepteur à la surface du globe. Un quatrième satellite permettra de définir l'altitude.

(source : <http://www.anr-prodige.com/index.php?n=Technologies.Geolocalisation>)

satellites. Le temps est multiplié par la vitesse de propagation du signal (vitesse de la lumière) pour obtenir la distance satellite - opérateur. Le calcul des quatre distances permettra par multilatération de définir les coordonnées X, Y et Z de l'opérateur. La précision de cette méthode peut atteindre au mieux 1 mètre si on lui associe des corrections de positionnement (correction différentielle). Des méthodes utilisant la phase de la porteuse et non le code ont été développées pour de nombreuses applications nécessitant des résolutions plus importantes (précision centimétrique).

3.2.2 Caméra

Une caméra est un capteur capable de percevoir la lumière émise par une zone restreinte de l'environnement dans la direction du capteur (figure 3.2). Les informations obtenues par ce genre de capteur sont souvent présentées sous forme d'une suite d'images dont les pixels contiennent les informations sur la lumière mesurée. La taille du champ de vision de la caméra peut être plus ou moins grande en fonction des technologies optiques utilisées. La quantité d'information qu'apporte ce genre de capteur est très importante mais l'extraction des données utiles est parfois coûteuse en calcul. Par exemple, il est possible de détecter des piétons ou des véhicules et d'estimer leur position dans l'image du capteur. Cependant, il s'agit ici d'une projection des objets observés dans le plan de l'image. L'information de distance n'est pas récupérable directement par le capteur. Il existe toutefois des solutions technologiques comme la stéréo-vision ou encore les caméras *time-of-flight* permettant de récupérer l'information de profondeur sous forme d'images.

Étant donné que ce type de capteur se base sur la lumière, il n'est possible d'obtenir des informations que sur les éléments de l'environnement dont la ligne de vue est directe. Si un obstacle passe devant l'objet, il ne sera alors plus possible de l'observer. Cela peut être problématique en extérieur lorsque le temps est pluvieux ou brumeux. De plus, lorsque une source



FIGURE 3.2 – Exemple de caméra de la marque Baumer.

lumineuse est dans le champ de vision de la caméra, des effets d'éblouissement peuvent perturber la qualité de la perception.

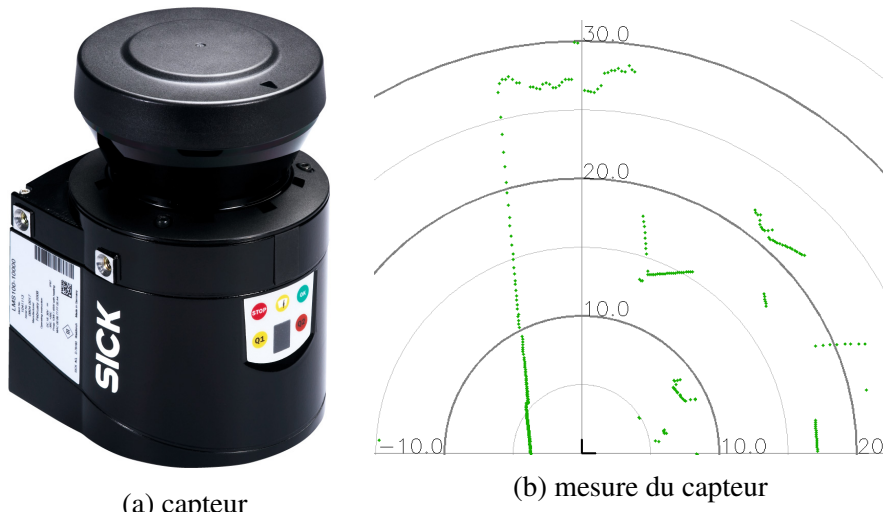
3.2.3 Lidar

Lidar est un acronyme pour l'expression anglaise *light detection and ranging*. Ce type de capteur travaille également avec la lumière mais permet cette fois-ci de mesurer des distances. Un laser infrarouge émet une onde lumineuse qui se réfléchit sur un objet et dont une partie de la lumière est renvoyée en direction d'un capteur infrarouge situé à proximité de l'émetteur. La plupart de ces capteurs effectue un balayage permettant au laser de couvrir une large zone de l'espace sur une distance pouvant aller jusqu'à 100 m pour certains capteurs. Ils permettent d'obtenir des points en coordonnées polaires dans le référentiel défini par l'axe de rotation du balayage. La figure 3.3 présente un exemple de lidar monocouche couramment utilisé pour la sécurité et la localisation de robots mobiles. Certains capteurs sont capables de couvrir un ou plusieurs plans sur 360°. Ils peuvent également mesurer la quantité de lumière réfléchie et ainsi connaître la qualité de réflexion des surfaces des objets.

Contrairement à la plupart des caméras, ce type de capteur émet sa propre lumière qu'il mesure. Il n'est donc pas contraint par l'éclairage ambiant et peut fonctionner dans le noir. Cependant, il possède les mêmes limitations de ligne de vue qu'une caméra et aura donc du mal à fonctionner en extérieur par temps brumeux. La quantité de données retournée par ce genre de capteur est plus faible qu'une caméra et le champ d'observation est plus important. Le traitement des données est donc plus léger, mais il est plus difficile de faire de la reconnaissance précise et de la classification d'objets dans l'environnement.

3.2.4 Radar

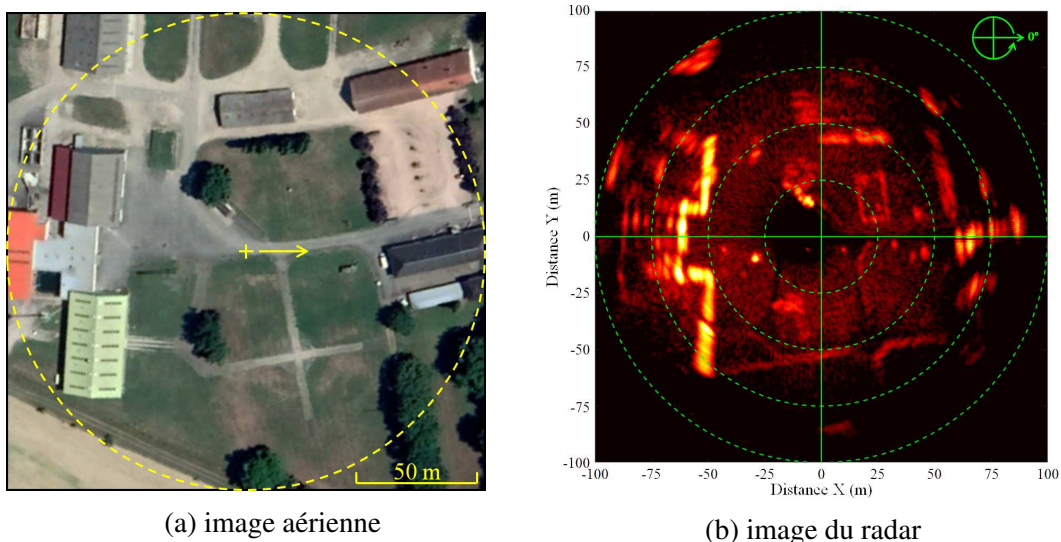
Un radar est un capteur émettant des ondes radios à hautes fréquences et écoutant le rebond de ces ondes sur les objets avoisinants. Les mesures effectuées sont des mesures de distance et



(a) capteur
 (b) mesure du capteur

FIGURE 3.3 – Le lidar Sick LMS150 et un exemple de mesure obtenue (exprimée en coordonnée polaire)

parfois de vitesse de l'objet dans la direction de l'antenne et selon un cône d'ouverture directement lié aux caractéristiques de cette dernière. Contrairement à la lumière, les ondes radios de ces capteurs ont l'avantage de pouvoir facilement traverser certaines matières. Ils sont donc capables d'observer des objets sur plusieurs plans dans la même direction. Les mesures de ce genre de capteur ne permettent pas d'obtenir des points précis sur les obstacles aux alentours. Après traitement, les données résultantes peuvent être exprimées sous forme d'une image de la zone observée dont les pixels contiennent la quantité d'onde réfléchié par l'environnement (figure 3.4). C'est le cas par exemple des radars développés à l'Inrae (figure 3.5), dont la rotation de l'antenne permet d'effectuer un balayage à 360° et couvrir une zone avec un rayon de 100 m.



(a) image aérienne
 (b) image du radar

FIGURE 3.4 – Exemple d'observation obtenue à partir d'un radar 360°. Source : (ROUVEURE et al. 2019, Fig. 2)

Ce genre de capteur n'est pas très adapté à l'observation de petits objets, car la faible quan-



FIGURE 3.5 – Le radar Pelican développé par l’Inrae.

tité d’informations d’une petite zone rend difficile l’utilisation des algorithmes de reconnaissance de forme. Cependant cet outil est très pratique pour la localisation du capteur à partir de cartes existantes ou en utilisant des solutions de type SLAM (*simultaneous localization and mapping*). Le capteur à l’avantage de pouvoir travailler efficacement en extérieur sans être contraint par la météo ou tout autre élément pouvant limiter la visibilité du robot.

3.2.5 UWB

L’acronyme UWB vient de l’expression anglaise *ultra-wideband* et correspond à une technologie de communication comme le Wifi ou le Bluetooth. L’UWB travaille dans des fréquences radios (3.5 à 6 GHz) proches de celles du Wifi et du Bluetooth, mais la particularité de ce protocole est qu’il a une très large bande de fréquence. Cela a l’avantage de permettre à cette technologie de faire des mesures précises de distance entre deux émetteurs/récepteurs UWB (figure 3.6). Cette technologie donne également la possibilité d’identifier l’objet mesuré puisqu’il est nécessaire de choisir la cible avec laquelle la mesure de distance est faite. Il n’est cependant pas possible d’effectuer plusieurs mesures simultanément. Cela implique qu’une seule paire de robot peut effectuer une mesure à un instant donné.



FIGURE 3.6 – Le capteur UWB développé par l’Inrae et l’Institut Pascal.

3.3 Présentation des approches de localisation coopérative

Il y a maintenant de plus en plus d'articles consacrés à la localisation coopérative mais parmi ces travaux, il est possible de distinguer deux approches différentes du problème :

- l'approche centralisée, dans laquelle les calculs sont effectués par un algorithme global
- l'approche décentralisée, dans laquelle chaque robot calcule individuellement son estimation de l'état du groupe.

Chacune de ces approches a des avantages et inconvénients qui vont dépendre de l'application souhaitée.

3.3.1 Approches centralisées

Dans une approche centralisée, un système central s'occupe de calculer la localisation de tous les robots. C'est une solution simple au problème de gestion des robots puisque les informations sont rassemblées dans un unique algorithme. Les premiers articles traitant de la localisation coopérative utilisent ce genre d'approche. Dans le cas de (KURAZUME et al. 1994), les informations de position de chaque robot sont calculées par la même machine. Cela implique que les informations des mesures entre chaque robot sont remontées à la machine s'occupant de faire le calcul de localisation. Il est donc nécessaire de mettre en place une architecture réseau sur les robots permettant de communiquer avec la machine de calcul. La figure 3.7 schématise cette communication.

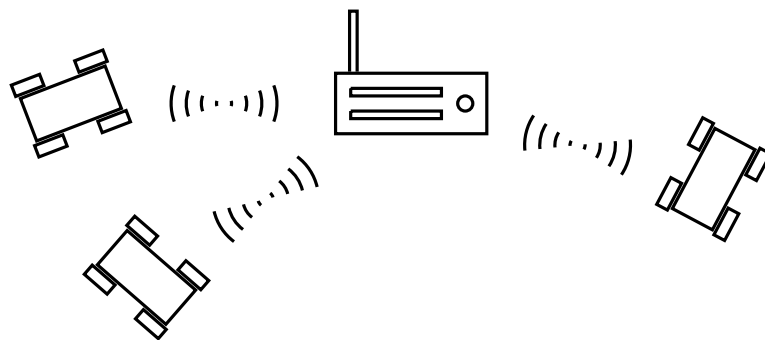


FIGURE 3.7 – Communication entre les robots et la machine dédiée au calcul de la localisation centralisée

En ayant rassemblé toutes les informations des robots, le problème de localisation est alors plus simple à résoudre. Il repose uniquement sur la modélisation de la fonction liant la pose de chaque robot avec les mesures réalisées. Cela correspond à considérer que la flotte de véhicules est un unique système complexe dont l'objectif est de déterminer l'état de chacun de ces éléments. Pour calculer une nouvelle estimation de l'état du système de robots, il est nécessaire d'avoir une communication permanente avec l'ensemble des acteurs du système. En effet,

l'approche centralisée ne permet pas de gérer facilement la perte de communication avec un ou plusieurs robots.

L'un des premiers algorithmes permettant de résoudre le problème de localisation d'un ensemble de robots est le filtre de Kalman (section 2.3.1). Il est utilisé pour obtenir une suite d'estimations gaussiennes de l'état du robot en prenant en compte les mesures internes et externes de ses capteurs. Les travaux de (ANDERSON et MOORE 1979) ont montré que ce filtre est optimal si l'on considère que les modélisations des robots et des observations correspondent exactement aux contraintes des mesures. Dans le contexte de la localisation coopérative, l'état à estimer est l'ensemble des positions et orientations des robots. Le filtre de Kalman peut donner une estimation optimale sous ces conditions. Dans la pratique, il est cependant rarement possible d'utiliser directement le filtre de Kalman. Les modèles de déplacement des robots ainsi que les modèles d'observation ne sont pas linéaires dans la plupart des cas. C'est pourquoi on lui préférera généralement le filtre de Kalman étendu (section 2.3.2) qui propose une linéarisation locale des modèles, ou d'autres méthodes plus complexes mais gérant plus efficacement ces problèmes.

Bien que le rassemblement des informations des robots dans une même machine simplifie les calculs de la localisation, ce scénario est rarement utilisé dans des applications concrètes. Lorsque le nombre de robots est important, la quantité d'information à traiter devient problématique. Pour que le calcul soit centralisé, il est en effet nécessaire de transmettre les informations des différents capteurs des robots au système de calcul. L'odométrie a souvent une fréquence de mesure assez élevée et les capteurs lidars ou caméras transmettent une quantité importante d'informations. Selon les technologies de communication et les algorithmes de localisation utilisés, la stratégie de localisation coopérative peut ne plus fonctionner en temps réel.

Une solution à ce problème consiste à mieux exploiter les capacités de calcul des robots. Au lieu de transmettre la totalité des informations des capteurs des robots, ces derniers peuvent calculer la partie de l'état qui les concerne. Les robots n'envoient donc plus les informations des capteurs mais calculent une estimation de leur propre état ainsi que les corrélations avec les états des autres robots. Ce calcul correspond à distribuer le calcul de l'état du système aux différents acteurs le composant. Cette approche a été utilisée dans (ROUMELIOTIS et BEKEY 2002) avec un filtre de Kalman. La fusion des informations propres à chaque robot est faite individuellement. Les observations entre les robots correspondent à des différences de poses. L'intégration de ces mesures dans l'estimation de l'état est également distribuée entre les robots en prenant soin de conserver les mêmes étapes de calcul que dans l'approche centralisée non-distribuée. Cette solution se sert du fait que la fusion d'une mesure entre deux robots ne fait pas intervenir la totalité des informations des robots mais uniquement la partie de l'état à estimer qui concerne ces deux robots. Ce calcul matriciel peut en effet être simplifié et déporté dans un algorithme individuel pour chaque robot.

Bien que le calcul d'une localisation centralisée puisse être réparti entre les robots, il reste cependant nécessaire de rassembler et transmettre les nouvelles informations calculées par les robots. Cet échange d'informations nécessite toujours une machine centrale dont le travail est cette fois-ci la synchronisation des données et la gestion de l'état global du système de robots. Quand le nombre de robots devient important, les problématiques de communication entre les robots se complexifient. Chaque robot doit pouvoir communiquer avec la machine centrale pour envoyer ses résultats et recevoir les informations des autres robots. Dans une solution basique de localisation centralisée, il est nécessaire que le système central envoie à tous les robots la nouvelle estimation de l'état après que chaque robot ait envoyé le résultat de ses calculs. En effet, il est nécessaire que chaque robot travaille avec l'état le plus à jour possible pour éviter les pertes d'informations.

Ce problème devient encore plus difficile à gérer lorsque le nombre de robots est dynamique. Cela peut survenir dans des contextes où les robots sont des véhicules se déplaçant dans une ville ou plus simplement quand les moyens de communication ne permettent pas d'avoir la totalité des robots à la portée du système central. Les robots qui ne peuvent plus communiquer avec le système central ne bénéficient plus de la localisation coopérative bien qu'ils pourraient continuer d'observer les robots voisins. Si ce système tombait en panne, ce serait la totalité de la flotte de robots qui retournerait à une localisation individuelle. Il reste cependant possible d'ajouter de la redondance dans le système central en multipliant le nombre de machines dédiées au rassemblement des informations mais cela implique une synchronisation des données entre ces machines. La dépendance à ce système est donc un inconvénient important pour des scénarios complexes.

3.3.2 Approche décentralisée

Pour éviter les problèmes de l'approche centralisée, beaucoup de travaux ont essayé de développer des solutions sans système central. Les robots ont alors une autonomie plus importante tout en conservant l'aspect coopératif de la localisation. Ce type d'approche est donc considéré comme décentralisé car chaque robot s'occupe de l'estimation de son propre état. Il est cependant possible que ces robots calculent également une estimation de l'état de leurs voisins mais ce calcul n'est plus groupé. Une solution de localisation décentralisée doit permettre de pouvoir enlever n'importe quel robot d'une expérimentation sans que le fonctionnement de l'algorithme des autres robots ne soit impacté.

Dans cette approche, l'échange d'informations se fait directement entre les robots. La figure 3.8 représente ce type de communication. La technologie de communication impliquée dans ce type de stratégie doit donc permettre de pouvoir parler directement avec les robots à leur portée. C'est le cas par exemple d'un réseau wifi avec point d'accès. Si tous les robots sont connectés

dans ce réseau, ils peuvent parler entre eux directement grâce à leur adresse IP. Cependant, le problème de centralisation existe toujours dans ce type de réseau mais sous une autre forme. En effet, si la machine faisant point d'accès wifi tombait en panne, la communication entre les robots ne pourrait plus être maintenue. La meilleure solution pour une approche entièrement décentralisée est donc d'utiliser une technologie de communication de type *peer-to-peer* qui offre l'avantage d'avoir une architecture équivalente à celle utilisée pour les échanges d'informations entre les robots. La déconnexion d'un robot n'aurait ainsi aucune répercussion sur les autres.

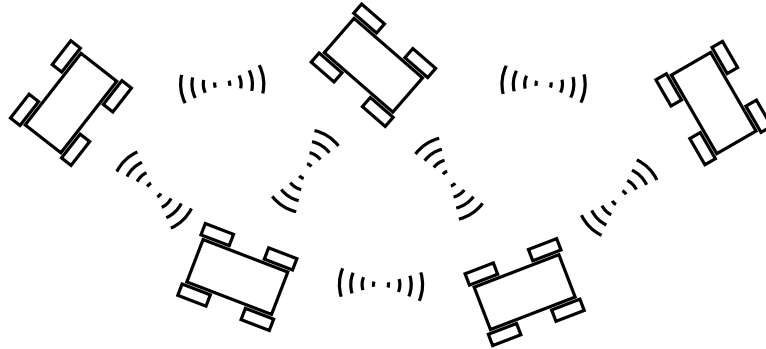


FIGURE 3.8 – Communication entre les robots à la portée les uns des autres, utilisée dans la localisation décentralisée

La problématique principale d'une solution de localisation décentralisée repose sur la gestion des états des autres robots. L'algorithme de localisation de chaque robot fait intervenir des mesures avec les robots voisins. Puisque ces mesures se basent sur l'état des robots voisins, il est donc nécessaire d'obtenir une estimation de cet état avant de pouvoir fusionner la mesure. L'estimation de l'état du voisin peut avoir été calculée par l'algorithme ou transmis par un autre robot (ou le robot mesuré lui-même). S'il s'agit d'un calcul local, il faut cependant que cette information prenne en compte les changements d'état du robot voisin. L'aspect coopératif de la localisation correspond alors au fait que les multiples estimations de l'état d'un même robot soient fusionnées pour ainsi prendre en compte le maximum d'informations possible.

Dans (HOWARD et al. 2003), les auteurs proposent une implémentation de cette solution basée sur des observations de différences de poses entre les robots. Chaque robot i met à jour un ensemble de filtres particulières estimant la pose des robots voisins. Les poses de ces robots sont exprimées dans le référentiel du robot i . Les informations échangées entre les robots correspondent à leur déplacement ainsi que les différences de poses mesurées. Dans cet article, l'aspect décentralisé de la localisation est bien présent puisque chaque robot construit une cartographie des poses des robots dans son référentiel. La suppression d'un robot du système n'empêche en aucun cas l'estimation des autres robots. Cette solution a cependant le problème d'avoir plusieurs estimations différentes de l'état d'un même robot. En effet, chaque robot met à jour l'état du robot observé en utilisant uniquement les mesures reçues des autres robots. Les robots ne transmettent donc jamais les estimations qu'ils ont calculées. Il n'y a donc pas de fusion entre les différentes estimations faites de ce même robot. Si l'on souhaite avoir une meilleure

estimation de chacun des robots, il est nécessaire de fusionner toutes les estimations existantes d'un même robot. Cependant, cette fusion n'est pas simple à mettre en place.

Considérons que nous souhaitons connaître l'état du robot i après fusion des estimations faites par tous les robots de l'expérimentation et étant donné que les mêmes mesures ont été utilisées pour chacune de ces estimations, il n'est pas envisageable de les fusionner directement. En effet, l'utilisation multiple d'une même observation dans une estimation provoque ce que l'on appelle de la *consanguinité* dans l'information. Le problème de la consanguinité est qu'elle provoque une sur-convergence de l'estimation d'un état. Cela signifie que l'intervalle de confiance que l'on accorde à une estimation peut devenir plus petit que ce qu'il devrait être compte tenu des informations disponibles. Avec le temps, l'intervalle de confiance peut même s'éloigner de la vraie valeur de l'état et ne plus la prendre en compte. On parle alors de perte d'intégrité. Le problème de consanguinité est donc assez important et doit être traité correctement pour qu'une solution de localisation coopérative décentralisée soit utilisable.

3.4 Gestion de la corrélation des données

La gestion des échanges d'information entre les robots est l'un des aspects les plus délicats à traiter en localisation coopérative. Dans les approches décentralisées, ces échanges sont souvent la première source d'erreur de localisation. Voici un exemple illustrant l'une des formes les plus simples de ce problème : la consanguinité de l'information. Nous avons deux robots immobiles A et B se trouvant à proximité l'un de l'autre. Ces robots sont capables de s'observer et de communiquer entre eux. À un instant k , chacun de ces robots possède une estimation assez grossière de sa position. Ces estimations sont échangées entre les robots. Le robot A effectue une observation du robot B et se sert de l'information de position de ce dernier pour améliorer la sienne. De la même façon, le robot B arrive à améliorer son estimation de position grâce à une observation de A . À l'instant $k + 1$, les nouveaux états de A et B sont transmis entre les robots. Les robots peuvent alors effectuer de nouvelles observations entre eux et améliorer leur estimation de position. Si nous répétons ces étapes en boucle, les estimations ne peuvent que s'améliorer puisque chaque nouvelle observation apporte de l'information. En utilisant un filtre de Kalman, ce genre de scénario conduirait à une sur-convergence des estimations. Les covariances deviendraient de plus en plus faibles mais les estimations ne seraient plus intègres. Les résultats seraient donc erronés.

Le problème vient ici de l'utilisation des estimations de positions des robots voisins. À chaque nouvelle mise à jour de son état, le robot utilise une information dont la corrélation avec son état est non négligeable. Le filtre de Kalman n'est alors pas approprié dans ce genre de scénario puisque celui-ci s'appuie sur l'hypothèse que les données à fusionner sont indépendantes de l'état. Il existe cependant des algorithmes capables de gérer correctement la corrélation des

données. Cette partie du manuscrit détaille de façon approfondie cette problématique et présente les solutions utilisées dans la littérature.

3.4.1 Corrélation des états des robots

Dans une approche décentralisée, chaque robot gère l'estimation de son état de façon indépendante. Nous pouvons reprendre l'exemple de la localisation de deux robots A et B ayant pour vecteur d'état \mathbf{x}_A et \mathbf{x}_B . Chaque robot exécute son propre filtre afin de faire évoluer l'estimation de son état. L'aspect coopératif de la localisation vient des observations entre les robots. Ce type d'observation s'appuie sur deux informations : la mesure de différence des états des robots effectuée par un capteur et l'estimation de l'état du robot observé. La mesure de différence d'états est un nouveau vecteur d'information \mathbf{z}_k obtenu à chaque mesure. L'estimation de l'état est une information transmise par un moyen de communication entre les robots. Ces informations ne s'utilisent pas de la même façon dans le modèle d'observation de l'équation (2.2) présenté dans la section 2.2 :

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k). \quad (3.1)$$

L'estimation de l'état partagée par le robot observé est une grandeur aléatoire. Cette information est donc apportée sous forme de bruit dans le modèle d'observation. La variable aléatoire \mathbf{v}_k est donc un vecteur contenant le bruit de mesure du capteur détectant le robot observé mais également l'incertitude de l'état de ce dernier. Si l'algorithme de localisation utilisé est un filtre de Kalman, le vecteur \mathbf{v}_k représentera l'espérance du bruit de mesure et de l'état partagé. La matrice de covariance \mathbf{R}_k associée à \mathbf{v}_k sera composée des matrices de covariance de ce bruit de mesure et de l'état partagé. Il y a cependant une contrainte sur ce vecteur de bruit qui ne peut être respectée : le bruit doit être blanc. En effet, la partie du bruit correspondant à l'état partagé ne peut être considéré comme blanc, car les observations issues de l'état partagé ne sont pas temporellement indépendantes. Chaque nouveau vecteur \mathbf{v}_k ne doit pas avoir de lien avec les vecteurs précédents \mathbf{v}_{k-1} , \mathbf{v}_{k-2} , etc. Or l'estimation de l'état du robot observé utilise un modèle d'évolution faisant intervenir son estimation précédente. Il y a donc clairement une corrélation dans le bruit \mathbf{v}_k .

Dans le cas d'une approche centralisée, le bruit du modèle d'observation n'a pas besoin d'être si complexe. Seule la mesure du capteur est prise en compte puisque l'information de l'état du robot observé est déjà présente dans le vecteur d'état que l'algorithme cherche à estimer. Si nous reprenons le cas de la localisation de deux robots A et B , l'unique vecteur d'état à estimer \mathbf{x}_k se composera des états \mathbf{x}_{Ak} et \mathbf{x}_{Bk} des robots :

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{Ak}^\top & \mathbf{x}_{Bk}^\top \end{bmatrix}^\top \quad (3.2)$$

Le bruit \mathbf{v}_k du modèle d'observation se limite ici uniquement au bruit de la mesure. La contrainte

de blancheur du bruit d'un filtre de Kalman peut donc être respectée. Dans cette modélisation, la fonction $h(\mathbf{x}_k, \mathbf{v}_k)$ utilise toujours l'estimation de l'état du robot observé. Cependant, cette estimation fait cette fois-ci partie du vecteur d'état \mathbf{x}_k . La fusion d'une observation entre le robot A et B génère toujours de la corrélation entre leurs états \mathbf{x}_{Ak} et \mathbf{x}_{Bk} , mais celle-ci est gérée correctement car l'information de corrélation est présente dans la matrice de covariance de l'état

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{Ak} & \mathbf{P}_{BAk}^\top \\ \mathbf{P}_{BAk} & \mathbf{P}_{Bk} \end{bmatrix}. \quad (3.3)$$

La sous-matrice \mathbf{P}_{BAk} représente la covariance entre les états de A et B . La corrélation entre ces états est donc stockée dans cette partie de la matrice. Chaque étape d'observation entre les robots utilise la covariance complète du système de robots. Le calcul reste donc cohérent peu importe le nombre d'observations.

La différence entre l'approche centralisée et décentralisée se trouve dans cette gestion de la corrélation. La version décentralisée n'a pas directement accès à \mathbf{P}_{BAk} puisque les états \mathbf{x}_{Ak} et \mathbf{x}_{Bk} sont calculés individuellement. Une solution basée sur des filtres de Kalman indépendants reviendrait à considérer que \mathbf{P}_{BAk} est nulle à chaque étape de l'algorithme. Le problème est que lorsque la corrélation entre les états des deux robots est forte, cette matrice est importante. Si cette information est perdue, les estimations des états des robots peuvent sur-converger.

3.4.2 Illustration du problème

Il est nécessaire de comprendre les conséquences que peut avoir une mauvaise gestion de la corrélation des données dans une approche de localisation coopérative. Pour cela, nous allons nous intéresser à une simulation de localisation par filtre de Kalman étendu (EKF). Nous allons tester une approche centralisée et une approche décentralisée sur deux robots A et B se déplaçant côte à côte sur une trajectoire plane relativement linéaire et effectuant des mesures de distance entre eux.

La figure 3.9 présente la trajectoire réelle de ces robots et une première estimation en utilisant uniquement des mesures d'odométrie. L'état initial des robots A et B est précis et l'incertitude augmente progressivement à cause de l'imprécision des mesures du modèle d'évolution. Le modèle utilisé ici est celui d'un robot différentiel se déplaçant dans le plan (x, y) : le robot mesure sa vitesse linéaire et angulaire (en z). Les robots démarrent avec une densité dont l'écart-type est de 10 cm en position et 5.7° en orientation. Les mesures d'odométrie ont un écart-type de 30 cm s^{-1} en vitesse linéaire et 4° s^{-1} en vitesse angulaire. Sur cette figure, nous observons le déplacement des robots sans qu'il n'échange aucune information de localisation et sans aucune détection ou mesure extéroceptive.

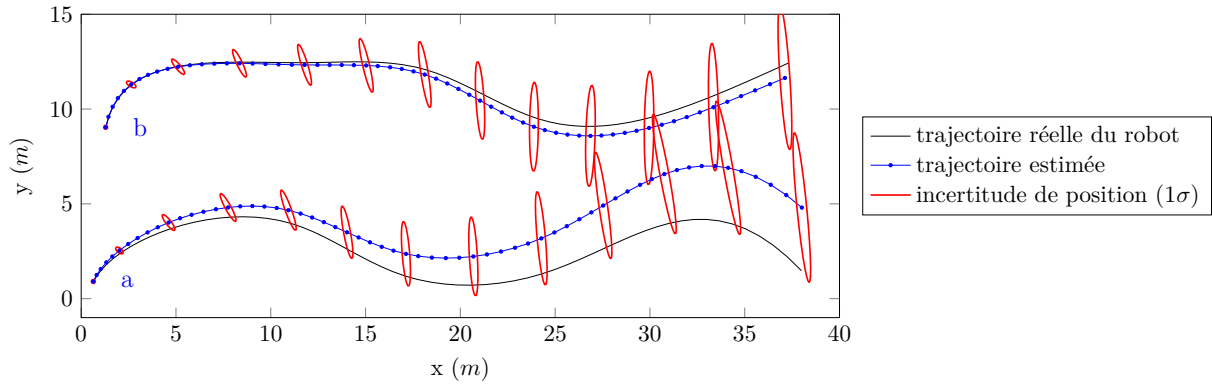


FIGURE 3.9 – Simulation de deux robots mobiles se déplaçant simultanément. Les robots se déplacent de gauche à droite. Les trajectoires estimées en bleue correspondent à la trajectoire des robots estimée en utilisant uniquement les mesures d’odométrie.

Approche centralisée

Dans la stratégie de localisation centralisée, les données des deux robots sont calculées par le même processus. L’état des robots à estimer à chaque étape de l’algorithme va donc correspondre à un vecteur de dimension 6 contenant la position (x, y) et l’orientation des robots A et B :

$$\mathbf{x} = [x_A \quad y_A \quad \theta_A \quad x_B \quad y_B \quad \theta_B]^T. \quad (3.4)$$

La matrice de covariance associée \mathbf{P} est donc de dimension 6×6 . Bien que les états des deux robots soient contenus dans le même vecteur, le calcul du modèle d’évolution reste sans conséquence sur les estimations obtenues puisqu’il n’y a aucun lien entre les deux robots dans ce type de mesures. Ce modèle d’évolution n’est donc rien d’autre qu’une concaténation de deux modèles de robot différentiel. Les vecteurs de mesure \mathbf{u} et de bruit \mathbf{w} sont alors de dimension 4, pour contenir les vitesses linéaires et angulaires et le bruit associé des deux robots.

Nous allons maintenant intégrer la mesure de distance entre les deux robots afin d’améliorer l’estimation de leur position par l’ajout de cette nouvelle information. Les mesures de distance entre les deux robots nécessitent un modèle d’observation spécifique à l’approche centralisée. En effet, celle-ci considère que l’information de position de l’objet observé est transmise en même temps que la mesure de distance. La fonction d’observation se résume donc à une simple distance euclidienne entre les robots :

$$h(\mathbf{x}, v) = \left\| [x_A \quad y_A]^T - [x_B \quad y_B]^T \right\| + v \quad (3.5)$$

Le vecteur de bruit v est ici un scalaire correspondant au bruit de la mesure de distance. L’écart-type de ce bruit est ici configuré à 1 m.

La figure 3.10 présente les résultats de la même simulation que la figure 3.9 mais en ajoutant les mesures de distance fusionnées par l’approche centralisée. Par rapport à la trajectoire

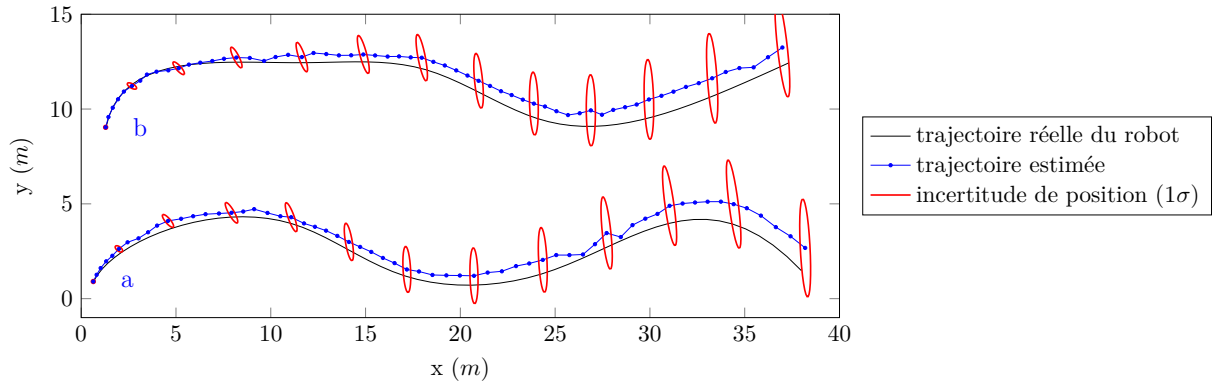


FIGURE 3.10 – Trajectoires obtenues avec l’EKF centralisé pour la même simulation que pour la figure 3.9 mais avec des mesures de distance entre les robots. Les robots effectuent des mesures de distance entre eux. Les trajectoires estimées en bleu ont été calculées avec un filtre de Kalman Étendu centralisé fusionnant l’odométrie et les mesures de distance.

estimée en utilisant uniquement les mesures d’odométrie, nous pouvons remarquer que l’erreur et l’incertitude de position sont plus faibles. L’incertitude reste cependant encore très importante par rapport aux résultats que nous aurions pu obtenir en utilisant des capteurs GPS et une localisation individuelle des robots. Cela vient du fait que la mesure de distance est une mesure relative à un autre objet. Dans le cas de la localisation coopérative, cet objet correspond à un robot dont la position est également incertaine. La mesure de distance apporte donc une quantité d’information qui dépend de l’information de l’objet observé. Étant donné que l’incertitude de l’état des robots augmente avec le trajet, les mesures de distance apporteront de moins en moins d’information. Cela signifie que sur une trajectoire de longueur infinie, l’incertitude de position et d’orientation des robots augmenterait indéfiniment.

Bien que ce type de scénario de localisation n’apporte pas suffisamment d’informations pour éviter le problème de divergence que l’on peut constater en odométrie pûre, l’approche centralisée permet cependant d’exploiter correctement les données de distance entre les robots. Cela permet de moyenniser les erreurs de position des robots et de réduire leurs incertitudes de position et d’orientation. Ce moyennage est observable dans la figure 3.10. L’erreur de position du robot *A* est plus faible qu’en odométrie pûre mais celle du robot *B* est plus importante. L’algorithme est capable de donner une estimation intègre des robots. Si les modèles d’évolution et d’observation étaient linéaires, la solution de ce filtre de Kalman centralisé serait optimale, car il n’y aurait aucune information perdue. Le caractère optimal de la solution impliquerait bien évidemment de considérer également les bruits des mesures comme parfaitement blanc et gaussien. C’est le cas dans cette simulation, mais ça ne l’est jamais en pratique.

Approche décentralisée

Dans une approche décentralisée, chaque robot calcule son estimation d'état sans se préoccuper de l'état des autres robots. Cela signifie, dans le cas de cette simulation, qu'il y a deux instances du filtre de Kalman qui s'exécutent en parallèle. Les deux robots ayant les mêmes caractéristiques, la suite de cette partie décrit les détails d'un seul robot mais les explications sont valables pour les deux.

Comme le filtre de Kalman est individuel, le vecteur d'état ne contient que les informations d'un seul robot :

$$\mathbf{x} = [x \quad y \quad \theta]^T. \quad (3.6)$$

Le modèle d'évolution utilisé est identique au précédent. Les étapes de prédiction ne provoquent donc pas de différences notables entre les états de chaque robot de l'approche décentralisée et les états de l'approche centralisée. S'il n'y avait que des étapes de prédiction, les deux algorithmes donneraient exactement le même résultat. Cependant, cela n'est plus vrai pour les étapes d'observation.

En effet, dans l'approche centralisée, une observation correspond à une simple mesure de distance entre les deux positions décrites dans le vecteur d'état. Dans le cas de la version décentralisée, la position de l'objet mesuré n'est pas directement connue par le robot et doit nécessairement être transmise avec la mesure de distance. Le modèle d'observation utilisé dans cette approche exploite donc la mesure de distance, la position du robot observé et une matrice de covariance modélisant l'incertitude de la mesure et de la position du robot au moment de l'observation. C'est donc dans la construction de ce type de mesure qu'intervient l'échange d'informations entre les deux robots. L'information de position du robot observé est directement extraite de son estimation courante. La position transmise dans la mesure correspond donc aux coordonnées x et y du vecteur d'état et à leur covariance associée.

La figure 3.11 présente les résultats de la simulation en utilisant l'approche décentralisée. La première chose que nous pouvons remarquer est que les incertitudes de position estimées par les deux robots sont très petites. Cela pourrait sembler être une bonne chose puisque cela signifierait que les robots considèrent savoir où ils se trouvent avec beaucoup de précision. Malheureusement les trajectoires estimées ne sont pas intègres. La vérité terrain s'éloigne de plus en plus des ellipses d'incertitudes. Comparée à la figure 3.10 obtenue avec l'approche centralisée, l'erreur de position de l'approche décentralisée est également plus importante. Le filtre de Kalman étendu a donc clairement un défaut pour gérer ce genre de scénario de localisation.

Nous pouvons constater dans cette simulation que les trajectoires des robots divergent assez rapidement. Ceci illustre l'effet de surconvergence dû à la corrélation des données entre les estimations des deux robots. Comme expliqué dans la section 3.4.1, l'algorithme utilisé ne gère pas la corrélation de l'état du robot observé dans l'étape de mise à jour puisque cette information

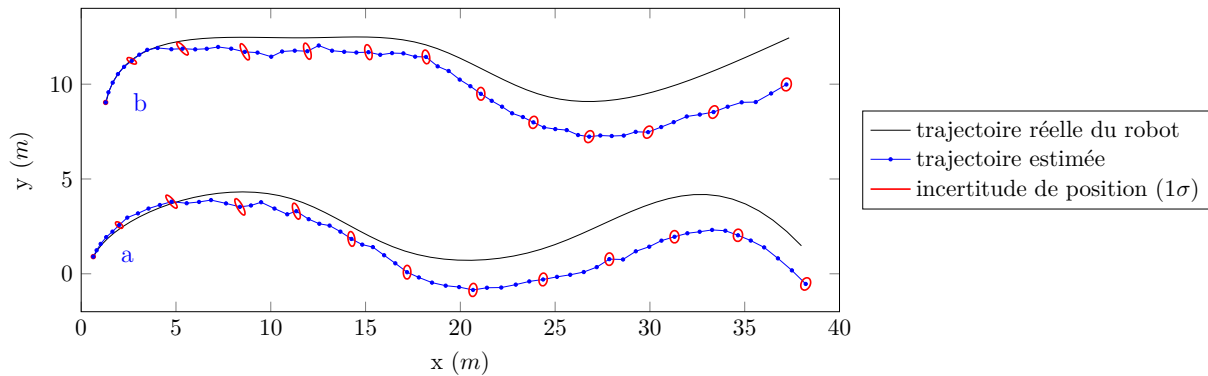


FIGURE 3.11 – Trajectoires obtenues avec l’EKF décentralisé pour la même simulation que pour la figure 3.9. Les trajectoires estimées en bleu ont été calculées avec un EKF individuel pour chaque robot fusionnant l’odométrie et les mesures de distance.

est considérée comme indépendante et entièrement nouvelle. L’estimation de l’état du robot est donc plus précise qu’elle ne devrait l’être.

Cet exemple illustre les problèmes liés à la corrélation des données dans une approche décentralisée d’une localisation coopérative de plusieurs robots. Dans la suite nous allons expliciter plusieurs des solutions proposées dans la littérature en précisant leurs avantages et leurs inconvénients.

3.4.3 Solutions existantes

Il existe plusieurs solutions pour gérer cette corrélation. Ces solutions peuvent se regrouper en deux catégories : celles qui cherchent à éviter de transmettre des informations pouvant conduire à une corrélation non contrôlée (HOWARD et al. 2003 ; ROHANI et al. 2015 ; KARAM et al. 2006) , et celles qui gèrent le problème de corrélation lors de la fusion des données (NERURKAR et al. 2009 ; CARRILLO-ARCE et al. 2013 ; WANASINGHE et al. 2014).

Évitement du problème

Beaucoup de travaux dans le domaine de la localisation coopérative décentralisée se basent sur les algorithmes de fusion les plus courants comme le filtre de Kalman ou le filtre particulaire. Puisque ces algorithmes ne prennent pas en compte la corrélation des mesures, la stratégie adoptée pour gérer la corrélation est de s’assurer que les données utilisées ne soient pas corrélées. Pour cela, il est par exemple possible de n’échanger que des données issues des capteurs des robots.

Dans (HOWARD et al. 2003), les auteurs proposent une solution de localisation où les observations entre robots sont faites à partir de mesures issues de caméras et de lidars. Ces données

capteurs permettent de calculer la position et l'orientation du robot observé dans le référentiel du robot observateur. Ces différences de poses sont alors transmises à tous les robots. À partir de ces données, chaque robot utilise alors un filtre particulière pour estimer son propre état ainsi que celui de tous ses voisins. Dans cette situation, il n'y a pas de problème de corrélation entre les différentes estimations des états des robots, car les données échangées n'ont pas d'autres dépendances que celles des capteurs. L'inconvénient ici est qu'il existe plusieurs estimations de la pose d'un même robot et qu'il n'est pas possible de les fusionner pour en obtenir une unique, plus précise. En effet, les estimations calculées ne peuvent être utilisées par les autres robots sans provoquer de la corrélation. C'est pour cette raison que chaque robot doit calculer une estimation de tous ses voisins pour pouvoir exploiter les différences de poses reçues.

Il existe cependant des solutions ne nécessitant pas le calcul de l'état des robots voisins. La méthode présentée par (ROHANI et al. 2015), se base sur des échanges de mesures GNSS et de distance entre les véhicules. Chaque véhicule calcule uniquement l'estimation de son propre état en utilisant un filtre de Kalman. Les observations utilisées dans l'étape de mise à jour se basent sur les mesures GNSS des robots voisins qui font ici office d'estimation de position de ces véhicules. Grâce aux mesures de distance avec ces derniers, il est alors possible d'en déduire une estimation de position du véhicule observateur (obtenue par filtre particulière) qui pourra alors être fusionnée à l'état. Cette solution se base également uniquement sur des échanges de mesures de capteurs et évite donc le problème de corrélation. Il n'y a également pas besoin de faire d'estimation des états des robots voisins puisque les données GNSS peuvent remplacer l'estimation de l'état des véhicules (l'orientation est ici inutile avec les mesures de distance). L'inconvénient de cette solution est que la qualité des observations venant de l'aspect coopératif se limite à celle des données GNSS.

L'approche de (KARAM et al. 2006) est une solution plus générique, capable de lever cette limitation. Dans celle-ci, il ne s'agit plus de transmettre des données capteurs entre les véhicules, mais des estimations d'états. L'idée développée ici est la construction d'états *locaux* et *fusionnés*. Chaque véhicule calcule, à l'aide d'un filtre de Kalman, une estimation de son état ainsi que celle de ses voisins s'il est capable de les détecter. Ces estimations correspondent à une estimation locale de la flotte de véhicules. Ce sont ces données qui seront transmises aux autres véhicules lorsqu'ils communiqueront entre eux. Après l'échange de données entre les véhicules, chacun est alors en possession d'un ensemble d'estimations locales de la flotte de véhicules. Il est alors possible de fusionner ces états locaux afin de construire un état fusionné plus précis. La corrélation des données entre les véhicules est gérée correctement dans cette situation puisque les données transmises ne sont que des estimations locales. Celles-ci sont donc indépendantes entre elles. L'état fusionné est la seule information faisant intervenir un mélange de données provenant de plusieurs véhicules. Il est donc important que ce résultat ne soit pas utilisé pour calculer un nouvel état. Cette solution a donc l'avantage de pouvoir abstraire la méthode de perception utilisée pour le calcul de l'estimation locale. N'importe quel capteur ou

ensemble de capteurs peut être utilisé pour calculer son état ainsi que l'état des voisins.

L'algorithme souffre cependant d'une limitation sur la qualité de l'aspect coopératif de la localisation. Lorsque un véhicule n'est plus à la portée des autres, son état fusionné ne peut conserver une qualité d'estimation supérieure à son état local. En effet, puisqu'il n'est pas possible d'avoir de dépendance entre un état fusionné et l'état fusionné précédent, la perte d'information est brutale lorsque le robot ne bénéficie plus d'observations extérieures. Pour que ce genre d'observation ne reste pas temporaire, il est nécessaire de le fusionner dans un processus ayant un modèle prédictif.

Algorithmes gérant la corrélation

Le meilleur moyen de fusionner de façon permanente les observations venant de l'aspect coopératif est de n'avoir qu'une seule estimation de l'état a posteriori pour un instant donné. Si nous souhaitons également éviter de devoir calculer localement des estimations des états des autres robots, alors ces observations doivent s'appuyer sur les estimations a posteriori obtenues par les autres robots. La corrélation entre les données est donc inévitable et il faut alors la prendre en compte dans les algorithmes.

Dans le cas des approches basées sur une localisation coopérative distribuée, la corrélation entre les états des robots est gérée correctement puisque la matrice de covariance complète est conservée. C'est le cas notamment de l'approche proposée par (NERURKAR et al. 2009). Un groupe de robots est localisé en utilisant une maximisation a posteriori de l'état de l'ensemble des robots. Le calcul est partagé entre les différents robots grâce à la transmission des informations concernant les estimations de chaque robot ainsi que les informations liant les robots entre eux. Les robots peuvent ensuite localement calculer la partie qui les concerne en utilisant les données reçues, les mesures d'odométrie et les mesure de distance et d'orientation des robots voisins. L'ensemble de ces résultats est alors utilisé par une méthode de gradient conjuguée afin d'obtenir l'estimation a posteriori des états des robots. L'information de corrélation entre ces états est bien conservée et le résultat ne souffre donc pas de sur-convergence. Cette solution repose néanmoins sur la contrainte très forte que la communication entre tous les robots soit parfaite et que le calcul soit synchronisé. Dans la pratique, il est assez difficile de respecter cette contrainte, car les observations entre robots ou les communications peuvent échouer.

Lorsque l'approche est entièrement décentralisée, l'information de corrélation entre les robots est difficile à obtenir et à conserver correctement. Il existe cependant des algorithmes permettant de se passer de cette information. La solution présentée par (CARRILLO-ARCE et al. 2013) s'appuie sur le filtre à intersection de covariance (CIF), dont l'algorithme est détaillé dans la section 2.3.4. Chaque robot calcule sa propre estimation d'état grâce à des mesures d'odométries et des observations de poses obtenues par communication avec les robots voisins. Ces observa-

tions sont contruitent à partir de l'état des robots et des mesures de différence de poses déduites de mesures extéroceptives. Dans cet algorithme, les états des robots sont utilisés dans les observations et sont donc fusionnés plusieurs fois dans l'estimation de l'état des robots voisins. Mais bien qu'il y ait une réutilisation de données, le CIF assure que la solution reste intègre. En effet, l'algorithme considère que les observations sont totalement corrélées à l'état. Dans la pratique, cette corrélation est rarement totale. Le résultat du CIF est donc moins précis qu'une solution prenant en compte la corrélation réelle entre les robots.

Il y a toutefois une alternative au CIF permettant de mieux prendre en compte cette dernière et ainsi améliorer légèrement la qualité de l'estimation. Les approches de (LI et NASHASHIBI 2013; WANASINGHE et al. 2014) se basent sur le filtre à intersection de covariance partitionné (SCIF, section 2.3.5) qui a l'avantage de pouvoir définir plus précisément le niveau de connaissance sur la corrélation d'une observation. Dans le cas d'observations de différence de poses, il est alors possible de définir que la pose d'un robot est une information fortement corrélée et que la mesure de différence de poses est une nouvelle information.

Les approches probabilistes basées sur le CIF ou le SCIF sont donc capables de se passer de l'information de corrélation entre les robots tout en évitant les problèmes de divergence liés à la réutilisation des estimations des états des robots. D'autres méthodes telles que les approches ensemblistes se basent sur une modélisation plus simple que des densités de probabilité pour représenter une estimation d'état. Celles-ci se limitent à une approximation de l'ensemble support d'une densité de probabilité. La manipulation d'ensembles a alors l'avantage de ne pas avoir besoin de gérer l'information de corrélation entre les robots pour ne pas surconverger. Ainsi, l'approche de (Jo et LEE 2008) utilise une méthode de propagation de contraintes afin de calculer les ensembles représentant les états des robots. Dans cet article, les robots sont équipés de capteurs proprioceptifs, de GNSS ou d'un capteur de distance inter-robot. La contrainte apportée par les mesures d'inter-distance permet de réduire les intervalles représentant les états des robots. (LASSOUED et al. 2014) propose une approche similaire mais ne nécessitant pas d'observation entre les robots. Celle-ci s'appuie sur l'hypothèse que, du fait de leur proximité, les robots partagent les mêmes biais dans leurs mesures GNSS. Chaque robot s'occupe de calculer une estimation de l'état de l'ensemble des robots ainsi que les biais de distance des satellites impliqués dans les mesures GNSS. L'aspect coopératif de cette localisation se manifeste alors par le partage des estimations des biais GNSS et des positions des robots. Bien que moins précises que les approches bayésiennes, les approches ensemblistes ont l'avantage de garantir mathématiquement l'intégrité des résultats.

3.5 Gestion de la non-linéarité des observations

La localisation coopérative de robots nécessite un échange d'informations entre les robots mais également une mesure de pose relative inter-robot. Dans nos travaux, cette mesure est issue d'un capteur fournissant une distance entre deux robots identifiés et communicants. Cette mesure de distance inter-robot situe le robot observé sur un arc de cercle dont l'étendue dépend de l'incertitude de position du robot observé. Dans notre approche bayésienne où l'incertitude de pose est représentée par une loi normale, la modélisation de cet arc par ce type de loi comporte une approximation qui peut poser un problème d'intégrité. En effet, si la modélisation par une loi normale se trouve trop éloignée de la forme réelle de l'estimation d'état, celle-ci risque de rapidement diverger, entraînant une perte d'intégrité. Ce genre de situation se présente lorsque le robot se retrouve dans des configurations particulières, mais il n'est pas rare de les rencontrer dans la pratique.

3.5.1 Problème des mesures de distance

Une partie de la difficulté dans la gestion des mesures de distance provient de la variété des formes que peuvent prendre les densités de probabilité de l'estimation de l'état d'un robot après fusion de la mesure. Cela vient du fait que l'estimation d'une mesure de distance prend la forme d'un cercle autour de la balise lorsque l'on projette la mesure dans l'espace 2D de la position du robot. En prenant en compte également le fait que cette mesure est incertaine et suit une loi normale, la forme de la densité possède alors une forme d'anneau. En utilisant des approches basées sur des filtres gaussiens, il est souvent difficile de conserver l'approximation gaussienne pour l'estimation de l'état dans ces conditions. La figure 3.12 présente les différentes situations qu'il est possible de rencontrer. Dans ces figures, l'objectif est d'obtenir la forme réelle de la densité d'un robot statique après une ou deux mesures de distance avec des balises statiques. L'estimation de l'état du robot avant la mesure est considérée comme gaussienne et est représentée par les ellipses orange en pointillés. Dans la figure 3.12a, l'estimation initiale de l'état possède une incertitude faible par rapport à la distance de la mesure. La densité obtenue après la mesure est assez groupée et peut facilement être modélisée par une loi normale. Si cette estimation initiale possédait une incertitude plus importante, la densité résultante peut alors se retrouver dans les situations de la figure 3.12b, dont la forme est incurvée ou la figure 3.12c, dont la distribution est bimodale. Dans les scénarios de localisation où l'on considère que la position initiale du robot est inconnue (incertitude très grande), l'estimation de l'état remplit alors la totalité de l'anneau (figure 3.12d). Si une seconde mesure avec une autre balise est possible, l'estimation de l'état peut alors devenir une distribution bimodale (figure 3.12e).

Bien que les approches gaussiennes ont l'avantage de pouvoir facilement représenter les

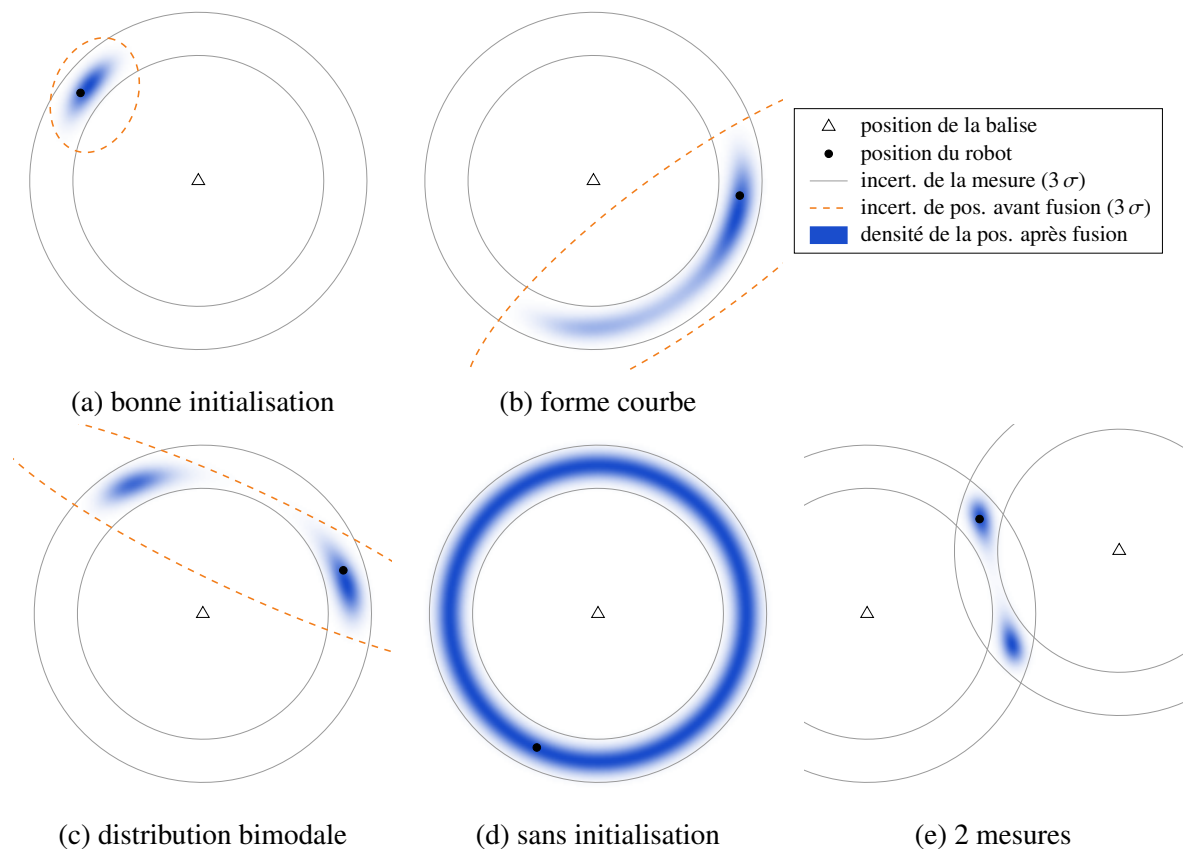


FIGURE 3.12 – Les différents cas d’estimation de la position d’un robot après la prise en compte de mesures de distance.

distributions par leur matrice de covariance et leur vecteur d’espérance, elles ne permettent pas d’obtenir de bonnes approximations des estimations en toutes circonstances. Ce type de filtres reste cependant couramment utilisé pour gérer la localisation de robots par mesure de distance. Nous pouvons par exemple citer (MENSING et al. 2010), (MOHAMMADABADI ET VALAEE 2014), (K. Guo et al. 2017), (Kexin GUO et al. 2017) ou encore (GABELA et al. 2019), dont les approches sont basées sur le filtre de Kalman étendu (EKF). Ces travaux n’évoquent pas les problèmes des approximations des estimations par des gaussiennes puisque leurs scénarios se placent dans des situations proches de la figure 3.12a. En effet, les robots présentés dans ces articles sont équipés de capteurs GNSS, sont initialisés à des positions précises ou ont beaucoup de mesures à leur portée. Ils conservent donc une incertitude de position faible par rapport à la distance des mesures. Si les conditions se dégradent et que l’incertitude de position des estimations augmente, la linéarisation que propose l’EKF ne permet pas d’assurer une bonne estimation de l’état. Nous pouvons le voir dans la figure 3.13 : l’ellipse orange représentant l’estimation *a posteriori* de l’état par un EKF n’inclue qu’une partie de la densité de l’estimation réelle. Les approches basées sur la linéarisation du modèle d’observation des mesures restent donc contraintes aux scénarios dont l’incertitude de position des robots reste durablement faible.

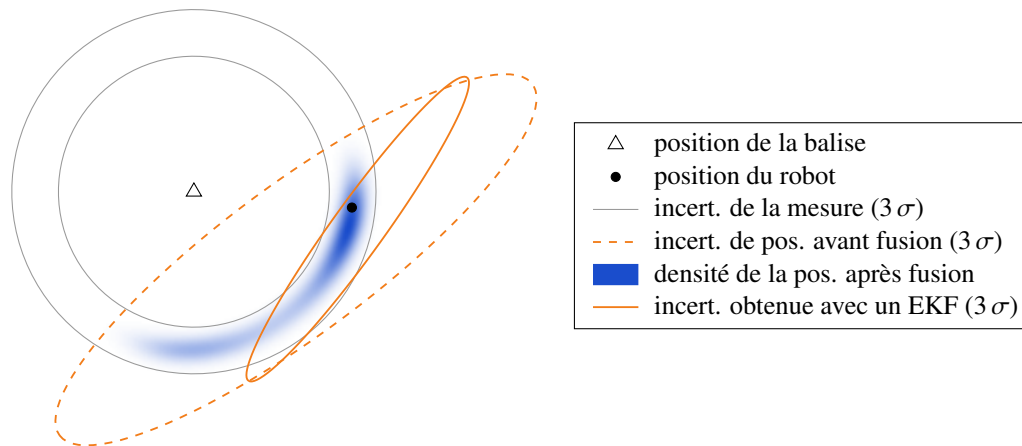


FIGURE 3.13 – Illustration du problème de la linéarisation du modèle d’observation par mesure de distance dans la localisation d’un robot. La densité de probabilité réelle de l’estimation de la position du robot n’est pas représentée correctement par une ellipse obtenue en faisant une simple linéarisation au niveau de l’espérance de l’estimation précédente (résultat d’un EKF).

3.5.2 Solutions au problème

L’ajout de contraintes sur les scénarios de localisation correspond plus à un contournement du problème de modélisation des estimations plutôt qu’à une vraie solution. Cette stratégie a également l’inconvénient de rendre l’approche plus fragile, car la non-linéarité du modèle d’observation peut rapidement avoir un impact sur l’intégrité de la localisation. Il existe cependant une alternative à l’EKF nommée *uncented Kalman filter* (UKF, section 2.3.3) permettant d’obtenir des estimations gaussiennes de l’état en étant plus robuste sur la méthode de linéarisation. Ce filtre est par exemple utilisé dans les travaux (BURCHETT 2019) ou (ARAKI et al. 2019) car il a l’avantage de donner de meilleures approximations gaussiennes de l’estimation réelle de l’état tout en conservant une consommation de ressources machines équivalente à l’EKF. Ces solutions restent cependant limitées aux cas où les estimations réelles sont unimodales et n’ont pas une courbure trop importante, car la modélisation gaussienne ne peut pas représenter convenablement ces formes.

Les situations où la courbure de l’estimation a un réel impact sur la qualité de la localisation sont lorsque l’incertitude de position est grande. C’est le cas par exemple à l’initialisation d’une expérimentation où la position du robot n’est pas connue précisément. Les auteurs de (DJUGASH et SINGH 2009) présente une solution basée sur un EKF, mais dont la paramétrisation de l’état du robot permet de modéliser correctement les estimations courbes ou recouvrant entièrement l’anneau de la mesure de distance. Dans cet article, la position d’un robot n’est pas représentée par la paire de coordonnées (x, y) , mais par 4 paramètres correspondant à une paramétrisation en coordonnées polaires (distance et angle) par rapport à un point (généralement la position de la balise ayant effectué la mesure). Cette configuration permet de modéliser de façon très fidèle les densités des estimations d’état en forme d’arc de cercle ou de cercle complet. Les au-

teurs proposent également une solution simple au problème des distributions bimodales grâce à l'utilisation de deux instances du filtre lorsque ces conditions sont rencontrées. L'un des deux filtres est ensuite éliminé lorsque les observations permettent de revenir à une distribution unimodale. La figure 3.14 représente les modélisations des estimations d'états (dans l'espace 2D de la position et dans l'espace polaire) dans les deux situations décrites ici. Cette méthode a l'avantage de pouvoir gérer correctement tous les cas que l'on peut rencontrer avec les mesures de distance. Cependant, puisque la paramétrisation de l'état est spécifique pour les mesures de distance, elle ne permet pas de gérer efficacement l'évolution du robot ainsi que les observations d'éventuels autres capteurs. Par exemple, lorsque l'estimation de l'état est représentée par un anneau, l'incertitude d'orientation du robot devrait provoquer une déformation de cet anneau lorsque le robot se déplace dans une direction particulière. La paramétrisation en coordonnées polaires ne permet pas de gérer correctement cette augmentation d'incertitude.

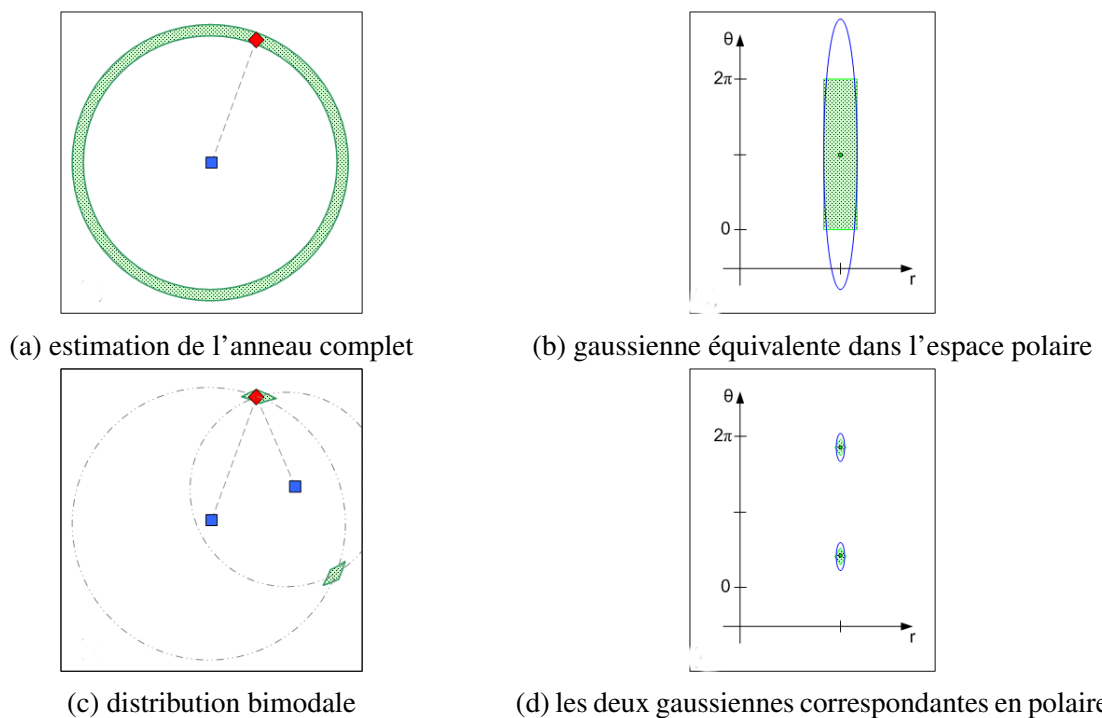


FIGURE 3.14 – Approximation des estimations d'état avec paramétrisation en polaire. La colonne de gauche représente les incertitudes de position dans l'espace 2D et la colonne de droite correspond aux mêmes estimations mais dans l'espace polaire. Les carrés bleus sont les positions des balises et les losanges rouges sont les positions des robots. (source : DJUGASH et SINGH 2009)

Il existe des alternatives aux filtres gaussiens permettant de s'affranchir de la contrainte de linéarité des modèles d'évolution et d'observation. Les distributions représentant les estimations d'états et des mesures peuvent alors être quelconques. Ces types de filtres offrent donc la possibilité de gérer de façon uniforme les problèmes des mesures de distance mais également les problèmes de toute autre méthode d'évolution ou d'observation. Dans le domaine de la localisation de robots mobiles, la famille de filtre la plus couramment utilisée est la famille des filtres particuliers (section 2.4). Nous pouvons par exemple citer (RISTIC et al. 2002), (HOWARD

et al. 2003), (CABALLERO, MERINO, MAZA et al. 2008), (ARAKI et al. 2019) ou encore (HAN et al. 2020). L'un des principaux problèmes de l'utilisation d'un filtre particulaire avec un modèle d'observation par mesures de distance est la dispersion des particules. Il faut souvent un nombre important de particules afin que les zones que couvrent les anneaux des estimations des mesures de distance puissent contenir suffisamment d'informations pour la modélisation des estimations des états. Les auteurs de (CABALLERO, MERINO, MAZA et al. 2008) proposent de réduire l'impact de ce problème en effectuant un échantillonnage particulier lors de l'initialisation. Une autre solution, présentée dans (HAN et al. 2020), permet de limiter la dispersion des particules grâce à une étape de ré-échantillonnage basée sur l'algorithme de colonies de fourmis. Bien qu'il soit possible d'améliorer les performances des filtres particulaires en les spécialisant pour les observations par mesures de distance, ces solutions ne sont pas souvent considérées comme la meilleure solution au problème. C'est le cas notamment dans (ARAKI et al. 2019), où la solution basée sur le filtre particulaire reste moins intéressante que les solutions basées sur l'UKF ou l'EKF.

Bien que la non-linéarité du modèle d'observation par mesure de distance provoque des problèmes de modélisation des estimations d'états importantes, ces situations ne sont rencontrées que sur des courtes durées lors des expérimentations. La plupart du temps, la gestion de la non-linéarité n'est utile que pour l'initialisation des robots lorsque les premières mesures de distance sont faites. L'expérimentation devient ensuite facilement gérable par un EKF ou un UKF, car la précision de l'estimation de position permet de rester dans les cas correctement linéarisables. C'est pourquoi les filtres particulaires ne sont pas les plus adaptés sur le long terme. Il existe cependant une autre famille de filtres non-gaussiens qui a l'avantage de pouvoir modéliser des distributions quelconques et de pouvoir ensuite se simplifier pour tendre vers les performances d'un filtre gaussien. Il s'agit de la famille des filtres à mixture de gaussiennes (section 2.5). Ces méthodes modélisent les distributions par une somme de gaussiennes pondérées afin de représenter au mieux la distribution réelle. Selon les algorithmes utilisés, un filtre gaussien peut n'être qu'un cas particulier d'un filtre à mixture de gaussiennes dont l'ensemble de particules gaussiennes ne contient qu'une seule particule. Ce type de filtre est utilisé dans (CABALLERO, MERINO et OLLERO 2010), (MERINO et al. 2010) et (FABRESSE et al. 2015) afin de localiser des balises observées par mesures de distance avec des objets dont la position est connue. Après une première observation, l'estimation de la position de la balise est initialisée par une mixture de gaussiennes couvrant l'anneau que génère la mesure de distance (figure 3.15a). Le nombre de particules gaussiennes (ou hypothèses gaussiennes) se réduit progressivement avec les mesures de distance effectuées. Nous pouvons voir que l'estimation de l'état passe par les différentes phases : distribution en arc de cercle (figure 3.15c), distribution bimodale (figure 3.15d) et pour finir, une distribution unimodale (figure 3.15e). La suite de l'expérimentation peut alors se dérouler comme un filtre gaussien classique. Contrairement aux filtres particulaires, les méthodes basées sur les filtres à mixture de gaussiennes ont l'avantage de pouvoir gérer les estimations

complexes provoquées par les mesures de distance sans avoir à rester dans une consommation importante de ressources lorsque les situations se simplifient.

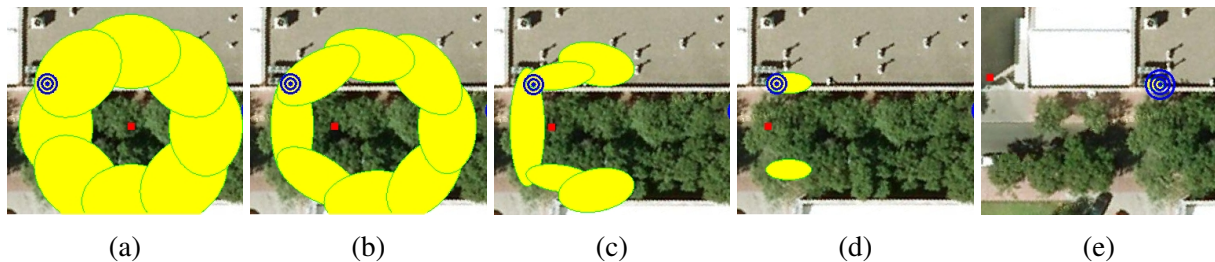


FIGURE 3.15 – Évolution d’une mixture de gaussiennes estimant l’état d’une balise statique. Le cercle représente la position de la balise, le carré rouge représente la position du robot effectuant les mesures et les ellipses jaunes représente l’incertitude des particules gaussiennes de l’estimation de la position de la balise. (source : CABALLERO, MERINO et OLLERO 2010)

3.6 Bilan

Ce chapitre a été consacré à la présentation de divers travaux réalisés dans le domaine de la localisation coopérative (LC) et de mettre en valeur les différents problèmes qui interviennent dans le domaine. La stratégie générale de la gestion de la localisation dépend de l’architecture mise en place pour la communication entre les robots et peut être découpée en deux familles : la LC centralisée et la LC décentralisée. Une majorité des travaux modernes opte cependant pour l’approche décentralisée, car elle offre une meilleure souplesse pour son utilisation dans des applications réelles.

L’un des premiers aspects que les approches de LC doivent gérer est la perception et l’identification des robots avoisinants. Bien qu’il soit possible d’exploiter des capteurs comme les caméras ou les lidars, l’identification reste un problème difficile à traiter. C’est pourquoi les travaux récents privilégient les technologies de radio-communication comme moyen de perception, car elles offrent l’avantage de fournir une identification précise des entités effectuant la mesure.

Pour prendre en compte l’information échangée entre les robots, les observations inter-robots font intervenir les estimations des états des deux robots. Ces interventions ont pour conséquence de générer de la corrélation entre ces états. Contrairement aux approches centralisées, les approches décentralisées n’ont pas de solution simple pour gérer entièrement le problème car l’information de corrélation n’est pas directement disponible. Certains travaux traitant cette problématique se concentrent sur la gestion des données évitant la génération de corrélation grâce à l’utilisation de 2 versions de l’estimation de l’état : une version locale (uniquement le robot) et fusionnée (prenant en compte les informations des autres robots). D’autres

solutions considèrent que les informations échangées sont corrélées et adaptent leur méthode de fusion pour éviter la surconvergence.

Les méthodes de perception basées sur les technologies de radio-communication peuvent souvent être considérées comme une mesure de distance entre les deux entités effectuant la mesure. Le modèle d'observation par mesure de distance à l'inconvénient d'être fortement non-linéaire et difficilement modélisable par les approches classiques basées sur les filtres gaussiens. La plupart des travaux utilisant les mesures de distance se base, malgré tout, sur ce type de filtre, car les non-linéarités du modèle restent cependant gérables lorsque les expérimentations sont faites dans de bonnes conditions. Ils existent cependant des approches développées pour gérer toutes les situations que peuvent provoquer les mesures de distance. Certains travaux adaptent leur modélisation des robots afin de mieux gérer les non-linéarités en restant sur des approches gaussiennes. D'autres utilisent des filtres plus généraux capables de représenter des distributions quelconques.

La localisation coopérative fait donc intervenir des problématiques sur beaucoup d'aspects de la localisation, mais il est souvent difficile de développer une approche permettant de gérer correctement tous ces points. En effet, bien que certains travaux se focalisent sur la localisation coopérative par mesures de distance, **il est très rare de voir des approches capables de gérer simultanément les problèmes de corrélation entre les états des robots et les problèmes de non-linéarité des modèles d'observation.**

Chapitre 4

Approche de localisation coopérative proposée

4.1 Introduction

L'approche que nous avons développée a pour objectif de résoudre les différentes problématiques évoquées dans l'introduction et dans l'état de l'art de la localisation coopérative. Pour rappel, l'idée générale de cette solution est d'être le plus souple possible pour son utilisation avec un nombre quelconque de robots terrestres. Elle doit pouvoir facilement s'intégrer dans des solutions de localisation individuelle afin de leur apporter un aspect coopératif. Les robots impliqués n'ont aucune contrainte sur leur stratégie de localisation individuelle. Ils peuvent donc être hétérogènes sur leurs capteurs ainsi que sur leur modèle d'évolution. L'approche doit également être suffisamment décentralisée pour permettre l'ajout ou la suppression de robots durant une expérimentation sans en perturber son fonctionnement.

La solution proposée peut se greffer facilement à une approche de localisation individuelle existante pour peu qu'elle repose sur un processus bayésien de type filtre de Kalman. Ce type d'approche se décompose généralement en deux grandes étapes : l'évolution de l'état du robot basée sur ses capteurs proprioceptifs et l'étape de mise à jour basée sur l'observation de son environnement. Notre approche consiste à ajouter un nouveau type d'observation issu de la détection des robots voisins et à adapter le filtre de kalman pour lui permettre de tenir compte des éventuelles non-linéarités et de la corrélation des données. L'aspect coopératif est donc vu comme une simple observation supplémentaire conférant à notre approche une certaine souplesse d'utilisation. Il n'est pas nécessaire de connaître le nombre de robots ni même quelles sont les caractéristiques des entités observées. L'objectif de ce chapitre est de décrire le fonctionnement de cette observation coopérative dans l'étape de mise à jour.

La problématique principale d'une observation coopérative est d'être capable d'identifier le robot observé et de pouvoir établir un lien entre leurs états. Dans le cas d'expérimentations avec des robots hétérogènes, l'identification se basant sur l'aspect du robot peut être difficile à gérer de façon robuste. Nous avons choisi d'utiliser la technologie UWB, qui offre le meilleur compromis dans l'équipement des robots et la qualité de l'identification. L'information fournie par ce type de capteur se limite cependant à une mesure de distance entre les deux entités. Puisqu'il s'agit d'une technologie de radio-communication, elle peut également servir de support pour l'échange d'informations entre les robots. Cependant, dans le cas de la fusion de ce type de mesure avec une estimation d'état présentant une forte incertitude, la non-linéarité induite rend difficile la modélisation de l'estimation résultante par un filtre gaussien. C'est pourquoi la solution que nous avons développée se focalise sur la résolution de ces points faibles : la corrélation entre les états des robots et la non-linéarité du modèle d'observation par mesures de distance. Ces points seront traités d'une façon suffisamment générale pour que la solution fonctionne avec tout autre capteur capable de fournir l'identification de l'entité observée et la détection par mesure de distance. L'algorithme proposé s'appuie sur une modélisation des états des robots par une mixture de gaussiennes, et sur une fusion de données par intersection de covariance partitionnée.

Dans un premier temps, ce chapitre présente la solution que nous avons développée ainsi que notre choix de modélisation des robots. Dans un second temps, l'algorithme sera présenté dans sa globalité, en expliquant la conception du filtre bayésien et la gestion temporelle des mesures. Les étapes d'évolutions et de mise à jour seront ensuite détaillées. Celles-ci ont la particularité de gérer le partitionnement de la mixture de gaussiennes en utilisant des solutions spécifiques aux modèles utilisés.

4.2 Modèle utilisé

L'algorithme de localisation coopérative développé ici s'appuie sur des observations entre les robots. Une observation correspond à une mesure de distance avec un robot dont on connaît l'identité. La technologie UWB est capable de fournir ce genre d'information, mais notre solution reste cependant suffisamment générale pour que tout autre capteur pouvant fournir le même type de donnée puisse fonctionner. La solution présentée ne prend donc pas en compte certaines caractéristiques qui pourraient être spécifiques aux capteurs UWB lorsque celles-ci peuvent être gérées dans une étape de traitement au préalable. La localisation des robots peut également faire intervenir d'autres capteurs dont la perception de l'environnement reste individuelle, mais nous nous intéresserons ici uniquement aux éléments liés à l'aspect coopératif de la localisation.

4.2.1 Modélisation de la flotte de robots

Le modèle de localisation proposé se limite aux robots mobiles terrestres. Nous considérons que l'environnement est relativement plat et les robots peuvent se déplacer sur sa surface (les axes x et y du repère global) et s'orienter (angle θ entre les axes x du robot et du repère global). Ce modèle est souvent appelé localisation en *2D et demi*, car les objets se déplacent dans un environnement 2D, mais prend également une dimension supplémentaire pour représenter leur orientation. L'objectif de chaque robot va donc être l'estimation de son état :

$$\mathbf{x}_k = \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix} \quad (4.1)$$

à un instant k dans un repère global. Afin de simplifier au maximum le problème de localisation décentralisée, chaque robot ne s'occupe d'estimer que son propre état. La modélisation des robots n'est donc pas différente d'une modélisation de localisation individuelle. L'aspect coopératif est vu ici comme une simple observation de distance faisant intervenir l'état d'un objet observé que nous appellerons *balise*. Elle se fusionne donc, dans un filtre, de la même façon qu'une observation d'amer par un lidar ou une caméra. Grâce à cette gestion particulière de l'aspect coopératif, il est alors facile de pouvoir l'ajouter à une solution de localisation individuelle existante.

Pour réaliser les observations de distance avec les robots voisins, chaque robot doit être équipé d'au moins un capteur UWB. L'intérêt de pouvoir équiper un robot de plusieurs capteurs UWB est que le robot peut alors obtenir des mesures depuis différents points de son châssis. Cela peut avoir un impact sur la convergence de son estimation d'orientation. La figure 4.1a montre un exemple de situation de localisation coopérative utilisant notre approche. Les robots ont la possibilité d'effectuer des mesures de distance avec les robots à proximité. La limite de portée du capteur ne permet cependant pas de pouvoir détecter les robots les plus éloignés. L'algorithme peut également être étendu aux expérimentations dans des environnements équipés de capteurs de distance statiques (figure 4.1b). Ces objets statiques observables sont alors considérés comme des robots immobiles. Si la position de ces balises est connue avec précision, ils deviennent alors des éléments de référence pour la localisation des robots. Il est également possible de définir des scénarios de localisation coopérative où les balises statiques ne connaissent pas leur position au départ. Ces objets peuvent alors exécuter le même algorithme que les robots, et ainsi améliorer leur estimation de position. Il n'y a donc aucune différence de modélisation entre un objet statique et un robot. L'information d'orientation des balises statiques n'a cependant aucun intérêt puisqu'elle n'intervient pas dans une mesure de distance faite par un robot.

Notre approche a été développée de manière à ce qu'elle reste compatible avec la plupart de nos travaux basés sur des modèles gaussiens. Néanmoins, un simple modèle gaussien n'est pas capable de gérer les fortes non-linéarités des mesures de distance. Il est parfois nécessaire d'uti-

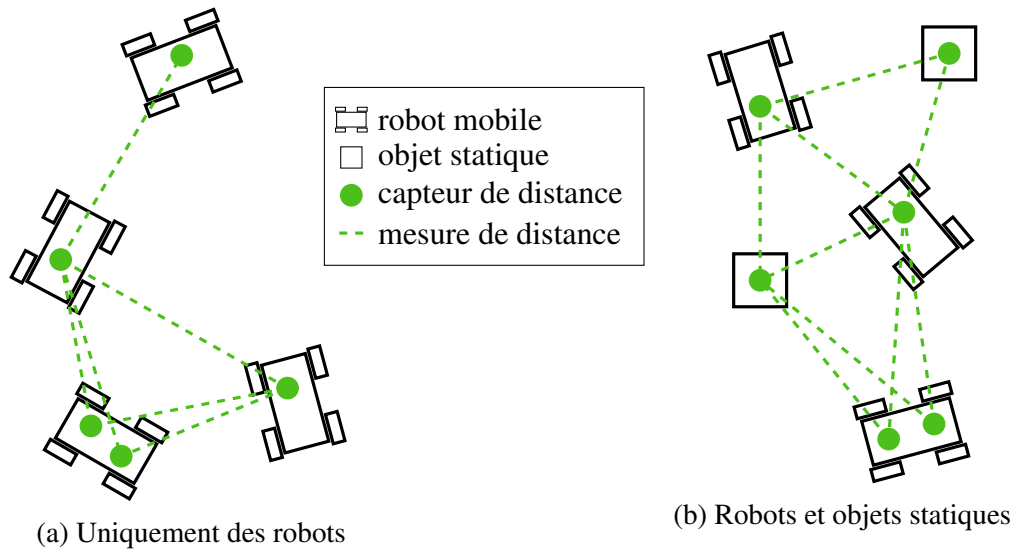


FIGURE 4.1 – Exemples de situation de localisation coopérative par mesures de distance

liser des modèles plus complexes afin de s’approcher de l’estimation réelle de l’état d’un robot. Ces problèmes ne concernent cependant que certains cas particuliers d’une expérimentation. La non-linéarité est importante dans les deux cas suivants :

- les premières observations d’un robot en position inconnue. Ce cas se modélise avec une importante incertitude de position du robot. Lorsque ce dernier effectuera une mesure de distance avec une balise (robot ou objet statique) dont la position est connue, il peut alors déterminer que sa position se trouve dans une zone circulaire autour de la balise. Si l’incertitude de position est modélisée par une gaussienne, elle ne pourra pas recouvrir correctement cette zone. Cette situation peut également se reproduire lorsque le robot se déplace sur un long trajet sans pouvoir faire d’observations.
- l’ambiguïté de la trajectoire d’un robot passant à proximité d’une balise statique. Dans le cas où un robot, ayant une importante incertitude d’orientation, se rapproche de la balise avec une trajectoire rectiligne, les mesures de distance effectuées avec la balise ne permettent pas de résoudre l’ambiguïté entre une trajectoire passant à droite ou à gauche de la balise.

Ce genre de situation, bien que problématique, ne se produit que dans de rares cas. La plupart du temps, les robots ont suffisamment d’informations pour qu’une modélisation gaussienne suffise. Pour que l’algorithme fonctionne dans toutes les situations, il n’est pas envisageable de négliger les cas particuliers. C’est pour cela que notre approche modélise les états des robots par une mixture de gaussiennes avec un nombre dynamique de particules gaussiennes. L’objectif est alors de faire en sorte que les états des robots soient modélisés par une unique particule gaussienne lorsque la non-linéarité est faible, mais de pouvoir augmenter ce nombre pour gérer les ambiguïtés et les zones circulaires. Les performances de l’algorithme peuvent ainsi être proches de celle d’un filtre gaussien dans les situations les plus courantes. Cette modélisation

de l'état a également l'avantage de pouvoir exploiter les observations de distance d'une façon similaire à une approche gaussienne.

4.2.2 Modèle d'observation des mesures de distance

Les observations utilisées dans cet algorithme ne reposent pas uniquement sur la mesure de distance. Il est en effet nécessaire de faire intervenir la position de l'objet observé pour que la mesure puisse être fusionnée à l'état d'un robot. Dans la plupart des approches existantes, cette position est considérée comme connue et parfaitement juste. C'est le cas par exemple de (DJUGASH et SINGH 2009), pour laquelle la localisation d'un robot est calculée grâce à des mesures avec des balises statiques et bien localisées. Nous pouvons également citer (CABALLERO, MERINO et OLLERO 2010), dont l'objectif est l'inverse mais avec une contrainte similaire : les balises doivent se localiser grâce aux observations de distance avec un robot mobile dont la position est connue. La particularité de notre approche est de considérer que la position des objets observés est incertaine mais connue par sa densité de probabilité. Nous nous limitons cependant aux cas où cette densité est une gaussienne. Il s'agit alors du cas où la mixture de gaussienne représentant l'état de la balise observée ne possède qu'une unique particule.

La dernière information importante à prendre en compte dans une observation de balise est la position du capteur de distance sur l'objet observateur. En effet, puisqu'un robot peut être équipé de plusieurs capteurs de distance, c'est cette information qui permettra de départager les données reçues des différents capteurs. Cette information est parfois nommée *bras de levier* et fait référence au décalage entre la position du capteur et la position du référentiel utilisé pour la localisation du robot. Elle peut avoir un impact conséquent sur la mesure lorsque ce décalage est important.

Pour résumer, une observation de balise à un instant k contient les informations suivantes :

- z_{rk} : la mesure de distance (dont le bruit est considéré comme blanc, gaussien et centré)
- σ_{rk} : l'écart-type du bruit de la mesure de distance
- \mathbf{p}_s : la position 3D du capteur dans le référentiel du robot faisant l'observation
- \mathbf{b}_k : l'espérance d'une loi normale représentant la position 3D du capteur de la balise observée dans le référentiel global. Sa position réelle \mathbf{p}_{bk} est inaccessible.
- \mathbf{B}_k : la matrice de covariance liée à la position de la balise observée

Bien que les estimations des états des robots soient calculées en 2D, les positions \mathbf{p}_s et \mathbf{p}_{bk} des capteurs sont exprimés en 3D afin de prendre en compte leur hauteur sur les robots et objets statiques. En fonction de la taille des robots et des objets, la hauteur peut changer. Cela peut avoir un impact non négligeable sur la mesure de distance quand celle-ci est courte. Comme

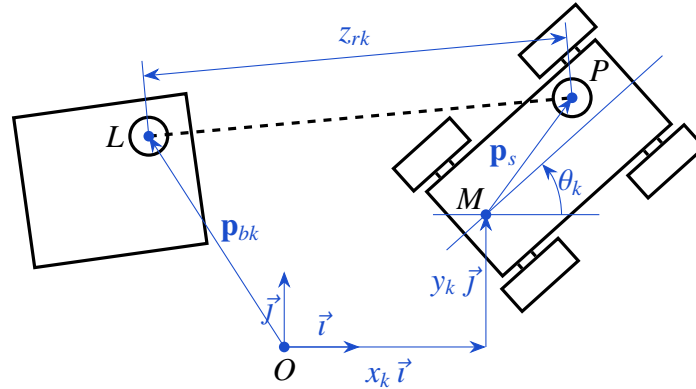


FIGURE 4.2 – Représentation du modèle géométrique utilisé pour la mesure de distance. L'observation est faite par le robot afin de calculer son état au point M . Le point P est la position du capteur. L'objet au point L est la balise observée. Le point O est l'origine du repère global.

présenté dans la section 2.2, l'équation du modèle d'observation est :

$$z_{rk} = h(\mathbf{x}_k, \mathbf{v}_k). \quad (4.2)$$

Le vecteur de bruit \mathbf{v}_k regroupe ici le bruit de la mesure de distance v_{rk} et de l'incertitude de position de la balise observée \mathbf{v}_{bk} . En plus de ces paramètres, la fonction d'observation fait intervenir le bras de levier du capteur et la position de la balise observée :

$$h : \mathbb{R}^3 \times \mathbb{R}^4 \rightarrow \mathbb{R} \quad (4.3)$$

$$(\mathbf{x}_k, \mathbf{v}_k) \mapsto \left\| \begin{bmatrix} x_k & y_k & 0 \end{bmatrix}^T + \text{rot}(\theta_k) \mathbf{p}_s - (\mathbf{p}_{bk} + \mathbf{v}_{bk}) \right\| + v_{rk}$$

avec

$$\text{rot}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.4)$$

La figure 4.2 représente graphiquement les éléments intervenant dans ce calcul. La prise en compte du bras de levier \mathbf{p}_s dans le calcul du modèle d'observation fait intervenir l'orientation de l'observateur. Cela signifie que si un robot connaît précisément son orientation, l'estimation de sa position va varier en fonction de son orientation lorsqu'une nouvelle mesure de distance sera fusionnée. Dans le cas des objets statiques, ce bras de levier doit être nul sur les axes x et y afin d'éviter que l'orientation de l'objet entre en jeu. La position de la balise observée \mathbf{p}_{bk} fait également intervenir de façon indirecte un bras de levier, mais il s'agit ici de celui de l'objet observé. En effet, l'information de position obtenue lors d'une observation d'une balise ne correspond pas à la position de la balise mais à la position de son capteur de distance. Si cette balise est un robot avec un important bras de levier pour son capteur, l'estimation de sa position sera différente de l'estimation de la position de l'état du robot. L'incertitude d'orientation de ce dernier va également augmenter la covariance de la position de son capteur.

L'utilisation d'une position de balise incertaine a une influence sur la forme de la densité de probabilité (*probability density function*, PDF) représentant l'observation. Habituellement, celle-ci possède une forme d'anneau dont l'épaisseur dépend de l'incertitude de la mesure de distance. La PDF de cette mesure correspond à une gaussienne dépendant de la distance r entre le robot et la balise observée :

$$p_r(r) = g(r; z_{rk}, \sigma_k^2) \quad (4.5)$$

avec $g(\mathbf{x}; \boldsymbol{\mu}, \mathbf{P})$, une fonction gaussienne de variable x , d'espérance $\boldsymbol{\mu}$ et de covariance \mathbf{P} . En utilisant la fonction d'observation, il est alors possible d'exprimer cette PDF dans le repère global. La variable n'est alors plus une distance r mais un vecteur de position $\mathbf{x} = [x \ y]^T$. L'expression de la vraisemblance devient alors :

$$p_r(\mathbf{x}) = g(\|\mathbf{x} + \mathbf{c} - \mathbf{p}_{rk}\|; z_{rk}, \sigma_k^2). \quad (4.6)$$

avec \mathbf{c} un vecteur représentant le bras de levier d'un capteur. Une représentation graphique de cette PDF est présentée dans la figure 4.3a. Le dégradé de couleur représente la densité. Plus la couleur est forte, plus la densité est importante. Nous pouvons voir dans cette carte de densité que les points ayant la même distance par rapport à la position de la balise possède une valeur similaire de densité. Dans le cas de notre approche, une observation de balise fait intervenir une seconde variable aléatoire représentant l'incertitude de la position de cette balise. Sa PDF est la suivante :

$$p_b(\mathbf{x}) = g(\mathbf{x}; \mathbf{b}_k, \mathbf{B}_k). \quad (4.7)$$

La forme complète de l'observation d'une balise est la somme de ces deux variables aléatoires. La PDF qui en résulte correspond donc au produit de convolution de ces deux densités :

$$p_z(\mathbf{x}) = p_r(\mathbf{x}) * p_b(\mathbf{x}). \quad (4.8)$$

La figure 4.3b représente la carte de densité correspondant à cette PDF. Nous pouvons voir que l'incertitude de position de la balise est très grande dans une direction. Cela a pour conséquence d'élargir l'anneau de la carte de densité précédente dans la direction principale de cette ellipse.

L'algorithme de localisation coopérative se base sur cette modélisation des observations de balises afin d'établir les règles permettant de créer de nouvelles particules gaussiennes dans l'estimation des états des robots. L'idée repose sur le découpage de cette PDF complexe en une somme de gaussiennes capable de la recouvrir. Contrairement à une approche de filtrage à base de mixtures de gaussiennes ou utilisant de simples particules pondérées, celle-ci est spécialisée dans le cas où les PDF des observations correspondent à celles décrites ici. Cela permet de simplifier le problème de la génération des mixtures de gaussiennes utilisées par les robots.

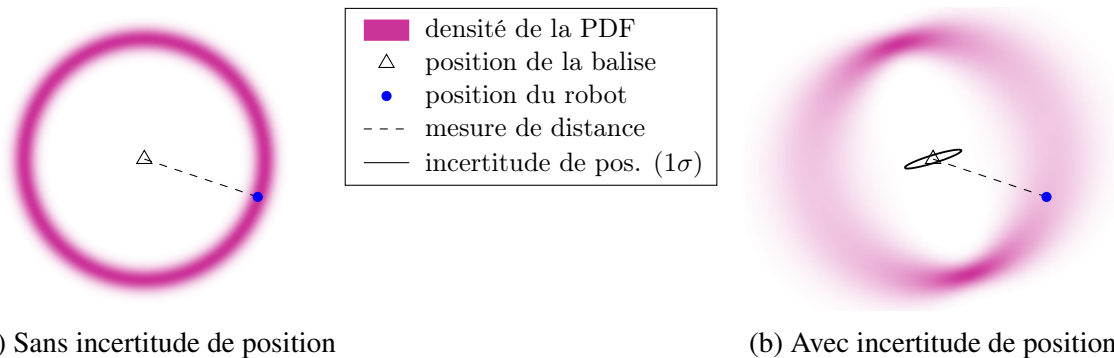


FIGURE 4.3 – Cartes de densité de probabilité de la mesure de distance entre un robot et une balise observée dans l’espace 2D représentant la position des objets.

4.3 Processus de fusion coopératif

Plusieurs problématiques sont résolues grâce au choix de la modélisation de la flotte de robots ainsi que de leurs outils d’observation coopérative par mesure de distance. L’objectif de l’algorithme de localisation va être d’exploiter les modèles présentés et de prendre en compte les dernières problématiques évoquées. L’approche développée est très proche de celle que l’on peut retrouver dans un filtre gaussien. Cela vient du fait que la plupart du temps, l’estimation de l’état peut être considérée comme une simple gaussienne. Les filtres comme le filtre à somme de gaussiennes (GSF, section 2.5.1) ou le filtre particulaire à somme de gaussiennes (GSPF, section 2.5.2) sont conçus pour conserver un même nombre de particules gaussiennes. Le filtre à mixture de gaussiennes que nous avons développé est assez proche du GSF puisqu’il s’appuie également sur une linéarisation des modèles d’évolution et d’observation. Cet algorithme peut également être vu comme un filtre de Kalman étendu, appliqué sur chaque particule. Notre approche se différencie cependant par sa gestion dynamique du nombre de particules et sa méthode de fusion des données. Celle-ci met en place des étapes de création, de suppression et de fusion de particules. Pour être également capable de gérer la corrélation des données des observations de balises, la méthode de fusion utilisée est l’intersection de covariance partitionnée. L’algorithme prend également en charge la latence des mesures à fusionner due à la communication réseau et aux éventuels temps de traitement.

4.3.1 Présentation du filtre

La plupart des approches de fusion de données se base sur l’utilisation de modèles d’évolution et d’observation. C’est le cas notamment de toutes les approches se basant sur une estimation bayésienne (présenté dans la section 2.2). Le modèle d’évolution est utilisé dans l’étape de prédiction pour la construction de l’estimation de l’état *a priori* $\mathbf{x}_{k|k-1}$ à un instant k , et le modèle d’observation intervient dans l’étape de mise à jour afin de construire l’estimation de

l'état *a posteriori* $\mathbf{x}_{k|k}$. Les présentations théoriques considèrent que ces étapes s'alternent en permanence avec le temps. Dans la pratique, cela ne se passe pas toujours de cette façon. En robotique mobile, le modèle d'évolution fait bien souvent intervenir des mesures de capteurs proprioceptifs. Une étape de prédiction est alors calculée uniquement après réception d'une mesure. La cadence des mesures proprioceptives étant souvent plus élevée que les observations extéroceptives, le filtre peut donc enchaîner plusieurs étapes de prédiction sans faire intervenir d'étapes de mise à jour. L'algorithme est alors considéré comme évènementiel : les évènements correspondent aux mesures d'évolution ou d'observation. Le type de l'évènement va influencer sur la façon dont les étapes de prédiction et de mise à jour sont appliquées.

Notre approche se décompose donc en deux sous-algorithmes regroupant les différentes étapes à effectuer pour l'évolution ou la mise à jour, et dont le fonctionnement sera détaillé dans les prochaines sections de ce chapitre. Leur objectif est d'obtenir une estimation *a priori* ou *a posteriori* sous forme de mixtures de gaussiennes. Comme présenté dans la section 2.5, ces mixtures représentent un ensemble de particules gaussiennes dont chacune est modélisée par un poids, un vecteur d'espérance et une matrice de covariance. Avec N particules, l'estimation de l'état *a priori* à l'instant k correspond à la mixture $\{(w_{k|k-1}^i, \mathbf{x}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i), i \in \llbracket 1, N \rrbracket\}$ et l'estimation de l'état *a posteriori* à l'instant k correspond à la mixture $\{(w_{k|k}^i, \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i), i \in \llbracket 1, N \rrbracket\}$.

Le schéma présenté dans la figure 4.4 résume les différentes actions impliquées dans les deux étapes du filtre. Contrairement à un filtre classique, la particularité de cette approche est l'ajout d'opérations particulières permettant la gestion d'un nombre dynamique de particules dans les mixtures représentant les estimations d'état. En effet, l'étape d'évolution s'occupe de tester l'incertitude d'orientation des différentes particules de l'état et va en créer de nouvelles (grâce à un partitionnement) lorsque cette incertitude est trop importante. L'estimation *a priori* est ensuite obtenue en calculant une prédiction de l'état grâce au modèle d'évolution. Dans le cas de l'étape de mise à jour, les particules de l'état sont soumises à un test de linéarité (basé sur un critère particulier) permettant de choisir entre l'application d'une mise à jour classique ou une mise à jour spécifique générant de nouvelles particules dans l'estimation *a posteriori* de l'état. Par la suite, une opération supplémentaire permet de réduire le nombre de particules grâce à l'élimination ou la simplification de groupe de particules de l'estimation de l'état.

4.3.2 Les étapes de l'algorithme

L'algorithme 1 présente les différentes étapes impliquées dans la phase d'évolution du filtre. Il se découpe en deux parties : le partitionnement des particules (instructions ?? à ??) et l'application du modèle d'évolution sur celles-ci (instructions ?? à ??). Le partitionnement sert à créer une mixture de gaussiennes intermédiaire dans laquelle, les particules dont l'incertitude d'orientation était importante ont été divisées en plusieurs particules tout en conservant un

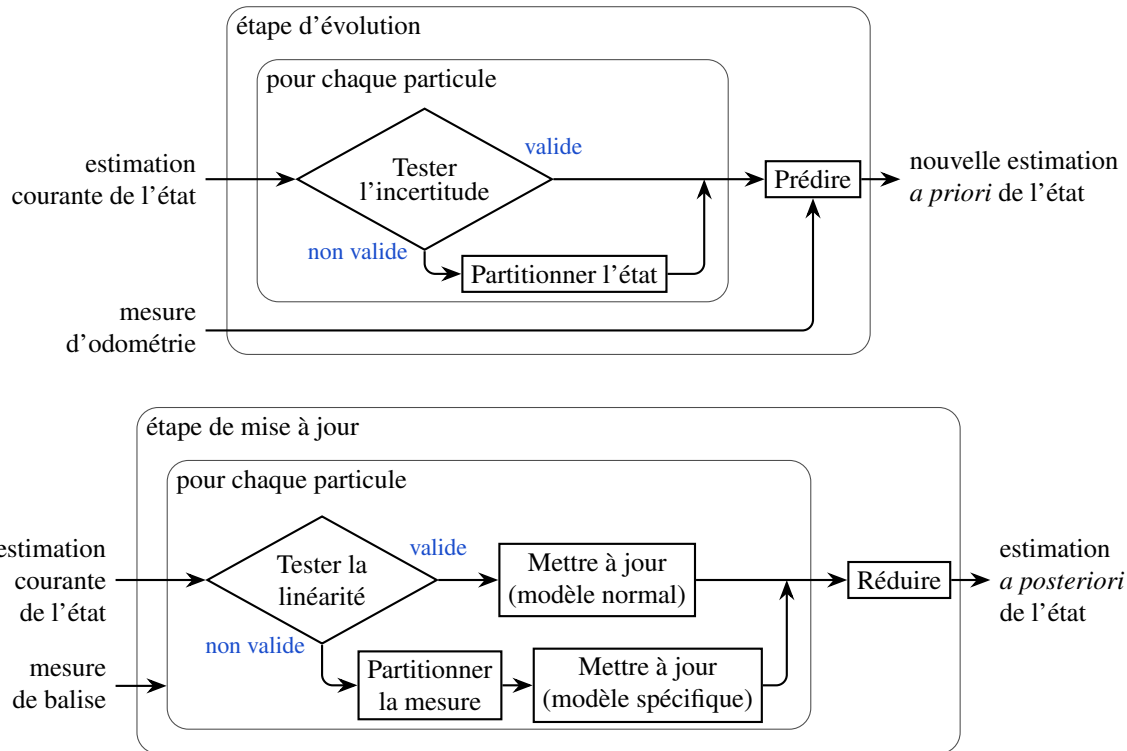


FIGURE 4.4 – Schéma représentant les étapes impliquées dans le calcul d'une nouvelle estimation d'état par notre approche.

bon recouvrement de la densité de probabilité initiale. Cette mixture intermédiaire est ensuite utilisée pour générer la mixture représentant l'estimation *a priori* de l'état. Ce calcul est une adaptation de l'étape d'évolution du filtre à intersection de covariance partitionné (SCIF) pour une utilisation avec un état représenté par une mixture de gaussienne.

L'algorithme 2 correspond aux étapes de la phase de mise à jour du filtre. Il fonctionne pour les observations de balises mais également pour d'autres types d'observation qui ne concernent que des informations propres au robot. Cependant le partitionnement d'une particule (instruction ??) est spécifique aux mesures de distance avec une balise. Il survient lorsque le test de non-linéarité (instructions ?? à ??) entre la mesure et une particule de l'état indique que la linéarisation est mauvaise. Ce test correspond à l'utilisation d'un critère permettant de quantifier l'écart entre l'approximation calculée et la vraisemblance réelle de la particule. De façon générale, il est possible d'utiliser la divergence de Cullback-Leibler comme critère de comparaison, mais il est rarement possible de trouver une forme analytique adéquate. Nous verrons dans la section 4.5.1 le critère que nous avons développé spécifiquement pour les observations de balises.

S'il n'y a pas de problème de linéarité, la mesure est fusionnée à la particule en cours de traitement en utilisant la procédure d'une mise à jour d'un SCIF avec le modèle d'observation présenté dans la section section 4.2.2 ainsi qu'un calcul de pondération. Le nombre de particules de l'estimation de l'état peut fortement augmenter après le partitionnement de l'étape de mise

Entrée : L'ensemble \mathcal{I} des particules de l'estimation précédente

$$\{(w_{k-1|k-1}^i, \mathbf{x}_{k-1|k-1}^i, \mathbf{P}_{k-1|k-1}^i)\}_{i \in \mathcal{I}}$$

Entrée : La mesure d'évolution

Sortie : L'ensemble \mathcal{O} des particules de l'estimation de l'état *a priori*

$$\{(w_{k|k-1}^i, \mathbf{x}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i)\}_{i \in \mathcal{O}}$$

Créer un ensemble de particules \mathcal{T} vide.

pour chaque particule i de l'ensemble \mathcal{I}

si l'incertitude d'orientation de la particule i est trop importante **alors**

 Partitionner la particule i en un nouvel ensemble de particules

 Ajouter les particules obtenues à l'ensemble \mathcal{T}

sinon

 Ajouter la particule i à l'ensemble \mathcal{T}

fin

fin

Créer un ensemble de particules \mathcal{O} vide.

pour chaque particule i de l'ensemble \mathcal{T}

 Calculer une nouvelle particule grâce à la particule i , la mesure d'évolution et les formules d'évolution du SCIF

 Ajouter cette nouvelle particule à l'ensemble \mathcal{O}

fin

Algorithme 1 : Étape d'évolution

à jour. Il est cependant possible de réduire ce nombre lorsque les particules sont dans certaines configurations particulières :

- si le poids d'une particule est très faible, elle n'a alors que peu d'influence dans la mixture de gaussiennes. Ces particules sont donc supprimées (instruction ??).
- si des particules sont très proches les unes des autres, il est alors possible de les fusionner afin d'éviter les calculs similaires. Pour cela, nous appliquons un algorithme de réduction de mixture de gaussiennes (instruction ??).

Les deux algorithmes présentés ici regroupent l'essentiel des étapes impliquées dans le filtre développé. Lorsqu'une nouvelle mesure sera disponible, le choix de l'algorithme utilisé va dépendre du type de mesure. Les mesures d'odométries seront traitées avec l'algorithme 1 et les autres mesures seront gérées dans l'algorithme 2. Cependant, les temps de traitements des données ne peuvent être négligés dans les approches de localisation coopérative à cause du temps de latence des outils de communication sans fil et du déplacement des robots.

4.4 Calcul de l'évolution

L'objectif de l'algorithme d'évolution est de fournir une estimation *a priori* de l'état du robot en se servant d'un modèle d'évolution des robots. Puisque la plupart des robots ne sont

Entrée : L'ensemble \mathcal{I} des particules de l'estimation *a priori* $\{(w_{k|k-1}^i, \mathbf{x}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i)\}_{i \in \mathcal{I}}$

Entrée : La mesure (observation de balise ou autre)

Sortie : L'ensemble \mathcal{O} des particules de l'estimation *a posteriori* $\{(w_{k|k}^i, \mathbf{x}_{k|k}^i, \mathbf{P}_{k|k}^i)\}_{i \in \mathcal{O}}$

Créer un ensemble de particules \mathcal{O} vide.

pour chaque particule i de l'ensemble \mathcal{I}

si c'est une mesure de balise **et** la non-linéarité de la mesure sur la particule i est trop importante **alors**

 Générer un ensemble de particules \mathcal{M} correspondant à la conversion de la mesure de balise dans le plan 2D.

 Calculer un nouvel ensemble de particules représentant la fusion entre la particule i et chaque particule de \mathcal{M} .

 Ajouter les particules obtenues à l'ensemble \mathcal{O}

sinon

 Appliquer l'étape de mise à jour du SCIF sur la particule i

 Calculer le nouvel état de la particule i grâce à la mesure et les formules d'observation du SCIF.

 Ajouter cette nouvelle particule à l'ensemble \mathcal{O}

fin

fin

Supprimer les particules de l'ensemble \mathcal{O} dont le poids est négligeable

Appliquer l'algorithme de réduction de mixture sur l'ensemble \mathcal{O}

Normaliser le poids des particules de l'ensemble \mathcal{O}

Algorithme 2 : Étape de mise à jour

pas holonomes, l'évolution de leur position est fortement contrainte par leur orientation. Le problème des modèles de direction courants des robots mobiles, comme le modèle différentiel ou encore le modèle Ackermann, est qu'ils ne sont pas linéaires par rapport à l'orientation. De plus, cette erreur se cumule avec le temps. Cela signifie que lorsque l'incertitude d'orientation est importante, une modélisation gaussienne ne sera pas une bonne approximation de l'état du robot. C'est pourquoi notre solution met en place une étape de partitionnement de la mixture de gaussiennes représentant l'état afin d'assurer une meilleure représentation de la densité de probabilité de l'état d'un robot lorsqu'il effectue un long trajet sans observation externe.

4.4.1 Partitionnement à l'étape d'évolution

Si aucune observation ne permet de corriger l'incertitude d'orientation d'un robot, cette incertitude va alors se propager à l'incertitude de position lorsque le robot se déplacera. La forme de la densité de probabilité de l'estimation de position du robot peut alors prendre une forme circulaire dans les cas où l'incertitude d'orientation n'évolue pas. La figure 4.5 illustre un exemple de forme que peut prendre cette densité avec un ensemble de particules d'états du robot tiré aléatoirement suivant la distribution de départ. L'estimation de l'état initial du robot

est modélisée par une loi normale ayant une très bonne précision sur sa position mais une importante incertitude d'orientation. Le robot se déplace sur une courte trajectoire et fait évoluer l'état des particules en leur appliquant le modèle d'évolution du robot. À la fin de la trajectoire, nous pouvons voir que ces particules se trouvent sur un arc de cercle dont le centre est la position initiale du robot. La forme circulaire représentée par ces particules est difficilement représentable par une densité de probabilité gaussienne. Nous pouvons le remarquer par l'approximation gaussienne obtenue par les étapes d'évolution d'un filtre de Kalman (représentée en orange dans la figure 4.5). Graphiquement, ce filtre ne peut donner au mieux qu'une ellipse tangente à une partie de l'arc de cercle, et donc couvrir qu'une petite partie de l'estimation obtenue avec un filtre gérant cette non-linéarité. Dans les situations où l'orientation du robot est considérée comme inconnue, cela peut se modéliser par une incertitude d'orientation suffisamment importante pour couvrir les 360° . La forme de la densité de probabilité peut alors être circulaire, et son approximation par une gaussienne n'est plus utilisable.

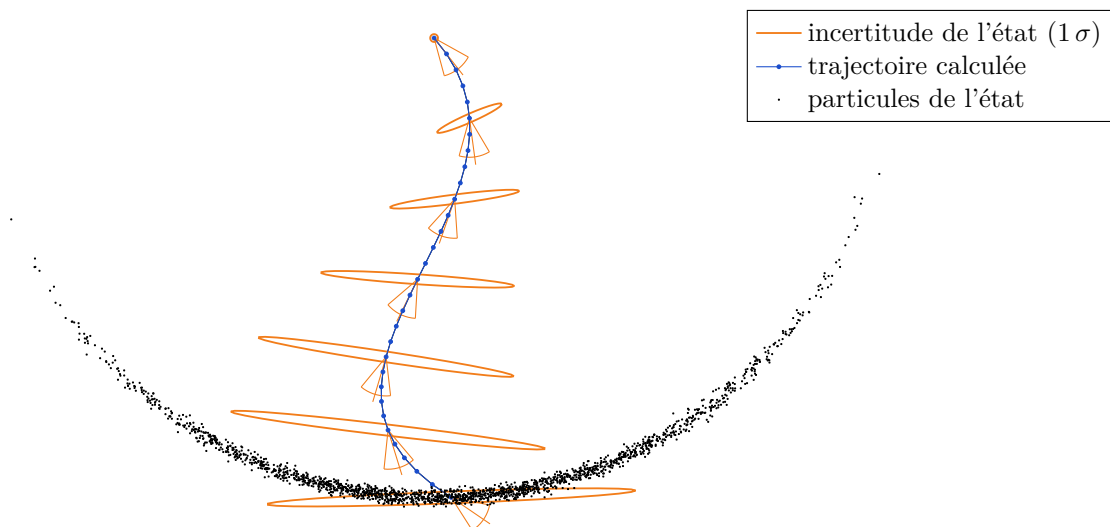


FIGURE 4.5 – Représentation de la non-linéarité du modèle d'évolution par un ensemble de 2000 particules tirées aléatoirement suivant la distribution initiale de l'état. Le robot est initialisé avec une importante incertitude d'orientation et effectue une courte trajectoire (en bleu). L'évolution d'une estimation de l'état par un filtre de Kalman est représentée en orange. Les ellipses représentent l'incertitude de position (à 1σ) et les cônes représentent l'incertitude d'orientation (à 1σ).

En exploitant la modélisation de l'état par une mixture de gaussienne, notre approche propose une approximation de la densité de probabilité (PDF) réelle de l'état du robot grâce à la création de nouvelles particules gaussiennes par une procédure de partitionnement de l'état. L'objectif est alors de limiter l'incertitude d'orientation des particules à une valeur maximale pour laquelle, la non-linéarité provoquée devient négligeable. Cela fonctionne de la façon suivante : supposons que nous ayons une particule $(w, \mathbf{x}, \mathbf{P})$ provenant de l'estimation courante de l'état. Appelons θ l'espérance son orientation et σ l'écart-type de celle-ci. Cette particule est considérée comme susceptible de provoquer une perte d'intégrité dans l'estimation de l'état si

σ dépasse un seuil maximal σ_{\max} d'incertitude d'orientation. Lorsque cette situation se produit, la particule est alors utilisée pour la génération d'un ensemble de N nouvelles particules gaussiennes. Elle sont réparties uniformément de façon à couvrir au mieux la PDF de la particule initiale. Ces particules possèdent toutes un même écart-type σ_{new} dont la valeur ne dépasse pas le seuil σ_{split} . Ce seuil est plus faible que σ_{\max} afin d'éviter que les nouvelles particules générées ne se re-partitionnent trop fréquemment avec l'augmentation de l'incertitude d'orientation.

La construction des nouvelles particules consiste aux calculs de leur nombre N , de leur écart-type σ_{new} , de leur orientation θ^i et de leur poids w^i . Afin de couvrir la majorité de la PDF de la particule initiale, les particules sont réparties en se basant sur une version tronquée du support de leur PDF (figure 4.6). En effet, puisqu'une gaussienne possède un support infini, il serait nécessaire de créer une infinité de particules afin de la couvrir avec des gaussiennes de plus faible variance. Le support tronqué se limite à un intervalle $[\theta - C_{\text{supp}} \sigma; \theta + C_{\text{supp}} \sigma]$ autour de l'espérance θ , avec C_{supp} un coefficient permettant de choisir la taille de la zone couverte. Si nous prenons $C_{\text{supp}} = 3$, nous couvrons 99.7 % de la probabilité de l'estimation. À l'inverse, si nous souhaitons couvrir 99 % de la probabilité, la loi du χ^2 nous permet de déterminer que $C_{\text{supp}} = 2.57$. Grâce à l'utilisation de supports finis, il est alors trivial de répartir les nouvelles particules afin de couvrir le support de la gaussienne initiale. Puisque la PDF des particules n'est pas uniforme sur leur support tronqué, l'approximation de la gaussienne initiale est de meilleure qualité lorsque ces supports se superposent. Le coefficient $C_{\text{mix}} \in [0; 1[$ permet de contrôler le taux de recouvrement de ces supports.

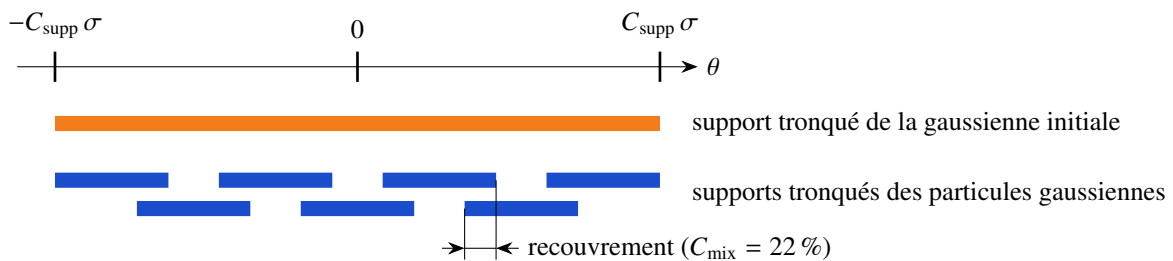


FIGURE 4.6 – Partitionnement du support de la gaussienne correspondant à l'incertitude d'orientation d'une particule de l'estimation de l'état, en un ensemble de supports de taille limitée. Ces supports sont utilisés par les particules gaussiennes approxinant la gaussienne initiale.

Le calcul du nombre de particules N à créer se fait de la façon suivante :

$$N = 2 \left\lceil \frac{\sigma - C_{\text{mix}} \sigma_{\text{split}}}{2(1 - C_{\text{mix}}) \sigma_{\text{split}}} \right\rceil + 1 \quad (4.9)$$

Le résultat de ce calcul est toujours un nombre impair afin d'éviter d'avoir un saut d'orientation dans la particule de plus grand poids dans la mixture de gaussiennes. En effet, lorsque ce nombre est impair, il y aura toujours une particule dont l'espérance de l'orientation correspondra à celle de la gaussienne initiale. À partir du nombre de particules, il est ensuite possible de calculer

leurs écart-types et leurs espérances :

$$\sigma_{\text{new}} = \frac{\sigma}{N + C_{\text{mix}}(1 - N)} \quad (4.10)$$

$$\theta^i = \theta + C_{\text{supp}}(\sigma_{\text{new}} - \sigma + 2(i-1)(1-C_{\text{mix}})\sigma_{\text{new}}) \quad \forall i \in \llbracket 1, N \rrbracket \quad (4.11)$$

Ces résultats sont utilisés pour la construction du troisième paramètre du vecteur d'état et de la matrice de covariance de chaque nouvelle particule. Les autres champs (position en x et y) sont copiés de la particule initiale de l'estimation courante de l'état. Afin de représenter correctement la gaussienne initiale, les nouvelles particules sont pondérées en suivant la PDF de cette gaussienne. Les poids w^i des particules sont calculés de la façon suivante :

$$w^i = c_n g(\theta^i; \theta, \sigma) \quad \forall i \in \llbracket 1, N \rrbracket \quad (4.12)$$

avec $g(\theta^i; \theta, \sigma)$ la PDF de l'orientation de la particule initiale exprimée aux différents angles des nouvelles particules. Le coefficient c_n sert à normaliser les poids de telle sorte que leur somme équivaut au poids w de la particule initiale.

La figure 4.7 présente un exemple de PDF obtenu pour une particule initiale dont l'incertitude d'orientation a un écart-type de 50° . Les supports tronqués couvrent 99 % de la probabilité, le recouvrement des supports est de 60 %, et les particules créées ont une orientation dont l'écart-type est inférieur ou égale à 20° . La mixture de gaussienne générée approxime la gaussienne initiale avec 5 particules. Nous pouvons voir que la forme de la PDF résultante n'est pas parfaite. En effet, la courbe possède une ondulation provoquée par les maximums des gaussiennes des particules. Les ondulations peuvent être réduites en rapprochant les particules (et donc en augmentant le taux de recouvrement), mais cela implique cependant la création d'un plus grand nombre de particules. Il est donc nécessaire de trouver le bon compromis dans le réglage du taux de recouvrement C_{mix} afin d'assurer la bonne qualité de l'approximation tout en limitant l'impact du partitionnement sur le temps de calcul.

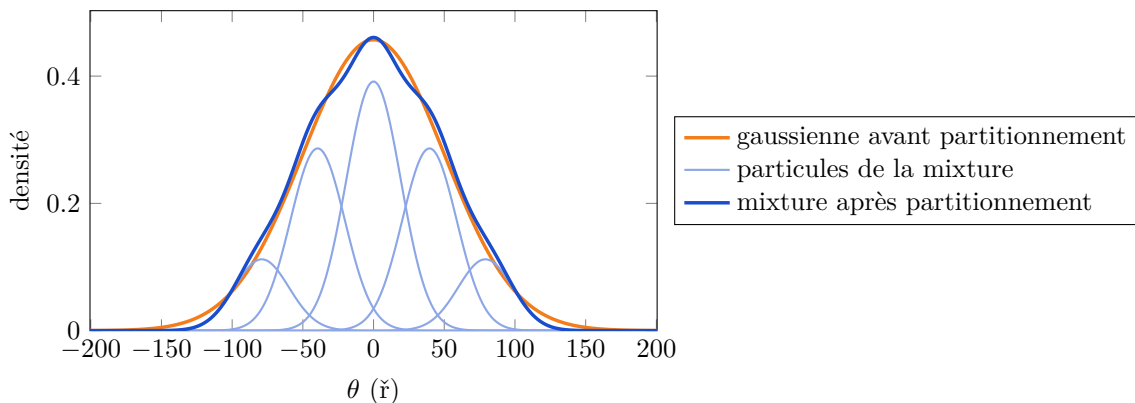


FIGURE 4.7 – Approximation d'une gaussienne représentant l'orientation d'une particule de l'estimation d'état par une mixture de 5 gaussiennes ayant des écart-types limités.

La figure 4.8 montre le résultat de cette configuration du partitionnement sur l'état complet d'un robot après une courte trajectoire en odométrie pure. Nous pouvons constater que la mixture de gaussiennes représentant l'état est proche de la forme circulaire présentée dans la figure 4.5. L'état du robot n'est partitionné qu'une seule fois avant la prise en compte de la première mesure d'odométrie. Les particules créées évoluent ensuite dans différentes directions tout en restant dans cette disposition circulaire. Puisque les mesures d'odométrie apportent également du bruit dans le modèle, l'incertitude d'orientation des particules augmente avec le temps et lorsque celle-ci redépassera le seuil d'incertitude maximale, les particules se partitionneront une nouvelle fois.

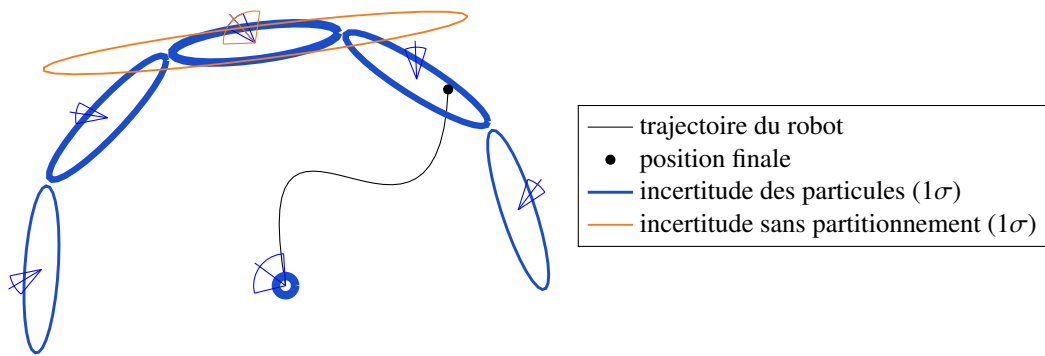


FIGURE 4.8 – Évolution d'un robot sur un court trajet après avoir généré plusieurs particules gaussiennes pour diminuer l'incertitude d'orientation de l'état. Comparaison avec un filtre gaussien n'effectuant pas de partitionnement. L'épaisseur du trait bleu est proportionnelle au poids des particules.

4.4.2 Étape d'évolution d'une particule

Dans l'algorithme d'évolution, l'étape de partitionnement se fait avant de prendre en compte la mesure d'odométrie et de calculer l'estimation *a priori* de l'état. Le résultat du partitionnement n'est donc qu'une approximation de la mixture de gaussiennes de la dernière estimation connue de l'état qui ne possède plus de particules avec d'importantes incertitudes d'orientation. Le calcul de l'évolution utilise ce résultat comme modélisation de l'état précédent. Nous nommerons

$$\{(w_{k-1|k-1}^i, \mathbf{x}_{k-1|k-1}^i, \mathbf{P}_{k-1|k-1}^i), i \in \llbracket 1, N \rrbracket\} \quad (4.13)$$

l'ensemble des N particules de cette mixture de gaussiennes. L'application du modèle d'évolution sur la mixture est très similaire à celle présentée dans le filtre à somme de gaussiennes (section 2.5.1). Cependant, le filtre développé ici s'appuie sur la méthode de fusion du filtre à intersection de covariance partitionné (SCIF, section 2.3.5) dont la modélisation de la covariance est différente. En effet, la matrice de covariance $\mathbf{P}_{k|k-1}^i$ de chaque particule se décompose

en deux matrices correspondant à la partie indépendante et dépendante de l'état :

$$\mathbf{P}_{k-1|k-1}^i = \mathbf{P}_{I,k-1|k-1}^i + \mathbf{P}_{D,k-1|k-1}^i. \quad (4.14)$$

Les mesures d'odométries \mathbf{u}_k sont considérées comme bruitées par un modèle gaussien centré dont la covariance correspond à la matrice $\mathbf{Q}_k = \mathbf{Q}_{I,k} + \mathbf{Q}_{D,k}$. Le calcul de l'estimation *a priori* de l'état correspond à une utilisation des équations de prédiction du SCIF pour chacune des particules :

$$w_{k|k-1}^i = w_{k-1|k-1}^i \quad (4.15)$$

$$\mathbf{x}_{k|k-1}^i = f(\mathbf{x}_{k-1|k-1}^i, \mathbf{u}_k, \mathbf{0}) \quad (4.16)$$

$$\mathbf{P}_{k|k-1}^i = \mathbf{J}_{f,x} \mathbf{P}_{k-1|k-1}^i \mathbf{J}_{f,x}^T + \mathbf{J}_{f,w} \mathbf{Q}_k \mathbf{J}_{f,w}^T \quad (4.17)$$

$$\mathbf{P}_{I,k|k-1}^i = \mathbf{J}_{f,x} \mathbf{P}_{I,k-1|k-1}^i \mathbf{J}_{f,x}^T + \mathbf{J}_{f,w} \mathbf{Q}_{I,k} \mathbf{J}_{f,w}^T \quad (4.18)$$

$$\mathbf{P}_{D,k|k-1}^i = \mathbf{P}_{k|k-1}^i - \mathbf{P}_{I,k|k-1}^i \quad (4.19)$$

avec

$$\mathbf{J}_{f,x} = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}_{k-1|k-1}, \mathbf{u}_k, \mathbf{0}) \quad \mathbf{J}_{f,w} = \frac{\partial f}{\partial \mathbf{w}}(\mathbf{x}_{k-1|k-1}, \mathbf{u}_k, \mathbf{0}). \quad (4.20)$$

Le poids des particules après prédiction reste inchangé car il n'y a pas d'interaction entre les particules pendant l'application du modèle d'évolution.

4.5 Calcul de la mise à jour

Lorsque la nouvelle mesure à traiter correspond à une observation de l'environnement, la procédure de mise à jour est utilisée afin d'obtenir une estimation *a posteriori* de l'état. Nous détaillerons dans cette section comment sont gérées les observations de balises. Le traitement de ce type de mesures peut se décomposer en différentes étapes : le partitionnement de la mesure (lorsque celle-ci devient trop non-linéaire), la fusion des données à l'estimation de l'état et la réduction de la mixture de gaussiennes représentant cet état.

4.5.1 Test de linéarité

De la même façon que lors d'une étape d'évolution, la non-linéarité du modèle d'observation n'a pas toujours un important impact sur la qualité de l'estimation de l'état. Puisque la création de nouvelles particules gaussiennes dans la mixture est une opération coûteuse, il est alors nécessaire de quantifier cet impact et de définir un critère permettant de décider quand il

faut partitionner la mixture de l'état. Le test de linéarité développé ici utilise un critère géométrique spécifique au type d'observation utilisé. Cela permet d'éviter l'utilisation d'un critère plus générique comme la divergence de Kullback-Leibler (section 2.7.7) pour laquelle il n'y a pas de solution analytique connue pour ce type d'observation. L'idée du critère géométrique présenté ici est d'estimer à quel point la particule gaussienne obtenue par linéarisation est loin de la forme réelle que devrait avoir cette PDF. Son calcul repose sur :

- la mesure de la courbure provoquée par l'observation
- l'allongement de la particule gaussienne de l'état dans la direction tangente à la forme circulaire de la PDF de l'observation.
- l'épaisseur de cette PDF dans la zone concernée

La figure 4.9 représente graphiquement les éléments qui interviennent dans le test de linéarité.

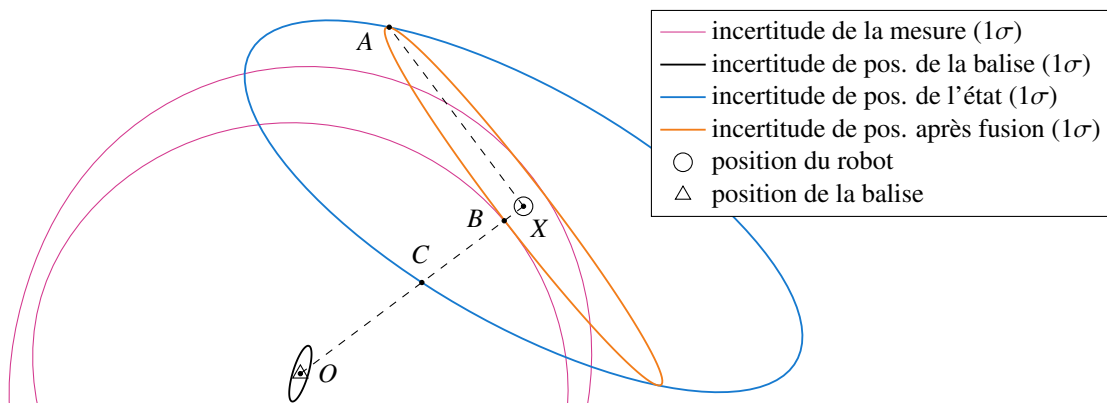


FIGURE 4.9 – Représentation des éléments géométriques utilisés dans l'élaboration du critère de partitionnement. L'ellipse orange correspond à l'estimation *a posteriori* de l'état du robot si l'observation était linéarisée lors de la fusion.

Le calcul du critère se fait à partir des données suivantes :

- $\mathbf{p}_{r,k}$ l'espérance de la position 2D du robot (avec prise en compte de la position du capteur)
- \mathbf{P}_k la covariance de cette position
- $\mathbf{p}_{b,k}$ l'espérance de la position de la balise
- $\mathbf{R}_{xy,k}$ la covariance de cette position
- $\mathbf{R}_{r,k}$ la variance de la mesure de distance

Les différents segments impliqués ont pour expression :

$$OX = \|\mathbf{d}\| \quad (4.21)$$

$$CX = \left(\sqrt{\mathbf{d}^\top \mathbf{P}_k^{-1} \mathbf{d}} \right)^{-1} \|\mathbf{d}\| \quad (4.22)$$

$$AX = \left(\sqrt{\mathbf{d}'^\top \mathbf{P}_k^{-1} \mathbf{d}'} \right)^{-1} \|\mathbf{d}'\| \quad (4.23)$$

$$BX = \sqrt{\frac{\mathbf{d}^\top \mathbf{R}_{xy,k} \mathbf{d}}{\mathbf{d}^\top \mathbf{d}} + \mathbf{R}_{r,k}} \quad (4.24)$$

avec

$$\mathbf{d} = \mathbf{p}_{r,k} - \mathbf{p}_{b,k} \quad \mathbf{d}' = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{d} \quad (4.25)$$

Le critère géométrique correspond alors au calcul d'un rapport entre ces segments :

$$s = \frac{OX^2 CX}{BX^2 AX} \quad (4.26)$$

Pour effectuer le test de linéarité, la valeur de ce critère est comparée à un seuil maximal au delà duquel, la non-linéarité est considérée comme trop importante et la particule de l'état sera alors partitionnée. Cependant, si le seuil choisi est trop bas, la mixture risque de se partitionner trop souvent. Cela nuira au bon fonctionnement de la localisation coopérative puisqu'elle repose sur le fait que, la plupart du temps, les mixtures estimant les états des robots ne possèdent qu'une seule particule gaussienne.

4.5.2 Mise à jour d'une particule Gaussienne

Si le résultat du test de linéarité indique qu'il n'y a pas besoin de partitionnement, l'étape de mise à jour consiste simplement à la fusion de la nouvelle observation avec l'estimation courante de l'état afin d'obtenir l'estimation *a posteriori*. Il s'agit de la procédure classique de mise à jour d'un SCIF appliquée à certaines particules de la mixture. Ces formules ne sont donc pas spécifiques aux observations de balises et peuvent s'appliquer à toutes les mesures provenant d'une localisation individuelle. Le calcul se base sur une particule de l'estimation *a priori* ($w_{k|k-1}^i, \mathbf{x}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i$) et dont la matrice de covariance se décompose de la même façon que lors de l'étape d'évolution : $\mathbf{P}_{k|k-1}^i = \mathbf{P}_{I,k|k-1}^i + \mathbf{P}_{D,k|k-1}^i$. La mesure \mathbf{z}_k est accompagnée d'un bruit gaussien centré, dont la matrice de covariance est $\mathbf{R}_k = \mathbf{R}_{I,k} + \mathbf{R}_{D,k}$. En reprenant les équations de mise à jour présentées dans la section 2.3.5, chaque nouvelle particule de l'estimation *a*

posteriori se calcule de la façon suivante :

$$\mathbf{P}_1 = \omega^{-1} \mathbf{P}_{D,k|k-1}^i + \mathbf{P}_{I,k|k-1}^i \quad (4.27)$$

$$\mathbf{P}_2 = (1 - \omega)^{-1} \mathbf{R}_{D,k} + \mathbf{R}_{I,k} \quad (4.28)$$

$$\mathbf{K} = \mathbf{P}_1 \mathbf{J}_{hx}^\top (\mathbf{J}_{hx} \mathbf{P}_1 \mathbf{J}_{hx}^\top + \mathbf{J}_{hv} \mathbf{P}_2 \mathbf{J}_{hv}^\top)^{-1} \quad (4.29)$$

$$\mathbf{x}_{k|k}^i = \mathbf{x}_{k|k-1}^i + \mathbf{K} (\mathbf{z}_k - \mathbf{h}(\mathbf{x}_{k|k-1}^i, \mathbf{0})) \quad (4.30)$$

$$\mathbf{P}_{k|k}^i = (\mathbf{I} - \mathbf{K} \mathbf{J}_{hx}) \mathbf{P}_1 \quad (4.31)$$

$$\mathbf{P}_{I,k|k}^i = (\mathbf{I} - \mathbf{K} \mathbf{J}_{hx}) \mathbf{P}_{I,k|k-1}^i (\mathbf{I} - \mathbf{K} \mathbf{J}_{hx})^\top + (\mathbf{K} \mathbf{J}_{hv}) \mathbf{R}_{I,k} (\mathbf{K} \mathbf{J}_{hv})^\top \quad (4.32)$$

$$\mathbf{P}_{D,k|k}^i = \mathbf{P}_{k|k}^i - \mathbf{P}_{I,k|k}^i \quad (4.33)$$

avec

$$\mathbf{J}_{hx} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}_{k|k-1}^i, \mathbf{0}) \quad \mathbf{J}_{hv} = \frac{\partial \mathbf{h}}{\partial \mathbf{v}}(\mathbf{x}_{k|k-1}^i, \mathbf{0}). \quad (4.34)$$

Le coefficient ω est optimisé pour minimiser le déterminant de $\mathbf{P}_{k|k}^i$. Le calcul du poids $w_{k|k}^i$ utilise la PDF de la particule de l'estimation *a priori* de l'état $p_{\mathbf{x}^i, k|k-1}(\mathbf{x})$, la PDF de l'observation exprimée dans l'espace d'état $p_{\mathbf{z}, k}(\mathbf{x})$ et le poids obtenu à l'estimation *a priori* $w_{k|k-1}^i$:

$$w_{k|k}^i = w_{k|k-1}^i p_{\mathbf{x}^i, k|k-1}(\mathbf{x}_{k|k}^i) p_{\mathbf{z}, k}(\mathbf{x}_{k|k}^i) \quad (4.35)$$

Le poids des particules obtenues diminue lorsque leur espérance est éloignée des zones à forte densité de l'ancien état et de l'observation. Dans le cas des observations de balises, la PDF $p_{\mathbf{z}, k}(\mathbf{x})$ doit correspondre à celle qui a été présentée précédemment à l'équation (4.8). Le problème de cette PDF est que son expression nécessite le calcul d'un produit de convolution. Puisqu'elle n'a pas de solution analytique connue, cela signifie qu'il faut alors utiliser une approche numérique coûteuse en ressources machines.

Pour éviter ce problème, nous utilisons une approximation de cette PDF. L'idée développée ici repose sur le fait que $p_{\mathbf{z}, k}(\mathbf{x})$ peut pratiquement être considérée comme gaussienne lorsque nous la projettons sur la demi-droite partant de la positions de la balise et passant par la position du robot (estimation *a priori*). Au voisinage de la position de ce dernier, le cercle que décrit le maximum de la PDF de la mesure de distance peut être approximé par la droite tangente à ce cercle et perpendiculaire à la demi-droite. Dans cette situation, la convolution entre la gaussienne de l'incertitude de position de la balise et la fonction de vraisemblance que représente l'approximation est uniforme dans la direction de la tangente. Le produit de convolution peut alors être exprimé dans l'espace de la mesure de distance (paramétrage sur la demi-droite). Il s'agit alors d'une convolution entre la gaussienne de la mesure de distance et la gaussienne représentant la marginale de la gaussienne 2D de l'incertitude de position de la balise dans la direction de la tangente. La solution du produit de convolution est alors une simple gaussienne

dont les espérances et les variances s'additionnent. L'expression de la PDF utilise le même calcul de variance que celui présent dans l'équation (4.24) :

$$p_{\mathbf{z},k}(\mathbf{x}_{k|k}^i) = g\left(\|\mathbf{x}_{k|k}^i - \mathbf{p}_{b,k}\|; \mathbf{z}_k, \mathbf{R}_{r,k}\right) * g\left(\mathbf{x}_{k|k}^i; \mathbf{p}_{b,k}, \mathbf{R}_{xy,k}\right) \quad (4.36)$$

$$\approx g\left(\|\mathbf{x}_{k|k}^i - \mathbf{p}_{b,k}\|; \mathbf{z}_k, \frac{(\mathbf{x}_{k|k}^i - \mathbf{p}_{b,k})^\top \mathbf{R}_{xy,k} (\mathbf{x}_{k|k}^i - \mathbf{p}_{b,k})}{(\mathbf{x}_{k|k}^i - \mathbf{p}_{b,k})^\top (\mathbf{x}_{k|k}^i - \mathbf{p}_{b,k})} + \mathbf{R}_{r,k}\right) \quad (4.37)$$

avec $\mathbf{p}_{b,k}$ la position de la balise et $\mathbf{R}_{xy,k}$ sa covariance, \mathbf{z}_k la mesure de distance et $\mathbf{R}_{r,k}$ sa variance.

4.5.3 Partitionnement à l'étape d'observation

Lorsqu'une particule de l'estimation *a priori* ne valide pas le test de linéarité, il est alors nécessaire d'utiliser une procédure de fusionnement différente de celle habituellement utilisée dans un SCIF. Le modèle d'observation par mesure de distance ne peut pas être linéarisé car son approximation serait trop éloignée de la forme réelle de sa PDF. La solution proposée consiste à convertir au préalable la PDF de la mesure en une mixture de gaussiennes exprimée dans l'espace 2D de la position du robot. Chacune des particules de la mixture de la mesure est ensuite fusionnée à la particule gaussienne de l'état par une procédure de mise à jour d'un SCIF afin de créer une nouvelle particule pour l'estimation *a posteriori* de l'état. Puisque la mixture représentant la mesure est directement exprimée dans l'espace d'état, le modèle de fusion ne dépend plus du type d'observation effectué. La gestion de la non-linéarité du modèle d'observation repose alors uniquement dans la construction de la mixture de gaussiennes qui la représente.

Pour le modèle d'observation par mesure de distance avec des balises, la mixture est construite avec un ensemble de gaussiennes représentant la forme d'anneau de la PDF de la mesure. Le nombre de particules N_{meas} de la mixture de la mesure est fixe et doit être suffisamment important pour que les différentes formes d'anneaux soient facilement représentables par les particules gaussiennes créées. La première étape de la construction de la mixture de la mesure consiste à créer un ensemble de gaussiennes $(\mathbf{z}^j, \mathbf{R}^j)$ exprimé dans un espace particulier de dimension 4 :

$$\mathbf{z}_k^j = \begin{bmatrix} r_k & \varphi_k^j & \mathbf{p}_{b,k} \end{bmatrix}^\top \quad \forall i \in \llbracket 1, N_{\text{meas}} \rrbracket. \quad (4.38)$$

Chaque particule est paramétrée par la mesure de distance r_k (qui provient directement de la mesure) et la position 2D de la balise $\mathbf{p}_{b,k}$. Elle possède également un paramètre angulaire φ_k^j qui est utilisé pour définir la position angulaire de la particule sur l'anneau :

$$\varphi_k^j = 2\pi(i-1)N_{\text{meas}}^{-1} + \varphi_k \quad \forall i \in \llbracket 1, N_{\text{meas}} \rrbracket \quad (4.39)$$

avec le décalage φ_k représentant l'angle de la direction du robot depuis la position de la balise. Ce décalage permet d'avoir une particule dont l'espérance sera très proche de la particule de l'estimation de l'état du robot. Les matrices de covariances $\mathbf{R}_k^j = \mathbf{R}_{I,k}^j + \mathbf{R}_{D,k}^j$ des particules de la mesure sont paramétrées de façon à prendre en compte les informations connues sur les bruits de l'observation. En effet, l'utilisation des matrices dépendantes et indépendantes nous permet de prendre en compte que l'incertitude de position des balises est une donnée fortement corrélée et que les mesures de distance sont totalement indépendantes. Ces matrices ont alors pour expression :

$$\mathbf{R}_{I,k}^j = \begin{bmatrix} \mathbf{R}_{r,k} & 0 & \mathbf{0} \\ 0 & (2\pi N_{\text{meas}}^{-1})^2 & \mathbf{0} \\ 0 & 0 & \mathbf{0} \end{bmatrix} \quad \mathbf{R}_{D,k}^j = \begin{bmatrix} 0 & 0 & \mathbf{0} \\ 0 & 0 & \mathbf{0} \\ 0 & 0 & \mathbf{R}_{xy,k} \end{bmatrix} \quad (4.40)$$

Les poids des gaussiennes de la mixture représentant la mesure n'ont pas de calcul particulier puisqu'ils sont uniformes. En effet, puisqu'elles sont toutes réparties à la même distance de la balise, il n'y a pas de direction φ_k^j privilégiée.

Avant d'être fusionnée à l'estimation de l'état du robot, la mixture de la mesure est d'abord convertie dans l'espace 2D de la position du robot grâce à l'utilisation de la fonction

$$f(\mathbf{z}_k^j) = \mathbf{p}_{b,k} + r_k \begin{bmatrix} \cos \varphi_k^j \\ \sin \varphi_k^j \end{bmatrix}. \quad (4.41)$$

Elle permet alors de générer la mixture contenant les particules gaussiennes $(\mathbf{m}_k^j, \mathbf{M}_k^j)$ et correspondant à la mesure après conversion dans l'espace 2D. Ces nouvelles particules ont pour expression :

$$\mathbf{m}_k^j = f(\mathbf{z}_k^j) \quad (4.42)$$

$$\mathbf{M}_k^j = \mathbf{J}_{fz} \mathbf{R}_k^j \mathbf{J}_{fz}^T \quad (4.43)$$

$$\mathbf{M}_{I,k}^j = \mathbf{J}_{fz} \mathbf{R}_{I,k}^j \mathbf{J}_{fz}^T \quad (4.44)$$

$$\mathbf{M}_{D,k}^j = \mathbf{M}_k^j - \mathbf{M}_{I,k}^j \quad (4.45)$$

avec

$$\mathbf{J}_{fz} = \frac{\partial f}{\partial \mathbf{z}^j}(\mathbf{z}_k^j). \quad (4.46)$$

La figure 4.10 représente un exemple de mixture de gaussiennes approximant la PDF de l'observation d'une balise dans l'espace 2D de la position du robot. La PDF de la mesure a été approximée par un ensemble de 16 particules. Nous pouvons voir que la représentation à 1σ des ellipses des gaussiennes recouvre bien la forme réelle de la densité de la mesure.

Une fois que l'observation de la balise a été convertie en une mixture de gaussiennes dans

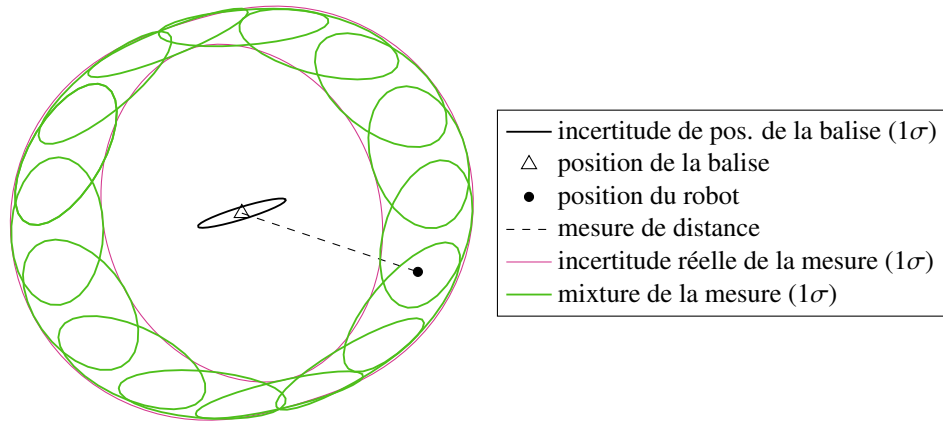


FIGURE 4.10 – Génération des particules pour représenter la mesure de distance

un espace proche de celui de l'état, il ne reste plus qu'à la fusionner avec chaque particule gaussienne de l'estimation *a priori* de l'état du robot. Cette fusion est réalisée grâce à l'utilisation de l'intersection de covariance partitionnée sur chaque particule de la mixture de la mesure. Puisque la mesure est exprimée dans l'espace de position du capteur du robot, le modèle d'observation n'est pas celui présenté à l'équation (4.3). Celui-ci est plus direct mais fait cependant intervenir le décalage de la position 2D du capteur \mathbf{p}_s :

$$h : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^2$$

$$(\mathbf{x}_k, \mathbf{v}_k) \mapsto \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} \cos \theta_k & -\sin \theta_k \\ \sin \theta_k & \cos \theta_k \end{bmatrix} \mathbf{p}_s + \mathbf{v}_k \quad (4.47)$$

Le calcul des particules de l'estimation *a posteriori* de l'état obtenu à partir de la particule de l'estimation *a priori* ($w_{k|k-1}^i, \mathbf{x}_{k|k-1}^i, \mathbf{P}_{k|k-1}^i$) utilise les équations (4.27) à (4.33). La différence est qu'elles sont appliquées pour chaque particule de la mesure ($\mathbf{z}_k^j, \mathbf{R}_k^j$). Le calcul du poids de chaque particule se base sur la PDF de la particule source de l'estimation *a priori* de l'état et sur la PDF de la particule de la mesure, exprimées toutes les deux pour l'espérance de l'état nouvellement calculé $\mathbf{x}_{k|k}^j$:

$$w_{k|k}^j = C p_{\mathbf{x}^i, k|k-1}(\mathbf{x}_{k|k}^j) g \left(\begin{bmatrix} x_{k|k}^j & y_{k|k}^j \end{bmatrix}^\top ; \mathbf{m}_k^j, \mathbf{M}_k^j \right) \quad (4.48)$$

avec C un coefficient permettant de normaliser les poids des nouvelles particules (la somme des poids obtenus correspond au poids de la particule source de l'estimation *a priori* $w_{k|k-1}^i$). Dans la pratique, une grande partie des particules générées ont un poids très faible puisque l'incertitude de position du robot peut ne couvrir qu'une petite zone de l'anneau que forme l'incertitude de la mesure de balise. Pour éviter le calcul de particules qui seront par la suite négligeables, un seuil de poids w_{\min} est utilisé afin de tester si la particule doit être calculée ou non.

La figure 4.11 présente un exemple d'estimation *a posteriori* de l'état que l'on obtient après

une mesure de distance avec une balise, lorsque l'estimation *a priori* de l'état est une gaussienne avec une importante incertitude de position (ellipse grise). Le test de linéarité conduit à un partitionnement de la mesure et l'estimation obtenue est alors composée de plusieurs particules gaussiennes (représentées en bleu). L'ellipse orange correspond au résultat que nous aurions obtenu sans partitionnement de la mesure. La mesure a été partitionnée en 16 particules, mais seulement 6 particules de la mixture *a posteriori* ont un poids suffisamment important pour ne pas être négligées. Le seuil de poids minimal choisi ici est $w_{\min} = 5 \times 10^{-4}$.

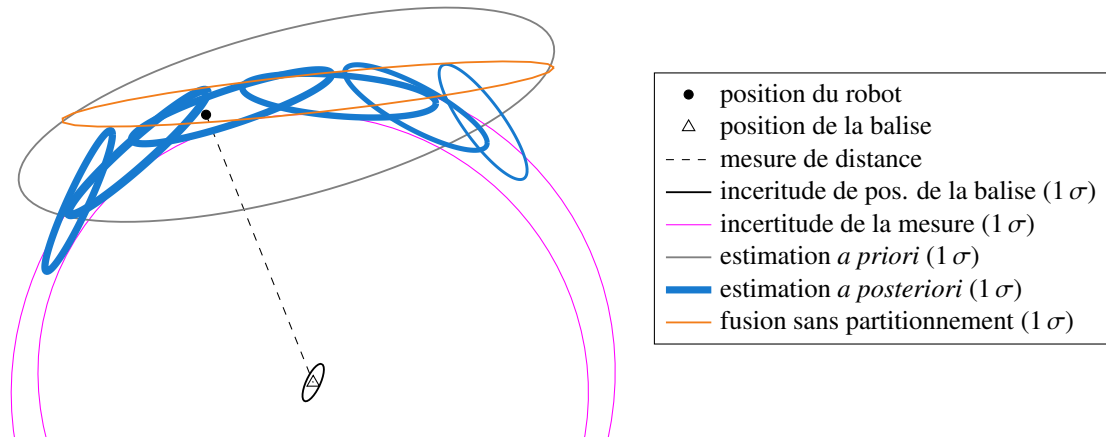


FIGURE 4.11 – Exemple d'estimation *a posteriori* de l'état du robot obtenu après le partitionnement de la mesure de distance avec une balise. L'ellipse orange correspond au résultat obtenu dans SCIF classique. L'épaisseur du trait bleu est proportionnel aux poids des particules de la mixture.

4.5.4 Réduction de la mixture de gaussiennes

Avec les différentes solutions de partitionnement présentées, le nombre de particules des mixtures représentant les états d'un robot ne peuvent qu'augmenter avec le temps. L'objectif étant de n'avoir qu'une seule particule représentant l'état lorsque les conditions sont bonnes, il est nécessaire de réduire le nombre de particules quand cela est possible. Une première réduction est faite en filtrant les particules dont le poids est négligeable. Ce filtrage est déjà réalisé lorsque l'observation à fusionner a été partitionnée afin d'éviter de calculer des particules qui seront par la suite éliminées. Le seuil utilisé w_{\min} est choisi suffisamment bas pour que la suppression de particules n'est pas d'impact sur la PDF que représente la mixture. Avec l'avancement d'une expérimentation et les observations successives, les particules de la mixture se trouvant loin de la réalité vont assez rapidement diminuer en poids et finir par être éliminées par cette étape de filtrage. Cependant, cet outil ne permet d'éliminer que les particules divergentes, mais il n'a aucun effet sur les particules dont la justesse est bonne.

Les particules qui sont proches de l'état réel du robot vont se ressembler de plus en plus avec la prise en compte successive des observations. L'idée est donc de réduire la mixture en

cherchant une autre mixture avec moins de particules mais donnant une qualité d'estimation équivalente. C'est un problème assez bien connu dans la littérature, et il existe actuellement beaucoup de solutions différentes. Par exemple il y a l'algorithme de Salmond (SALMOND 1989) ou l'algorithme de Runnalls (RUNNALLS 2007). Il existe également des améliorations de solutions existantes comme le GMRC (*Gaussian Mixture Reduction via Clustering*) (SCHIEFERDECKER et HUBER 2009) ou le COWA (*Constraint Optimized Weight Adaptation*) (CHEN et al. 2010). Certaines de ces méthodes ont été comparées par les auteurs de (CROUSE et al. 2011), et ils ont montré que les approches se basant sur la divergence de Kullback-Leibler (c'est à dire la solution de Runnalls et le GMRC) donnaient de bons résultats. De plus, l'amélioration de GMRC n'est pas significative par rapport de la méthode de Runnalls. La solution que nous avons utilisée pour résoudre notre problème est donc la méthode de Runnalls dont l'algorithme est suffisamment simple et flexible pour être adapté à notre configuration.

L'objectif de la méthode de réduction de Runnalls est d'approximer une mixture de gaussiennes contenant N particules par une mixture contenant $M < N$ particules tout en minimisant la divergence de Kullback-Leibler entre ces deux mixtures. La réduction consiste à approximer le couple de particules minimisant une fonction de coût particulière par une nouvelle particule dont le poids, l'espérance et la covariance correspondent à la mixture que représente le couple de particules. Cette approximation est appelée *réduction de mixture standard* et est décrite dans la section 2.7.2. L'opération est répétée jusqu'à ce que la mixture n'ait plus que M particules. La fonction de coût utilisée correspond au calcul d'une borne supérieure de la divergence de Kullback-Leibler. Avec $(w^i, \mathbf{x}^i, \mathbf{P}^i)$, $(w^j, \mathbf{x}^j, \mathbf{P}^j)$ le couple de particules et $(w^m, \mathbf{x}^m, \mathbf{P}^m)$ la particule obtenue après réduction, le coût a pour expression :

$$c_{i,j} = \frac{1}{2}(w^m \log |\mathbf{P}^m| - w^i \log |\mathbf{P}^i| - w^j \log |\mathbf{P}^j|). \quad (4.49)$$

Dans la version de l'algorithme de réduction de Runnalls que nous utilisons ici (algorithme 3), le test de fin de réduction est différent. Plutôt que de réduire la mixture à un nombre M de particules, nous réduisons la mixture tant que les coûts de réduction des couples de particules sont supérieurs à un seuil minimal c_{\min} . Cela permet d'être plus dynamique sur le niveau de réduction que nécessite la mixture. Cette version évite également de simplifier les couples de particules si jamais la particule résultante ne respecte pas le test de linéarité. Une réduction trop importante de la mixture pourrait en effet nuire à l'objectif initial qui était de modéliser au mieux l'estimation de l'état du robot après la prise en compte d'une observation de balise.

Grâce à cette étape de réduction, le nombre de particules de la mixture converge toujours vers une unique particule lorsque le robot reçoit suffisamment d'informations des balises environnantes. Dans la pratique, le nombre de particules est élevé (plus de 10 particules) au début d'une expérimentation, lorsque l'état du robot est initialisé avec une très importante incertitude de position et d'orientation. La première observation de balise va alors générer un grand nombre

de particules à différentes positions (en cercle) pour chaque particule créée lors du partitionnement à l'évolution. Pour que cette observation ait un impact sur l'estimation de l'état du robot, il faut cependant que la balise observée corresponde à un robot ou un objet statique ayant déjà une bonne estimation de sa position. Si le robot peut observer une autre balise ayant une bonne localisation, le nombre de particules de l'estimation de l'état va alors descendre à 2 particules, correspondant à l'ambiguïté classique de l'intersection de deux cercles. Le nombre peut, par la suite, passer à 1 particule après quelques observations supplémentaires pendant le déplacement du robot. Il est également possible que l'estimation descende à 1 particule en observant qu'une seule balise sous certaines conditions qui seront illustrées dans le prochain chapitre. La réduction de la mixture est donc une bonne solution pour remplir l'objectif d'une unique particule pour représenter l'état quand les conditions le permettent.

Entrée : Un ensemble \mathcal{I} de particules gaussiennes $\{(w^i, \mathbf{x}^i, \mathbf{P}^i)\}_{i \in \mathcal{I}}$

Entrée : le coût minimal d'une réduction c_{\min}

Sortie : Un ensemble \mathcal{O} de particules gaussiennes $\{(w^i, \mathbf{x}^i, \mathbf{P}^i)\}_{i \in \mathcal{O}}$

Copier les particules de l'ensemble \mathcal{I} dans l'ensemble \mathcal{O}

tant que il y a plus d'une particule dans l'ensemble \mathcal{O}

pour chaque couple de particules $\{(i, j) \in \mathcal{O}^2, i \neq j\}$

 Calculer la particule réduite $(w^m, \mathbf{x}^m, \mathbf{P}^m)$ à partir de $(w^i, \mathbf{x}^i, \mathbf{P}^i)$ et $(w^j, \mathbf{x}^j, \mathbf{P}^j)$

 Calculer le coût $c_{i,j}$ de la réduction.

si la particule m valide le test de linéarité et que $c_{i,j}$ est actuellement le coût minimal **alors**

 Conservé les paramètres i, j, m et $c_{i,j}$ comme i', j', m' et $c'_{i,j}$

fin

fin

si aucune fusion n'a été effectuée, ou $c'_{i,j} > c_{\min}$ **alors**

 Quitter

fin

 Ajouter m' à l'ensemble \mathcal{O}

 Supprimer i' et j' de l'ensemble \mathcal{O}

fin

Algorithme 3 : Réduction de mixture de gaussiennes

Chapitre 5

Expérimentations et résultats

5.1 Introduction

Ce chapitre présente différents résultats d'expérimentation aussi bien en simulation qu'en conditions réelles. Nous nous attacherons particulièrement à illustrer l'apport de l'approche proposée à travers différentes situations de complexité croissante. Les résultats obtenus seront analysés en les comparant aux vérités terrains à chaque fois qu'elles sont accessibles. Dans un premier temps, des cas canoniques seront proposés afin de faciliter la compréhension des différents apports. Notre approche sera comparée aux filtres EKF et SCIF dans des situations particulières mettant en valeur les différentes problématiques rencontrées. Dans un second temps des situations plus proches de la réalité seront proposées afin de valider le fonctionnement à plus grande échelle de notre algorithme. Ces expérimentations comportera deux simulations réalisées dans un environnement 3D réaliste et une expérimentation réelle sur la plateforme Pavin de l'Institut Pascal.

5.2 Illustration des cas canoniques

Après avoir étudié l'approche que nous avons développée pour traiter le problème de la localisation coopérative décentralisée par mesure de distance, nous allons maintenant observer son comportement dans de courtes simulations. L'objectif de ces simulations est de présenter des situations particulières dans lesquelles les approches basiques ne peuvent assurer une bonne intégrité des résultats. Nous présentons ici trois scénarios différents permettant de mettre en valeur le fonctionnement de notre approche sur les aspects suivants :

- la corrélation entre les estimations d'états des robots
- l'impact de l'incertitude d'orientation d'un robot

— la fusion d’observations de distance lorsque le robot est mal localisé.

Chacune de ces simulations se focalise sur l’illustration d’un aspect particulier de cette liste, tout en limitant l’influence des autres. Cela permet d’observer le comportement de notre solution de localisation dans des situations canoniques spécifiques.

5.2.1 Corrélation des informations échangées entre les robots

Dans cette première simulation, nous nous intéressons à la gestion de la corrélation entre les estimations des états de deux robots se déplaçant dans la même direction sur une courte trajectoire. Nous supposons que la position et l’orientation initiales de chaque robot sont précisément connues et les modèles des robots simulés sont considérés comme étant de type Ackermann. Cela signifie que les mesures d’odométrie permettent d’estimer l’angle de braquage et la vitesse linéaire du train avant du robot (les détails du modèle d’évolution sont présentés dans la section 5.3.2). Ces robots effectuent également des mesures d’inter-distance qui seront utilisées comme méthode d’observation dans les solutions de localisation coopérative. Les mesures d’odométrie et de distance sont simulées par l’ajout d’un bruit blanc gaussien sur l’odométrie réelle des robots.

Cette simulation a déjà été présentée dans la section 3.4.1 de ce manuscrit afin de comparer les approches centralisée et décentralisée de la localisation coopérative en utilisant un filtre de Kalman étendu (EKF). La figure 5.1 est un rappel des résultats obtenus pour ces deux approches. La trajectoire estimée de chaque robot est dessinée en orange et la trajectoire réelle est en noir. Les incertitudes de position à différents instants de la simulation sont représentées en orange par des ellipses (à 1σ) et les incertitudes d’orientation, par des cônes (à 1σ) autour de la direction du robot. En utilisant un EKF centralisé, les états des robots sont traités simultanément et l’information de corrélation entre leurs états est ainsi conservée pendant toute la simulation. L’approche basée sur l’EKF décentralisé calcule les estimations des états des robots séparément et ignore la corrélation existante entre leurs états. Cette perte d’information provoque une surconvergence des estimations et, par conséquent, une perte d’intégrité. Pour qu’une approche décentralisée soit considérée comme correcte, il ne faut pas qu’elle converge plus que le résultat obtenu par l’approche centralisée. En effet, cette dernière conserve toute l’information sur l’état des robots et est donc plus proche de l’optimal.

Notre approche est également décentralisée mais se base sur le filtre à intersection de covariance partitionnée (SCIF) qui a l’avantage de prendre en compte la corrélation des états. Lorsqu’une observation de distance est faite entre les robots, l’information de position du robot observée est alors considérée comme complètement corrélée à l’estimation précédente de l’état du robot effectuant l’observation. Nous pouvons alors voir dans la figure 5.2 que l’intégrité de la localisation des robots reste correcte. L’incertitude de position des différentes estimations

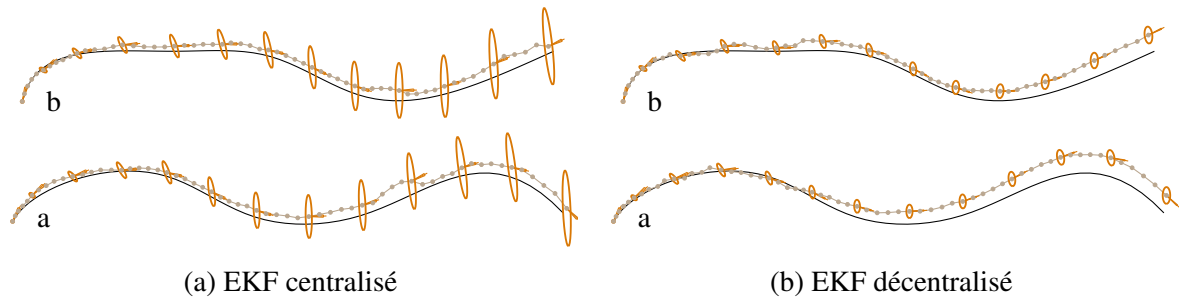


FIGURE 5.1 – Rappel des simulations de la localisation coopérative par mesures de distance entre deux robots se déplaçant de gauche à droite. Les estimations (à 1σ) de la position et de l'orientation sont représentées en orange à différents instants de la simulation. La trajectoire réelle est en noir.

est alors très similaire à celle que l'on peut avoir dans l'approche centralisée. Nous pourrions cependant remarquer que la distance entre la position estimée et la position réelle de chaque robot est plus importante que celle obtenue dans l'approche centralisée. Les mixtures de gaussiennes représentant les estimations des états des robots ne font évoluer qu'une seule particule pendant toute la trajectoire, car la non-linéarité du modèle d'évolution et d'observation n'a ici que peu d'impact. En effet, les observations de balises sont appliquées perpendiculairement à la trajectoire des robots et l'incertitude des états des robots est très étroite dans la direction des trajectoires.

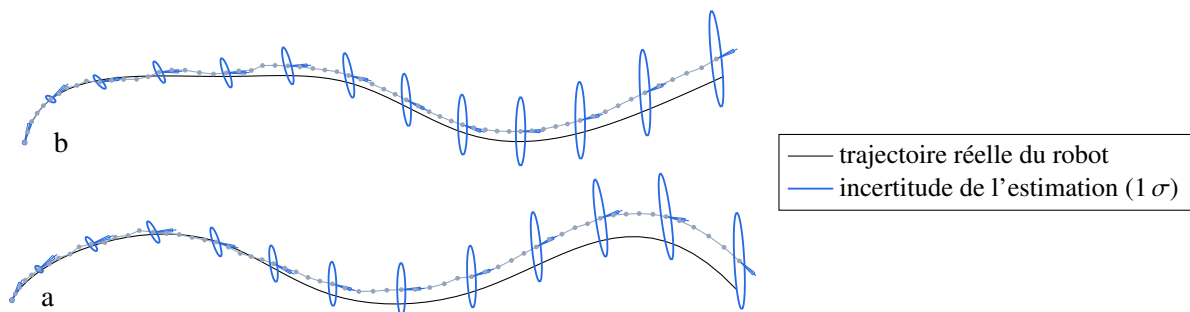


FIGURE 5.2 – Trajectoires de la simulation en utilisant notre approche de localisation coopérative. Chaque robot correspond à une instance différente de l'algorithme. Les ellipses représentent les incertitudes de position des robots à différents instants de la simulation.

L'utilisation de la méthode de fusion du SCIF est donc une bonne solution pour gérer le problème de corrélation entre les états des robots lorsque les observations de balises sont appliquées. Cette solution a l'inconvénient d'être moins précise sur les estimations car une partie des informations de corrélation entre les robots est perdue, mais les estimations ne peuvent cependant pas converger davantage qu'une solution optimale. Puisque les mixtures de gaussiennes n'ont ici qu'une seule particule, l'apport de notre approche ne concerne que le modèle d'observation coopératif par mesure de distance.

5.2.2 Prise en compte de la non-linéarité du processus de fusion basé sur des mesures de distance

Partitionnement dû à l'incertitude d'orientation du robot

Cette seconde simulation se focalise cette fois-ci sur l'influence de l'incertitude de l'orientation d'un robot dans la localisation coopérative par mesure de distance. Bien qu'il n'existe pas de lien direct entre les observations de balises et l'orientation des robots, cette dernière peut avoir une influence non négligeable sur la localisation des robots. Cela vient du fait que les modèles d'évolution des robots de type Ackermann créent une forte corrélation entre leur position et leur orientation. Dans certaines configurations, l'incertitude d'orientation peut provoquer une ambiguïté sur l'estimation de la trajectoire du robot lorsqu'il n'y a que très peu de balises observables. L'ambiguïté correspond ici au fait qu'il n'est pas possible de distinguer les trajectoires linéaires passant symétriquement à gauche et à droite d'une balise observable. Cette simulation permet d'étudier cette situation en utilisant un robot mobile et une balise statique. Le robot effectue une trajectoire rectiligne en passant à proximité de la balise et se termine par un virage à droite. La position de la balise statique est connue et le robot est initialisé à une position connue et précise mais avec une importante incertitude d'orientation.

La trajectoire simulée et les résultats de la localisation sont présentés dans la figure 5.3. Chaque sous-figure représente un instant différent de la simulation et permet ainsi de voir son évolution sur la trajectoire complète. Les résultats de notre approche (SCI-GMF), représentés en bleu, sont comparés à l'EKF (en orange) et au SCIF (en violet). L'épaisseur du trait bleu des ellipses est proportionnel au poids des particules de la mixture. À cause de l'importante incertitude d'orientation, l'estimation de l'état du SCI-GMF est partitionnée en un ensemble de 6 particules. Au début de la simulation (figure 5.3a), le SCI-GMF commence par un partitionnement de son unique particule gaussienne afin de gérer l'importante incertitude d'orientation de l'estimation initiale de l'état du robot. Les particules de la mixture ainsi générée sont orientées de manière à couvrir au mieux les orientations possibles du robot et ont une incertitude d'orientation réduite. Les estimations initiales de l'EKF et du SCIF ne sont pas des mixtures de gaussiennes et conservent donc l'importante incertitude d'orientation.

Après quelques étapes d'évolution et d'observation de balises, les particules de la mixture se retrouvent étalées perpendiculairement à la trajectoire, car les orientations des particules les plus extrêmes se sont écartées de la trajectoire mais les mesures de distance ont progressivement corrigé leur orientation ou éliminé les particules lorsque leurs pondérations sont devenues trop faibles (figure 5.3b). Après quelques observations supplémentaires, la mixture représente une distribution bimodale correspondant à l'ambiguïté que provoque cette configuration (figure 5.4c). Certaines particules se trouvent à proximité de la vérité terrain mais un autre ensemble de particules se situe sur la position symétrique (dont l'axe passe par la position initiale

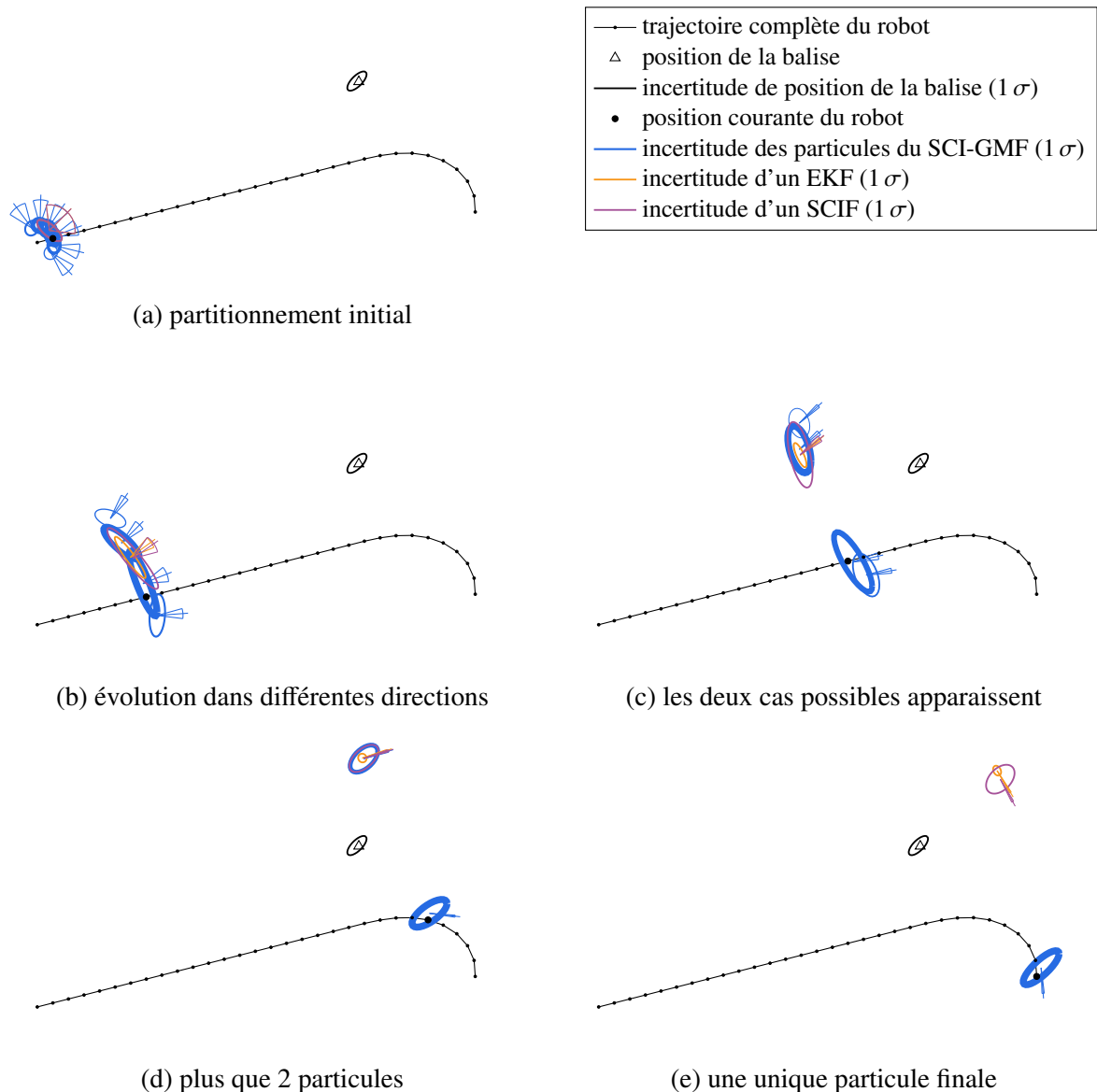


FIGURE 5.3 – Simulation de la localisation d’un robot à partir de mesures de distance avec une balise statique. Les 5 sous-figures représentent différents instants de la simulation. La trajectoire et l’initialisation de l’état du robot provoquent une ambiguïté sur l’estimation de position du robot.

du robot et la position de la balise) de ce point. En effet, les mesures de distance seraient exactement les mêmes sur cette trajectoire symétrique. Nous pourrions noter que les estimations de l’EKF et du SCIF ne se trouvent pas à proximité de la bonne position. Dans la suite de la simulation (figure 5.3d), l’ambiguïté n’est toujours pas résolue, mais il ne reste plus que deux particules dans la mixture. Les estimations de l’EKF et du SCIF sont très proches de la particule du SCI-GMF. Cependant l’incertitude donnée par l’EKF montre que cette solution surconverge (l’ellipse d’incertitude est plus petite que celle de la balise statique).

À la fin de la simulation, il ne reste plus qu'une seule particule dans l'estimation du SCI-GMF, car le changement de trajectoire du robot a permis de résoudre l'ambiguïté (figure 5.3e). Le robot a tourné à droite et s'est éloigné de la balise et la particule de la trajectoire symétrique a également emprunté cette direction mais s'est alors approchée de la balise statique. Puisque la distance des mesures a augmenté, le poids de la mauvaise particule a diminué jusqu'à devenir négligeable face au poids de la particule se trouvant à proximité de la position réelle. Dans le cas de l'EKF et du SCIF, le changement de trajectoire n'a pas suffi à corriger l'importante déviation qu'a provoqué l'ambiguïté de cette situation.

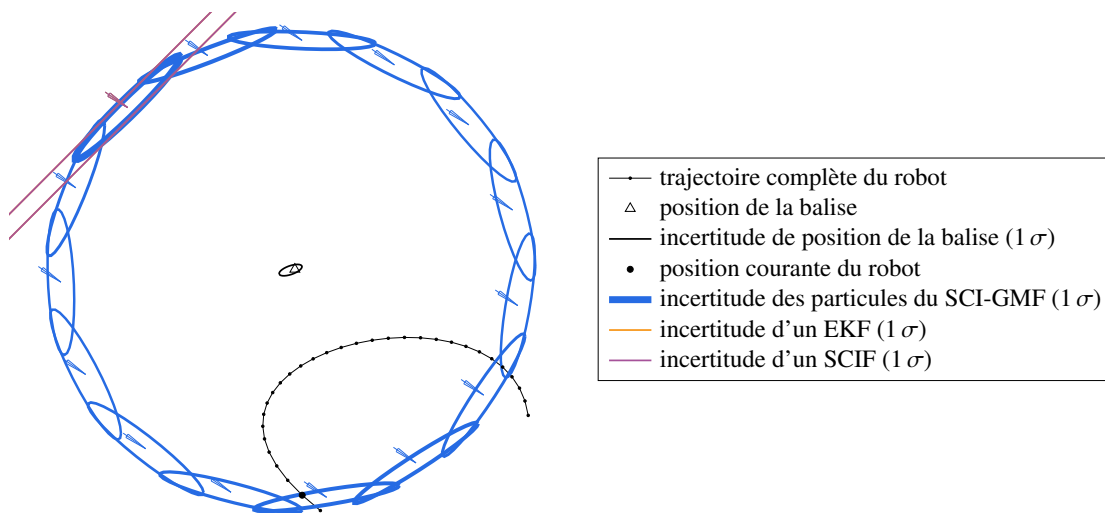
L'approche que nous avons développée est donc capable de gérer les ambiguïtés que provoquent les mesures de distance lorsque les particules de l'estimation de l'état du robot représentent correctement la distribution réelle. Dans le cas présent, l'ambiguïté a pris la forme d'un double choix de trajectoires symétriques, mais il peut parfois y avoir d'autres types d'ambiguïté impliquant 3 trajectoires possibles ou plus autour d'une balise statique en fonction de la configuration de l'état initial du robot. Elle ne se limite également pas aux trajectoires rectilignes puisque les balises observées peuvent être mobiles (lorsqu'il s'agit d'observations de robots).

Partitionnement lié au test de linéarité de la mesure

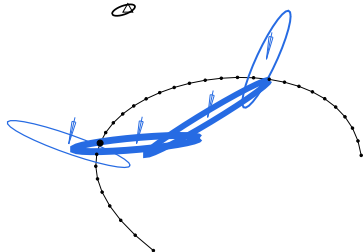
Cette troisième simulation se consacre cette fois-ci à la non-linéarité du modèle d'observation de balises. Cette non-linéarité a un impact important sur la forme que va prendre la densité de probabilité. Comme nous l'avons vu dans les chapitres précédents, cela peut aller d'une forme en arc de cercle jusqu'au cercle complet en fonction de la taille de l'incertitude de position du robot. Cette simulation fait encore intervenir un unique robot et une balise statique bien localisée, mais cette fois-ci, l'orientation initiale du robot est précise mais sa position est inconnue. La position inconnue est modélisée par une initialisation du vecteur d'état du robot à une position aléatoire dans l'espace de la simulation et à une incertitude de position suffisamment grande pour couvrir l'espace complet avec une densité de probabilité relativement homogène. Le robot effectue alors une trajectoire courbe à proximité de la balise et reçoit des mesures de distance avec cette dernière.

De la même façon que dans la simulation précédente, les résultats de notre approche (SCI-GMF) sont comparés à un EKF et un SCIF à plusieurs instants de la simulation (figure 5.4). Après la première observation de la balise, la mixture de gaussienne représentant l'estimation de l'état du robot dans le SCI-GMF est partitionnée en un ensemble de particules recouvrant entièrement l'anneau que forme la mesure de distance (figure 5.4a). L'espérance des estimations de l'EKF et du SCIF correspond au point du cercle de la mesure de distance le plus proche de l'état initial du robot. En raison de la forte non-linéarité du modèle d'observation, cette estimation n'a que très peu de chance d'être juste si la position initiale du robot est choisie

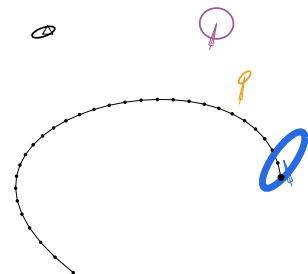
aléatoirement dans l'espace 2D de la simulation. Après quelques observations de balises supplémentaires, le nombre de particules qui compose la mixture a diminué et il ne reste maintenant plus que quelques particules à proximité de la vérité de terrain (figure 5.4b). Cela vient du fait que, bien que le robot se soit approché de la balise, les particules se trouvant de l'autre côté de celle-ci se sont éloignées. Les observations de balises n'ont alors pas de données de bonnes estimations et leurs poids ont diminué jusqu'à ce que la particule devienne négligeable. Les estimations des EKF et SCIF ne sont toujours pas justes mais sont cependant plus précises après la fusion de ces nouvelles observations. Cela montre que la non-linéarité du modèle d'observation n'est pas prise en compte et que ces solutions divergeraient dans la plupart des situations où cette non-linéarité ne serait plus négligeable.



(a) recouvrement complet de la mesure



(b) élimination des particules de poids faibles



(c) une unique particule finale

FIGURE 5.4 – Simulation de la localisation d'un robot bien orienté mais dont la position initiale est inconnue. Les observations de distance sont faites avec une unique balise statique bien localisée.

À la fin de la simulation, la mixture de gaussienne ne possède plus qu'une seule particule dont la justesse est correcte mais la précision reste améliorable. Les dernières particules ont été éliminées par les nouvelles observations bien que la direction de la mesure de distance avec la balise ne donne pas les mesures les plus précises. En effet, l'incertitude de position de la balise étant allongée dans une certaine direction, la précision de la mesure est alors moins importante lorsque le robot se trouve dans cette direction. La qualité de la localisation d'un robot à l'aide d'une seule balise statique nécessite donc des observations avec des points de vue très variés. Les résultats finaux de l'EKF et du SCIF restent erronés mais la justesse s'est améliorée grâce à la forte corrélation existante entre l'orientation et la position du robot.

Les résultats de cette simulation montrent que le SCI-GMF est capable de gérer la non-linéarité d'une observation de distance avec une balise dans sa globalité. Si l'estimation *a priori* de l'état du robot est vraiment imprécise, la distribution de la mesure sera entièrement utilisée pour l'estimation de l'état et les observations successives pourront alors l'améliorer progressivement. Il est cependant nécessaire d'avoir suffisamment d'informations pour que l'estimation de l'état du robot puisse ensuite converger vers une unique particule gaussienne. Par exemple, si le robot est initialisé avec une position inconnue mais également une orientation inconnue, il n'est alors plus possible de converger vers une estimation précise puisque tous les points des cercles des mesures de distance sont des solutions valides au problème de localisation.

5.3 Outils pour les expérimentations

5.3.1 Le simulateur 4D-Virtualiz

L'une des principales difficultés de la réalisation d'expérimentations en localisation coopérative est la complexité de la préparation du terrain et de l'utilisation des véhicules. Cela requiert du temps et de la ressource humaine. Afin d'alléger les tests et de gagner du temps sur le développement des approches, il est cependant possible d'utiliser de la simulation sur ordinateur. Pour cela, nous avons utilisé les produits de l'entreprise Clermontoise 4D-Virtualiz. Les logiciels développés permettent de simuler des capteurs, des robots et des environnements proches des conditions réelles. Les calculs de la simulation sont réalisés grâce à l'utilisation d'un moteur physique (PhysX) permettant de reproduire le mouvement et les efforts des robots réels. L'environnement graphique de la simulation est également suffisamment détaillé pour que des algorithmes de perception basés sur l'utilisation de caméras puissent fonctionner correctement. Des exemples d'images obtenues dans l'environnement de simulation sont présentés dans la figure 5.5. Pour que les simulations soient les plus réalistes possibles, les capteurs sont simulés de façon très fidèle à la réalité. Les bruits des mesures sont générés en prenant en compte les défauts et les particularités des capteurs réels.



(a) Environnement agricole



(b) La mini-ville *Pavin*

FIGURE 5.5 – Exemples d’images obtenues dans le simulateur de 4D-Virtualiz

Nous avons utilisé ce simulateur dans l’objectif de définir des scénarios de localisation coopérative proche de ceux que nous pourrions faire avec de vrais véhicules. 4D-Virtualiz met à disposition une version simulée de l’environnement nommé *pavin*, sur lequel nous avons l’habitude de travailler (figure 5.5b). Les véhicules nommés *Ipcar* ont également leur modélisation équivalente dans le simulateur ainsi que les capteurs UWB. Cette solution offre donc la possibilité de tester un scénario de localisation en version réelle et en version simulée avec une très bonne fidélité.

5.3.2 Véhicule

Dans ces expérimentations, nous utiliserons trois véhicules pouvant être modélisés par le modèle Ackermann. Ces véhicules sont :

- l’IPCar (figure 5.6a), qui est un petit véhicule urbain de laboratoire utilisé par les chercheurs de l’Institut Pascal. Ce véhicule *laboratoire* intègre différentes technologies capteurs (GPS, caméra, Lidar, balise UWB, centrale inertielle) qui sont utilisées seules ou combinées
- Aroco (figure 5.6b), qui est un robot tout terrain d’une taille proche d’une voiture, est utilisé principalement par les chercheurs de l’INRAE et construit par la société Robo-soft. Il intègre les mêmes technologies que les IPCars, mais il est capable d’évoluer en environnement ouvert.
- un tracteur agricole (figure 5.6c). Ce véhicule sera utilisé uniquement en simulation pour des raisons de praticité et de sécurité. Il est équipé des mêmes capteurs que l’Aroco ou l’IPCar.

Le modèle Ackermann est très connu puisqu’il correspond au modèle utilisé pour la plupart des voitures. La figure 5.7 illustre un exemple de robot de type Ackermann. Dans ce modèle,



(a) IPCar (b) Aroco (c) tracteur
 FIGURE 5.6 – Robots utilisés pour les différentes expérimentations

le robot est équipé de 4 roues positionnées en rectangle. Les deux roues avant ont la possibilité de pivoter selon l'axe passant par A et B . Le robot peut alors décrire des trajectoires circulaires autour d'un centre instantané de rotation I . La mesure du mouvement du véhicule est basée sur deux paramètres : la vitesse curviligne \dot{s} mesurée grâce à un odomètre et l'angle de braquage φ correspondant à l'angle d'une roue avant virtuelle se trouvant au point E avec la direction des roues arrières. Le vecteur de bruit \mathbf{w} correspond une fois de plus au bruit de ces deux paramètres :

$$\mathbf{u} = \begin{bmatrix} \dot{s} \\ \varphi \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (5.1)$$

La trajectoire du robot est calculée au point O . En utilisant l'angle de braquage φ et l'emplacement $l = OE$, il est possible de calculer le rayon de courbure

$$OI = \frac{l}{\tan \varphi} \quad (5.2)$$

La fonction d'évolution f est définie de la façon suivante :

$$f : \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$(\mathbf{x}, \mathbf{u}, \mathbf{w}) \mapsto \mathbf{x} + \delta t \begin{bmatrix} (\dot{s} + w_1) \cos \theta \\ (\dot{s} + w_1) \sin \theta \\ \frac{\dot{s} + w_1}{l} \tan(\varphi + w_2) \end{bmatrix} \quad (5.3)$$

Ce modèle est approximatif puisqu'il considère que les vitesses de translation et de rotation sont constantes entre chaque mesure. Il est donc tout à fait possible de l'améliorer en considérant par exemple que ces vitesses évoluent de façon linéaire. Le modèle serait alors à accélération constante. Bien que ces modèles ne soient pas exacts, ils permettent néanmoins de calculer les trajectoires des robots avec une estimation correcte sur de courtes distances (environ 20 m) lorsque l'intervalle de temps δt entre deux mesures est très court. Cependant, l'accumulation des erreurs de mesures entraîne une dérive de la trajectoire estimée sur les longues distances.

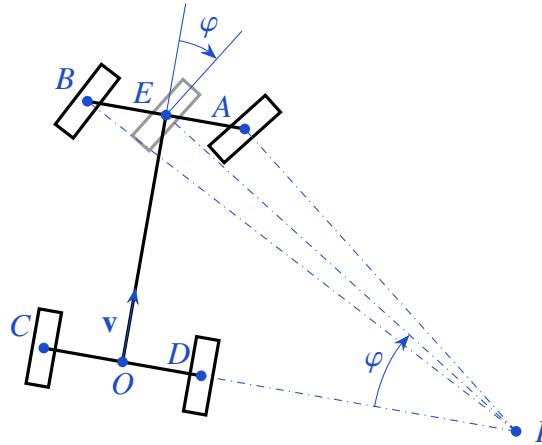


FIGURE 5.7 – Représentation d’un véhicule de type Ackermann. Les roues A et B peuvent pivoter pour que le véhicule puisse tourner autour d’un centre instantané de rotation I.

5.3.3 Modèle d’observation par GNSS

L’objectif des expérimentations est de montrer le fonctionnement de notre approche de localisation coopérative basée sur des mesures de distances mais ce type d’observation ne permet que d’échanger des informations entre les robots. Il est également nécessaire d’apporter de l’information de l’environnement extérieur grâce à l’utilisation d’autres moyens de perception. Pour cela, nous avons choisi d’utiliser des capteurs GNSS qui ont l’avantage d’être très simple à utiliser et d’avoir un modèle d’observation très direct.

Ce type de capteur fournit généralement des mesures de position exprimées dans les coordonnées géographiques (les mesures angulaires : latitude et longitude, et l’altitude). Ces coordonnées sont d’abord converties dans l’espace cartésien local de l’expérimentation en utilisant une méthode de projection comme la projection Transverse universelle de Mercator (UTM) ou encore la projection conique Lambert93. Les expérimentations présentées dans ce manuscrit utilisent la projection UTM.

Une fois que la mesure d’un GPS est convertie dans le plan du robot, la fonction d’observation h correspondant à ce capteur devient très simple. Cette mesure correspond à un vecteur position \mathbf{z} de dimension 2 et le vecteur \mathbf{v} , de même dimension, correspond au bruit de la mesure. Cette position est directement liée à la position du robot $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$ si le capteur GPS est placé au niveau du point d’origine du robot (point O). Pour gérer les différentes configurations des robots, la position du capteur \mathbf{p} est prise en compte dans la fonction d’observation. Cette fonction a donc pour expression

$$h : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$(\mathbf{x}, \mathbf{v}) \mapsto \begin{bmatrix} x \\ y \end{bmatrix} + \mathbf{v} + \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \mathbf{p} \quad (5.4)$$

Nous pouvons constater que l'orientation du robot a une influence importante sur la mesure du capteur embarqué sur un robot lorsque ce dernier est loin de l'origine du robot (effet de bras de levier). La position de la mesure va donc varier avec l'orientation du robot. Sans changer de position, la mesure du GPS peut varier jusqu'à une distance de deux fois la distance entre la position de l'antenne et l'origine du robot si celui-ci effectue une rotation de 180°. L'incertitude de l'orientation du robot a donc un impact direct sur la précision de la mesure du GPS. Le bras de levier que représente le placement de l'antenne GNSS n'est cependant pas la seule source de cette influence puisque la corrélation entre la position et l'orientation a également un impact.

5.3.4 Les balises UWB

Dans ces expérimentations, nous utiliserons des balises UWB (*Ultra-Wideband*) conçues et réalisées au sein du labex ImobS³ par les chercheurs de l'institut Pascal et l'INRAE (figure 5.8). Ces balises intègrent un module radio Decawave ainsi qu'un microcontrôleur. Elles proposent principalement deux modes de fonctionnement : un mode maître (anchor) et esclave (tag). Leur principe d'utilisation réside dans une interrogation du maître vers l'esclave pour déterminer la distance qui les sépare (technologie *time of flight*). Le principal inconvénient de ces balises est qu'une seule mesure de distance peut être demandée à la fois. De plus, la propagation du signal radio peut être soumise à des perturbations telles que des réflexions multiples (*multipath*) ou des masquages.



FIGURE 5.8 – Capteur de distance UWB développé par l'institut Pascal et l'INRAE

5.3.5 Gestion temporelle du filtre

Les données obtenues par un capteur sont rarement directement utilisées comme des mesures pouvant être fusionnées par un filtre. En effet, certains capteurs comme les caméras ou les lidars fournissent de grandes quantités d'informations ayant subies peu de traitement par le système embarqué du capteur. En fonction de leur utilisation, il est souvent nécessaire de mettre en place une étape de détection permettant d'extraire des éléments particuliers qui pourront ensuite être utilisés dans le filtre. Le temps induit par cette extraction va alors dépendre de

la complexité du détecteur ainsi que de la puissance de la machine effectuant le calcul. D'autres technologies de capteurs peuvent également prendre un certain temps d'acquisition. C'est le cas notamment de la mesure de distance par UWB qui nécessite un certain nombre d'échanges radio entre les émetteurs UWB. Dans le cas des approches coopératives, le temps de transmission des données entre les entités vient également s'ajouter. Si nous nous limitons aux aspects théoriques des méthodes de filtrage, la latence n'est pas évoquée. Cependant elle peut avoir un impact conséquent sur les résultats. Par exemple, s'il s'écoule 300 ms entre la mesure d'un capteur et sa prise en compte dans un robot mobile se déplaçant à 10 m s^{-1} , la mesure sera décalée de 3 m. L'estimation de l'état du robot est alors biaisée par cette latence.

Pour gérer correctement cette latence, nous avons utilisé la méthode présentée dans (TESSIER et al. 2006). Cette solution permet de fusionner les mesures en respectant leur date d'acquisition. Pour ce faire, l'algorithme gère l'historique des mesures et des estimations calculées afin de pouvoir replacer temporellement les nouvelles mesures. L'ensemble des mesures et estimations d'état est conservé chronologiquement sur une période de quelques secondes pour que toutes les mesures ajoutées aient une estimation d'état qui la précède. Lorsqu'une nouvelle mesure est ajoutée, celle-ci est placée à la bonne date dans la liste des observations, et l'ensemble des estimations d'état est recalculé à partir de cette mesure. La figure 5.9 représente l'ordre chronologique réel de l'ajout des mesures dans le filtre après leur temps de traitement, ainsi que la liste chronologique basée sur la date d'acquisition, utilisée pour le calcul. Chaque estimation de l'état *a posteriori* est calculée à la date d'acquisition d'une mesure. Les étapes de prédiction s'occupent de faire évoluer l'état estimé entre chaque mesure, en prenant en compte le temps qui sépare ces mesures. Pour que cet algorithme fonctionne au mieux, il est donc nécessaire d'enregistrer la date des mesures le plus tôt possible après son acquisition pour que le recalage temporel soit le plus réaliste possible. Cela nécessite également une synchronisation des horloges des robots pour que les échanges de mesures horodatées restent cohérents.

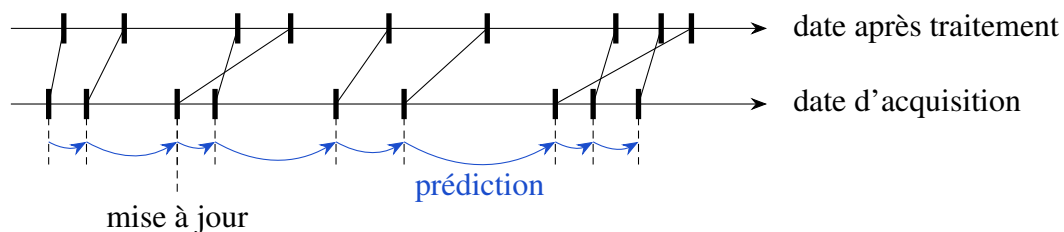


FIGURE 5.9 – Liste d'observations de l'algorithme de TESSIER et al. (2006) avec recalage temporel des mesures. Les traits noir épais correspondent à plusieurs mesures à différents instants. Les états futurs sont recalculés lorsqu'une observation est plus vieille que la dernière.

L'utilisation de la liste d'observation permet de gérer en temps réel la latence des mesures. La méthode présentée s'appuie ensuite sur les algorithmes de prédiction et de mise à jour pour calculer les estimations d'état.

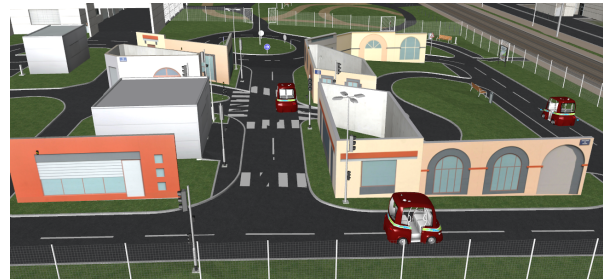
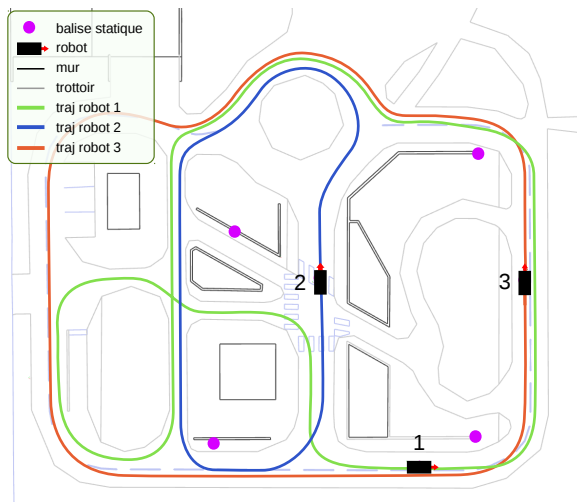
5.4 Simulations réalistes et expérimentations réelles

Les expérimentations présentées dans cette section ont pour objectif de tester le fonctionnement de l'approche de localisation coopérative dans des situations réalistes mais dont les conditions restent difficiles pour le calcul d'une estimation précise. L'idée ici est de créer des scénarios où les robots et les objets statiques à localiser n'ont que très peu d'informations disponibles mais les solutions de localisation doivent rester intègres. Afin d'étudier la qualité des résultats obtenus, les estimations des états des robots seront comparées à une vérité de terrain. Généralement, cette comparaison est faite par le calcul de l'erreur de position et d'orientation entre l'espérance de l'estimation calculée et la vérité de terrain. Dans le cas de l'utilisation d'une mixture de gaussiennes pour l'estimation, ce calcul d'erreur peut ne pas être pertinent sur la qualité de l'estimation. En effet, notre approche de localisation peut être considérée comme une solution multi-hypothèses où chaque particule gaussienne représente une possibilité d'estimation de l'état. Puisque notre objectif est de s'assurer que les localisations restent intègres, nous avons choisi de modéliser les différentes erreurs par le calcul de la distance entre la vérité de terrain et la particule ayant la meilleure justesse (la plus petite distance de Mahalanobis). Les algorithmes développés pour ces expérimentations ont été écrits en C++ et fonctionnent en temps réel avec une faible charge de calcul.

5.4.1 Localisation dans un environnement connu

La première expérimentation est une simulation de Monte-Carlo de 50 itérations basées sur l'utilisation du logiciel 4D-Virtualiz pour la simulation de la physique du robot et des mesures des capteurs. Pour chaque itération, le bruit sur les mesures des capteurs est calculé par un générateur de nombres pseudo-aléatoires dont l'initialisation varie à chaque simulation. Les bruits générés suivent les lois de probabilité normales qui ont été utilisées pour la configuration des algorithmes de localisation. L'environnement retenu est la plateforme Pavin de l'Institut Pascal (figure 5.10b) qui sera équipée de quatre balises UWB de positions connues fixées sur les murs et sur laquelle évolueront trois IPCars (figure 5.10a). Le premier IPCar (1) est équipé d'une balise UWB et d'un GPS RTK qui permet d'être localisé précisément. Ce véhicule fait alors office de balise mobile pour la localisation des autres robots. Les deux autres véhicules (2 et 3) seront équipés d'un GPS naturel (incertitude de 10 m) et d'une balise UWB. L'objet de cette expérimentation est de permettre à ces deux véhicules de se localiser avec une bonne précision en utilisant les communications qu'ils peuvent avoir : soit avec les balises de l'infrastructure, soit avec l'IPCar qui embarque un GPS RTK. On montrera ainsi que la localisation utilisant des mesures de distance est viable que ce soit avec des balises fixes ou mobiles.

Les simulations sont effectuées dans des conditions proches d'un environnement réel. Chaque



(a) Position initiale des éléments

(b) Aperçu dans le simulateur

FIGURE 5.10 – Représentation de l'initialisation de la simulation

robot est piloté automatiquement grâce à une loi de commande (THUILLOT et al. 2001) permettant de suivre en boucle une trajectoire prédéfinie. Les données odométriques calculées par le simulateur sont récupérées comme données d'entrée pour les étapes d'évolution des processus de localisation de chaque robot. Nous noterons cependant que ces données sont d'assez mauvaise qualité, car le moteur physique a tendance à provoquer beaucoup de glissement des robots. La configuration de la plupart des paramètres de la simulation est disponible dans le tableau 5.1. Chaque robot est initialisé à une position inconnue mais avec une orientation correcte. Cette position inconnue est représentée par une incertitude de position très grande (1 km) par rapport à la taille du terrain (environ $60 \text{ m} \times 50 \text{ m}$). Au démarrage, les calculs de localisation des robots ne prennent en compte que l'odométrie et les mesures GPS. Cela permet de laisser le temps au robot 1 d'obtenir une estimation précise de son état. Les mesures de distance entre les robots et les balises statiques démarrent 4 secondes après le début de la simulation. Lors de son déplacement, le robot 1 effectue une trajectoire en huit, passant par certaines petites ruelles de Pavin. Le robot 2 effectue une boucle passant par les rues les plus au centre de la zone d'expérimentation. Cela lui permet de pouvoir observer deux voire trois balises statiques simultanément. La trajectoire du robot 3 fait le tour de la zone en passant par la route la plus extérieure. Cette trajectoire ne lui permet que dans de très rares cas, de faire des observations de deux balises statiques simultanément.

Les résultats de la simulation sont présentés dans la figure 5.11 pour les robots 2 et 3. Les résultats du robot 1 ne sont pas présentés car l'utilisation d'un GPS RTK donne des erreurs de position et d'orientation négligeables par rapport aux autres robots. Les courbes correspondent aux erreurs de position, d'orientation ainsi que la distance de Mahalanobis entre les estimations obtenues et la vérité de terrain des robots 2 et 3. Nous pouvons remarquer que, de manière générale, les erreurs géométriques du robot 2 sont plus faibles que celles du robot 3. Cela vient du

TABLE 5.1 – Configuration de la première simulation

Configuration des robots		
Paramètre	Variable	Valeur
écart type de la position initiale	σ_{x0} & σ_{y0}	1 km
écart type de l'orientation initiale	$\sigma_{\theta0}$	34°
écart type de la vitesse linéaire (odométrie)		0.3 m s ⁻¹
écart type de l'angle de braquage (odométrie)		1.7°
vitesse de déplacement		1.7 m s ⁻¹
écart type des mesures GNSS en x et y		12 m
écart type des mesures GNSS précise (RTK) du robot 1		5 cm
Partitionnement à l'étape d'évolution		
coefficient de limitation de taille des supports	C_{supp}	2.57
coefficient de recouvrement des supports	C_{mix}	0.6
écart type maximal avant partitionnement	σ_{max}	30°
borne sup. de l'écart type à la création des particules	σ_{split}	20°
Partitionnement durant une observation de distance		
nombre de particules générées pour représenter une mesure	N_{meas}	14
seuil du critère de linéarité	s_{max}	6
Réduction de la mixture de gaussiennes		
poids minimal avant filtrage d'une particule	w_{min}	1×10^{-4}
seuil minimal du coup de réduction	c_{min}	0.03
Balises statiques		
portée maximale des balises		20 m
écart type des positions		20 cm
écart type des mesures de distance		15 cm
fréquence des mesures		10 Hz

fait que le robot 2 se trouvait dans de meilleures conditions de localisation puisqu'il se trouvait à la portée de plusieurs balises statiques bien localisées. Son erreur de position était inférieure à 30 cm dans ces situations. Quand un robot ne peut voir qu'une unique balise, il devient difficile d'assurer une bonne précision de l'estimation de l'état. En fonction de la direction de la trajectoire du robot par rapport à la position de la balise, la mesure de distance peut n'avoir que peu d'impact sur la correction de l'estimation. C'est le cas notamment pour le robot 3 qui effectue souvent de longues trajectoires linéaires perpendiculaires à la direction des mesures de distance. Dans ces situations, les mesures ne peuvent pas corriger l'incertitude dans la direction de la trajectoire et l'erreur de position augmente. Cette situation est également problématique lorsque la trajectoire du robot est dans la même direction que les observations de l'unique balise. Le robot 3 se trouve dans cette configuration dans l'intervalle 12 s à 20 s. L'incertitude d'orientation ne peut pas être réduite par les mesures et le robot peut parfois dévier de sa trajectoire. Nous pouvons en effet voir que l'erreur d'orientation du robot 3 peut alors atteindre 40°.

Le calcul des distances de Mahalanobis permet de vérifier l'intégrité des estimations obtenues. Dans le cas de notre approche, il s'agit de s'assurer qu'il y ait toujours une particule de

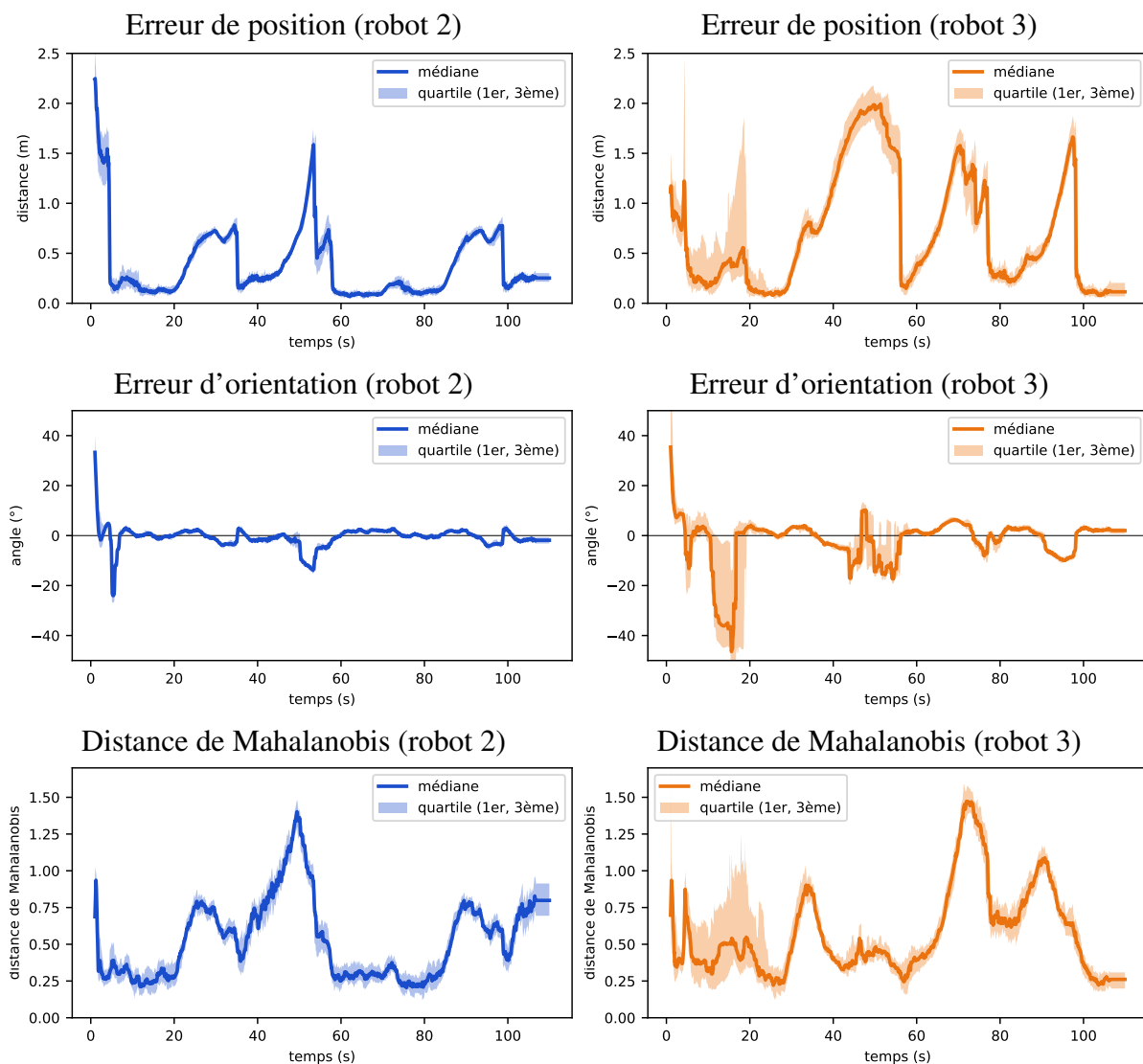


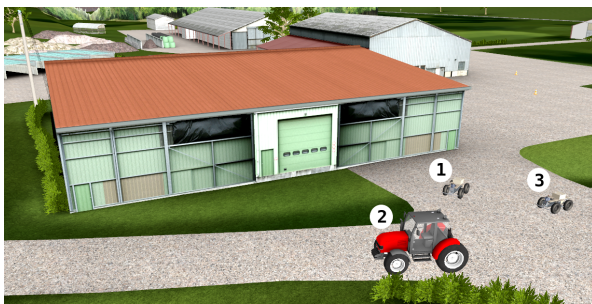
FIGURE 5.11 – Résultats obtenus pour la localisation des robots 2 et 3 (sur 50 simulations)

la mixture à proximité de la vérité de terrain. En utilisant la procédure présentée dans la section 2.7.8, la solution est considérée comme intègre si la distance de Mahalanobis est inférieure, dans 95 % des cas, à la limite de la loi du χ^2 pour 3 degrés de liberté : $\sqrt{\chi_3^2(0.95)} = 2.79$. Nous pouvons donc considérer que les localisations des robots 2 et 3 restent intègres puisque leurs distances de Mahalanobis ne dépassent pas 1.6. Nous pourrions cependant signaler que cette distance est parfois importante lorsque la mixture se partitionne et que la vérité de terrain se trouve entre deux particules.

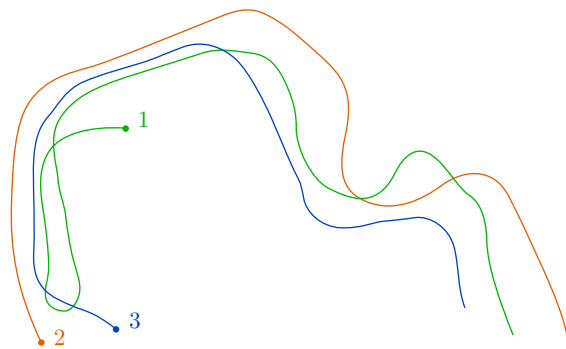
5.4.2 Groupe de robots mobiles

La seconde expérimentation est également une simulation exécutée 50 fois sur le simulateur 4DV-simulator. Le scénario présenté ici correspond à la localisation d'un groupe de robots

mobiles se déplaçant sur une distance d'environ 200 m dans un environnement inconnu : il n'y a cette fois-ci aucune balise statique. La figure 5.12 présente un aperçu de l'environnement de simulation ainsi que les trajectoires des trois robots. De la même façon que pour la simulation précédente, le robot 1 fait office de balise mobile bien localisée grâce à la fusion de mesures d'un GPS RTK dont l'incertitude est de 30 cm. Les robots 2 et 3 n'ont cette fois-ci aucun GNSS et ne peuvent donc qu'effectuer des observations de distance entre eux et le robot 1. L'objectif de cette simulation est de montrer que notre approche de localisation coopérative peut fonctionner avec des robots hétérogènes et avec très peu d'informations disponibles. En effet, le modèle de robot utilisé varie : les robots 1 et 3 sont des Aroco (petits robots par rapport au tracteur) et le robot 2 est un tracteur (plus gros). Le robot 3 a également la particularité d'être équipé de deux capteurs UWB. Cela signifie qu'il peut effectuer des observations des autres robots avec deux points de vue différents. Pour les robots 1 et 2, ces deux capteurs UWB sont simplement perçus comme deux balises observables dont les positions sont proches l'une de l'autre.



(a) Les robots (deux Arocos et un tracteur)



(b) trajectoire des trois robots (pos. initiale)

FIGURE 5.12 – Aperçu de l'environnement et des robots simulés ainsi que leurs trajectoires

La configuration de cette seconde simulation est présentée dans le tableau 5.2. Les paramètres des algorithmes de localisation n'ont pas été renseignés car ils sont similaires à ceux de la précédente simulation. La principale difficulté dans le calcul d'estimations précises des états des robots vient du fait que la seule entité apportant une source d'information précise est le robot 1. Puisqu'il n'y a cette fois-ci aucune autre balise bien localisée, il est difficile d'obtenir une bonne correction des estimations de position dans toutes les directions. Le robot 1 va faire varier sa vitesse et sa trajectoire afin de permettre l'observation des autres robots sous différents angles. Il ne s'agit donc pas d'un simple convoi de véhicules où les robots sont placés les uns derrière les autres. Le problème du convoi est qu'il ne permettrait pas de corriger l'incertitude d'orientation et de position dans la direction perpendiculaire à la trajectoire lorsque les robots se déplacent en ligne droite. Au démarrage de la simulation, le robot 1 ne sera pas immédiatement à la portée des capteurs UWB des robots 2 et 3, et il arrivera parfois, pendant l'expérimentation, que le robot quitte le champ de perception de ces robots.

Les résultats de la localisation des robots 2 et 3 sont présentés dans la figure 5.13. Les résultats du robot 1 ne sont pas présentés car l'utilisation d'un GPS RTK donne des erreurs

TABLE 5.2 – Configuration de la première simulation

Configuration des robots	
Paramètre	Valeur
écart type de la position initiale	1 km
écart type de l'orientation initiale	34°
écart type de la vitesse linéaire (odométrie)	0.3 m s ⁻¹
écart type de l'angle de braquage (odométrie)	1.7°
écart type des observations GNSS (RTK) du robot 1	30 cm
écart type des mesures de distance	15 cm
portée maximale des capteurs UWB	20 m
fréquence des observations de balises	10 Hz

de position et d'orientation négligeables par rapport aux autres robots. Les courbes en orange correspondent aux erreurs de position, d'orientation et la distance de Mahalanobis du tracteur (robot 2) et les courbes en bleu sont les erreurs de l'Aroco équipé de deux capteurs UWB (robot 3). Pendant les 10 premières secondes, le robot 1 n'est pas encore suffisamment proche pour être observable et les estimations des positions restent encore proche de l'initialisation. Les erreurs de position sont donc très grandes (environ 80 m) au début des simulations. Aux premières observations de distance avec le robot 1, les mixtures de gaussiennes des robots 2 et 3 vont se partitionner pour représenter l'incertitude de position en anneau autour du robot 1. Les graphiques montrent que l'erreur de position devient alors très faible, mais elle remonte progressivement pour le robot 2. En effet, la trajectoire de ce dernier se trouve alignée avec la position du robot 1 pendant une courte période et sort ensuite de sa portée. La légère erreur d'orientation des particules gaussiennes de son estimation vont alors s'éloigner de la vérité de terrain. Les premières observations ont également perturbé l'orientation du robot 2 dont l'erreur descend parfois à -40° . Ce robot se trouve cependant dans de meilleures conditions d'observation et cette erreur se corrige vite. Dans la suite de l'expérimentation, les robots 2 et 3 vont alterner entre des phases où le robot 1 est à leur portée et des phases où il ne l'est pas. Il arrive parfois que le robot 1 ne soit pas à portée du robot 3, mais le robot 2 fait alors office d'intermédiaire car celui-ci peut observer le robot 1 et son estimation devient suffisamment précise pour que les observations du robot 3 avec celui-ci aient un impact sur l'estimation de l'état du robot 3.

D'un point de vue global, la précision des estimations des états des robots n'est pas très bonne. L'utilisation d'un unique robot mobile bien localisé n'est pas une source d'information suffisante pour la localisation de ce groupe de trois robots, car il est rarement possible de corriger les erreurs de position dans toutes les directions. Nous pouvons cependant noter que notre approche arrive à gérer ce type de situation et reste intègre pendant toute la durée des simulations : la distance de Mahalanobis des robots reste inférieure au seuil de 2.79 plus de 95 % du temps. Il n'y a également que très peu de différence entre les résultats du robot 2 et du robot 3 bien que ce dernier soit équipé de deux capteurs UWB. La différence des bras de levier des

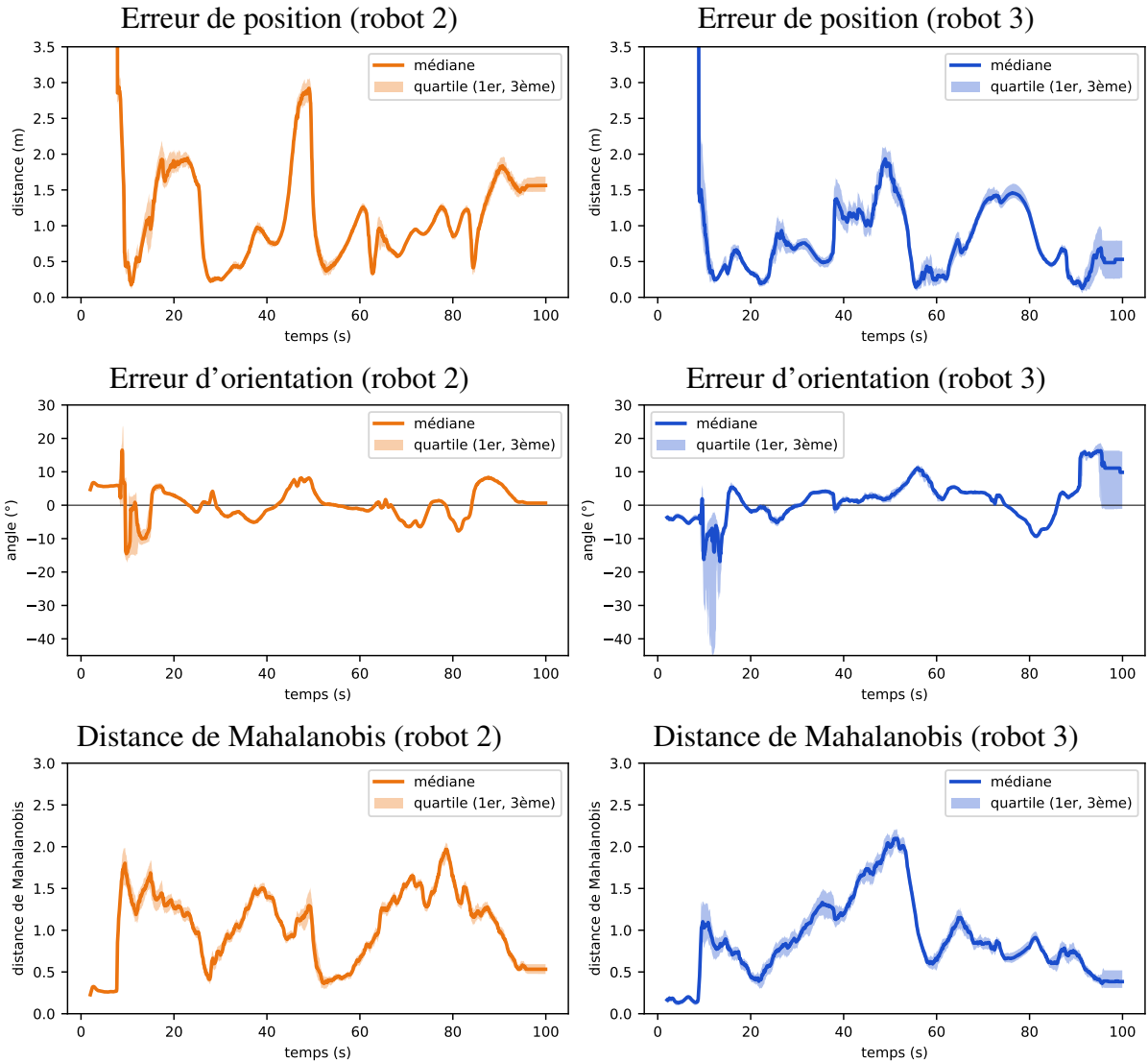


FIGURE 5.13 – Résultat obtenu pour la localisation des robots 2 et 3 (sur 50 simulations)

capteurs aurait pu avoir un impact sur la précision de l'orientation du robot 3 lorsque le robot 1 se trouve à côté de lui. Le problème ici est que la distance entre les deux capteurs (qui est de 86 cm) n'est pas suffisante par rapport à la distance des mesures. Les résultats auraient été de bien meilleure qualité si cet écart était de plusieurs mètres, mais il est rarement possible de les placer ainsi sur des robots de petite taille.

5.4.3 Localisation de balises statiques (expérimentation réelle)

Cette dernière expérimentation n'est cette fois-ci pas une simulation mais une **expérimentation réelle** réalisée grâce à l'utilisation de deux robots IPCar et de l'environnement Pavin se trouvant à proximité de l'Institut Pascal. Elle également fait l'objet d'une démonstration pour la visite HCERES de l'Institut Pascal en Janvier 2020.

L'objectif ici est assez similaire à la simulation précédente puisqu'il s'agit de localiser un ensemble d'entités grâce à l'unique source d'informations précises d'un robot mobile (1) équipé d'un GPS RTK (qui ne fournit pas l'orientation). L'environnement est donc encore considéré comme inconnu, mais nous y disposons 5 balises statiques dont la position est inconnue. Ces objets statiques sont considérés comme des robots et chacun va estimer sa position grâce aux observations de distance avec les robots à proximité. L'expérimentation comportera également un second robot (2) de position inconnue et ne pouvant effectuer que des observations de balises ainsi que des mesures d'odométrie. La figure 5.14 montre le placement des balises statiques ainsi que les positions initiales des robots 1 et 2. Comme le montre la figure 5.14, les balises statiques correspondent à de simples capteurs UWB montés sur des trépieds à une hauteur d'approximativement 1.70 m afin de faciliter leurs observations. Puisque leurs positions sont inconnues, les balises statiques peuvent être placées n'importe où dans l'environnement avant le démarrage de l'expérimentation. À la différence des capteurs UWB utilisés en simulation, les capteurs réels que nous avons à disposition ne peuvent fonctionner simultanément en émetteur et récepteur. Les balises statiques sont équipées de récepteur UWB : ils ne peuvent donc que répondre à des requêtes de distance. Le robot 1 est équipé d'un unique émetteur UWB et peut donc effectuer des requêtes avec les balises statiques et le robot 2. Le robot 2 possède un émetteur et un récepteur, lui permettant ainsi de répondre et envoyer des requêtes. La configuration de l'expérimentation est disponible dans le tableau 5.3. Les paramètres des algorithmes de localisation sont identiques à la première simulation.

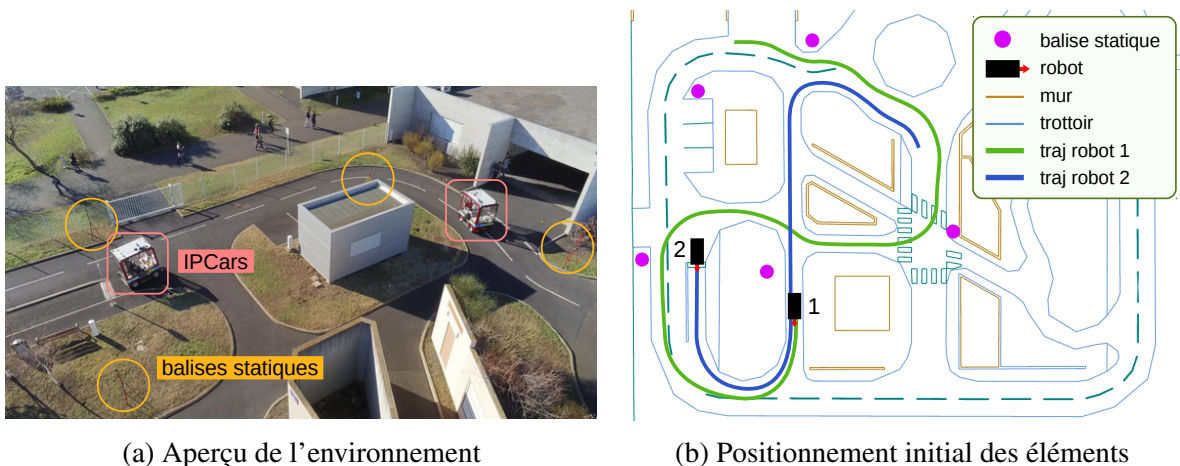


FIGURE 5.14 – Aperçu de l'environnement d'expérimentation et du placement initial des éléments dans la carte

Cette expérimentation est similaire à la simulation précédente en termes de quantité d'information extérieure apportée puisqu'il n'y a que les mesures GNSS précises du robot 1, mais l'ajout de balises statiques dans l'environnement a un impact important sur la qualité de la localisation de toutes les entités. En effet, grâce aux observations de distance avec le robot 1, les balises statiques ne peuvent que se localiser de mieux en mieux. Une fois que leurs estimations seront précises, elles feront office de bons points d'observation pour le robot mobile 2. Le

scénario de cette expérimentation peut donc être découpé en deux phases :

- le robot 1 se déplace dans l’environnement, localise l’ensemble des balises statiques, puis s’arrête et peut ainsi être considéré comme une balise statique
- le robot 2 est démarré lorsque le robot 1 termine son trajet. Il se déplace ensuite dans l’environnement et effectue des observations de distance avec les balises avoisinantes.

La portée maximale des capteurs UWB peut parfois monter à 50 m lorsque les conditions de visibilité sont bonnes. Elle est donc plus importante que dans les simulations (20 m) car ces dernières se limitent aux situations générales. La qualité des mesures n’est cependant pas toujours idéale et ces mesures sont parfois exclues lorsque la visibilité de la balise n’est pas bonne. L’incertitude des mesures de distance est également plus importante (1 m au lieu de 15 cm) car les mesures par ondes radios sont parfois soumises à des perturbations électromagnétiques causées par la présence d’objets métalliques aux alentours.

TABLE 5.3 – Configuration de la troisième expérimentation

Configuration des robots	
Paramètre	Valeur
écart type de la position initiale	1 km
écart type de l’orientation initiale	23°
écart type de la vitesse linéaire (odométrie)	0.1 m s ⁻¹
écart type de l’angle de braquage (odométrie)	2.3°
écart type des observations GNSS (RTK) du robot 1	10 cm
Balises statiques	
écart type de la position initiale	1 km
portée maximale des capteurs UWB	50 m
écart type des mesures de distance	1 m
fréquence des observations de balises	10 Hz

Les résultats de l’expérimentation sont présentés dans les graphiques de la figure 5.15. Ces courbes permettent d’étudier la qualité de l’estimation de l’état du robot 2 pendant son trajet au milieu des balises statiques qui ont été localisées auparavant par le robot 1. La justesse de la localisation des balises statiques n’est pas disponible puisque elles n’ont pas de vérité de terrain, mais nous pouvons cependant étudier la précision de leur localisation. Les courbes du premier graphique montrent l’évolution du logarithme décimal du déterminant de la somme des matrices de covariance des particules des mixtures de gaussiennes. Le déterminant est proportionnel aux volumes des ellipsoïdes représentant les covariances. Ces courbes permettent donc de se rendre compte de la taille des incertitudes des différentes estimations. Le robot 1 obtient des mesures GNSS très précises et possède donc un déterminant très faible dès la fusion de la première mesure. Étant donné qu’il s’agit de l’unique source d’information extérieure, les balises statiques ne pourront au mieux que converger vers la valeur du robot 1. Nous noterons cependant que la borne inférieure réelle est plus basse que le déterminant du robot 1 dans la période 2 s à 50 s puisque le robot est en mouvement et ajoute, par conséquent, une incertitude

due à l'imprécision de son odométrie. Lorsque les premières mesures de distance sont effectuées entre le robot 1 et les balises statiques, ces dernières vont partitionner leurs estimations d'état pour couvrir la totalité de l'anneau que représente la mesure. Avec le déplacement du robot, les mesures suivantes permettent d'éliminer progressivement les mauvaises particules jusqu'à ne conserver que la particule la plus juste. Cela peut prendre jusqu'à 40 s pour les balises les plus éloignées du robot, car il est nécessaire que le robot effectue un important déplacement avant de pouvoir éliminer toutes les ambiguïtés.

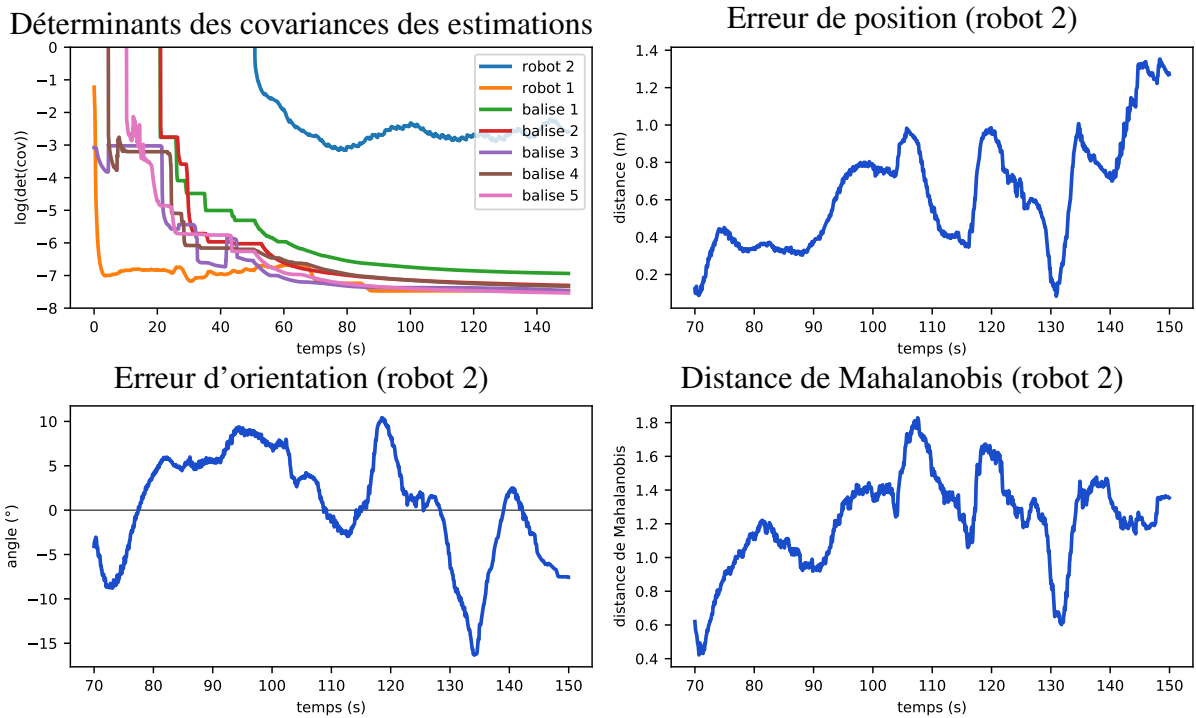


FIGURE 5.15 – Résultat obtenu pour l'expérimentation réelle avec 2 robots et 5 balises statiques. Les robots et les balises sont localisées grâce à l'aide d'un troisième robot équipé d'un GPS RTK

Au démarrage de l'algorithme de localisation du robot 2 (à 50 s), le robot reçoit des observations de distance avec plusieurs balises statiques qui sont maintenant bien localisées. La mixture de gaussiennes de son estimation d'état descend donc très rapidement à une unique particule. Le robot 2 commence son déplacement à 70 s et l'évolution de ses erreurs de position, d'orientation et sa distance de Mahalanobis est présentée dans les graphiques suivants de la figure 5.15. De façon générale, la qualité de la localisation du robot 2 est meilleure que celles que l'on pouvait avoir dans la simulation précédente bien qu'il s'agisse dans les deux cas d'expérimentations n'utilisant qu'un seul robot bien localisé. Cependant, après que le robot 1 ait localisé l'ensemble des balises statiques, cette expérimentation a beaucoup plus de points en commun avec la première simulation. Le robot 2 présenté ici devrait donc avoir une qualité de localisation proche de celle que pouvait avoir le robot 2 dans cette première simulation (figure 5.11). L'erreur de position est assez faible (de l'ordre de 40 cm) dans l'intervalle 70 s à 95 s

de sa trajectoire, mais cette erreur va ensuite augmenter alors que le robot reste à la portée d'au moins 4 balises différentes. Cela vient du fait que les mesures avec certaines balises sont biaisées lorsque le robot se trouve à certains endroits de l'environnement. La ligne de vue directe avec le capteur UWB n'est alors pas forcément disponible et le signal radio peut alors rebondir sur certains murs ou clôture métallique se trouvant à proximité. Ce phénomène est connu sous le nom de *multipath* dans la littérature scientifique. Bien qu'il soit possible de détecter les grosses variations dans les mesures et ainsi éviter leur prise en compte, cela est beaucoup plus difficile lorsque le biais est très faible. Puisque le bruit des mesures de distance est considéré comme blanc dans notre approche de localisation, les erreurs de position et d'orientation sont alors importantes quand les conditions électromagnétiques sont mauvaises pour les mesures.

5.5 Bilan

Ce chapitre a été consacré à l'illustration du fonctionnement de l'approche de localisation coopérative que nous avons développée. De courtes simulations ont permis d'observer le comportement du filtre dans les situations problématiques concernant la corrélation des états des robots, les ambiguïtés des trajectoires ou les initialisations très incertaines des objets à localiser. Des expérimentations plus réalistes ont ensuite été étudiées pour vérifier l'intégrité des localisations dans les situations où il est difficile d'obtenir de bonnes estimations des états des robots. Les résultats des deux simulations et de l'expérimentation réelle ont montré que notre algorithme reste intègre dans la plupart des configurations. La qualité de la précision des robots reste cependant assez faible, car ces derniers se trouvaient assez régulièrement dans des situations où ils se localisaient en odométrie pure ou en ne pouvant observer qu'une seule balise bien localisée. Dans le cas de l'expérimentation réelle, l'imprécision des estimations d'état du robot à localiser venait principalement du fait que certaines mesures de distance étaient biaisées à cause du rebond du signal sur des parois métalliques. D'un point de vue global, les résultats sont satisfaisants car notre approche peut gérer convenablement des situations dans lesquelles des filtres basiques divergeraient.

Chapitre 6

Conclusion et perspectives

6.1 Conclusion

Nous allons terminer ce manuscrit par un bref résumé des problématiques et des solutions que nous proposons dans le domaine de la localisation coopérative de robots mobiles. L'objectif principal de notre approche est de fournir une solution de localisation robuste, intègre et facilement intégrable dans des algorithmes de localisation individuelle existants. Cela signifie qu'elle doit fonctionner dans des expérimentations ayant le moins de contraintes possibles sur les robots et l'environnement :

- robots terrestres hétérogènes (modèles et capteurs)
- environnement connu ou inconnu
- fonctionnement pour tous les types de scénarios
- groupe dynamique de robots (ajout ou suppression de robots pendant une expérimentation).

L'étude de l'état de l'art (chapitre 3) nous a amené à choisir une solution de localisation coopérative décentralisée dont les observations inter-robots sont basées sur les mesures de distance de capteurs UWB. Le choix de cette stratégie nous a conduit à résoudre les problèmes de corrélation entre les états des robots (consanguinité de l'information) et de non-linéarité des modèles d'évolution et d'observation par mesures de distance.

L'aspect *décentralisé* de notre approche de localisation coopérative repose sur une modélisation particulière des différentes entités à localiser. Une entité peut correspondre à un robot mobile ou un objet statique (qui est alors considéré comme un robot sans modèle d'évolution) et doit être équipée d'au moins 1 capteur UWB. Un robot peut également exploiter les informations d'autres capteurs qui seront alors utilisés comme observations *locales*. Chaque entité ne s'occupe d'estimer que son propre état grâce à la fusion des observations qui sont à sa dis-

position. Le côté *coopératif* de la localisation vient alors du fait que le modèle d'observation des capteurs UWB fait intervenir l'estimation de position de l'entité observée. L'algorithme que nous avons développé est un filtre bayésien spécialisé pour la gestion des problèmes de non-linéarité des modèles et de la corrélation des données. L'utilisation d'une méthode bayésienne nous donne l'avantage de pouvoir facilement exploiter des algorithmes de localisation individuelle existants. La corrélation entre les estimations des états des robots est gérée par l'utilisation de la méthode de fusion SCI (intersection de covariance partitionnée) et la non-linéarité des modèles d'évolution et d'observation est gérée par une modélisation de l'estimation d'état d'un robot par une mixture de gaussiennes. Cette modélisation est plus difficile à gérer qu'un simple filtre de Kalman, mais notre algorithme a pour objectif d'imiter le fonctionnement d'un filtre gaussien lorsque la situation le permet, et de se complexifier lorsqu'il existe des ambiguïtés dans les estimations. Pour ce faire, nous avons mis en place des règles permettant d'augmenter ou diminuer le nombre de particules gaussiennes composant la mixture. Des critères ont été développés pour juger de la qualité de l'approximation générée par la linéarisation du modèle d'évolution et du modèle d'observation par mesure de distance. Lorsque la qualité est mauvaise, de nouvelles particules sont créées pour que l'approximation de l'estimation soit plus fidèle. Les méthodes de création des particules sont cependant spécifiques aux modèles utilisés. La réduction du nombre de particules de la mixture correspond à une élimination des particules ayant un poids trop faible et à une fusion des particules similaires.

Grâce à l'utilisation d'une grande variété de travaux dans différents domaines, nous avons pu développer un algorithme capable de valider l'ensemble des contraintes présentées. Le fonctionnement et la qualité des résultats de localisation ont été analysés grâce à l'utilisation de plusieurs simulations et expérimentations réelles avec un ensemble de robots mobiles et d'objets statiques placés dans l'environnement. L'objectif de ces expérimentations était de vérifier que les solutions de localisation ne divergeaient pas dans les situations les plus contraignantes : les robots à localiser démarraient leurs processus de localisation à des positions inconnues (c'est-à-dire avec une très grande incertitude de position) et ne pouvait que très rarement observer plus d'une balise UWB. Les résultats ont montré que notre approche est robuste et intègre dans ces conditions, bien que le peu d'information disponible puisse parfois provoquer d'importantes imprécisions dans les estimations. La dernière expérimentation montre qu'il est possible d'obtenir de bonnes conditions de localisation si la flotte de robots comporte au moins un robot capable de se localiser seul et précisément, et que l'on dispose de balises statiques à localiser dans l'environnement.

6.2 Contributions

Les travaux actuellement réalisés dans le domaine de la localisation coopérative s'orientent de plus en plus vers la méthode décentralisée, car elle offre plus de souplesse pour son utilisation dans des expérimentations réelles. Dans ce manuscrit, nous avons tenté de rendre cette utilisation la plus simple possible en abstrayant son intégration dans une approche bayésienne. La stratégie de développement de notre solution s'est basée sur l'idée que l'aspect coopératif de la localisation doit transparaître derrière une approche de localisation individuelle. Il ne s'agit ici que d'un type d'observation particulier faisant intervenir la position d'un objet externe. Cela rend alors l'approche de localisation coopérative très fortement décentralisée puisque le problème de coopération est alors assimilé à une étape de détection d'un capteur d'inter-distance intelligent. Cela nécessite cependant que l'approche de localisation individuelle soit capable de gérer la corrélation des mesures.

Une des particularités de notre approche concerne le modèle d'observation des balises par mesures de distance. Généralement, lorsqu'un filtre utilise une unique mesure de distance dans l'objectif d'améliorer l'estimation de position d'un objet, le modèle d'observation ne fait intervenir que le bruit de la mesure. Ce type de modèle est habituellement utilisé lorsque l'on souhaite localiser un robot à partir d'un ensemble d'objets observables dont la position est connue. Ce scénario peut cependant être étendu aux situations où la position de l'objet observé est imprécise. Notre méthode d'observation de balises a été conçue dans l'objectif de pouvoir considérer que les objets observés ont une estimation de leur position exprimée par une distribution normale. Cela revient à modéliser le bruit du modèle d'observation par un vecteur comprenant le bruit de la mesure de distance et le bruit de position (en x et y) de l'objet observé. En fonction de la configuration des estimations initiales des objets, il est alors possible d'envisager le scénario de localisation d'un robot à partir d'objets bien localisés, mais il est également possible de gérer les situations d'observations entre des robots.

Les problématiques de corrélations des états des robots et de non-linéarité du modèle d'observation par mesures de distance sont bien connues dans le domaine de la localisation coopérative, mais il est assez rare de voir des travaux traitant ces questions simultanément. Parmi les méthodes actuellement présentées, il existe par exemple les travaux de (LASSOUED et al. 2014) basés sur une méthode ensembliste qui a l'avantage de gérer la non-linéarité, mais les capteurs utilisés (GNSS) ne font ici pas intervenir ce problème. Notre approche est donc une des premières propositions d'algorithme de localisation coopérative gérant ces deux problématiques simultanément pour un grand nombre de situations.

L'exploitation de la modélisation par mixture de gaussiennes est assez rarement utilisée pour la localisation d'objets ou de robots mobiles par mesures de distance. Leurs utilisations se limitent aux situations où les estimations de position initiales des objets est inconnue. La

mixture permet alors de modéliser l'estimation de l'état recouvrant l'anneau de la mesure de distance ou la distribution bimodale qui apparaît après la fusion d'une seconde mesure. Lorsque la non-linéarité des modèles a un impact pendant toute une expérimentation, les travaux existants ont tendance à privilégier les filtres particuliers. Cela vient du fait qu'il est difficile de faire évoluer une mixture de gaussiennes autrement qu'en réduisant son nombre de particules. Grâce au développement des méthodes de partitionnement des mixtures, notre approche est capable de réaliser une évolution de la mixture de gaussiennes et permet ainsi d'obtenir une bonne approximation de la distribution réelle en toute circonstance.

6.3 Perspectives

Les quelques expérimentations que nous avons présentées ont permis de mettre en valeur les points les plus intéressants de notre approche, mais il reste encore beaucoup de champs à explorer d'un point de vue technique mais également scientifique.

Prise en compte des observations de balises modélisées par des mixtures de gaussiennes quelconques

Les observations de balises ne sont actuellement prises en compte que lorsque la position de la balise observée peut être estimée par une simple loi normale. Cela signifie que si la mixture de gaussiennes représentant l'estimation de l'état de la balise possède plusieurs particules, la mesure est ignorée. Il pourrait être intéressant d'élargir la capacité d'observation des balises aux situations où celles-ci sont estimées par des mixtures de gaussiennes, mais l'application de la mesure ne serait pas simple à gérer. En effet, lorsque l'on fusionne une mixture avec une autre, le nombre de particules résultant peut très rapidement augmenter. Une solution possible, simple à mettre en place, serait d'approximer la mixture par une unique gaussienne en utilisant la réduction de mixture standard (section 2.7.2) ou l'union de covariance (section 2.7.3). Les résultats ne seraient pas aussi précis que l'utilisation d'une mixture, mais cela permettrait d'apporter de l'information dans des conditions qui ne sont actuellement pas gérées.

Scénario de localisation pour le rejeu d'une trajectoire

La grande flexibilité de notre approche permet de réaliser des scénarios de localisation très variés. Un des scénarios pouvant être intéressant à étudier serait d'imiter le type d'applications généralement résolues par une méthode de SLAM (*simultaneous localization and mapping*). L'idée ici est de tester le rejeu d'une trajectoire d'un robot dans un environnement intérieur inconnu (ex : bâtiment, entrepôt, ...). Le robot serait piloté manuellement lors de son premier

passage afin d'enregistrer sa trajectoire, puis le robot re-parcourrait cette trajectoire en étant guidé par une loi de commande. L'objectif est alors de fournir la localisation nécessaire à ce scénario grâce à notre approche en utilisant un ensemble de balises statiques mal localisées placées dans l'environnement. Au premier passage du robot, celui-ci est considéré comme bien localisé (bien qu'il n'utilise que son odométrie) et permet ainsi la localisation des balises statiques grâce aux observations de distance. Au second passage, le robot est alors considéré comme mal localisé et exploite les observations avec les balises pour se localiser et rester sur sa trajectoire. Puisque cette cartographie des balises statiques se base sur l'odométrie du robot, il y a de fortes chances qu'elle ne soit pas juste, mais si la répétabilité de l'expérimentation est bonne, cela ne devrait pas être gênant pour le rejeu de la trajectoire du robot.

Amélioration de la gestion dynamique du groupe de robots

Notre approche permet actuellement de pouvoir démarrer ou stopper la localisation de n'importe quelle entité pendant une expérimentation, mais il est cependant nécessaire de connaître à l'avance l'ensemble des objets impliqués dans la localisation coopérative. En effet, les capteurs UWB se basent sur l'utilisation d'un identifiant pour savoir avec quelle balise ils doivent effectuer leurs mesures, et cette liste est actuellement fixée au début de l'expérimentation. Cependant, comme la plupart des technologies radios, les modules UWB sont capables d'effectuer des requêtes en *broadcast* (c'est-à-dire à toutes les balises à portée de l'émetteur). Il est alors possible pour un robot de mettre à jour régulièrement sa liste des balises observables, et de n'effectuer des requêtes de mesure avec uniquement les robots qui sont à sa portée. La solution de localisation deviendrait ainsi suffisamment dynamique pour être capable de fonctionner dans des scénarios urbains impliquant un très grand nombre de véhicules ou d'objets statiques.

Communication par UWB

La technologie UWB a été utilisée ici pour la réalisation de mesures de distance entre deux capteurs, mais cette technologie est avant tout un outil de radiocommunication. Notre solution de localisation nécessite actuellement la mise en place d'un réseau Wifi pour transmettre toutes les informations complémentaires à la mesure. Cela comprend l'estimation de la position de la balise observée, la covariance de cette position et la variance de la mesure. L'ensemble de ces informations peut tenir dans un message de 24 octets qui pourrait être transmis en même temps que la mesure. La solution de localisation coopérative deviendrait ainsi entièrement autonome et décentralisée d'un point de vue technologique.

Traitement du problème de *multipath* des capteurs UWB

L'expérimentation réelle nous a permis d'observer que les mesures de distance sont parfois biaisées lorsque le robot se trouve dans certaines zones de l'environnement. Cela vient du fait que l'environnement interagit avec les ondes électromagnétiques émises par les capteurs UWB et peut provoquer une mauvaise mesure lorsqu'il n'y a pas de ligne de vue directe entre l'émetteur et le récepteur. Ce phénomène est connu sous le nom de *multipath* et est étudié par les chercheurs travaillant dans le domaine de la radiocommunication (FOERSTER 2001), mais également en robotique mobile (GONZÁLEZ et al. 2009). Une des solutions possibles est d'ajouter un paramètre supplémentaire à estimer, correspondant au biais de la mesure avec une balise. Une autre solution, proposée par (WYMEERSCH et al. 2012), est beaucoup plus simple à mettre en place puisqu'il ne modifie pas le fonctionnement de l'algorithme de localisation. La solution présentée est de traiter le problème directement à l'étape de perception grâce à une analyse du signal reçu et à un algorithme de *machine learning* permettant d'atténuer l'erreur.

Bibliographie

- ANDERSON, Brian D. O. et John B. MOORE (1979). *Optimal Filtering*. eng. Prentice-Hall Information and System Sciences. Englewood Cliffs, NJ : Prentice-Hall.
- ARAKI, Brandon et al. (2019). « Range-Based Cooperative Localization with Nonlinear Observability Analysis ». In : *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, p. 1864-1870.
- BAR-SHALOM, Yaakov, X. Rong LI et Thiagalingam KIRUBARAJAN (2004). *Estimation with Applications to Tracking and Navigation : Theory Algorithms and Software*. John Wiley & Sons.
- BHATTACHARYYA, A. (1943). « On a Measure of Divergence between Two Statistical Populations Defined by Their Probability Distributions ». In : *Bull. Calcutta math. Soc.* 35, p. 99-109.
- BURCHETT, Bradley T. (2019). « Unscented Kalman Filters for Range-Only Cooperative Localization of Swarms of Munitions in Three-Dimensional Flight ». In : *Aerospace Science and Technology* 85, p. 259-269.
- CABALLERO, F., L. MERINO, I. MAZA et al. (2008). « A Particle Filtering Method for Wireless Sensor Network Localization with an Aerial Robot Beacon ». In : *2008 IEEE International Conference on Robotics and Automation*, p. 596-601.
- CABALLERO, F., L. MERINO et A. OLLERO (2010). « A General Gaussian-Mixture Approach for Range-Only Mapping Using Multiple Hypotheses ». In : *International Conference on Robotics and Automation*. IEEE, p. 4404-4409.
- CARRILLO-ARCE, Luis C. et al. (2013). « Decentralized Multi-Robot Cooperative Localization Using Covariance Intersection ». In : *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, p. 1412-1417.
- CHEN, H. D., K. C. CHANG et Chris SMITH (2010). « Constraint Optimized Weight Adaptation for Gaussian Mixture Reduction ». In : *Signal Processing, Sensor Fusion, and Target Recognition XIX*. T. 7697. International Society for Optics and Photonics, 76970N.
- CHONG, Chee-Yee et Shozo MORI (2001). « Convex Combination and Covariance Intersection Algorithms in Distributed Fusion ». In : *Proc. of the 4th International Conference on Information Fusion 2001*.
- CROUSE, D. F. et al. (2011). « A Look at Gaussian Mixture Reduction Algorithms ». In : *14th International Conference on Information Fusion*, p. 1-8.

- DJUGASH, Joseph et Sanjiv SINGH (2009). « A Robust Method of Localization and Mapping Using Only Range ». en. In : *Experimental Robotics*. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, p. 341-351.
- FABRESSE, F. R., F. CABALLERO et A. OLLERO (2015). « Decentralized Simultaneous Localization and Mapping for Multiple Aerial Vehicles Using Range-Only Sensors ». In : *International Conference on Robotics and Automation (ICRA)*. IEEE, p. 6408-6414.
- FOERSTER, Jeffrey R. (2001). « The Effects of Multipath Interference on the Performance of UWB Systems in an Indoor Wireless Channel ». In : *Vehicular Technology Conference (VTC)*. T. 2. IEEE, p. 1176-1180.
- GABELA, Jelena et al. (2019). « Experimental Evaluation of a UWB-Based Cooperative Positioning System for Pedestrians in GNSS-Denied Environment ». en. In : *Sensors* 19.23, p. 5274.
- GAIKWAD, Dhananjay Sudhakar et Mukesh ZAVERI (2011). « VANET Routing Protocols and Mobility Models : A Survey ». en. In : *Trends in Network and Communications*. Communications in Computer and Information Science. Berlin, Heidelberg : Springer, p. 334-342.
- GONZÁLEZ, Javier et al. (2009). « Mobile Robot Localization Based on Ultra-Wide-Band Ranging : A Particle Filter Approach ». In : *Robotics and autonomous systems* 57.5, p. 496-507.
- GUO, K., D. HAN et L. XIE (2017). « Range-Based Cooperative Localization with Single Landmark ». In : *International Conference on Control Automation (ICCA)*. IEEE, p. 588-593.
- GUO, Kexin et al. (2017). « Ultra-Wideband Based Cooperative Relative Localization Algorithm and Experiments for Multiple Unmanned Aerial Vehicles in GPS Denied Environments ». en. In : *International Journal of Micro Air Vehicles* 9.3, p. 169-186.
- HAN, Yongqiang et al. (2020). « A Novel Cooperative Localization Method Based on IMU and UWB ». en. In : *Sensors* 20.2, p. 467.
- HOWARD, Andrew, Maja J. MATARIC et Gaurav S. SUKHATME (2003). « Putting the 'I' in 'team' : An Ego-Centric Approach to Cooperative Localization ». In : *International Conference on Robotics and Automation (ICRA)*. T. 1. IEEE, p. 868-874.
- JAULIN, Luc (2000). « Le Calcul Ensembliste Par Analyse Par Intervalles et Ses Applications ». PhD Thesis.
- Jo, Kyoung-Hwan et Jihong LEE (2008). « Cooperative Localization of Multiple Robots with Constraint Propagation Technique ». In : *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 3477-3482.
- JULIER, S.J. et J.K. UHLMANN (2001). « Simultaneous Localisation and Map Building Using Split Covariance Intersection ». In : *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*. T. 3, 1257-1262 vol.3.
- JULIER, Simon J. et Jeffrey K. UHLMANN (1997). « A Non-Divergent Estimation Algorithm in the Presence of Unknown Correlations ». In : *In Proceedings of the American Control Conference*. Citeseer.

- JULIER, Simon J., Jeffrey K. UHLMANN et David NICHOLSON (2004). « A Method for Dealing with Assignment Ambiguity ». In : *American Control Conference, 2004. Proceedings of the 2004*. T. 5. IEEE, p. 4102-4107.
- KARAM, Nadir et al. (2006). « Collective Localization of Communicant Vehicles Applied to Collision Avoidance ». In : *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*. IEEE, p. 442-449.
- KOTECHA, J.H. et P.M. DJURIC (2003). « Gaussian Sum Particle Filtering ». In : *IEEE Transactions on Signal Processing* 51.10, p. 2602-2612.
- KULLBACK, S. et R. A. LEIBLER (1951). « On Information and Sufficiency ». In : *The Annals of Mathematical Statistics* 22.1, p. 79-86.
- KURAZUME, R., S. NAGATA et S. HIROSE (1994). « Cooperative Positioning with Multiple Robots ». In : *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1250-1257 vol.2.
- LASSOUED, Khaoula et al. (2014). « Mobile Robots Cooperation with Biased Exteroceptive Measurements ». In : *International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE, p. 1835-1840.
- LI, Hao et Fawzi NASHASHIBI (2013). « Cooperative Multi-Vehicle Localization Using Split Covariance Intersection Filter ». In : *IEEE Intelligent transportation systems magazine* 5.2, p. 33-44.
- MENSING, Christian, Stephan SAND et Armin DAMMANN (2010). « Hybrid Data Fusion and Tracking for Positioning with GNSS and 3GPP-LTE ». In : *International Journal of Navigation and Observation* 2010.
- MERINO, L., F. CABALLERO et A. OLLERO (2010). « Active Sensing for Range-Only Mapping Using Multiple Hypothesis ». In : *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, p. 37-42.
- MOHAMMADABADI, P. H. et S. VALAEE (2014). « Cooperative Node Positioning in Vehicular Networks Using Inter-Node Distance Measurements ». In : *25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, p. 1448-1452.
- NERURKAR, Esha D., Stergios I. ROUMELIOTIS et Agostino MARTINELLI (2009). « Distributed Maximum a Posteriori Estimation for Multi-Robot Cooperative Localization ». In : *Robotics and Automation, 2009. ICRA'09. IEEE International Conference On*. IEEE, p. 1402-1409.
- PREMVUTI, S. et S. YUTA (1990). « Consideration on the Cooperation of Multiple Autonomous Mobile Robots ». In : *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, 59-63 vol.1.
- REECE, S. et S. ROBERTS (2010). « Generalised Covariance Union : A Unified Approach to Hypothesis Merging in Tracking ». In : *IEEE Transactions on Aerospace and Electronic Systems* 46.1, p. 207-221.

- RISTIC, Branko, Sanjeev ARULAMPALAM et James McCARTHY (2002). « Target Motion Analysis Using Range-Only Measurements : Algorithms, Performance and Application to ISAR Data ». In : *Signal Processing* 82.2, p. 273-296.
- ROHANI, Mohsen et al. (2015). « A New Decentralized Bayesian Approach for Cooperative Vehicle Localization Based on Fusion of GPS and VANET Based Inter-Vehicle Distance Measurement ». In : *IEEE Intelligent Transportation Systems Magazine* 7.2, p. 85-95.
- ROUMELIOTIS, S.I. et G.A. BEKEY (2002). « Distributed Multirobot Localization ». In : *IEEE Transactions on Robotics and Automation* 18.5, p. 781-795.
- ROUVEURE, Raphaël et al. (2019). « Robot Localization and Navigation with a Ground-Based Microwave Radar ». In : *2019 International Radar Conference (RADAR)*, p. 1-4.
- RUIZ-SARMIENTO, J.R., C. GALINDO et J. GONZALEZ-JIMENEZ (2017). « Robot@home, a Robotic Dataset for Semantic Mapping of Home Environments ». en. In : *The International Journal of Robotics Research* 36.2, p. 131-141.
- RUNNALLS, A. R. (2007). « Kullback-Leibler Approach to Gaussian Mixture Reduction ». In : *IEEE Transactions on Aerospace and Electronic Systems* 43.3, p. 989-999.
- SALMOND, D. J. (1989). « Mixture Reduction Algorithms for Target Tracking ». In : *IEE Colloquium on State Estimation in Aerospace and Tracking Applications*, p. 7/1-7/4.
- SCHIEFERDECKER, D. et M. F. HUBER (2009). « Gaussian Mixture Reduction via Clustering ». In : *2009 12th International Conference on Information Fusion*, p. 1536-1543.
- TESSIER, Cédric et al. (2006). « A Real-Time, Multi-Sensor Architecture for Fusion of Delayed Observations : Application to Vehicle Localization ». In : *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*. IEEE, p. 1316-1321.
- THUILOT, B. et al. (2001). « Automatic Guidance of a Farm Tractor along Curved Paths, Using a Unique CP-DGPS ». In : *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*. T. 2, 674-679 vol.2.
- WANASINGHE, T. R., G. K. I. MANN et R. G. GOSINE (2014). « Decentralized Cooperative Localization for Heterogeneous Multi-Robot System Using Split Covariance Intersection Filter ». In : *Canadian Conference on Computer and Robot Vision (CRV)*, p. 167-174.
- WANG, Xuedong et al. (2017). « A Survey of Recent Advances in Particle Filters and Remaining Challenges for Multitarget Tracking ». en. In : *Sensors* 17.12, p. 2707.
- WYMEERSCH, H. et al. (2012). « A Machine Learning Approach to Ranging Error Mitigation for UWB Localization ». In : *IEEE Transactions on Communications* 60.6, p. 1719-1728.