



HAL
open science

Localisation ensembliste de drones à l'aide de méthodes par intervalles

Ide Flore Kenmogne Fokam

► **To cite this version:**

Ide Flore Kenmogne Fokam. Localisation ensembliste de drones à l'aide de méthodes par intervalles. Robotique [cs.RO]. Université de Rennes, 2019. Français. NNT : 2019REN1S084 . tel-02921454

HAL Id: tel-02921454

<https://theses.hal.science/tel-02921454>

Submitted on 25 Aug 2020

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.

THESE DE DOCTORAT DE

L'UNIVERSITE DE RENNES 1
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
*Mathématiques et Sciences et Technologies
de l'Information et de la Communication*
Spécialité : *Informatique*

Par

Ide Flore KENMOGNE FOKAM

**Localisation ensembliste de drones
à l'aide de méthodes par intervalles**

Thèse présentée et soutenue à Rennes, le février 2019
Unité de recherche : IRISA / INRIA Rennes Bretagne Atlantique

Rapporteurs avant soutenance :

Philippe BONNIFAIT Professeur, L2S, CNRS, Supélec, Université Paris Sud
Michel KIEFFER Professeur, Sorbonne Universités, Université de Technologie de Compiègne

Composition du Jury :

Président : (*à préciser après la soutenance*)
Examineurs : Philippe BONNIFAIT Professeur, Sorbonne Universités, Université de Technologie de Compiègne
 Luc JAULIN Professeur, ENSTA Bretagne, Lab-STICC
 Michel KIEFFER Professeur, L2S, CNRS, Supélec, Université Paris Sud
 Louise TRAVÉ-MASSUYÈS Directeur de recherche, LASS-CNRS Toulouse
Dir. de thèse : Éric MARCHAND Professeur, Université de Rennes 1
Co-enc. de thèse : Vincent DREVELLE Maître de Conférences, Université de Rennes 1

Table des matières

1. Introduction	1
1.1. Contexte	1
1.2. Objectifs et contributions	2
1.3. Plan du document	3
2. Estimation de pose d'un drone basée sur la vision	5
2.1. Introduction	5
2.2. Estimation de pose basée vision	5
2.2.1. Géométrie de prise de vue	6
2.2.1.1. Modèle géométrique	6
2.2.1.2. Modèle algébrique	7
2.2.1.3. Modèle de distorsion	10
2.2.1.4. Étalonnage d'une caméra	11
2.2.2. Paramétrisation des rotations	11
2.2.2.1. Quaternion unitaire	11
2.2.2.2. Vecteur-angle	12
2.2.2.3. Angles d'Euler	12
2.2.3. Estimation de pose à partir d'un modèle 3D connu	13
2.2.3.1. Formulation du problème	13
2.2.3.2. P3P	14
2.2.3.3. DLT	15
2.2.3.4. POSIT	16
2.2.3.5. Vers des calculs plus efficaces du PnP	17
2.2.3.6. Minimisation de la norme de l'erreur de re-projection	17
2.3. Estimation de la pose d'un drone basée vision	19
2.3.1. Déterminer la pose du drone à partir de la caméra	19
2.3.2. Assistance des capteurs embarqués	19
2.4. Conclusion	20
3. Estimation ensembliste par intervalles	21
3.1. Introduction	21
3.2. Calcul et analyse par intervalles	22
3.2.1. Définition des intervalles	22
3.2.2. Arithmétique des intervalles	23
3.2.3. Vecteurs et Matrices d'intervalles	24
3.2.4. Notion de pessimisme	25

Table des matières

3.2.5.	Fonctions d’inclusion	26
3.2.5.1.	Fonction d’inclusion vectorielle	26
3.2.5.2.	Fonction d’inclusion centrée	28
3.2.5.3.	Fonction d’inclusion de Taylor	29
3.2.6.	Sous-pavage	29
3.3.	Inversion ensembliste par intervalles	30
3.3.1.	Définition d’un problème d’inversion ensembliste	30
3.3.2.	Test d’inclusion	31
3.3.2.1.	Intervalle booléen	31
3.3.2.2.	Test d’inclusion	31
3.3.2.3.	Test d’inclusion et fonction d’inclusion	32
3.3.3.	Algorithme SIVIA	32
3.4.	Contracteurs	35
3.4.1.	Problème de satisfaction de contraintes	35
3.4.2.	Définition et propriétés des contracteurs	36
3.4.2.1.	Quelques propriétés d’un contracteur	36
3.4.3.	Exemples de contracteurs	37
3.4.3.1.	Contracteur trivial	37
3.4.3.2.	Contracteur par propagation et rétropropagation	37
3.4.3.3.	Contracteur par découpage : Shaving	39
3.4.3.4.	N-B consistance	40
3.4.4.	SIVIA avec contracteurs	41
3.5.	Inversion ensembliste q -relaxée	42
3.5.1.	Intersection q -relaxée	42
3.5.2.	Inversion ensembliste robuste par intervalles	42
3.5.3.	Détection de mesures aberrantes	44
3.6.	Conclusion	46
4.	Caractérisation d’un domaine de localisation pour un drone	47
4.1.	Introduction	47
4.2.	Modélisation du drone utilisé	48
4.2.1.	Repères et pose d’un drone	48
4.2.2.	Modèle cinématique	49
4.3.	Estimation de pose à partir d’un modèle de mesures à erreurs bornées	50
4.3.1.	Mesures à erreurs bornées	50
4.3.2.	Ensemble de contraintes liés à l’image	51
4.3.3.	Méthode de résolution du problème	52
4.3.4.	Définition des CSP et de l’ensemble solution correspondant	52
4.3.4.1.	CSP pour l’estimation des 6DoF	53
4.3.4.2.	CSP enrichi, estimation de 3 DoF	53
4.3.5.	Contracteur utilisé	54
4.3.6.	Calcul d’une estimée ponctuelle	54

Table des matières

4.4.	Validation et comparaison avec un filtre de Kalman étendu	54
4.4.1.	Validation en simulation	54
4.4.1.1.	Estimation des 6DoF	55
4.4.1.2.	Estimation de 3DoF	59
4.4.2.	Validation expérimentale	60
4.4.2.1.	Dispositif expérimental	60
4.4.2.2.	Calcul d'un domaine de la pose	63
4.4.2.3.	Détection d'inconsistances	64
4.4.3.	Comparaison à un filtre de Kalman étendu	66
4.4.4.	Évolution du domaine d'incertitude sur la pose en fonction de l'incertitude sur les mesures dans l'image	68
4.4.5.	Influence de l'incertitude de position des amers sur la taille du domaine	68
4.5.	Conclusion	69
5.	Prise en compte de l'incertitude de position des amers par élimination de quantificateurs	71
5.1.	Introduction	71
5.2.	Problème d'inversion ensembliste quantifié	72
5.2.1.	Problème d'inversion ensembliste 2D	72
5.2.1.1.	Re-formulation de l'ensemble solution	73
5.2.1.2.	Résultats du problème d'inversion avec N=1 amer	74
5.2.1.3.	Résultats du problème d'inversion avec N=3 amers	76
5.2.2.	Problème d'inversion ensembliste avec incertitude sur la position des amers	77
5.3.	Résolution par élimination du quantificateur existentiel	79
5.3.1.	Résolution d'un problème quantifié	80
5.3.1.1.	Définition	80
5.3.1.2.	Méthodes existantes	80
5.3.1.3.	QCSP et l'inversion ensembliste par intervalles	82
5.3.2.	Réécriture du problème d'estimation de pose avec amers incertains sous forme non quantifiée	82
5.4.	Résultats de simulation	86
5.4.1.	Résultats pour un amer observé	86
5.4.2.	Résultats avec trois amers	88
5.4.3.	Discussion	89
5.5.	Conclusion	89
6.	Localisation coopérative ensembliste de drones	91
6.1.	Introduction	91
6.2.	Modélisation d'un groupe de drones communicants	92
6.2.1.	Scénario	92
6.2.2.	Ensemble de contraintes liées aux mesures	93
6.2.2.1.	Contraintes liées aux mesures dans l'image	94

Table des matières

6.2.2.2.	Contrainte liée à la mesure de distance à la base	94
6.2.2.3.	Contraintes liées aux mesures d'inter-distances	94
6.2.2.4.	Contrainte liée à la mesure de l'altitude	95
6.3.	Localisation coopérative par partage de mesures	96
6.3.1.	Ensemble solution avec partage de mesures	96
6.3.2.	Résultats en simulation et comparaison à la méthode des moindres carrés	97
6.3.2.1.	Environnement de simulation	97
6.3.2.2.	Sous-pavages résultats dans le cas d'un robot isolé	98
6.3.2.3.	Localisation coopérative avec visibilité totale des amers	99
6.3.2.4.	Localisation coopérative avec visibilité réduite	102
6.3.3.	Résultats expérimentaux et comparaison	105
6.3.3.1.	Scénario expérimental	106
6.3.3.2.	Résultats et comparaison avec un filtre de Kalman étendu	106
6.3.4.	Problème de fusion centralisée (non passage à l'échelle du problème)	108
6.4.	Estimation coopérative par partage de domaines de position	108
6.4.1.	CSP lié aux mesures propres d'un robot (vision et distance à la base)	109
6.4.2.	CSP lié aux mesures d'inter-distances pour la coopération	110
6.4.2.1.	Contraction de la pose	110
6.4.2.2.	Atteinte d'un point fixe	111
6.4.3.	Résultats expérimentaux	111
6.4.3.1.	Influence de la coopération sur la taille du domaine	112
6.4.3.2.	Influence de la coopération en cas de visibilité réduite	113
6.5.	Modèle d'estimation/prédiction	114
6.5.1.	Réduction du domaine initial à chaque instant	114
6.5.2.	Présence des inconsistances	115
6.6.	Conclusion	117
7.	Conclusion et perspectives	120
7.1.	Conclusion	120
7.2.	Perspectives	121
Annexes		124
A.	Types de drones à voilure tournante	125
A.1.	Drones à un rotor	125
A.2.	Drones double rotors	127
A.3.	Drones quadrirotor	128
A.4.	Autres	129
A.5.	Drones utilisés dans ces travaux	130
A.5.1.	Quadrirotors utilisés	130
A.5.2.	Composants d'un drone	130

Table des matières

B. Architecture de l'environnement expérimental	133
B.1. architecture générale	133
B.2. Système de capture de mouvement Vicon	134
B.3. Drones expérimentaux	134
B.3.1. Drone de classe MK-Quadro	135
B.3.2. Drone de classe AR.Drone	135
C. Estimation robuste en présence de mesures aberrantes dans l'image	137
C.1. Solution q -relaxée	137
C.2. Résultats de la q -relaxée dans l'image	137

Liste des notations

Conventions typographiques

x	Réel
$[x]$	Intervalle $[x] = [\underline{x}, \bar{x}]$
$\{x\}$	Intervalle ponctuel $\{x\} = [x, x]$
\underline{x}	Borne inférieure d'un intervalle $[x]$
\bar{x}	Borne supérieure d'un intervalle $[x]$
\mathbf{x}	Vecteur
$[\mathbf{x}]$	Vecteur d'intervalles
\mathbf{X}	Matrice
$[\mathbf{X}]$	Matrice d'intervalles
$\bar{\mathbf{X}}$	Point en coordonnées non homogènes $\bar{\mathbf{X}} = (X, Y, Z)^\top$

Notations algébriques

\mathbb{R}^n	Espace des réels de dimension n
\mathbb{IR}^n	Espace des intervalles de dimension n
\mathbb{S}	Ensemble solution d'un problème
\mathcal{H}	Problème de satisfaction de contraintes (CSP)

Paramètres cinématiques

ϕ	Roulis
θ	Tangage
ψ	Lacet
p	Position du robot $p = (x, y, z)^\top$
q	Attitude du robot $q = (\phi, \theta, \psi)^\top$
\mathbf{r}	Pose d'un robot $\mathbf{r} = (x, y, z, \phi, \theta, \psi)^\top$

1. Introduction

Les avancées technologiques de cette dernière décennie ont suscité une attention croissante pour la robotique aérienne. Que ce soit dans le cadre militaire ou civil, les véhicules aériens sans pilote (ou drones) ont connu un intérêt fulgurant conduisant à des applications diverses. Nous pouvons par exemple citer, dans le cadre militaire, les missions de soutien opérationnel, les missions militaires de reconnaissance, etc. Dans le cadre civil, ces derniers sont de plus en plus utilisés, en particulier par l'industrie du divertissement (prises de vues aériennes, courses, spectacles), mais aussi pour la cartographie rapide de domaines tels que des champs et plantations (pour l'agriculture de précision) ou de zones sinistrées pour l'évaluation des dégâts. La manœuvrabilité de ces véhicules les destine également à l'inspection d'environnements dangereux ou inaccessibles par l'homme, d'ouvrages d'arts, de bâtiments, de mines, ...

La difficulté de naviguer dans de tels environnements encombrés et dont la structure est complexe se présente très rapidement. Ainsi, pour naviguer et accomplir ces différentes tâches, les drones ont besoin de se localiser par rapport à l'environnement. Pour ce faire, les mesures GPS seules deviennent insuffisantes ou inefficaces car elles sont facilement perturbées par le milieu environnant (masquage des satellites par les arbres et bâtiments, navigation en intérieur). Une alternative est d'utiliser un capteur externe comme une caméra. Les méthodes d'estimation de pose basée sur la vision peuvent alors fournir un positionnement du drone par rapport à son environnement.

En d'autres termes, afin de garantir la sûreté de l'engin et la réussite de la mission qui leur est confiée, ces véhicules ont besoin de développer un mécanisme qui leur permettent de se localiser précisément dans leur environnement. Il est aussi nécessaire de quantifier l'incertitude de positionnement afin de fournir une zone de sécurité. Cette quantification de l'incertitude permet de garantir le bon déroulement des différentes tâches de décollage, navigation, atterrissage et d'évitement d'obstacles.

1.1. Contexte

Les méthodes classiques d'estimation de pose (position et orientation) consistent généralement à trouver la valeur optimale de la pose en optimisant une fonction de coût qui lie les observations dans l'image à la pose. Dans le cas d'approches probabilistes, on cherche ainsi généralement à maximiser la vraisemblance de l'estimée par rapport aux mesures. Ces méthodes reposent sur la supposition qu'une description statistique du bruit de mesure est disponible, et permettent de déterminer une estimée optimale de la

1. Introduction

pose en utilisant un algorithme d'optimisation ponctuelle. Toutefois, dans certaines applications, il est difficile (parfois impossible) de décrire précisément les lois de probabilité des perturbations. Il est alors nécessaire de recourir à une hypothèse de bruits blancs Gaussiens indépendants. Dans ce cas, il peut être plus judicieux de considérer que les différentes erreurs (de mesures et de modèle) sont bornées et de bornes connues (qui bien-sûr tiennent compte des différents bruits de mesure et de modélisation).

On ne recherche donc plus une estimation ponctuelle de la pose optimisant un critère de coût, mais plutôt un ensemble de valeurs de pose *acceptables* pour les mesures disponibles. Cet ensemble est alors garanti de contenir toutes les poses consistantes avec les mesures à erreurs bornées (dont les bornes sont connues *a priori*).

Dans ce contexte où il faut quantifier et éventuellement borner l'erreur de positionnement, l'étude de *méthodes ensemblistes* pour la localisation se justifie par le fait qu'elles permettent de décrire l'ensemble des solutions compatibles avec les mesures, compte tenu de leurs incertitudes. Dans ces approches, les ensembles sont représentés à l'aide de domaines englobants simples à manipuler, comme des ellipsoïdes et les polytopes par exemple dans le cas linéaire, ou encore comme des unions de pavés intervalles dans le cas général. Nous avons fait le choix pour cette thèse de représenter nos ensembles par des unions de pavés, aussi appelées sous-pavages. Les méthodes basées sur l'analyse par intervalles permettent par ailleurs de calculer un résultat garanti, même dans le cas non-linéaire (ce qui est le cas du modèle d'observation sténopé utilisé dans cette thèse). Ces méthodes étant garanties, la preuve d'existence ou non d'une solution pour un problème est rendue possible, conduisant ainsi à la détection de mesure aberrantes dans l'ensemble des données utilisées.

Les méthodes ensemblistes par intervalles, étaient utilisées dès la fin des années 90 pour l'estimation à erreurs bornées (JAULIN et al. 1997 ; KIEFFER 1999), la planification de chemins (JAULIN et al. 1999), l'exploration sous marine (JAULIN et al. 2001), l'étude de stabilité et la commande robuste ; mais elles ont récemment connues un certain intérêt dans les domaines de la robotique mobile (DREVELLE 2011 ; GUYONNEAU 2013), le calcul d'espace de navigation (MORAS et al. 2012), ou encore la robotique aérienne (VOGES et al. 2018).

1.2. Objectifs et contributions

Ce travail de thèse a été réalisé dans le cadre du projet ARED Locoflot « *Localisation collaborative en flotille* ». Il a été co-financé par la région Bretagne et l'INRIA, et mené au sein des équipes-projets LAGADIC puis RAINBOW.

Il vise à démontrer que les méthodes ensemblistes par intervalles sont applicables aux problèmes de localisation pour des drones aériens, en mettant un accent particulier sur l'estimation de pose basée sur la vision par ordinateur, qui n'avait pratiquement pas été abordée dans le contexte de l'analyse par intervalles.

1. Introduction

Pour réaliser ce travail, les objectifs que nous nous sommes fixés sont de caractériser un domaine de pose pour un drone à partir de mesures dans l'image des positions des amers géoréférencés. La pose étant caractérisée, nous souhaitons quantifier l'incertitude de positionnement du drone dans un environnement contraint en fonction de l'incertitude sur la position des amers. Nous souhaitons ensuite effectuer une comparaison de l'approche ensembliste pour l'estimation de pose avec les méthodes classiques présentes dans la littérature. Enfin, nous résoudrons le problème de localisation coopérative ensembliste à partir d'une fusion des données des capteurs proprioceptifs et extéroceptifs dans un groupe de drones communicants.

La réalisation de ces objectifs a donné lieu aux contributions suivantes, publiées dans des conférences internationales :

- le calcul garanti du domaine de la pose d'un drone à partir de l'observation d'amers géoréférencés, sous hypothèse de mesures à erreurs bornées (KENMOGNE et al. 2017),
- l'amélioration de la prise en compte de l'incertitude de position des amers géoréférencés (dans le cas 2D), par élimination de quantificateur (KENMOGNE et al. 2018b)
- et, le calcul collaboratif de domaines de position dans une flottille de drones équipés de caméras et de mesure d'interdistances, d'abord par échange de mesures (KENMOGNE et al. 2018a) puis par échange de domaines (KENMOGNE et al. 2018c).

1.3. Plan du document

Après cette introduction générale, ce manuscrit commence par un premier chapitre dédié à l'estimation de pose basée sur la vision. Nous y présentons d'abord le modèle de formation d'image d'une caméra qui est le modèle sténopé. Par la suite, quelques méthodes classiques d'estimation de pose de la caméra à partir d'un modèle 3D connu sont présentées. Enfin, ce premier chapitre aborde comment déterminer la pose du drone à partir de celle de la caméra.

Le second chapitre présente l'estimation ensembliste par intervalles. Les outils de base de l'analyse par intervalles ainsi que l'outil de propagation de contraintes appelé contracteur sont détaillés. Ces notions sont illustrées par un exemple de localisation planaire à partir de balises. L'inversion ensembliste par intervalles est exposée et son extension pour la prise en compte de mesures aberrantes. Enfin, ce chapitre montre comment ces outils sont utilisés dans le processus d'estimation pour détecter les anomalies dans les données.

Le calcul d'un domaine de localisation pour un drone à partir de la vision est traité au chapitre trois. Le problème d'estimation de pose est modélisé sous forme d'un problème de satisfaction de contraintes. Dans un premier temps en considérant comme mesures les données issues de l'image uniquement et les contraintes associées. Par la suite une modélisation plus enrichie en considérant les données issues de certains capteurs proprioceptifs, contraignant ainsi trois des six degrés de libertés de la pose du drone. Les

1. Introduction

résultats obtenus sont présentés et commentés et une comparaison entre les deux cas de figure est effectuée.

Le problème traité au chapitre précédant ne prenant pas en compte explicitement l'incertitude sur la position des amers 3D géolocalisés, au chapitre quatre, une méthode d'estimation permettant de les intégrer en procédant à l'élimination de quantificateur est proposée. Un bref aperçu de l'influence de cette incertitude est présentée dans le cas simplifié de localisation 2D. Par la suite, la solution quantifiée est modélisée et les méthodes classiques de résolution des problèmes quantifiés sont exposées. La méthode d'élimination de quantificateur dans le cas 2D est détaillée et un ensemble solution non-quantifié est proposé. Ce chapitre se termine par quelques résultats en simulation qui montrent la pertinence de la solution proposée en ce qui concerne le temps de calcul de la méthode et la finesse du domaine calculé.

De plus en plus, des groupes de mini-drones aériens sont utilisés en collaboration pour accomplir des tâches de recouvrement ou des missions dans des environnements contraints. L'une des justifications est le besoin d'augmenter la capacité de perception en cas de défaillance d'un module. Ainsi, au chapitre cinq, nous présentons une solution au problème de localisation collaborative ensembliste. Dans un premier temps, en supposant un échange de mesures entre drones, une solution est proposée ainsi qu'une comparaison par rapport à la méthode des moindres carrés. Une deuxième méthode basée sur des échanges de domaines de pose solutions entre robots est ensuite proposée. Les résultats obtenus et une comparaison aux résultats d'un filtre de Kalman dans le cas à 2 robots sont présentés. Pour finir, les résultats obtenus pour deux, trois, quatre robots sont détaillés dans le cas d'une estimation simple et lorsqu'une procédure d'estimation-prédiction est implémentée.

Ce manuscrit se termine par une conclusion rappelant les contributions et les résultats majeurs de cette thèse ainsi que quelques perspectives envisagées.

2. Estimation de pose d'un drone basée sur la vision

2.1. Introduction

Nous ne pouvons traiter l'estimation ensembliste de pose d'un drone basée sur la vision sans aborder la notion d'*estimation de pose*. D'après HARTLEY et al. (2001) et MARCHAND et al. (2016), estimer la pose d'une caméra consiste à calculer la position et de l'orientation d'une caméra (calibrée) en utilisant un ensemble de mesures extraites des images. Ces mesures sont le plus souvent des projections 2D sur le plan image.

Une image en vision par ordinateur est le résultat de la projection du monde réel sur les cellules photosensibles d'une caméra à travers une optique (HARTLEY et al. 2001). Ainsi, si nous considérons un drone sur lequel est montée une caméra, pour estimer la pose (position et orientation) du drone, deux étapes sont nécessaires. La première consistera à procéder à l'estimation de la pose de la caméra et la seconde à déterminer celle du drone à partir de la pose de la caméra. Si l'on suppose que la transformation entre la pose de la caméra et celle du robot est connue par calibration, il nous suffira ici de porter une attention particulière aux méthodes d'estimations de pose de la caméra existantes dans la littérature.

Dans ce chapitre nous présentons quelques notions élémentaires de vision par ordinateur ainsi que la description du problème d'estimation de pose (section 2.2). On y détaillera le modèle de formation d'une image numérique ; après quoi le choix de la paramétrisation des rotations pour représenter l'attitude de la caméra est présentée ; le problème d'estimation de la pose connaissant un modèle 3D est ensuite formalisé et les méthodes existantes pour le résoudre seront présentés. La section 2.3 montre comment la pose du drone est déterminée à partir de celle de la caméra. Étant donné que les méthodes classiques d'estimation de pose fournissent une estimée ponctuelle comme solution, nous terminons ce chapitre par un positionnement des méthodes ensemblistes par intervalles comme un outil pour l'estimation garantie de la pose et la caractérisation d'un domaine de pose en tenant compte des incertitudes sur mesures.

2.2. Estimation de pose basée vision

Tel que défini par MANZANERA (2005), «*La vision par ordinateur désigne la compréhension d'une scène ou d'un phénomène à partir d'informations image, liant intimement*

2. Estimation de pose d'un drone basée sur la vision

perception, comportement et contrôle.» Son but n'est pas de comprendre ou de reproduire la vision humaine, mais principalement de construire un modèle algorithmique possédant des propriétés semblables. Un système de vision numérique est constitué d'une scène, d'une caméra numérique pour l'acquisition d'une image de la scène et d'un ordinateur pour son interprétation tel que le montre la figure 2.1. Une caméra numérique quant-à elle est constituée d'un capteur CCD ou CMOS permettant l'acquisition de l'image.

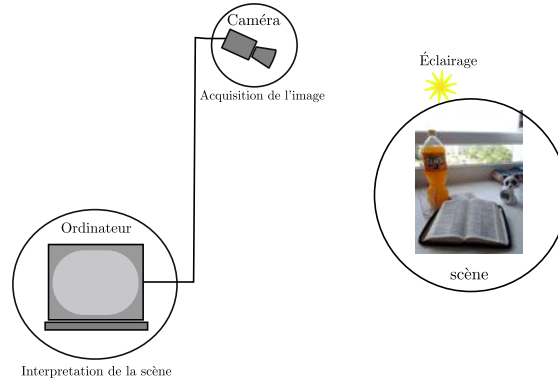


FIGURE 2.1. – Composants d'un système de vision par ordinateur.

Afin de pouvoir estimer la pose (la position et l'orientation) d'une caméra numérique à partir des informations extraites d'une image acquise, nous avons besoin d'un modèle pour décrire comment un point de la scène se projette sur le plan image de la caméra.

2.2.1. Géométrie de prise de vue

Il existe différents modèles permettant de caractériser une caméra numérique, tels que présentés par HARTLEY et al. (2004). Dans ces travaux, nous utiliserons le *modèle sténopé* (ou *pinhole model* en anglais) qui décrit une projection perspective (FAUGERAS 1993; HARTLEY et al. 2001).

Il permet de transformer les coordonnées d'un point 3D d'un objet de la scène en sa projection 2D dans l'image. La simplicité de ses équations et sa bonne approximation du modèle physique d'une caméra font de lui le modèle typiquement utilisé en vision par ordinateur.

2.2.1.1. Modèle géométrique

Le modèle sténopé est géométriquement constitué d'un *centre optique* (ou centre de projection) et d'un *plan image* (rétine). Il repose sur le principe selon lequel, tout point 3D de la scène se projette sur le plan image de la caméra au travers d'un rayon lumineux qui passe nécessairement par le centre optique. Le point issue de l'intersection du rayon avec le plan image est le *point image*. La *distance focale* f est la distance entre le centre

2. Estimation de pose d'un drone basée sur la vision

optique et le point principal qui est l'intersection entre la ligne de projection de O_c suivant l'axe optique. Ces notions sont illustrées à la figure 2.2.

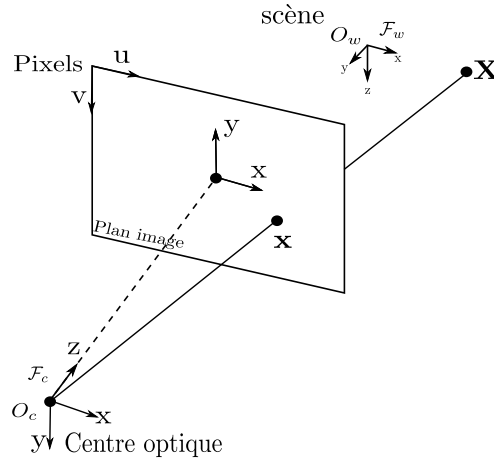


FIGURE 2.2. – Le modèle sténopé et repères utilisés.

2.2.1.2. Modèle algébrique

À partir du modèle géométrique, dérivent les équations de projection du modèle sténopé qui sont le résultat d'une série de transformations entre les repères représentés à la figure 2.2. Nous avons le repère cartésien $\mathcal{F}_c = (O_c, x, y, z)$ rattaché au centre optique de la caméra, le repère cartésien $\mathcal{F}_w = (O_w, x, y, z)$ qui est le repère inertiel ou le repère monde, le plan rétinien dont les axes sont parallèles aux axes x et y de \mathcal{F}_c et le plan image.

En vision par ordinateur, les coordonnées homogènes sont souvent utilisés pour représenter les équations du modèle sténopé. Elles permettent de l'exprimer par des relations linéaires. Dans la suite, les coordonnées homogènes de chaque point seront tels que : ${}^w\mathbf{X} = (X, Y, Z, 1)^\top$, ${}^c\mathbf{X} = ({}^cX, {}^cY, {}^cZ, 1)^\top$ et $\mathbf{x} = (x, y, 1)^\top$.

Transformation de repère La première transformation est un changement de repère entre \mathcal{F}_w et \mathcal{F}_c , associant à un point 3D de coordonnées ${}^w\mathbf{X}$ dans le repère monde ses coordonnées ${}^c\mathbf{X}$ dans le repère caméra. Le passage de ${}^w\mathbf{X}$ à ${}^c\mathbf{X}$ peut être exprimée à travers la fonction affine :

$${}^c\bar{\mathbf{X}} = {}^c\mathbf{R}_w {}^w\bar{\mathbf{X}} + {}^c\mathbf{t}_w \quad (2.1)$$

2. Estimation de pose d'un drone basée sur la vision

où ${}^c\bar{\mathbf{X}} = ({}^cX, {}^cY, {}^cZ)^\top$ et ${}^w\bar{\mathbf{X}} = (X, Y, Z)^\top$ représente les coordonnées non homogènes des points.

Cette transformation peut également être exprimée sous forme homogène grâce à la transformation rigide ${}^c\mathbf{T}_w$ caractérisée par la rotation ${}^c\mathbf{R}_w$ et la translation ${}^c\mathbf{t}_w$ tel que :

$${}^c\mathbf{X} = {}^c\mathbf{T}_w {}^w\mathbf{X} \quad (2.2)$$

avec la matrice homogène ${}^c\mathbf{T}_w$ définie par :

$${}^c\mathbf{T}_w = \begin{bmatrix} {}^c\mathbf{R}_w & {}^c\mathbf{t}_w \\ 0_{3 \times 1} & 1 \end{bmatrix} \quad (2.3)$$

Les paramètres de cette matrice sont appelés les *paramètres extrinsèques* de la caméra. ${}^c\mathbf{t}_w$ dénote la position de l'origine du système de coordonnées de la scène exprimée dans le repère \mathcal{F}_c centré sur la caméra. La position de la caméra dans le repère global peut être déterminée par ${}^w\mathbf{t}_r = -{}^c\mathbf{R}_w^\top {}^c\mathbf{t}_w$.

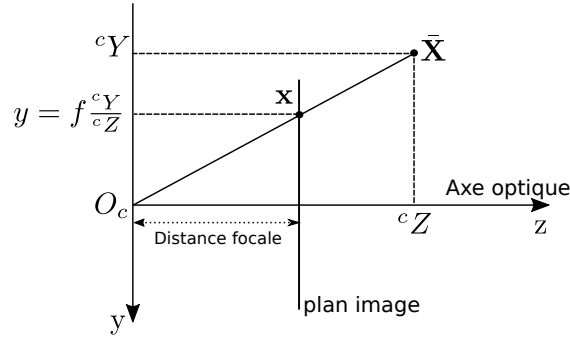


FIGURE 2.3. – Modèle projectif du sténopé.

Projection La deuxième transformation consiste à projeter le point ${}^c\mathbf{X}$ du repère caméra dans le plan rétinien ou capteur en un point 2D \mathbf{x} . Le calcul des coordonnées de \mathbf{x} (en unité métriques) se fait à partir des équations non-linéaires de projection perspective du modèle sténopé. Ces équations sont obtenues par une application simple du théorème de Thalès (illustrés à la figure 2.3) et exprimées comme suit :

$$\begin{aligned} x &= f \frac{{}^cX}{{}^cZ} \\ y &= f \frac{{}^cY}{{}^cZ} \end{aligned} \quad (2.4)$$

En coordonnées homogènes, l'équation 2.3 est donnée par :

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} {}^cX \\ {}^cY \\ {}^cZ \\ 1 \end{pmatrix} \quad (2.5)$$

2. Estimation de pose d'un drone basée sur la vision

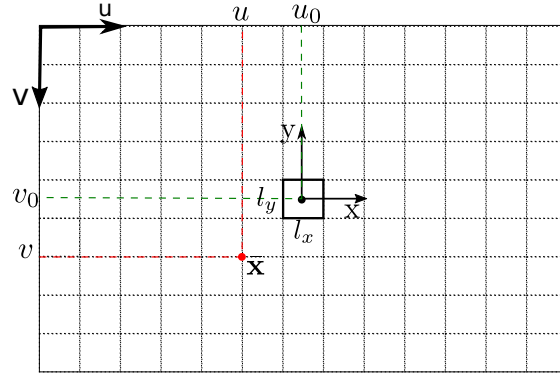


FIGURE 2.4. – Transformation en coordonnées pixelisées dans l'image.

Discretisation La dernière transformation consiste à passer des coordonnées métriques de \mathbf{x} en coordonnées pixelisées (u, v) dans le plan image. L'origine de ce plan est situé à l'extrémité de l'image et non au point central (voir figure 2.4). Un point dans l'image s'exprimera donc comme suit :

$$\begin{aligned} u &= u_0 + k_x \cdot x \\ v &= v_0 + k_y \cdot y \end{aligned} \quad (2.6)$$

En coordonnées homogènes, cette équation s'exprime de la manière suivante :

$$\underbrace{\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}}_{\bar{\mathbf{x}}} = \begin{pmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2.7)$$

où :

- (u_0, v_0) sont les coordonnées en pixels du point central, qui est l'intersection entre l'axe optique avec le plan image (qui est généralement le centre de l'image).
- $(k_x = \frac{1}{l_x}, k_y = \frac{1}{l_y})$ représentent les paramètres de changement d'échelle pixel-mètre, l_x étant la largeur d'un pixel et l_y sa hauteur.

A partir des équations 2.5 et 2.7, nous pouvons écrire :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} {}^cX \\ {}^cY \\ {}^cZ \\ 1 \end{pmatrix} \quad (2.8)$$

$$\bar{\mathbf{x}} = \begin{pmatrix} fk_x & 0 & u_0 \\ 0 & fk_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\Pi} {}^c\mathbf{X} \quad (2.9)$$

2. Estimation de pose d'un drone basée sur la vision

Posons $\alpha_u = fk_x$ et $\alpha_v = fk_y$, nous avons donc :

$$\bar{\mathbf{x}} = \underbrace{\begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\mathbf{\Pi}} {}^c\mathbf{X} \quad (2.10)$$

$$\bar{\mathbf{x}} = \mathbf{K} \mathbf{\Pi} {}^c\mathbf{X} \quad (2.11)$$

où la matrice \mathbf{K} représente la matrice des *paramètres intrinsèques* de la caméra et $\mathbf{\Pi}$ la matrice de projection perspective. La composition de ces transformations peut être résumée par la figure 2.5 et exprimée par l'équation suivante :

$$\bar{\mathbf{x}} = \mathbf{K} \mathbf{\Pi} {}^c\mathbf{T}_w {}^w\mathbf{X} \quad (2.12)$$

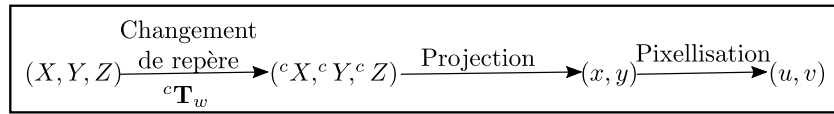


FIGURE 2.5. – Modèle sténopé complet.

2.2.1.3. Modèle de distorsion

Le modèle sténopé présenté ci-dessus suppose que la caméra soit idéale. Toutefois en fonction du système optique utilisé, des déformations ou distorsions peuvent apparaître sur l'image tel qu'illustré sur la figure 2.6.



FIGURE 2.6. – Image avec et sans distorsion.

Ces distorsions peuvent être radiale, prismatiques ou décentrées (WENG et al. 1992; ZELLER et al. 1996; GARCIA 2001). Lorsque ces distorsions deviennent importantes, il est nécessaire de les prendre en compte dans le modèle sténopé. Étant généralement de type radial, elles peuvent être modélisées de la façon suivante :

$$\begin{aligned} x &= x_d(1 + k_1r^2 + k_2r^4) \\ y &= y_d(1 + k_1r^2 + k_2r^4) \end{aligned} \quad (2.13)$$

2. Estimation de pose d'un drone basée sur la vision

où (x_d, y_d) représente les coordonnées métriques du point distordu, $r^2 = x^2 + y^2$ est la distance entre le point distordu et le centre de l'image, k_1 et k_2 sont les paramètres de distorsion. Les paramètres intrinsèques du modèle avec une image distordu sont donc $(u_0, v_0, \alpha_u, \alpha_v, k_1, k_2)$.

2.2.1.4. Étalonnage d'une caméra

Le modèle étant défini, calibrer la caméra reviendra donc à déterminer les paramètres intrinsèques et extrinsèques de celle-ci. Il s'agira principalement de calculer ses paramètres intrinsèques qui sont propre à la caméra utilisée et ne changent pas si l'on la déplace. On parle d'étalonnage hors ligne de la caméra (BROWN 1971 ; TSAI 1986 ; FAUGERAS et al. 1987 ; STEIN 1997 ; ZHANG 2000 ; BRADSKI et al. 2008).

En ce qui concerne les paramètres extrinsèques, des méthodes spécifiques (dites de localisation) existent pour les calculer. Dans la suite, nous supposons connus les paramètres intrinsèques (nous disposons donc d'une caméra calibrée).

2.2.2. Paramétrisation des rotations

La matrice de rotation ${}^c\mathbf{R}_w$ (représentant l'attitude de la caméra) appartient au groupe spécial orthogonal de dimension 3 ou $SO(3)$ défini par :

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}_3, \det(\mathbf{R}) = 1 \right\}$$

Il existe plusieurs représentations permettant de paramétrer de façon minimale la matrice de rotation.

2.2.2.1. Quaternion unitaire

Un quaternion unitaire est un quadruplet constituant une représentation non minimale de la matrice de rotation \mathbf{R} (ALTMANN 1986). Soit $\mathbf{q} = a + bi + cj + dk$ un quaternion unitaire tel que $|\mathbf{q}| = 1$, la paramétrisation de la rotation par \mathbf{q} nous permet d'écrire la matrice de rotation ${}^c\mathbf{R}_w$ comme suit :

$$\begin{pmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2ac + 2bd \\ 2ad + 2bc & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & a^2 - b^2 - c^2 + d^2 \end{pmatrix} \quad (2.14)$$

Les avantages de cette représentation sont : qu'elle est la plus compacte que la représentation matricielle, elle représente toujours une matrice de rotation même en cas d'erreurs et elle permet d'éviter le blocage de cardan (*Gimbal lock*) dont souffre la représentation par les angles d'Euler.

2. Estimation de pose d'un drone basée sur la vision

Malgré ses avantages, cette représentation n'est pas celle que nous avons choisi dans ce travail. En effet, les capteurs proprioceptifs des drones mesurent des angles et par conséquent ont des erreurs de mesures sur des angles, qui ne sont pas représentables en utilisant les quaternions. De plus, nous ne pouvons représenter des intervalles d'angles en utilisant les quaternions, car outre les valeurs d'angles contenues dans les intervalles, les intervalles de quaternions contiendront des valeurs indéterminées, qui ne correspondent pas toujours à une rotation.

2.2.2.2. Vecteur-angle

Encore appelée représentation $\theta\mathbf{u}$ où $\mathbf{u} = (u_x, u_y, u_z)^\top$ est un vecteur unitaire de dimension 3 et θ un scalaire. Elle est constituée en un vecteur de rotation instantané, se traduisant comme la rotation de θ autour d'un axe de rotation \mathbf{u} . C'est une forme intuitive qui permet de facilement lever les ambiguïtés. Le passage de la matrice \mathbf{R} à cette paramétrisation se fait comme suit :

$$\mathbf{R} = e^{[\theta\mathbf{u}]_\times} \quad (2.15)$$

où $[\theta\mathbf{u}]_\times$ est une matrice antisymétrique dite *matrice de pré-produit vectoriel* associée à $\theta\mathbf{u}$ définie par

$$[\theta\mathbf{u}]_\times = \begin{pmatrix} 0 & -u_z\theta & u_y\theta \\ u_z\theta & 0 & -u_x\theta \\ -u_y\theta & u_x\theta & 0 \end{pmatrix} \quad (2.16)$$

À partir de la formule de Rodrigues, la rotation peut être réécrite comme suit :

$${}^c\mathbf{R}_w = \mathbf{I}_3 + (1 - \cos\theta)\mathbf{u}\mathbf{u}^\top + \sin\theta[\mathbf{u}]_\times \quad (2.17)$$

avec

$$[\mathbf{u}]_\times = \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix} \quad (2.18)$$

2.2.2.3. Angles d'Euler

Représentation utilisée dans ce manuscrit, ce sont trois angles couramment notés ϕ , θ et ψ (appelés tangage, roulis et lacet) qui représentent les angles de rotation de la caméra suivant ses trois axes X, Y et Z. C'est la paramétrisation la plus répandue dans l'aviation.

Pour construire de manière explicite la matrice ${}^c\mathbf{R}_w$ en fonction des angles d'Euler, nous considérons initialement le repère caméra aligné avec le repère global (monde) \mathbf{R} . ${}^c\mathbf{R}_w$ est le résultat de trois rotations consécutives à l'aide de deux repères intermédiaires. Soit $\mathbf{R}(a, \alpha)$, la rotation d'angle α autour de l'axe unité a .

La chaîne des rotations est la suivante :

2. Estimation de pose d'un drone basée sur la vision

1. rotation d'angle ψ autour de Z donnée par

$$\mathbf{R}_z(\psi) = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2. rotation d'angle θ autour de Y donnée par

$$\mathbf{R}_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

3. rotation d'angle ϕ autour de X

$$\mathbf{R}_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix}$$

La matrice ${}^c\mathbf{R}_w$ est égale à la matrice de rotation avec l'agencement des trois rotations suivant le formalisme $\mathbf{R}_{zyx}(\phi, \theta, \psi) = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$ décrit par :

$${}^c\mathbf{R}_w = \begin{pmatrix} \cos \psi \cos \theta & \sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & -\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{pmatrix} \quad (2.19)$$

L'avantage d'utiliser les angles d'Euler c'est qu'ils fournissent un modèle d'angles le plus proche de la réalité physique du mouvement d'une caméra dans une scène. Toutefois cette paramétrisation peut produire un phénomène de blocage physique provoqué lorsque la deuxième rotation aligne les deux autres axes de rotation engendrant ainsi une singularité. On parle alors de *gimbal lock*, ou blocage de cardan.

Bien que la représentation par les angles d'Euler présente une singularité, son utilisation dans ces travaux se justifie par le fait qu'elle est la plus facile à interpréter physiquement. Elle permet également l'utilisation directe des incertitudes sur les angles issues du capteur (centrale inertielle) permettant d'avoir les mesures des angles de rotation du drone. La singularité ne posera pas de problèmes en pratique, l'orientation des robots considérés dans cette thèse (drones évoluant quasiment à l'horizontale) se situant à bonne distance du *gimbal lock* (nez pointant vers le zénith ou le nadir).

Les rotations ainsi paramétrées, l'estimation de la pose de la caméra consistera donc à estimer les paramètres extrinsèques qui sont l'attitude ϕ_c, θ_c, ψ_c à partir de ${}^c\mathbf{R}_w$ et la position x_c, y_c, z_c donnés par ${}^w\mathbf{t}_r = -{}^c\mathbf{R}_w^\top {}^c\mathbf{t}_w = (x_c, y_c, z_c)^\top$.

2.2.3. Estimation de pose à partir d'un modèle 3D connu

2.2.3.1. Formulation du problème

Considérons une caméra calibrée telle que la matrice de paramètres intrinsèques \mathbf{K} , est connue. Calculer la pose à partir d'un modèle 3D connu consiste à déterminer la

2. Estimation de pose d'un drone basée sur la vision

transformation ${}^c\mathbf{T}_w$ entre le repère global \mathcal{F}_w et le repère de la caméra \mathcal{F}_c en fonction des mesures fournies par la caméra grâce à l'équation 2.12. Si nous avons N points 3D ${}^w\mathbf{X}_i$ (avec $i = 1 \dots N$) observés, pour lesquels les mesures correspondantes 2D dans l'image $\bar{\mathbf{x}}_i$ sont connues, ${}^c\mathbf{T}_w = ({}^c\mathbf{R}_w, {}^c\mathbf{t}_w)$ peut être calculé en résolvant le système formé par l'ensemble des équations issues de l'application du modèle sténopé à chacun des couples de positions points 2D-3D connus.

Il s'agit d'un problème inverse connu sous le nom de problème de *Projection Perspective à partir de N points* ou PnP (*Perspective-n-Point* en anglais).

Plusieurs techniques pour résoudre le problème d'estimation de pose existent. Dans la suite, nous présenterons deux méthodes classiquement utilisées pour l'estimation de pose ainsi que quelques méthodes plus efficaces. En cas de présence de mesures aberrantes, une estimation robuste est rendue possible en couplant ces méthodes avec l'algorithme RANSAC (FISCHLER et al. 1981).

2.2.3.2. P3P

Cette approche consiste à considérer le plus petit nombre de points pour lequel on a un nombre fini de solutions. Lagrange fut le premier à fournir une méthode en utilisant $N = 3$ points 3D observés ; ce qui conduit à trouver quatre solutions. Or, à partir de $N = 4$ points, il n'y a qu'une seule solution.

L'approche de résolution est constituée de deux étapes.

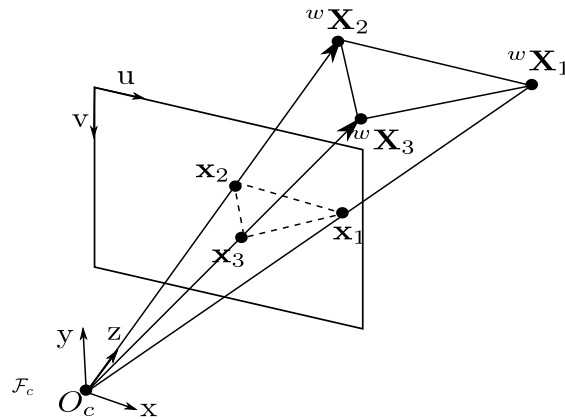


FIGURE 2.7. – Projection perspective des points observés dans l'image. Illustration des 3 triangles utilisés pour l'application de la méthode des cosinus.

- Étape 1 : elle consiste à déterminer la profondeur cZ_i de chaque point dans le repère de la camera en résolvant un système de 3 équations linéaires quadratiques avec 3 inconnues (QUAN et al. 1999 ; GAO et al. 2003) obtenu en appliquant la loi des cosinus aux angles entre les trois triangles formés par le centre de projection \mathbf{O}_c et chaque paire de 2 points 3D sur les 3 observés (${}^w\mathbf{X}_1 \mathbf{O}_c {}^w\mathbf{X}_2$, ${}^w\mathbf{X}_2 \mathbf{O}_c {}^w\mathbf{X}_3$ et

2. Estimation de pose d'un drone basée sur la vision

${}^w\mathbf{X}_3 \mathbf{O}_c {}^w\mathbf{X}_1$, illustrés à la figure 2.7). Posons $d_{ij}^2 = \|{}^w\mathbf{X}_i - {}^w\mathbf{X}_j\|_2$ et $\|O_c {}^w\mathbf{X}_i\|_2 = d_i$, l'application de la loi des cosinus conduit à :

$$d_i^2 + d_j^2 - 2d_i d_j \cos \delta_{ij} = d_{ij}^2 \quad (2.20)$$

où δ_{ij} est l'angle entre les triangles connue. En appliquant cette lois aux trois triangles, nous arrivons à déterminer la profondeur de chaque point. Le reste du problème consistera donc à déterminer la rotation, donc un problème d'orientation absolue.

$$\mathbf{x}_i = {}^c\mathbf{R}_w {}^w\mathbf{X}_i + {}^c\mathbf{t}_w \quad (2.21)$$

- Étape 2 : elle revient à déterminer la transformation ${}^c\mathbf{T}_w$ qui associe aux coordonnées de chaque point 3D dans le repère \mathcal{F}_w , les coordonnées d'un point 3D dans le repère de la caméra \mathcal{F}_c . Ainsi, si $\mathbf{r}_c \in se(3)$ est une paramétrisation de ${}^c\mathbf{T}_w$, ayant déterminé la profondeur, cette deuxième étape consiste à calculer la valeur de \mathbf{r}_c telle que

$$\hat{\mathbf{r}}_c = \operatorname{argmin}_{\mathbf{r}_c} \sum_{i=1}^N ({}^c\mathbf{X}_i - {}^c\mathbf{T}_w(\mathbf{r}_c) {}^w\mathbf{X}_i)^2 \quad (2.22)$$

En utilisant une paramétrisation par des quaternions, \mathbf{r}_c peut être calculée de manière analytique (HORN 1987) ou par la méthode des moindres carrés utilisant la décomposition en valeurs singulières (SVD) (AMELLER et al. 2000).

La méthode P3P a l'avantage de ne pas nécessiter d'initialisation. Elle n'est pas précise car quatre solutions différentes sont obtenues. L'ajout d'un quatrième point permet de lever l'ambiguïté.

2.2.3.3. DLT

Bien-que le PnP soit un problème non-linéaire, il est possible de le résoudre à base d'un système d'équations linéaires. Cette technique est appelée la transformation linéaire directe (DLT pour *Direct Linear Transform*) (SUTHERLAND 1974 ; HARTLEY et al. 2001) et fournit une estimation de la pose mais ne considère malheureusement pas toutes les contraintes non-linéaires présentes dans la matrice de rotation. Pour un nombre $N \geq 3$ points, cette méthode est généralement utilisée car un plus grand nombre de points augmente la précision de la pose estimée.

Le principe est d'exprimer la relation de projection sous forme d'un système linéaire tel que

$$\mathbf{x}_i = \mathbf{A} \mathbf{y}_i \quad \text{où } i \in [1, N] \quad (2.23)$$

$\mathbf{x}_i, \mathbf{y}_i$ sont des vecteurs connus et \mathbf{A} la matrice des inconnues. Ainsi, pour chaque correspondance 3D-2D, deux équations linéairement indépendantes sont obtenues, données

2. Estimation de pose d'un drone basée sur la vision

par :

$$x_i = \frac{a_{11}X + a_{12}Y + a_{13} + a_{14}}{a_{31}X + a_{32}Y + a_{33} + a_{34}} \quad (2.24)$$

$$y_i = \frac{a_{21}X + a_{22}Y + a_{23} + a_{24}}{a_{31}X + a_{32}Y + a_{33} + a_{34}} \quad (2.25)$$

où les coefficients a_{ij} sont les éléments de la matrice de transformation

$$\mathbf{A} = ({}^c\mathbf{R}_w, {}^c\mathbf{t}_w) = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix} \quad (2.26)$$

Après cette étape, une méthode standard de résolution d'un système linéaire est utilisée pour déterminer \mathbf{A} en exprimant l'équation 2.23 sous la forme $\mathbf{B}\mathbf{p} = 0$ où \mathbf{p} est le vecteur constitué des 12 inconnues et la matrice \mathbf{B} donnée par :

$$\mathbf{B} = \begin{pmatrix} X_i & Y_i & Z_i & 0 & 0 & 0 & 0 & -X_i x_i & -Y_i x_i & -Z_i x_i & -x_i \\ 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -X_i y_i & -Y_i y_i & -Z_i y_i & -y_i \end{pmatrix} \quad (2.27)$$

La solution de ce système est le vecteur propre correspondant à la plus petite valeur propre possible de la décomposition SVD de \mathbf{B} .

Les avantages de cette méthode sont qu'elle ne nécessite pas d'initialisation et qu'elle est directe (en une étape). Toutefois elle est très sensible aux bruits, sa paramétrisation comparativement au P3P n'est pas minimale et les contraintes géométriques ne sont pas directement appliquées. Aucune contrainte n'étant imposée sur le paramètre définissant la rotation (a_{ij} $i, j \leq 3$) il est nécessaire d'effectuer a posteriori une orthogonalisation de \mathbf{R} . Une solution qui tient compte explicitement des contraintes non-linéaires du système est préférable.

2.2.3.4. POSIT

Publiée par DEMENTHON et al. (1995), POSIT est une solution alternative et très élégante, qui prend en compte les contraintes non-linéaires du problème d'estimation de pose. Elle est capable de calculer la pose de la caméra à partir d'une seule image et ne nécessite aucune valeur de initiale de la pose. Pour ce faire, un nombre minimum de $N = 4$ correspondances entre points 3D et 2D est nécessaire. Cette méthode est scindée en 2 étapes :

- la première étape est POS (*Pose from Orthography and Scaling*), qui effectue une approximation de la projection perspective par une *projection orthographique à l'échelle* afin de déterminer la matrice de rotation et le vecteur de translation à partir d'un ensemble d'équations linéaires.

2. Estimation de pose d'un drone basée sur la vision

- La deuxième étape consiste à utiliser de manière itérative le facteur d'échelle pour chaque point afin d'améliorer la projection orthographique trouvée et ensuite d'utiliser POS sur les nouveaux points au lieu des points originaux jusqu'à ce qu'un seuil soit atteint. Ce seuil peut correspondre à un certain nombre d'itérations ou à la moyenne de l'erreur 2D entre deux itérations sur toutes les correspondances 2D.

POSIT estime la profondeur de chaque point 2D, en partant de la projection orthographique pure et en l'améliorant ensuite de façon itérative pour estimer la projection en perspective du point permettant ainsi de déterminer la pose.

L'un des avantages de cette méthode c'est qu'elle est robuste au bruit (sauf cas de bruit très grand) et très rapide. Elle n'est cependant pas adaptée pour des points coplanaires.

2.2.3.5. Vers des calculs plus efficaces du PnP

Lorsque l'accent est mis sur la complexité, les algorithmes PnP non-itératifs ayant une complexité linéaire ont été proposés.

Une première solution $O(n)$ précise pour le PnP était EPnP (pour *Efficient PnP*) (LEPETIT et al. 2009). Elle consiste à exprimer chaque point monde comme une somme pondérée de 4 points virtuels (dits de contrôle). Le problème de pose est alors réduit à l'estimation des coordonnées de ces points de contrôle dans le cadre de la caméra.

Le principal avantage de cette dernière approche est sa complexité réduite, qui est linéaire en le nombre de points. Plusieurs autres variantes ayant une complexité en $O(n)$ dont le principe est similaire à celui de l'EPnP ont été proposées, telles que l'OPnP (pour Optimal solution to PnP) (HAJDER 2017), et l'UPnP (une solution optimale $O(n)$ au problème de pose absolue avec une applicabilité universelle) (KNEIP et al. 2014).

2.2.3.6. Minimisation de la norme de l'erreur de re-projection

D'après MARCHAND et al. 2016, la « solution standard » consiste à estimer les six paramètres $\mathbf{r}_c \in se(3)$ de la transformation ${}^c\mathbf{T}_w$ en minimisant la norme de l'erreur de re-projection tel que présenté à l'équation 2.28).

$$\hat{\mathbf{r}}_c = \arg \min_{\mathbf{r}_c} \sum_{i=1}^N d(\mathbf{x}_i, \mathbf{\Pi} {}^c\mathbf{T}_w(\mathbf{r}_c)^w \mathbf{X}_i)^2 \quad (2.28)$$

Où $d(x, x')$ est la distance euclidienne entre deux points x et x' .

Lorsque le bruit sur les mesures (coordonnées des points $\bar{\mathbf{x}}_i$) est supposé gaussien, cette minimisation fournit l'estimation du maximum de vraisemblance. Une approche de minimisation non-linéaire telle que la méthode Gauss-Newton ou l'algorithme de Levenberg-Marquardt peut être utilisée pour la résolution.

2. Estimation de pose d'un drone basée sur la vision

Résoudre l'équation 2.28, reviendra donc à minimiser une fonction de coût donnée par :

$$E(\mathbf{r}_c) = \|\mathbf{e}(\mathbf{r}_c)\|^2 \quad (2.29)$$

$$= \mathbf{e}(\mathbf{r}_c)^\top \mathbf{e}(\mathbf{r}_c), \text{ avec } \mathbf{e}(\mathbf{r}_c) = \mathbf{x}(\mathbf{r}_c) - \mathbf{x} \quad (2.30)$$

où $\mathbf{x}(\mathbf{r}_c)$ représente les coordonnées métriques issues de la projection des points 3D dans la caméra (défini à l'équation 2.4).

La solution est obtenue en linéarisant l'équation $\mathbf{e}(\mathbf{r}_c) = 0$. Le développement de Taylor au premier ordre de l'erreur de re-projection $\mathbf{e}(\mathbf{r}_c)$ est donné par :

$$\mathbf{e}(\mathbf{r}_c + \delta\mathbf{r}) \approx \mathbf{e}(\mathbf{r}_c) + \mathbf{J}(\mathbf{r}_c)\delta\mathbf{r} \quad (2.31)$$

où $\mathbf{J}(\mathbf{r}_c)$ est la matrice Jacobienne de $\mathbf{e}(\mathbf{r}_c)$ en \mathbf{r}_c , de dimensions $2N \times 6$. Dans le cas où \mathbf{r}_c représente la pose de la caméra par un vecteur de translation et une rotation sous forme vecteur-angle, soit $\mathbf{r}_c = ({}^c\mathbf{t}_w, \theta\mathbf{u})^\top$, l'expression de la matrice jacobienne est la suivante :

$$\mathbf{J} = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -\frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_i y_i & -(1 + y_i^2) & y_i \\ 0 & -\frac{1}{Z_i} & \frac{y_i}{Z_i} & 1 + y_i^2 & -x_i y_i & -x_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \quad (2.32)$$

La méthode de Gauss-Newton consiste à minimiser $E(\mathbf{r}_c + \delta\mathbf{r})$, telle que :

$$E(\mathbf{r}_c + \delta\mathbf{r}) = \|\mathbf{e}(\mathbf{r}_c + \delta\mathbf{r})\|^2 \approx \|\mathbf{e}(\mathbf{r}_c) + \mathbf{J}(\mathbf{r}_c)\delta\mathbf{r}\|^2 \quad (2.33)$$

La minimisation se fait en annulant le gradient de la fonction coût approchée, soit $\mathbf{J}(\mathbf{r}_c)^\top (\mathbf{e}(\mathbf{r}_c) + \mathbf{J}(\mathbf{r}_c)\delta\mathbf{r}) = 0$, on obtient

$$\delta\mathbf{r} = -\mathbf{J}(\mathbf{r}_c)^+ \mathbf{e}(\mathbf{r}_c),$$

avec \mathbf{J}^+ la pseudo-inverse de la matrice Jacobienne. La minimisation se fait itérativement aux moindres carrés. L'algorithme consiste à partir d'une hypothèse initiale de \mathbf{r}_c « assez proche » de la solution afin de converger vers une solution optimale globale. À chaque itération, la valeur de la pose \mathbf{r}_c est mise à jour par l'incrément δr :

$$\mathbf{r}_{k+1} = \mathbf{r}_k \oplus \delta\mathbf{r} \quad (2.34)$$

L'opérateur \oplus correspond à la composition sur $se(3)$. À la convergence de l'algorithme, le gradient de l'erreur de re-projection tend vers zéro.

L'algorithme de Levenberg-Marquardt permet d'améliorer le domaine de convergence de la minimisation non linéaire, en ajoutant un terme d'amortissement $\lambda \geq 0$. Le calcul de l'incrément $\delta\mathbf{r}$ se fait en résolvant

$$\left(\mathbf{J}(\mathbf{r}_c)^\top \mathbf{J}(\mathbf{r}_c) + \lambda \mathbf{I} \right) \delta\mathbf{r} = -\mathbf{J}(\mathbf{r}_c)^\top \mathbf{e}(\mathbf{r}_c)$$

2. Estimation de pose d'un drone basée sur la vision

On réalise ainsi un compromis entre la méthode de Gauss-Newton (quand $\lambda = 0$) et la descente de gradient (λ grand). Le terme λ est adapté au cours de la minimisation : augmenté si nécessaire, de façon à assurer une amélioration de la fonction coût à chaque pas ; diminué si possible, de façon à accélérer la convergence vers l'optimum.

Un des avantages de l'approche par minimisation de l'erreur de reprojection est qu'elle permet d'intégrer facilement les corrélations non-linéaires induites par le problème PnP , et qu'elle offre une solution optimale au problème. Néanmoins, du fait qu'elle nécessite une bonne localisation initiale (démarrer proche de la solution), il est possible de converger vers un minimum local ou même de ne pas converger vers cette solution optimale.

2.3. Estimation de la pose d'un drone basée vision

Dans cette thèse, l'application des méthodes d'estimation que nous ferons sera effectuée à partir de drones à voilures tournantes équipés d'une caméra.

Un drone désigne donc ici un appareil aérien sans pilote et sans humain à bord, capable de s'élever et de se mouvoir en altitude, au sein de l'atmosphère terrestre. On le désigne couramment sous l'acronyme UAV, pour *Unmanned Aerial Vehicle*. Les drones de type VTOL (*Vertical Take-Off and Landing*) sont ceux capables de décoller et d'atterrir verticalement. Ils sont très souvent utilisés dans le cadre civil ou militaire et servent principalement à étendre la vision humaine afin d'assurer l'exploration des zones à risques, comme par exemple, l'inspection d'ouvrages d'arts comme les ponts ou d'accomplir des missions diverses dans des environnements hostiles.

2.3.1. Déterminer la pose du drone à partir de la caméra

Après avoir déterminé la pose de la caméra, déterminer la pose du drone reviendra juste à trouver la matrice de transformation robot-monde grâce à la relation ${}^c\mathbf{T}_w = {}^c\mathbf{T}_r \cdot {}^r\mathbf{T}_w(\mathbf{r})$. En supposant connue la matrice rigide de transformation entre le repère robot et le repère de la caméra ${}^c\mathbf{T}_r$, par calibration, la pose du drone est définie par

$${}^r\hat{\mathbf{T}}_w(\mathbf{r}) = {}^c\mathbf{T}_r^{-1} \cdot {}^c\hat{\mathbf{T}}_w \quad (2.35)$$

Cette équation permet d'avoir la pose du drone \mathbf{r} dans le repère global de référence. Dans les approches présentées au chapitres 4, 5 et 6, les systèmes de calcul de pose définis ont été paramétrés de manière à intégrer les variables représentant la pose du robot directement dans la définition du problème traité.

2.3.2. Assistance des capteurs embarqués

Les deux modèles de drones à voilure tournante que nous utiliserons dans ces travaux, ainsi que l'ensemble des capteurs qui y sont embarqués sont présentés à l'annexe A.

2. Estimation de pose d'un drone basée sur la vision

La centrale de mesure inertielle (IMU pour « *Inertial Measurement Unit* » en anglais), est le capteur embarqué permettant la mesure de la position angulaire (encore appelée attitude : roulis, tangage et lacet), de l'accélération linéaire (selon les trois axes orthogonaux) et de la vitesse angulaire (vitesse de rotation autour des trois axes orthogonaux) du drone. Sous hypothèse que le drone navigue en régime *quasi-stationnaire*, il est possible à partir de l'IMU d'avoir une mesure précise du roulis ϕ , du tangage θ et une mesure plus grossière du lacet ψ .

La hauteur z du drone (par rapport au sol) est également mesurée. Elle résulte de la fusion des données d'un altimètre barométrique et d'un télémètre à ultrasons. Enfin, la vitesse linéaire du drone par rapport au sol est mesurée par l'information de flot optique d'une caméra orientée vers le sol.

La localisation du drone dans son environnement sera donc assistée, rendue facile par la connaissance de ces mesures proprioceptives du drone. Ainsi, pour les méthodes proposées, une analyse des résultats avec et sans assistance de ces capteurs sera faite.

2.4. Conclusion

Dans ce chapitre, nous avons présenté la formulation classique du problème d'estimation de pose à partir d'un modèle 3D connu. Différentes méthodes existantes dans la littérature pour le résoudre ont été présentées. La plupart de ces méthodes reposent sur la minimisation d'une fonction de coût dépendante l'erreur de reprojection des positions des amers 3D géo-référencés dans l'image. Nous avons vu que chacune de ces méthodes fournissent une estimation ponctuelle de la pose. Dans cette thèse, sous l'hypothèse de mesures à erreurs bornées, ces méthodes ne sont plus utilisables car il s'agira de résoudre le problème d'estimation de pose en prenant en compte les incertitudes sur ces mesures. ceci reviendra à calculer un domaine de toutes les poses compatibles avec les mesures à erreurs bornées.

Dans le chapitre 3, nous présentons l'estimation ensembliste par intervalles comme un outil pour l'estimation garantie de pose. Les différentes approches par intervalles pour quantifier l'incertitude sont passées en revue.

3. Estimation ensembliste par intervalles

3.1. Introduction

L'objet de cette thèse est l'estimation de la pose de drones à partir de mesures bruitées et/ou biaisées et d'un modèle d'observation aux paramètres incertains. Il est donc primordial de pouvoir quantifier ou borner l'incertitude sur la pose estimée. Pour ce faire, une solution consiste à rétro-propager l'erreur d'observation à travers le modèle incertain pour obtenir une estimation de l'incertitude sur la position.

Les approches probabilistes sont classiquement utilisés à cette fin. Elles consistent à représenter les quantités incertaines (mesures et paramètres) par des variables aléatoires. Les méthodes basées simulation comme celle de Monte Carlo (BUCHER 1988) ou l'échantillonnage préférentiel (MELCHERS 1989) peuvent traiter une grande classe de problèmes mais s'adaptent difficilement aux problèmes de grande dimension. La méthode dite *First-order Second-moment* (FOSM) est basée sur le développement de Taylor au premier ordre et les deux premiers moments statistiques des grandeurs incertaines. Cette approche n'est adaptée que lorsque la non-linéarité est modérée sur la plage des incertitudes considérées. D'autres méthodes comme le filtre de Kalman « sans parfum » (*Unscented Kalman Filter*, WAN et al. 2000) peuvent être utilisées pour une meilleure gestion de la non-linéarité. L'utilisation de ces approches nécessite généralement des connaissances préalables sur la distribution des erreurs. Une hypothèse courante lors de l'utilisation de FOSM pour la propagation est que le vecteur d'erreur suit une distribution normale multivariée centrée et que la matrice de covariance est connue. En l'absence d'une telle information, l'hypothèse des erreurs indépendantes et identiquement distribuées est souvent prise.

Contrairement à ces méthodes probabilistes, les méthodes non-probabilistes, comme l'analyse par intervalles (MOORE et al. 2009) reposent sur des représentations différentes de l'incertitude. Les méthodes ensemblistes par intervalles supposent que les erreurs sont bornées et que ces bornes sont connues. Les bornes de l'erreur de mesure peuvent ainsi être propagées afin de déterminer l'erreur d'estimation au pire cas. La seule hypothèse faite à propos de la distribution de probabilité de l'erreur de mesure est que son support est borné.

Ce chapitre est consacré à l'approche ensembliste par intervalles. Ce sera l'outil principal de cette thèse, pour la propagation garantie des incertitudes sous hypothèse de mesures à erreurs bornées. La première section rappelle la définition des intervalles, l'extension aux intervalles de l'arithmétique sur les réels et l'extension des intervalles aux vecteurs et aux matrices. L'évaluation des fonctions sur les intervalles est abordée à travers la

3. Estimation ensembliste par intervalles

notion de fonction d'inclusion (voir KIEFFER 1999 ; JAULIN et al. 2001 ; DREVELLE 2011 pour plus de détails). La deuxième section présente l'inversion ensembliste comme problème d'estimation de paramètres à partir de mesures à erreurs bornées, et sa résolution par l'algorithme SIVIA (JAULIN et al. 1993). La troisième section introduit la notion de contracteurs pour la propagation garantie de contraintes à travers une fonction d'observation reliant les variables. Enfin, la dernière section traite de l'estimation en présence de mesures aberrantes grâce à l'inversion ensembliste q -relaxée.

3.2. Calcul et analyse par intervalles

3.2.1. Définition des intervalles

Un *intervalle* $[x]$ est un sous-ensemble connexe et borné de \mathbb{R} (MOORE 1966). L'ensemble des intervalles de \mathbb{R} est noté \mathbb{IR} . L'intervalle $[x]$ représente l'ensemble des réels x compris entre la *borne inférieure* \underline{x} et la *borne supérieure* \bar{x} de $[x]$ tel que :

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

Un intervalle est dit dégénéré ou ponctuel lorsque $\underline{x} = \bar{x}$ ($x = [x, x] = \{x\}$).

Le *centre* d'un intervalle non vide et borné est définie par :

$$mid([x]) = \frac{\underline{x} + \bar{x}}{2}.$$

La *largeur* d'un intervalle non vide $[x]$ est définie par :

$$w([x]) = \bar{x} - \underline{x}.$$

Elle est toujours positive ou nulle et peut être interprétée comme l'incertitude sur la variable représentée par cet intervalle. La figure 3.1 permet d'illustrer ces notions.

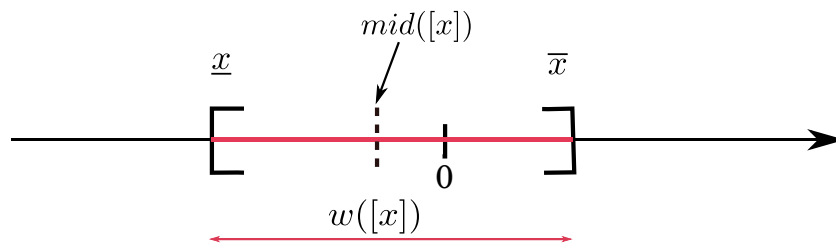


FIGURE 3.1. – Illustration d'un intervalle.

3. Estimation ensembliste par intervalles

Par convention, la largeur d'un intervalle vide est infiniment négative ($w(\emptyset) = -\infty$) et la largeur d'un intervalle dégénéré est nulle ($w(\{x\}) = 0$).

On définit l'opérateur *enveloppe intervalle* $[\cdot]$ qui, à un sous ensemble \mathbb{X} de \mathbb{R} , associe le plus petit intervalle fermé $[\mathbb{X}]$ contenant toutes les valeurs de \mathbb{X} . On a ainsi, pour $\mathbb{X} \subset \mathbb{R}$, $[\mathbb{X}] = [\inf \mathbb{X}, \sup \mathbb{X}]$.

L'*intersection* de deux intervalles $[x]$ et $[y]$ est un intervalle, défini par :

$$[x] \cap [y] = \{z \in \mathbb{R} \mid z \in [x] \wedge z \in [y]\}.$$

L'*union* de deux intervalles, quant-à elle, n'est pas toujours un intervalle. Elle est définie par :

$$[x] \cup [y] = \{z \in \mathbb{R} \mid z \in [x] \vee z \in [y]\}.$$

Ainsi, l'ensemble des intervalles \mathbb{IR} n'est pas stable par l'union ensembliste \cup : par exemple, $[4, 7] \cup [9, 15] \notin \mathbb{IR}$.

Afin que l'ensemble des intervalles soit stable par rapport à une opération d'union, on définit l'opérateur d'*union intervalle* \sqcup . L'union intervalle est définie comme l'enveloppe intervalle de l'union de deux intervalles. Pour deux intervalles $[x]$ et $[y]$, on a

$$[x] \sqcup [y] = [[x] \cup [y]].$$

3.2.2. Arithmétique des intervalles

Les opérations élémentaires binaires sur les réels, notées $\diamond = \{+, -, *, /\}$, peuvent être étendues aux intervalles.

$$[x] \diamond [y] = [\{x \diamond y \in \mathbb{R} \mid x \in [x], y \in [y]\}] \quad (3.1)$$

Les calculs se font comme suit :

- l'addition : $[x] + [y] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$
- la soustraction : $[x] - [y] = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]$
- la multiplication : $[x] * [y] = [\min\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}]$
- la division $[x]/[y] = [x] * (1/[y])$

Le calcul de la division est effectué à partir du calcul de l'inverse donné par

$$1/[y] = \begin{cases} \emptyset & \text{si } [y] = [0, 0], \\ [1/\bar{y}, 1/\underline{y}] & \text{si } 0 \notin [y], \\ [1/\bar{y}, \infty[& \text{si } \underline{y} = 0 \text{ et } \bar{y} > 0, \\] - \infty, 1/\underline{y}] & \text{si } \underline{y} < 0 \text{ et } \bar{y} = 0, \\] - \infty, \infty[& \text{si } \underline{y} < 0 \text{ et } \bar{y} > 0. \end{cases}$$

Tout comme les opérations élémentaires, les fonctions élémentaires telles que sin, cos, exp, tan, etc s'étendent également aux intervalles de réels. Soit f une fonction définie de \mathbb{R} vers \mathbb{R} , son équivalent intervalles $[f]$ satisfait :

$$[f]([x]) = [\{f(x) \mid x \in [x]\}]. \quad (3.2)$$

3. Estimation ensembliste par intervalles

Si f est continue, le calcul de $[f]([x])$ revient à calculer l'ensemble image de $[x]$ par f (c-à-d $f([x])$); ce qui équivaut à un calcul réel sur les bornes de l'intervalle.

Exemple 3.1. *Extension d'une fonction usuelle aux intervalles*

$$\begin{aligned} [\exp([x])] &= [\exp(\underline{x}), \exp(\bar{x})] \\ [\exp([0, 2])] &= [\exp(0), \exp(2)] = [1, e^2] \end{aligned}$$

Pour des intervalles fermés $[x], [y] \in \mathbb{IR}$, le résultat des opérations binaires entre $[x]$ et $[y]$ est obtenu en manipulant leurs bornes (JAULIN et al. 2001). Si $[x]$ et $[y]$ sont non-vides, leur union intervalle satisfait

$$\forall [x] \in \mathbb{IR}, \forall [y] \in \mathbb{IR}, [x] \sqcup [y] = [\min\{\underline{x}, \underline{y}\}, \max\{\bar{x}, \bar{y}\}] \quad (3.3)$$

et l'intersection satisfait

$$\forall [x] \in \mathbb{IR}, \forall [y] \in \mathbb{IR}, [x] \cap [y] = \begin{cases} [\max\{\underline{x}, \underline{y}\}, \min\{\bar{x}, \bar{y}\}] & \text{si } \max\{\underline{x}, \underline{y}\} \leq \min\{\bar{x}, \bar{y}\} \\ \emptyset & \text{sinon.} \end{cases} \quad (3.4)$$

3.2.3. Vecteurs et Matrices d'intervalles

Un *vecteur d'intervalles* encore appelé *pavé* ou *boîte* est un sous-ensemble de \mathbb{R}^n défini comme le produit cartésien de n intervalles de réels :

$$[\mathbf{x}] = [x_1] \times [x_2] \times [x_3] \times \dots \times [x_n] = [\underline{x}_1, \bar{x}_1] \times [\underline{x}_2, \bar{x}_2] \times [\underline{x}_3, \bar{x}_3] \times \dots \times [\underline{x}_n, \bar{x}_n].$$

L'ensemble des boîtes de n -dimensions est appelé \mathbb{IR}^n . Les boîtes non vides sont des hyper-rectangles n -dimensionnels alignés sur les axes. Par exemple, pour $n = 2$, nous avons $[\mathbf{x}] = [x_1] \times [x_2]$ tel que l'illustre la figure 3.2.

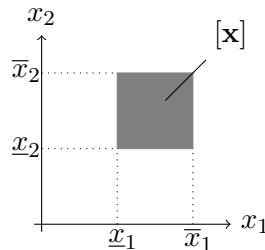


FIGURE 3.2. – Illustration d'une boîte $[\mathbf{x}]$ de \mathbb{IR}^2 .

Les notions de base sur les intervalles de réels ont une extension directe pour les vecteurs d'intervalles. Par exemple :

3. Estimation ensembliste par intervalles

- $[x_i]$ est la $i^{\text{ème}}$ composante de $[\mathbf{x}]$, c'est aussi la projection de $[\mathbf{x}]$ sur le $i^{\text{ème}}$ axe
- une *boîte ponctuelle* est une boîte dont toutes les composantes sont ponctuelles
- les bornes inférieure $\underline{\mathbf{x}}$ et supérieure $\overline{\mathbf{x}}$ d'une boîte $[\mathbf{x}]$ sont définies par : $\underline{\mathbf{x}} = (\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n)^\top$ et $\overline{\mathbf{x}} = (\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)^\top$
- la largeur $width([\mathbf{x}]) = w([\mathbf{x}]) = \max_{1 \leq i \leq n} (w([x_i]))$
- le centre $mid([\mathbf{x}]) = (mid(x_1), mid(x_2), \dots, mid(x_n))^\top$

Le vecteur d'intervalles $[\mathbf{x}]$ est *vide* si au moins une de ses composantes est vide. Par exemple, le vecteur $[\mathbf{x}] = ([2, 5]; [0, 1]; \emptyset)$ est vide étant donné que son 3^{ème} axe est vide.

Les opérations élémentaires sur les intervalles de réels peuvent également s'étendre aux vecteurs d'intervalles. Ceci revient à opérer les calculs sur chaque composante du vecteur. Les intervalles de réels peuvent également s'étendre aux *matrices d'intervalles*. Une matrice d'intervalles de dimensions $m \times n$ est un sous ensemble de $\mathbb{IR}^{m \times n}$ constitué de $m \times n$ intervalles fermés et les opérations élémentaires sur les matrices peuvent également être étendues.

3.2.4. Notion de pessimisme

Il peut arriver que le résultat de l'application de plusieurs opérations sur les intervalles produise un résultat qui n'est pas minimal. Dans ces cas, le résultat obtenu est dit *pessimiste*. Le pessimisme est principalement dû à la *dépendance* entre les variables d'une équation et à l'*effet d'enveloppe*.

Soit l'expression $z = x - x$ et l'intervalle $[x] = [-1, 0]$. L'ensemble des valeurs de z , pour $x \in [x]$ peut s'écrire de la façon suivante

$$\{x - x \mid x \in [x]\} = \{0\}, \quad (3.5)$$

soit l'intervalle dégénéré $\{0\}$.

Appliquons maintenant l'arithmétique des intervalles sur l'expression de z pour déterminer un intervalle $[z]$:

$$[z] = [x] - [x] = [-1, 0] - [-1, 0] = [-1, 1] \neq \{0\} \quad (3.6)$$

Dans le calcul par intervalles, chaque intervalle est considéré comme une variable différente, conduisant ainsi à perdre la dépendance entre les deux occurrences de la même variable intervalle. L'expression ensembliste de $[z]$ est ainsi

$$[x] - [x] = \{x - y \mid x \in [x], y \in [x]\} \quad (3.7)$$

Le résultat évalué à l'équation 3.5 est en général contenu dans celui de l'expression ensembliste de l'équation 3.7. Ce problème est connu sous le nom du *problème de dépendance* qui est une source de pessimisme en cas de présence multiples d'une même variable dans une expression.

3. Estimation ensembliste par intervalles

La représentation d'un ensemble quelconque par une boîte peut être source de pessimisme. Prenons par exemple la ligne courbe de la figure 3.3, sa représentation par une boîte se fait en l'enveloppant par le rectangle rouge dont les côtés sont parallèles aux axes du repère. Nous observons bien que la taille du rectangle est de loin plus grande que celle de l'ensemble représenté. De plus si l'on applique une transformation consistant à effectuer une rotation de l'enveloppe de cette courbe, nous observons que la boîte enveloppe résultante est plus grande que celle initiale (rectangle *en bleu* sur la figure 3.3) ; ceci illustre bien le fait que l'effet d'enveloppe est une source de pessimisme.

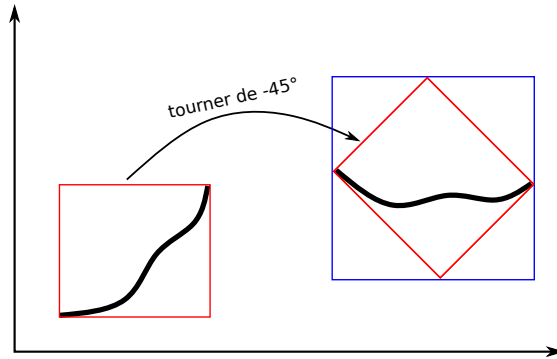


FIGURE 3.3. – Illustration de la notion de l'effet d'enveloppe.

3.2.5. Fonctions d'inclusion

L'image d'une boîte par une fonction n'est pas nécessairement une boîte. Ainsi, il est nécessaire de trouver des fonctions intervalles dites *fonctions d'inclusion*, qui permettent d'englober dans une boîte l'image d'une boîte donnée.

Définition Soit une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$, la fonction intervalle $[f] : \mathbb{IR} \rightarrow \mathbb{IR}$ est une fonction d'inclusion pour f si et seulement si

$$\forall [x] \in \mathbb{IR}, f([x]) \subset [f]([x]), \quad (3.8)$$

où $f([x]) = \{f(x), x \in [x]\}$ est l'image par f de l'intervalle $[x]$. L'intervalle $[f]([x])$ est garanti de contenir l'image de $[x]$ par f quelque soit la forme de $f([x])$.

3.2.5.1. Fonction d'inclusion vectorielle

Une fonction vectorielle à valeurs réelles dans \mathbb{R}^m est une fonction \mathbf{f} qui à tout vecteur \mathbf{x} de \mathbb{R}^n associe un vecteur de taille m dont chacune des composantes est une fonction scalaire définie comme suit :

$$\begin{aligned} \mathbf{f} : \mathbb{R}^n &\longrightarrow \mathbb{R}^m \\ \mathbf{x} &\longmapsto \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^\top \end{aligned} \quad (3.9)$$

3. Estimation ensembliste par intervalles

La fonction d'inclusion $[\mathbf{f}]([\mathbf{x}])$ retourne une boîte contenant l'image de la boîte $[\mathbf{x}]$ par la fonction vectorielle \mathbf{f} . $[\mathbf{f}]$ est définie de $\mathbb{I}\mathbb{R}^n$ vers $\mathbb{I}\mathbb{R}^m$ telle que $\forall [\mathbf{x}] \in \mathbb{I}\mathbb{R}^n, \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}])$.

Propriétés La figure 3.4 montre que $[\mathbf{f}]([\mathbf{x}])$ peut donner une évaluation très pessimiste de $\mathbf{f}([\mathbf{x}])$. Ainsi il existe plusieurs propriétés spécifiques permettant de caractériser une fonction d'inclusion.

Une fonction d'inclusion $[\mathbf{f}]$ est dite *fine* si pour tout intervalle vectoriel ponctuel $[\mathbf{x}] = \mathbf{x}$, son image par $[\mathbf{f}]$ est ponctuelle

$$[\mathbf{f}](\mathbf{x}) = \mathbf{f}(\mathbf{x}).$$

$[\mathbf{f}]$ est *convergente* si pour toute suite de boîtes $[\mathbf{x}](k)$,

$$\lim_{k \rightarrow \infty} w([\mathbf{x}](k)) = 0 \implies \lim_{k \rightarrow \infty} w([\mathbf{f}]([\mathbf{x}](k))) = 0. \quad (3.10)$$

Une fonction d'inclusion convergente est nécessairement fine. Cette propriété est requise pour la convergence des algorithmes utilisant l'analyse par intervalles.

La fonction d'inclusion $[\mathbf{f}]$ est *monotone pour l'inclusion* si

$$[\mathbf{x}] \subset [\mathbf{y}] \implies [\mathbf{f}]([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{y}]). \quad (3.11)$$

Une fonction d'inclusion $[\mathbf{f}]$ est *minimale* si $\forall [\mathbf{x}], [\mathbf{f}]([\mathbf{x}])$ est la plus petite boîte contenant $\mathbf{f}([\mathbf{x}])$. Elle est unique et notée $[\mathbf{f}]^*$.

$$[\mathbf{f}]^*([\mathbf{x}]) = [\{\mathbf{f}(\mathbf{x}), \mathbf{x} \in [\mathbf{x}]\}].$$

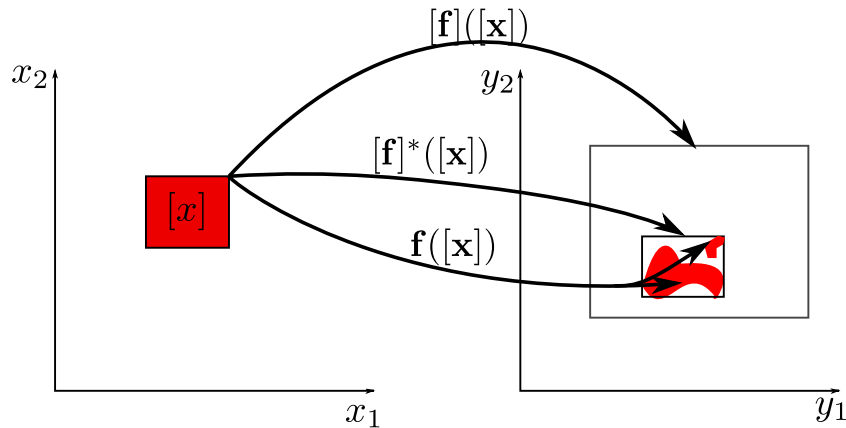


FIGURE 3.4. – Illustration d'une fonction d'inclusion. Image d'une boîte par une fonction $\mathbf{f}([\mathbf{x}])$, et deux de ses fonctions d'inclusion $[\mathbf{f}]$ et $[\mathbf{f}]^*$ (qui est minimale).

Il existe donc plusieurs fonctions d'inclusions $[\mathbf{f}]$ pour une fonction vectorielle \mathbf{f} , tel que présenté à la figure 3.4.

3. Estimation ensembliste par intervalles

Fonction d'inclusion triviale Une manière de définir une fonction d'inclusion pour \mathbf{f} est de définir la fonction d'inclusion dite *triviale* donnée par :

$$[\mathbf{f}]([\mathbf{x}]) = \begin{cases} \mathbb{R} & \text{si } w([\mathbf{x}]) > 0 \\ \mathbf{f}(\mathbf{x}) & \text{si } w([\mathbf{x}]) = 0 \end{cases} \quad (3.12)$$

Cette fonction très pessimiste n'est d'aucune utilité en pratique.

Fonction d'inclusion naturelle La fonction d'inclusion naturelle d'une fonction vectorielle $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, est obtenue en remplaçant les variables réelles x_i de \mathbf{f} par leurs variables intervalles correspondantes $[x_i]$ et en remplaçant chaque opération élémentaire de \mathbf{f} par son extension aux intervalles. La fonction d'inclusion naturelle est minimale si \mathbf{f} est continue et chaque variable x_i n'apparaît qu'une seule fois dans l'expression de \mathbf{f} (voir JAULIN 2001).

Exemple 3.2. : Soit la fonction réelle $f : \mathbb{R} \rightarrow \mathbb{R}$, indifféremment définie par les 4 expressions suivantes :

$$\begin{aligned} f_1(x) &= x^2 - 2x \\ f_2(x) &= (x - 1)^2 - 1 \\ f_3(x) &= x(x - 2) \\ f_4(x) &= x.x - 2x \end{aligned}$$

L'évaluation de leurs fonctions d'inclusion naturelles obtenues pour chaque expression de f donne, pour $[x] = [0, 1]$:

$$\begin{aligned} [f_1]([x]) &= [x]^2 - 2[x] = [-2, 1] \\ [f_2]([x]) &= ([x] - 1)^2 - 1 = [-1, 0] \\ [f_3]([x]) &= [x]([x] - 2) = [-2, 0] \\ [f_4]([x]) &= [x].[x] - 2[x] = [-2, 1]. \end{aligned}$$

Nous remarquons que la largeur des intervalles obtenus est dépendante de l'expression utilisée. Ainsi, bien-qu'étant toutes équivalentes dans l'arithmétique des réels, leurs fonctions d'inclusions naturelles ne le sont pas. La variable x n'apparaît qu'une seule fois dans l'expression de f_2 , ceci rend sa fonction d'inclusion $[f_2]$ minimale.

3.2.5.2. Fonction d'inclusion centrée

Soient $f : \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction scalaire différentiable sur une boîte $[\mathbf{x}]$ et $\mathbf{m} = \text{mid}([\mathbf{x}])$. Le théorème des accroissements finis conduit à écrire

$$\forall \mathbf{x} \in [\mathbf{x}], \exists \mathbf{z} \in [\mathbf{x}] \mid f(\mathbf{x}) = f(\mathbf{m}) + \mathbf{g}^\top(\mathbf{z})(\mathbf{x} - \mathbf{m}) \quad (3.13)$$

3. Estimation ensembliste par intervalles

où \mathbf{g} est le gradient de la fonction f . Si $[\mathbf{g}^\top]$ est une fonction d'inclusion de \mathbf{g}^\top alors nous avons

$$\forall \mathbf{x} \in [\mathbf{x}] \mid f(\mathbf{x}) \in f(\mathbf{m}) + [\mathbf{g}^\top]([\mathbf{x}])(\mathbf{x} - \mathbf{m}) \quad (3.14)$$

$$\iff f([\mathbf{x}]) \subseteq f(\mathbf{m}) + [\mathbf{g}^\top]([\mathbf{x}])([\mathbf{x}] - \mathbf{m}) \quad (3.15)$$

Il existe de ce fait une fonction intervalle $[f_C]$, appelée *fonction d'inclusion centrée* définie par

$$[f_C]([\mathbf{x}]) \triangleq f(\mathbf{m}) + [\mathbf{g}^\top]([\mathbf{x}])([\mathbf{x}] - \mathbf{m}) \quad (3.16)$$

Cette fonction permet de réduire considérablement le pessimisme par rapport à la fonction d'inclusion naturelle lorsque la largeur de la boîte $[\mathbf{x}]$ tend vers zéro.

3.2.5.3. Fonction d'inclusion de Taylor

Similairement à la forme centrée, il est possible de définir une *fonction d'inclusion de Taylor* qui utilise le développement en série de Taylor pour approcher f . Notons $[f_T]$ cette fonction d'inclusion, en considérant une expansion d'ordre deux, elle est définie par :

$$[f_T]([\mathbf{x}]) = f(\mathbf{m}) + \mathbf{g}^\top(\mathbf{m})([\mathbf{x}] - \mathbf{m}) + \frac{1}{2}([\mathbf{x}] - \mathbf{m})^\top [\mathbf{H}]([\mathbf{x}])([\mathbf{x}] - \mathbf{m}) \quad (3.17)$$

Où $[\mathbf{H}]([\mathbf{x}])$ est une fonction d'inclusion de la *matrice Hessienne intervalle* de f . La convergence de cette fonction est quadratique ; ce qui rend cette forme de fonction d'inclusion plus adaptée que les autres lorsque la largeur des boîtes est très grande.

Il existe des formes mixtes des fonctions d'inclusion détaillées dans JAULIN et al. 2001.

3.2.6. Sous-pavage

L'arithmétique des intervalles et les fonctions d'inclusion rendent facile la manipulation des intervalles et des boîtes. Toutefois, les solutions d'un problème ensembliste sont généralement des ensembles de formes quelconques. Les intervalles et les boîtes ne sont pas toujours appropriées pour donner une bonne approximation de ces ensembles solutions, surtout lorsqu'ils ne sont pas connexes.

Un *sous-pavage* est une union de boîtes non chevauchantes et de largeurs non-nulles. Il permet une représentation plus fine d'un ensemble, tout en permettant d'appliquer l'arithmétique des intervalles décrite précédemment. Un ensemble \mathbb{S} peut ainsi être encadré par deux sous-pavages $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$ tels que :

$$\underline{\mathbb{S}} \subset \mathbb{S} \subset \overline{\mathbb{S}} \quad (3.18)$$

3. Estimation ensembliste par intervalles

L'approximation intérieure $\underline{\mathbb{S}}$ est contenue dans \mathbb{S} , tandis que \mathbb{S} est inclus dans l'approximation extérieure $\overline{\mathbb{S}}$. Une forme classique est $\overline{\mathbb{S}} = \underline{\mathbb{S}} \cup \Delta\mathbb{S}$, avec $\Delta\mathbb{S}$ le sous-pavage contenant la frontière de \mathbb{S} .

La figure 3.5 présente un exemple d'approximation de l'ensemble d'un ensemble quelconque \mathbb{S} .

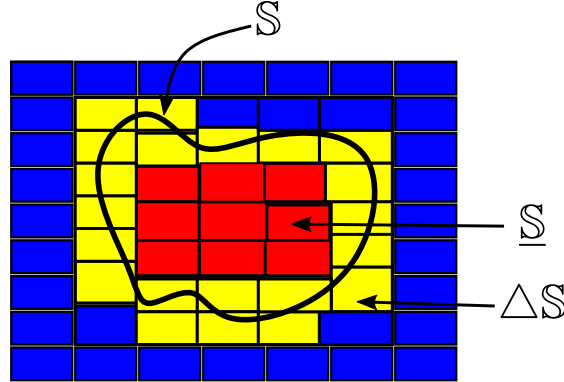


FIGURE 3.5. – Illustration de l'encadrement de l'ensemble \mathbb{S} . *Rouge* : sous-pavage intérieur $\underline{\mathbb{S}}$, *bleu* : boîtes n'appartenant pas à \mathbb{S} , *jaune* : frontière $\Delta\mathbb{S}$.

Ces deux sous-pavages permettent de donner des informations sur \mathbb{S} :

- si $\overline{\mathbb{S}} = \emptyset$ alors $\mathbb{S} = \emptyset$;
- si $\underline{\mathbb{S}} \neq \emptyset$ alors \mathbb{S} n'est pas vide ;
- $\Delta\mathbb{S}$ contient l'ensemble des boîtes pour lesquelles aucune conclusion n'a pu être établie, il représente l'incertitude sur l'approximation de \mathbb{S} ou encore la précision de l'approximation.

3.3. Inversion ensembliste par intervalles

L'algorithme SIVIA, pour *Set Inversion Via Interval Analysis* (JAULIN et al. 1993 ; JAULIN et al. 2001), est un algorithme récursif de type « *branch and bound* », utilisé pour résoudre un problème d'inversion ensembliste par le calcul d'une approximation de l'ensemble solution \mathbb{S} de ce problème en l'encadrant par deux sous-pavages $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$.

3.3.1. Définition d'un problème d'inversion ensembliste

Considérons une fonction continue et différentiable $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Nous cherchons à déterminer l'ensemble solution \mathbb{S} contenant toutes les valeurs de $\mathbf{x} \in \mathbb{R}^n$ tel que $\mathbf{f}(\mathbf{x})$ est contenue dans l'ensemble \mathbb{Y} (qui est un sous ensemble fermé de \mathbb{R}^m). Résoudre ce *problème d'inversion ensembliste* revient à caractériser l'*image réciproque* $\mathbf{f}^{-1}(\mathbb{Y})$ de \mathbb{Y} par \mathbf{f} :

$$\mathbb{S} = \mathbf{f}^{-1}(\mathbb{Y}) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} \quad (3.19)$$

3. Estimation ensembliste par intervalles

La caractérisation exacte de \mathbb{S} n'est pas toujours possible. Toutefois, l'algorithme SIVIA est utilisé pour calculer une approximation intérieure et extérieure de \mathbb{S} . Dans cet algorithme, pour déterminer l'appartenance ou non d'une boîte $[\mathbf{x}]$ à l'ensemble solution \mathbb{S} , il est nécessaire d'utiliser un test d'inclusion.

3.3.2. Test d'inclusion

Les tests d'inclusions permettent de montrer que tous les points d'une boîte donnée satisfont une propriété donnée, ou qu'aucun de ces points ne la satisfait (JAULIN et al. 2001). Pour définir les tests d'inclusions, les intervalles de booléens sont utilisés.

3.3.2.1. Intervalle booléen

Un *intervalle booléen* est un sous-ensemble de l'ensemble des booléens $\mathbb{B} = \{false, true\} = \{0, 1\}$. L'ensemble de tous les intervalles de booléens est $\mathbb{IB} = \{\emptyset, [0, 0], [1, 1], [0, 1]\}$ avec,

- \emptyset pour *impossible*
- $[0, 0]$ (encore noté 0) pour *faux*
- $[1, 1]$ (encore noté 1) pour *vrai*
- $[0, 1]$ pour *indéterminé*.

Les opérateurs logiques ET (\wedge) et OU (\vee) peuvent leur être appliqués. Le tableau 3.1 présente les tables de vérité de ces deux opérations logiques

\wedge	1	0	$[0, 1]$
1	1	0	$[0, 1]$
0	0	0	0
$[0, 1]$	$[0, 1]$	0	$[0, 1]$

\vee	1	0	$[0, 1]$
1	1	1	1
0	1	0	$[0, 1]$
$[0, 1]$	1	$[0, 1]$	$[0, 1]$

TABLE 3.1. – Opérations logiques entre intervalles booléens.

Étant donné que les intervalles booléens sont des ensembles, les opérations d'union et d'intersection sur les ensembles peuvent également leur être appliquées.

3.3.2.2. Test d'inclusion

Un *test* est une fonction booléenne $t : \mathbb{R}^n \rightarrow \mathbb{B}$. Un *test d'inclusion* pour un test t est une fonction $[t] : \mathbb{IR}^n \rightarrow \mathbb{IB}$ telle que $\forall [\mathbf{x}] \in \mathbb{IR}^n$,

$$[t]([\mathbf{x}]) = \begin{cases} 1 & \text{si } \forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 1 \\ 0 & \text{si } \forall \mathbf{x} \in [\mathbf{x}], t(\mathbf{x}) = 0 \\ [0, 1] & \text{sinon} \end{cases} \quad (3.20)$$

Un test d'inclusion $[t]$ est dit *fin* si

$$\forall \mathbf{x} \in \mathbb{IR}^n, [t](\{\mathbf{x}\}) = t(\mathbf{x}),$$

3. Estimation ensembliste par intervalles

et *minimal* si

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, [t]([\mathbf{x}]) = \{t(\mathbf{x}), \mathbf{x} \in [\mathbf{x}]\}. \quad (3.21)$$

Si un test d'inclusion $[t]$ est minimal, alors il est forcément fin.

Les tests d'inclusions sont utilisés par la suite, dans l'algorithme SIVIA pour déterminer si un vecteur \mathbf{x} auquel on applique une fonction d'inclusion donnée satisfait ou non la contrainte d'appartenance à l'ensemble image \mathbb{Y} d'un problème d'inversion ensembliste donné.

3.3.2.3. Test d'inclusion et fonction d'inclusion

Soit le test t vérifiant l'appartenance de l'image d'un vecteur \mathbf{x} par une fonction f à un domaine \mathbb{Y} , tel que

$$t(f(\mathbf{x}), \mathbb{Y}) = \begin{cases} 1 & \text{si } (f(\mathbf{x})) \in \mathbb{Y} \\ 0 & \text{sinon} \end{cases} \quad (3.22)$$

A partir de la notion de fonction d'inclusion, il est facile de construire un test d'inclusion pour le test t . Ce test d'inclusion $[t]$ est exprimé à partir de la fonction d'inclusion de $[f]$ de la fonction f . Il est défini par

$$[t]([f]([\mathbf{x}]), \mathbb{Y}) = \begin{cases} 1 & \text{si } [f]([\mathbf{x}]) \subset \mathbb{Y} \\ 0 & \text{si } [f]([\mathbf{x}]) \cap \mathbb{Y} = \emptyset \\ [0, 1] & \text{sinon} \end{cases}$$

3.3.3. Algorithme SIVIA

Considérons une boîte initiale $[\mathbf{x}]$ dans laquelle la solution du problème d'inversion ensembliste $\mathbf{f}^{-1}(\mathbb{Y})$ (équation 3.19) est garantie de se trouver. L'algorithme SIVIA, de type *branch and bound*, consiste à l'appel récursif et successif d'un test d'inclusion à l'aide de la fonction d'inclusion $[f]$ et de bisections de l'espace des solutions $[\mathbf{x}]$. On obtient en sortie un encadrement de l'ensemble solution entre deux sous-pavages $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$ tels que

$$\underline{\mathbb{S}} \subset \mathbf{f}^{-1}(\mathbb{Y}) \cap [\mathbf{x}] \subset \overline{\mathbb{S}} \quad (3.23)$$

1. Si $[t]([f]([\mathbf{x}]), \mathbb{Y}) = 0$, alors $[f]([\mathbf{x}])$ est disjointe de \mathbb{Y} c'est-à-dire que $[\mathbf{x}]$ n'appartient pas à l'ensemble solution \mathbb{S} , la boîte $[\mathbf{x}]$ est donc mise de côté (figure 3.6a)
2. Si $[t]([f]([\mathbf{x}]), \mathbb{Y}) = 1$, alors $[f]([\mathbf{x}])$ est dans \mathbb{Y} c-à-d que $[\mathbf{x}]$ appartient à \mathbb{S} , la boîte $[\mathbf{x}]$ est ajoutée aux sous-pavages $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$ (figure 3.6b)
3. Dans les autres cas $[t]([f]([\mathbf{x}]), \mathbb{Y}) = [0, 1]$, alors $[\mathbf{x}]$ est indéterminé. (figure 3.6c)
 - a) Si la largeur $w([\mathbf{x}])$ est inférieure à un réel ϵ , alors $[\mathbf{x}]$ est trop petit pour être bisecté et est ajouté au sous-pavage extérieur $\overline{\mathbb{S}}$ ($[\mathbf{x}]$ appartient à la frontière $\Delta\mathbb{S}$)

3. Estimation ensembliste par intervalles

- b) si $w([\mathbf{x}]) > \epsilon$, $[\mathbf{x}]$ est bissecté en $[\mathbf{x}_1]$ et $[\mathbf{x}_2]$ et l'algorithme SIVIA est appliqué récursivement aux deux boîtes.

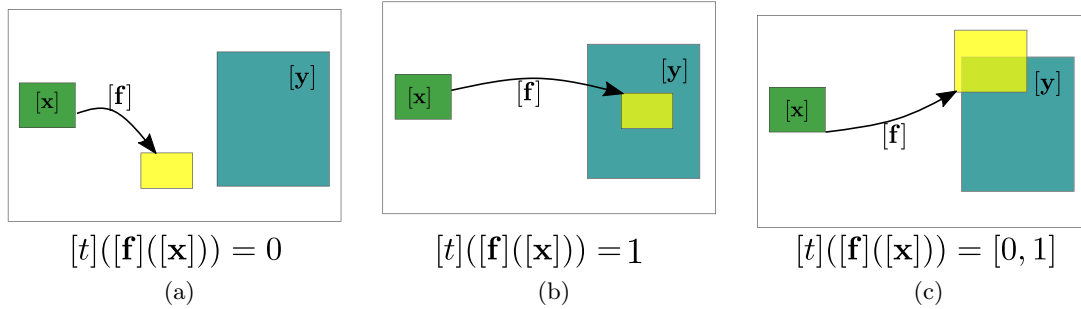


FIGURE 3.6. – Illustration d'un test d'inclusion

L'algorithme 3.10 illustre les étapes de l'algorithme SIVIA que nous venons d'expliquer. Le critère d'arrêt de cet algorithme est dépendant de la taille des boîtes ϵ . Les sous-pavages $\underline{\mathbb{S}}$ et $\overline{\mathbb{S}}$ sont initialisés à l'ensemble vide.

```

1 : if  $[t]([\mathbf{f}]([\mathbf{x}]), \mathbb{Y}) = 0$  then return ;
2 : if  $[t]([\mathbf{f}]([\mathbf{x}]), \mathbb{Y}) = 1$  then
3 :    $\{\underline{\mathbb{S}} := \underline{\mathbb{S}} \cup [\mathbf{x}]; \overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [\mathbf{x}];$  return ;}
4 : if  $w([\mathbf{x}]) < \epsilon$  then
5 :    $\{\overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [\mathbf{x}];$  return ;}
6 : bissecter  $[\mathbf{x}]$  en  $[\mathbf{x}_1]$  et  $[\mathbf{x}_2]$  ;
7 : SIVIA( $[\mathbf{x}_1]$ ,  $[\mathbf{f}]$ ,  $\mathbb{Y}$ ,  $\epsilon$ ,  $\underline{\mathbb{S}}$ ,  $\overline{\mathbb{S}}$ )
8 : SIVIA( $[\mathbf{x}_2]$ ,  $[\mathbf{f}]$ ,  $\mathbb{Y}$ ,  $\epsilon$ ,  $\underline{\mathbb{S}}$ ,  $\overline{\mathbb{S}}$ )

```

Algorithme 1 : SIVIA(*in* : $[\mathbf{x}]$, $[\mathbf{f}]$, \mathbb{Y} , ϵ ; *inout* : $\underline{\mathbb{S}}$, $\overline{\mathbb{S}}$)

L'algorithme 1 explore en profondeur l'espace des solutions. Il est également possible d'implémenter un parcours en largeur, ce qui permet d'utiliser le temps d'exécution ou le nombre de bisections comme critère d'arrêt supplémentaire.

Exemple 3.3. Localisation planaire d'un robot à partir de mesures de distances

Considérons un robot, de position $\mathbf{r} = (x_r, y_r)^\top$ inconnue, qui se déplace dans le plan et est capable de mesurer des distances (d_1 , d_2 et d_3) par rapport à 3 balises (B_1 , B_2 et B_3) dont les positions sont connues. Le modèle d'observation est décrit par :

$$\sqrt{(x_r - x_i)^2 + (y_r - y_i)^2} = d_i, i = 1 \dots 3 \quad (3.24)$$

3. Estimation ensembliste par intervalles

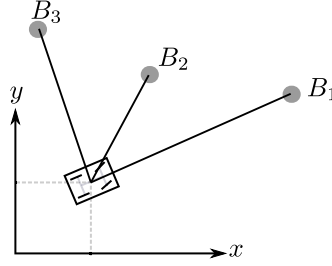


FIGURE 3.7. – Situation de localisation planaire

où $(x_i, y_i)^\top$ est la position de la balise B_i . Ce qui correspond à la fonction d'observation

$$\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} \sqrt{(x_r - x_1)^2 + (y_r - y_1)^2} \\ \sqrt{(x_r - x_2)^2 + (y_r - y_2)^2} \\ \sqrt{(x_r - x_3)^2 + (y_r - y_3)^2} \end{pmatrix}$$

Considérons que l'erreur de mesure des distances est bornée. On a pour chaque distance mesurée $d_i \in d_i^{meas} + [e_d]$, $i = 1 \dots 3$ et $[e_d] = 1$. La figure 3.8 présente le résultat de l'algorithme SIVIA appliqué au problème d'inversion ensembliste consistant à déterminer toutes les positions (x, y) compatibles avec les mesures de distances incertaines, pour différentes valeurs du paramètre ϵ .

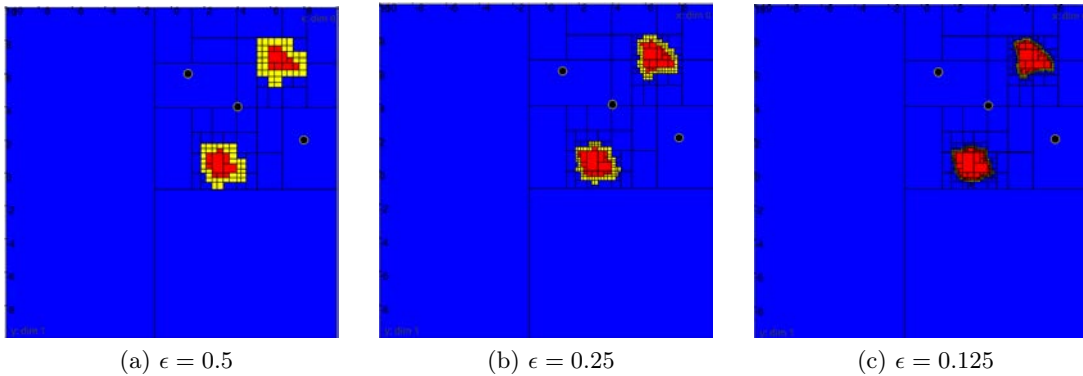


FIGURE 3.8. – Sous-pavages résultats de l'inversion ensembliste pour une exemple de localisation avec 3 balises

Les pavés rouges représentent le sous pavage intérieur \mathbb{S} faisant partie de la solution, les pavés bleus représentent la partie non consistante avec les mesures et la partie en jaune, le sous pavage frontière (indéterminé) $\Delta\mathbb{S}$. Comme les trois balises sont presque alignées, deux hypothèses de localisation sont possibles ; ceci correspond aux 2 composantes connexes de l'ensemble solution.

3. Estimation ensembliste par intervalles

Lorsque la dimension de l'espace des paramètres est grande, le nombre de bisections requises peut rendre SIVIA trop coûteux en temps de calcul pour une application pratique. En effet, comme démontré dans JAULIN et al. 2001, le nombre de bisections avant terminaison de l'algorithme est borné par

$$\left(\frac{w([x](0))}{\epsilon} + 1 \right)^n .$$

Le temps de calcul de SIVIA augmente exponentiellement avec le nombre de dimensions de bisection n de x .

3.4. Contracteurs

Dans cette section, nous présentons ce qu'est un *problème de satisfaction de contraintes* (en anglais *Constraint Satisfaction Problem*, noté CSP), ensuite comment le domaine des variables est réduit grâce aux opérateurs de réduction nommés *contracteurs*, et enfin certains exemples de contracteurs couramment utilisés.

3.4.1. Problème de satisfaction de contraintes

Lorsque les composantes d'un vecteur \mathbf{x} sont reliés par des *relations* ou *contraintes*, nous pouvons définir un problème de satisfaction de contraintes. Un CSP consiste à déterminer l'ensemble solution $\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}], \mathbf{g}(\mathbf{x}) = 0\}$, où $[\mathbf{x}] = [x_1] \times \dots \times [x_m]$ est le domaine à priori (ou espace de recherche) des variables et $\mathbf{g}(\mathbf{x}) = 0$ la fonction vectorielle représentant les contraintes que doivent satisfaire les éléments de l'ensemble solution \mathbb{S} .

Le problème de localisation à 3 balises décrit dans la section précédente peut ainsi s'écrire comme le CSP suivant :

$$\left\{ \begin{array}{ll} x_r \in [x_r], y_r \in [y_r] & \text{variables et domaines de la position du robot} \\ d_1 \in [d_1^{meas}], d_2 \in [d_2^{meas}], d_3 \in [d_3^{meas}] & \text{variables et domaines des mesures} \\ d_1 = \sqrt{(x_r - x_1)^2 + (y_r - y_1)^2} \\ d_2 = \sqrt{(x_r - x_2)^2 + (y_r - y_2)^2} & \text{contraintes} \\ d_3 = \sqrt{(x_r - x_3)^2 + (y_r - y_3)^2} \end{array} \right.$$

Le formalisme $\mathbf{g}(\mathbf{x}) = 0$ est général. En effet, dans les cas où les contraintes d'un CSP sont des inégalités, ces dernières peuvent être transformées en égalités en introduisant des variables positives nommées *variables d'écart*.

Exemple 3.4. *Transformation d'une inégalité en égalité par l'introduction d'une variable d'écart .*

Considérons le système d'équation et inéquation suivant

$$\left\{ \begin{array}{l} x_1 + 2x_2 = 0 \\ x_2 - x_3 \geq 0 \end{array} \right. \quad (3.25)$$

3. Estimation ensembliste par intervalles

Si l'on y introduit une variable d'écart positive x_4 , il se transforme en

$$\begin{cases} x_1 + 2x_2 & = 0 \\ x_2 - x_3 - x_4 & = 0 \end{cases} \quad (3.26)$$

avec $[x_1] = [x_2] = [x_3] = \mathbb{R}$ et $[x_4] = [0, \infty[$.

Les inégalités ayant été transformées, pour résoudre un CSP \mathcal{H} , les algorithmes de propagation de contraintes utilisant des contracteurs peuvent être utilisés.

3.4.2. Définition et propriétés des contracteurs

Un *contracteur* est une opération permettant de réduire le domaine des variables d'un CSP \mathcal{H} , c'est à dire, remplacer le domaine initial $[\mathbf{x}]$ des variables de \mathcal{H} par un domaine $[\mathbf{x}']$ tel que l'ensemble solution \mathbb{S} du CSP reste inchangé.

La figure 3.9 présente un exemple de contraction d'un domaine. Le domaine obtenu est réduit en ne perdant aucune solution.

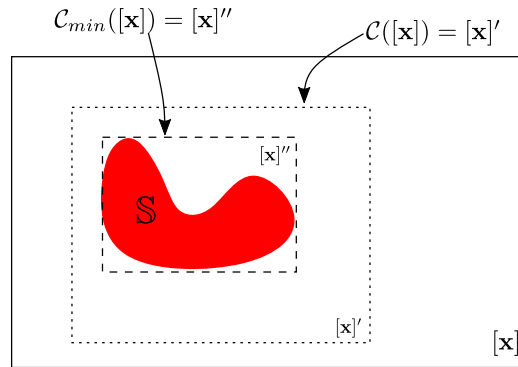


FIGURE 3.9. – Illustration d'un contracteur pour \mathbb{S}

Un opérateur $\mathcal{C} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$ est un *contracteur* (JAULIN et al. 2008; CHABERT et al. 2009) si :

- $\forall [\mathbf{x}] \in \mathbb{IR}^n, \mathcal{C}([\mathbf{x}]) \subseteq [\mathbf{x}]$ (contractance)
- $(\forall \mathbf{x} \in [\mathbf{x}], \mathcal{C}(\{\mathbf{x}\}) = \{\mathbf{x}\}) \Rightarrow \mathbf{x} \in \mathcal{C}([\mathbf{x}])$ (consistance)
- $\mathcal{C}(\{\mathbf{x}\}) = \emptyset \iff (\exists \epsilon > 0, \forall [\mathbf{x}] \in B(\mathbf{x}, \epsilon), \mathcal{C}([\mathbf{x}]) = \emptyset)$ (convergence)

où $B(\mathbf{x}, \epsilon)$ représente la boule de centre \mathbf{x} et de rayon ϵ .

3.4.2.1. Quelques propriétés d'un contracteur

Un contracteur est :

3. Estimation ensembliste par intervalles

- *minimal* lorsque la boîte résultant de l'opération de contraction est égale à la plus petite boîte qui contient $\mathbb{S} \cap \mathcal{C}([\mathbf{x}])$.

$$\forall [\mathbf{x}] \in \mathbb{I}\mathbb{R}^n, \mathcal{C}([\mathbf{x}]) = [\mathbb{S} \cap \mathcal{C}([\mathbf{x}])]$$

Dans l'exemple de la figure 3.9, la boîte $[\mathbf{x}]''$ est la plus petite boîte pouvant contenir $\mathbb{S} \cap \mathcal{C}([\mathbf{x}])$ donc \mathcal{C}_{min} est minimal.

- *monotone* si $[\mathbf{x}] \subset [\mathbf{y}] \implies \mathcal{C}([\mathbf{x}]) \subset \mathcal{C}([\mathbf{y}])$

3.4.3. Exemples de contracteurs

Il existe plusieurs méthodes de construction des contracteurs :

- par rapport à un ensemble de contraintes linéaires (Gauss-Seidel intervalles et les itérations de Krawczyk intervalles (NEUMAIER 1990))
- par rapport à un ensemble de contraintes non-linéaires (opérateur Newton Intervalles (HANSEN 1978))
- par une généralisation des techniques de programmation de contraintes telles que : la propagation (WALTZ 1972), la réfutation (encore appelée *Shaving* ou *3B-consistance*) (MOORE 1979; CLEARY 1987; HANSEN 1992; BENHAMOU et al. 1994) ou la construction disjonctive (NEVEU et al. 2015; TROMBETTONI et al. 2007).

Dans nos travaux, nous avons principalement utilisé des contracteurs de la troisième classe ; basés sur les techniques de propagation de contraintes.

3.4.3.1. Contracteur trivial

Un contracteur trivial $\mathcal{C}_{\mathbb{S}}$ (JAULIN et al. 2001) pour une fonction \mathbf{f} peut être défini à l'aide de la fonction d'inclusion $[\mathbf{f}]([\mathbf{x}])$

$$\mathcal{C}_{\mathbb{S}} : \begin{cases} \mathbb{I}\mathbb{R}^n \rightarrow \mathbb{I}\mathbb{R}^n \\ [\mathbf{x}] \mapsto \begin{cases} \emptyset & \text{si } [\mathbf{f}]([\mathbf{x}]) \cap [\mathbf{y}] = \emptyset \\ [\mathbf{x}] & \text{sinon} \end{cases} \end{cases} \quad (3.27)$$

La réduction, de type tout-ou-rien, dépend du résultat d'un test d'inclusion.

3.4.3.2. Contracteur par propagation et rétropropagation

Le contracteur par *propagation-rétropropagation*, noté $\mathcal{C}_{\downarrow\uparrow}$ et appelé *HC4Revise* ou encore *Forward-Backward*, est un algorithme classique de propagation de contraintes sur les intervalles proposé par BENHAMOU et al. (1999) et BENHAMOU et al. (2006). Basé sur la propagation de contraintes (WALTZ 1972), il consiste à appliquer chacune des contraintes de \mathcal{H} de manière indépendante en la décomposant en *contraintes primitives* principalement constituées d'opérations/fonctions élémentaires ($+$, $-$, $*$, $/$, \sin , \cos , ...).

3. Estimation ensembliste par intervalles

Exemple 3.5. Reprenons l'exemple 3.3 de localisation. À partir de la fonction d'observation, la contrainte associée à chaque balise B_i est donné par

$$d_i = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2}$$

Cette contrainte se décompose en contraintes primitives en introduisant des variables intermédiaires de la façon suivante

$$\begin{cases} a_1 &= -x_i \\ a_2 &= x_r + a_1 \\ a_3 &= a_2^2 \\ a_4 &= -y_i \\ a_5 &= y_r + a_4 \\ a_6 &= a_5^2 \\ a_7 &= a_3 + a_6 \\ d_i &= \sqrt{a_7} \end{cases} \quad (3.28)$$

Les domaines des variables intermédiaires ($a_1, a_2, a_3, a_4, a_5, a_6$ et a_7) sont initialisés à \mathbb{R} . Après cette décomposition, si l'on a les domaines ($[x_i], [y_i]$) des positions des amers et les domaines $[d_i]$ des mesures de distances, les domaines des autres variables (en particulier de la position du robot) peuvent être réduits en appliquant ces contraintes primitives de l'équation 3.28 (*propagation*).

Les expressions des contraintes de l'équation 3.28 peuvent être réécrites afin de *retropropager* les contraintes pour mettre à jour tous les domaines des variables du problème jusqu'à ce que les contractions des domaines soient inefficaces. Par exemple, une contrainte d'addition de la forme $a_2 = x_r + a_1$ peut être réécrite de 3 façons différentes

$$\begin{cases} x_r &= a_2 - a_1 \\ a_1 &= a_2 - x_r \\ a_2 &= x_r + a_1 \end{cases} \quad (3.29)$$

Ainsi, les contractions suivantes pourront donc être appliquées

$$\begin{cases} [x_r] &:= ([a_2] - [a_1]) \cap [x_r] \\ [a_1] &:= ([a_2] - [x_r]) \cap [a_1] \\ [a_2] &:= ([x_r] + [a_1]) \cap [a_2] \end{cases} \quad (3.30)$$

Toutes les contraintes de l'équation 3.28, sont ainsi écrites afin de permettre à l'algorithme de rétropropager partant de la racine vers les contraintes aux extrémités.

Ces deux étapes sont appliquées successivement jusqu'à ce que les contraintes soient inefficaces (les domaines ne soient plus significativement réduits après application des

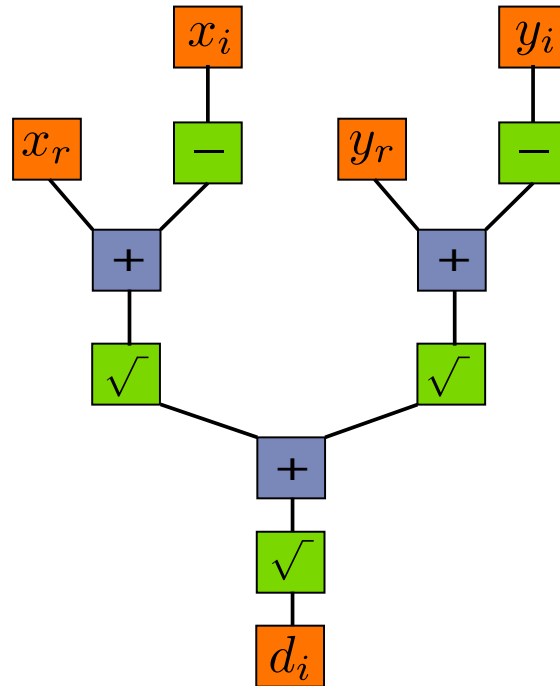


FIGURE 3.10. – Décomposition en contraintes primitives de la contrainte associée à la mesure de distance d'une balise B_i .

contraintes), c'est à dire l'obtention d'un point fixe (BESSIERE 2006). La figure 3.10 présente le graphe de contraintes de notre problème après une décomposition en contraintes primitives pour une balise. Lorsque ce graphe est acyclique, l'application de la propagation et la rétropropagation mène à la contraction optimale et l'algorithme 2 peut être écrit.

L'une des faiblesses de cette méthode est qu'en présence de multi-occurrences d'une variable dans les expressions des contraintes d'égalités (inégalités), les contractions sont moins efficaces car le graphe de contraintes est cyclique. Toutefois, ce contracteur peut être utilisé comme contracteur atomique.

3.4.3.3. Contracteur par découpage : Shaving

Le contracteur par découpage consiste à subdiviser le domaine des variables en sous-domaines appelés tranches de largeur w correspondant à une précision. Le contracteur atomique $\mathcal{C}_{\downarrow\uparrow}$ est appliqué à chaque tranche. Une tranche est inconsistante et donc éliminée de l'ensemble de départ si l'application du contracteur $\mathcal{C}_{\downarrow\uparrow}$ produit l'ensemble vide. La $3B(w)$ consistance (LHOMME 1993) du domaine entier est vérifiée si et seulement si les tranches aux bornes du domaine ne peuvent pas être éliminées par $\mathcal{C}_{\downarrow\uparrow}$.

La figure 3.11 illustre les notions de tranches et différents découpages.

3. Estimation ensembliste par intervalles

Propagation

$$\begin{aligned}
[a_1] &:= [x_i] \\
[a_2] &:= [x_r] + [a_1] \\
[a_3] &:= [a_2]^2 \\
[a_4] &:= -[y_i] \\
[a_5] &:= [y_r] + [a_4] \\
[a_6] &:= [a_5]^2 \\
[a_7] &:= [a_3] + [a_6] \\
[d_i] &:= \sqrt{[a_7]} \cap [d_i]
\end{aligned}$$

Rétropropagation

$$\begin{aligned}
[a_7] &:= [d_i]^2 \cap [a_7] \\
[a_6] &:= ([a_7] - [a_3]) \cap [a_6] \\
[a_3] &:= ([a_7] - [a_6]) \cap [a_3] \\
[a_5] &:= (\text{sqr}^{-1}[a_6]) \cap [a_5] \\
[a_4] &:= ([a_5] - [y_r]) \cap [a_4] \\
[y_r] &:= ([a_5] - [a_4]) \cap [y_r] \\
[y_i] &:= -[a_4] \cap [y_i] \\
[a_2] &:= (\text{sqr}^{-1}[a_3]) \cap [a_2] \\
[a_1] &:= ([a_2] - [x_r]) \cap [a_1] \\
[x_r] &:= ([a_2] - [a_1]) \cap [x_r] \\
[x_i] &:= -[a_1] \cap [x_i]
\end{aligned}$$

Algorithme 2 : $\mathcal{C} \downarrow \uparrow(\text{inout} : [\mathbf{x}])$

3.4.3.4. N-B consistance

Une généralisation de l'algorithme de consistance d'arc simple (*Singleton Arc Consistency*, noté SAC) permet de construire des contracteurs $NB(w)$ consistants (BESSIERE et al. 2008). Il consiste à utiliser les contracteurs de niveau $N - 1$ comme sous contracteur à appliquer à une tranche du domaine.

- Le contracteur de consistance $3B$, \mathcal{C}_{3B} utilise le contracteur $\mathcal{C}_{\downarrow \uparrow}$ comme sous-contracteur
- Le contracteur de consistance $4B$, \mathcal{C}_{4B} utilise le contracteur \mathcal{C}_{3B}
- Le contracteur de consistance $5B$, \mathcal{C}_{5B} utilise le contracteur \mathcal{C}_{4B}
- La NB utilise le contracteur $\mathcal{C}_{(N-1)B}$ (qui est le contracteur de niveau de consis-

3. Estimation ensembliste par intervalles

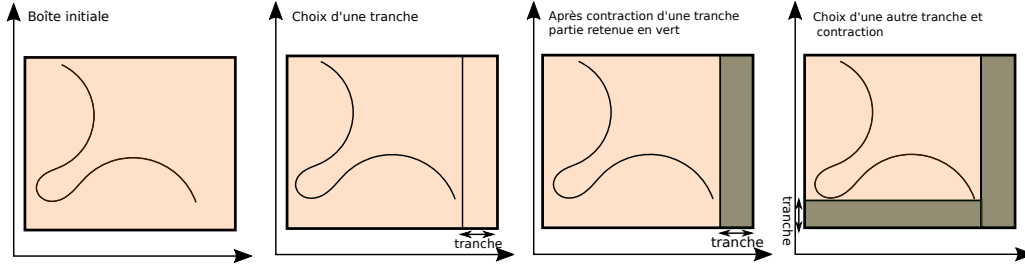


FIGURE 3.11. – Principe du contracteur par découpage. Illustration du découpage en tranche et du résultat après contraction.

tance $N - 1$).

Dans la section 4.4, nous présenterons l'application de chacun de ces contracteurs au problème d'estimation de pose et les résultats obtenus.

3.4.4. SIVIA avec contracteurs

Considérons le CSP :

$$\mathcal{H} : \mathbf{f}(\mathbf{x}) = \mathbf{y}, \mathbf{x} \in [\mathbf{x}], \mathbf{y} \in [\mathbf{y}]$$

Lorsque le domaine initial $[\mathbf{x}]$ est très grand, l'algorithme d'inversion ensembliste SIVIA (voir section 3.3.3), peut être coûteux en temps de calcul. Un moyen de pallier à ce manquement est de limiter le nombre de bisections de $[\mathbf{x}]$ en utilisant un contracteur \mathcal{C}_S .

```

1 :  $[\mathbf{x}] := \mathcal{C}_S([\mathbf{x}])$ 
2 : if  $[\mathbf{x}] = \emptyset$  then return ;
3 : if  $[\mathbf{f}]([\mathbf{x}]) \subset \mathbb{Y}$  then
4 :    $\{\underline{\mathbb{S}} := \underline{\mathbb{S}} \cup [\mathbf{x}]; \overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [\mathbf{x}];$  return ;}
5 : if  $w([\mathbf{x}]) < \epsilon$  then
6 :    $\{\overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [\mathbf{x}];$  return ;}
7 : bissecter  $[\mathbf{x}]$  en  $[\mathbf{x}_1]$  et  $[\mathbf{x}_2]$  ;
8 : SIVIAP(in :  $[\mathbf{x}_1]$ ,  $\mathcal{C}_S$ ,  $[\mathbf{f}]$ ,  $\mathbb{Y}$ ,  $\epsilon$ ,  $\underline{\mathbb{S}}$ ,  $\overline{\mathbb{S}}$ )
9 : SIVIAP(in :  $[\mathbf{x}_2]$ ,  $\mathcal{C}_S$ ,  $[\mathbf{f}]$ ,  $\mathbb{Y}$ ,  $\epsilon$ ,  $\underline{\mathbb{S}}$ ,  $\overline{\mathbb{S}}$ )

```

Algorithme 3 : SIVIAP(in : $[\mathbf{x}]$, \mathcal{C}_S , $[\mathbf{f}]$, \mathbb{Y} , ϵ ; inout : $\underline{\mathbb{S}}$, $\overline{\mathbb{S}}$)

L'algorithme 3, SIVIAP, consiste donc à remplacer le test d'inclusion de l'algorithme SIVIA sur une boîte $[\mathbf{x}]$ par une étape de contraction afin d'éliminer les parties du domaine qui sont inconsistantes.

- Si la boîte après contraction est vide, la boîte $[\mathbf{x}]$ est rejetée.
- Si elle est non vide et prouvée incluse dans la solution, $[\mathbf{x}]$ est rajouté aux approximations intérieure et extérieure ($\underline{\mathbb{S}}$, $\overline{\mathbb{S}}$).

3. Estimation ensembliste par intervalles

— Si la largeur de $[\mathbf{x}]$ est inférieure à ϵ alors la boîte est indéterminée et appartient au sous pavage extérieur $\bar{\mathbb{S}}$

Considérons l'exemple de localisation planaire 3.3, en utilisant le contracteur extérieur par propagation et rétro-propagation présenté plus haut, nous obtenons les résultats de la figure 3.12.

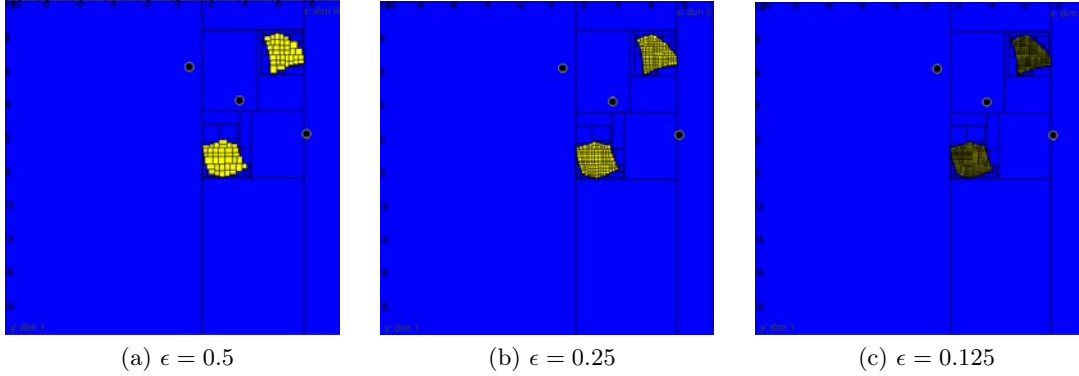


FIGURE 3.12. – Résultats de l'algorithme SIVIA avec le contracteur $\mathcal{C}_{\downarrow\uparrow}$ pour différentes valeurs du paramètre de précision ϵ .

3.5. Inversion ensembliste q -relaxée

L'inversion ensembliste q -relaxée est une méthode d'estimation ensembliste robuste (JAULIN et al. 1996). Lorsque nous avons un ensemble de m contraintes imposées par m mesures (qui peuvent être erronées), la q -intersection est un outil permettant de fusionner ces mesures tout en tolérant un certain nombre q de mesures aberrantes. La solution de l'inversion ensembliste q -relaxée est l'ensemble des paramètres compatibles avec au moins $m - q$ mesures.

3.5.1. Intersection q -relaxée

La q -intersection de m sous ensembles $\mathbb{X}_1, \dots, \mathbb{X}_m$ de \mathbb{R}^n est l'ensemble noté $\mathbb{X}^{\{q\}} = \bigcap^{\{q\}} \mathbb{X}_i$, des paramètres $\mathbf{x} \in \mathbb{R}^n$ appartenant à au moins $m - q$ sous ensembles \mathbb{X}_i . La figure 3.13 en illustre le principe avec 4 ensembles pour différentes valeurs de q .

Lorsque $q = 0$, on retourne l'intersection des ensembles solutions. Lorsque $q = m - 1$ (illustration la plus à droite de la figure 3.13), on retourne l'union des ensembles solutions.

3.5.2. Inversion ensembliste robuste par intervalles

L'algorithme RSIVIA pour *Robust Set Inversion Via Interval Analysis* (JAULIN 2009) permet d'estimer de manière garantie un ensemble solution q -relaxé. RSIVIA consiste

3. Estimation ensembliste par intervalles

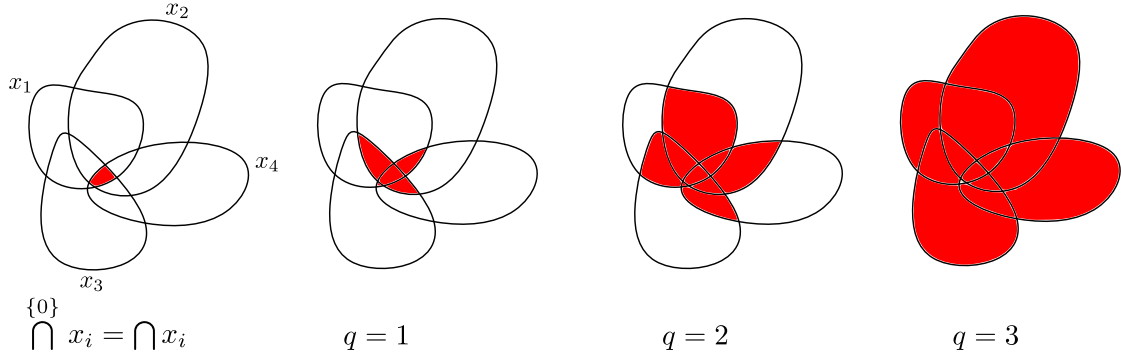


FIGURE 3.13. – Illustration de l'intersection q -relaxée pour sur 4 ensembles pour $q = 0, 1, 2, 3$

à déterminer l'intersection q -relaxée des sous-ensembles solution rattachés à chaque composante $f_i^{-1}([y_i])$ de \mathbf{f} .

Connaissant une fonction d'inclusion pour \mathbf{f} , nous allons définir un test d'inclusion pour l'inversion ensembliste q -relaxée. La résolution du problème pourra ainsi s'effectuer en adaptant l'algorithme SIVIA (Alg. 1). Le test se base sur les notions de degrés d'inclusion et de séparation.

Degré d'inclusion et degré de séparation Le degré d'inclusion « incl » d'une boîte $[\mathbf{x}]$ dans une boîte $[\mathbf{y}]$, est le nombre de composantes intervalles $[x_i]$ qui vérifient l'inclusion dans $[y]$. Il est défini par

$$\text{incl}([\mathbf{x}], [\mathbf{y}]) = \sum_i \text{incl}([x_i], [y_i])$$

$$\text{où } \text{incl}([x_i], [y_i]) = \begin{cases} 1 & \text{si } [x_i] \subset [y_i] \\ 0 & \text{sinon} \end{cases}.$$

Le degré de séparation « sep » quant-à lui représente le nombre de composantes intervalles $[x_i]$ disjointes de $[y]$. Il est défini par

$$\text{sep}([\mathbf{x}], [\mathbf{y}]) = \sum_{i=1}^m \text{sep}([x_i], [y_i])$$

$$\text{avec } \text{sep}([x_i], [y_i]) = \begin{cases} 1 & \text{si } [x_i] \cap [y_i] = \emptyset \\ 0 & \text{sinon} \end{cases}.$$

Ainsi le test d'inclusion q -relaxé d'une boîte $[\mathbf{x}]$ dans RSIVIA est donné par :

$$\begin{cases} \text{sep}([\mathbf{f}]([\mathbf{x}]), [\mathbf{y}]) > q & [\mathbf{x}] \text{ ne contient aucune solution} & \{q\} \\ \text{incl}([\mathbf{f}]([\mathbf{x}]), [\mathbf{y}]) \geq m - q & [\mathbf{x}] \text{ est dans l'ensemble solution (} [\mathbf{x}] \in \bigcap f_i^{-1}([y_i]) \text{)} \\ \text{sinon} & [\mathbf{x}] \text{ est indéterminé} \end{cases} \quad (3.31)$$

3. Estimation ensembliste par intervalles

Le test d'inclusion ainsi conçu permet d'adapter l'algorithme SIVIA pour le rendre robuste à la présence d'au plus q mesures aberrantes. Ainsi, RSIVIA (Algorithme 4) permet d'encadrer le résultat de l'inversion ensembliste q -relaxée entre un sous-pavage intérieur $\underline{\mathbb{S}}$ et un sous-pavage extérieur $\overline{\mathbb{S}}$.

```

1 :  incl := 0 ; sep := 0
2 :    for  $i = 1 \dots m$  do
3 :      if  $[f_i]([\mathbf{x}]) \subset [y_i]$  then
4 :         $incl := incl + 1$ 
5 :      else if  $[f_i]([\mathbf{x}]) \cap [y_i] = \emptyset$  then
6 :         $sep := sep + 1$ 
7 :      end if
8 :    end for
9 :    if  $incl > m - q$  then
10 :       $\{\underline{\mathbb{S}} := \underline{\mathbb{S}} \cup [\mathbf{x}] ; \overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [\mathbf{x}] ; \text{return} ;\}$ 
11 :    else if  $sep > q$  then
12 :       $\text{return} ;$ 
13 :    else if  $w([\mathbf{x}]) < \epsilon$  then
14 :       $\{\overline{\mathbb{S}} := \overline{\mathbb{S}} \cup [\mathbf{x}] ; \text{return} ;\}$ 
15 :    else
16 :      bissecter  $[\mathbf{x}]$  en  $[\mathbf{x}_1]$  et  $[\mathbf{x}_2]$  ;
17 :      SIVIAP(in :  $[\mathbf{x}_1]$ ,  $\mathcal{C}_S$ ,  $[\mathbf{f}]$ ,  $\mathbb{Y}$ ,  $\epsilon$ ,  $\underline{\mathbb{S}}$ ,  $\overline{\mathbb{S}}$ )
18 :      SIVIAP(in :  $[\mathbf{x}_2]$ ,  $\mathcal{C}_S$ ,  $[\mathbf{f}]$ ,  $\mathbb{Y}$ ,  $\epsilon$ ,  $\underline{\mathbb{S}}$ ,  $\overline{\mathbb{S}}$ )
19 :    end if

```

Algorithme 4 : RSIVIA(in : $[\mathbf{x}]$, $[\mathbf{f}]$, $[\mathbf{y}]$, ϵ , q ; inout : $\underline{\mathbb{S}}$, $\overline{\mathbb{S}}$)

Dans le cadre de SIVIA avec contracteurs, le problème est défini sous forme de contraintes, la solution revient à construire m contracteurs $\mathcal{C}_1 \dots \mathcal{C}_m$. Un ensemble solution est calculée à partir de chaque contrainte et l'intersection q -relaxée est appliquée à ces sous ensembles afin d'avoir la solution q -relaxée avec contracteurs.

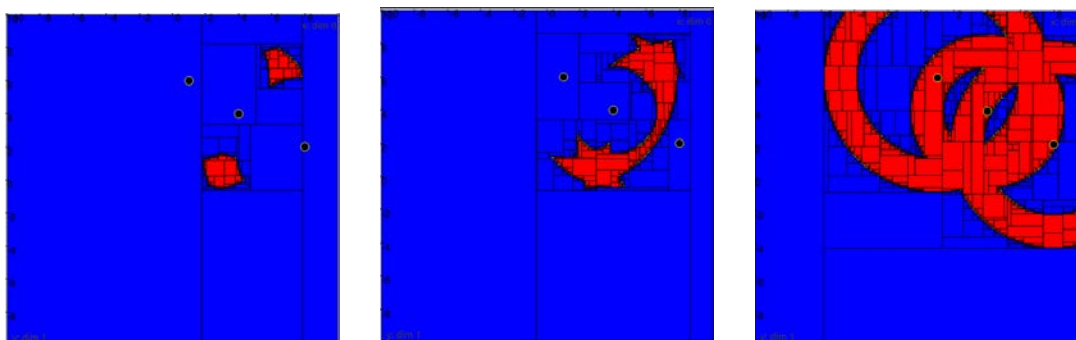
Reprenons l'exemple de localisation, sans introduire de mesure aberrantes, la figure 3.14 présente le résultat de RSIVIA pour différentes valeurs de q . Nous calculons ainsi toutes les positions (x_r, y_r) compatibles avec au moins $3 - q$ balises et les mesures de distances à erreurs bornées associées.

Dans les trois cas présentée dans cette figure, l'ensemble solution n'est jamais vide car aucune des mesures n'est en réalité fausse. Dans la section suivante, nous verrons comment détecter la présence de mesures aberrantes grâce à cet algorithme.

3.5.3. Détection de mesures aberrantes

Lorsque que toutes les bornes d'erreur des mesures d'un problème d'inversion ensembliste sont correctes, la solution donnée par l'algorithme SIVIA ne peut être vide.

3. Estimation ensembliste par intervalles



(a) Solution compatible avec les 3 mesures de distances ($q = 0$).
 (b) Solution compatible avec au moins 2 mesures de distances ($q = 1$).
 (c) Solution compatible avec au moins 1 mesure de distance ($q = 2$).

FIGURE 3.14. – Résultats des l’algorithme RSIVIA en considérant un nombre variable de mesures aberrantes et le paramètre $\epsilon = 0.125$

Cette solution est vide lorsqu’aucun paramètre de l’espace solution n’est consistant avec les mesures. Ainsi, la présence de mesures aberrantes est détectée lorsque l’ensemble solution donné par SIVIA ou RSIVIA pour $q = 0$ est vide.

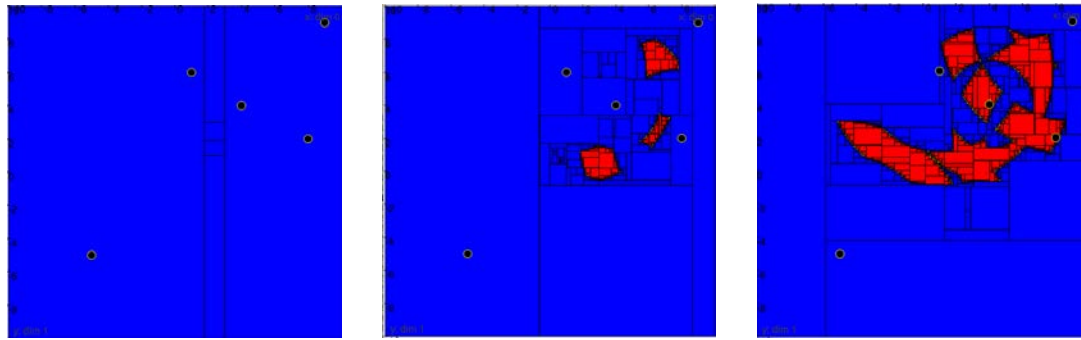
Dans ce cas, RSIVIA peut être utilisé en supposant qu’une mesure est aberrante. Cela conduira à calculer $\mathbb{X}^{\{1\}}$. Si cette information est correcte, alors l’ensemble solution sera non vide, sinon, l’ensemble calculé sera vide (signifiant que le nombre de mesures aberrantes est supérieur à q , soit au moins égal à 2). On peut répéter cette procédure en incrémentant q jusqu’à obtenir la preuve d’une solution non-vide. Cette stratégie constitue la méthode GOMNE (*Guaranteed Outlier minimum number estimator*) (JAULIN et al. 1996).

Rajoutons maintenant deux balises B_4 et B_5 de positions respectives $(9, 9)$ et $(-5, -5)$ au problème de localisation planaire précédent en leur affectant des mesures de distances erronées (la distance réelle n’étant pas contenue dans l’intervalle renseigné). En appliquant RSIVIA à ces données en supposant $q = 0$ (donc qu’aucun mesure n’est aberrante), nous obtenons le sous-pavage de la figure 3.15a. L’ensemble solution est vide (bleu), cela signifie qu’aucune solution n’est compatible avec l’ensemble des mesures. Le résultat est le même pour $q = 1$. À partir de $q \geq 2$ (figures 3.15b et 3.15c) nous avons un ensemble solution non vide.

Ainsi grâce à ce résultat, il est possible de conclure que sur les 5 mesures de distances des balises, au moins 2 sont fausses.

Nous utiliserons, dans cette thèse, l’algorithme SIVIA pour calculer le domaine de la pose d’un ou de plusieurs drones. L’obtention d’ensembles vides, indiquant la présence de mesures erronées, devra être traitée spécifiquement (réinitialisation, q -relaxée, ...).

3. Estimation ensembliste par intervalles



(a) $q = 0, 1$. L'ensemble solution est vide

(b) $q = 2$, Une solution non vide est obtenue, correspondant à l'ensemble des positions compatibles avec au moins $m - q = 3$ mesures.

(c) $q = 3$

FIGURE 3.15. – Résultats de l'inversion ensembliste q -relaxée pour l'exemple de localisation en présence de 5 balises dont 2 sont des *outliers*.

3.6. Conclusion

Dans ce chapitre, nous avons introduit les notions de base sur l'analyse par intervalles ainsi que le principe de l'inversion ensembliste, illustré par un exemple pratique de localisation planaire avec des balises dont les positions sont connues.

Les outils de résolution des problèmes de satisfaction de contraintes tels que les contracteurs sont utilisés dans un algorithme d'inversion ensembliste pour déterminer une approximation de la solution par un sous-pavage, dont la finesse est déterminée par le paramètre ϵ . Les contracteurs par propagation et rétropropagation des contraintes et ceux par réfutation ont principalement été présentés.

Une version améliorée de l'algorithme d'inversion ensembliste SIVIA notée SIVIAP a été présentée ainsi qu'une version permettant l'estimation robuste RSIVIA basée sur l'intersection q -relaxée. Nous avons montré comment cette dernière version permet d'estimer des paramètres tout en tolérant la présence d'un certain nombre de mesures aberrantes.

4. Caractérisation d'un domaine de localisation pour un drone

4.1. Introduction

Pour naviguer et accomplir leurs tâches, les robots doivent non seulement se situer par rapport à l'environnement, mais aussi ont besoin d'avoir une information fiable sur l'incertitude de leur position. Dans le cas d'un drone aérien, la solution standard consiste à utiliser un GPS, une centrale inertielle, un baromètre et une boussole. Cette solution n'est pourtant pas appropriée dans un environnement où les signaux GPS sont très bruités ou bloqués tel que, l'intérieur, la proximité de grands bâtiments, ou près des structures en acier qui compromettent les lectures de la boussole comme évoqué au chapitre 1.

Comme un grand nombre de drones disponibles sur le marché sont désormais équipés d'une caméra vidéo, utilisée principalement pour fournir un retour visuel au pilote et effectuer des acquisitions aériennes de vidéos, nous utiliserons la vision pour remédier à ce problème de localisation.

En vision par ordinateur, comme abordé au chapitre 2, de nombreuses solutions pour effectuer une estimation de pose à partir d'un ensemble de points connus existent (voir, par exemple, HARTLEY et al. (2001), LEPETIT et al. (2005) et MARCHAND et al. (2016)). Toutefois, ces méthodes fournissent classiquement une estimation ponctuelle de la position du robot. L'analyse par intervalles telle que présentée au chapitre précédent est donc un outil puissant pour la propagation rigoureuse de l'incertitude, utilisé par exemple pour une application de vision 3D (TELLE et al. 2003), la localisation sous-marine (JAULIN 2009) et pour le calcul d'un domaine d'incertitude de la position pour des robots mobiles (DREVELLE et al. 2013; COLLE et al. 2013).

Dans ce chapitre, nous nous intéresserons donc à caractériser un domaine contenant la pose d'un robot équipé d'une caméra et de capteurs proprioceptifs, en utilisant des techniques d'inversion ensembliste et de propagation de contraintes basées sur les intervalles. Plus précisément, pour quantifier l'incertitude de la pose de robot, nous proposons une approche ensembliste par intervalles (JAULIN et al. 2001), qui calcule systématiquement un sous-pavage de l'ensemble solution contenant la pose exacte du robot en prenant en compte les incertitudes sur les mesures 2D dans les images et les positions des points 3D géo-référencés (connus).

Ce chapitre est organisé comme suit. Le modèle cinématique du type de drones utilisés dans cette thèse est décrit à la section 4.2. La section 4.3 explique tout d'abord le

problème d'estimation de pose à partir de mesures à erreurs bornées. Notre approche basée sur les notions d'inversion ensembliste et de contracteurs pour résoudre ce problème est ensuite détaillée; elle présente une formalisation en problème de satisfaction de contraintes estimant les six degrés de libertés de la pose et un autre enrichi avec des mesures proprioceptives pour réduire la caractérisation à trois des six degrés de liberté. La section 4.4 expose par la suite quelques résultats en simulation et expérimentaux obtenus avec un drone de type quadrirotor, ainsi qu'une comparaison de la méthode proposée avec un filtre de Kalman étendu.

4.2. Modélisation du drone utilisé

Nous considérons ici un drone de type quadrirotor équipé d'une caméra. Ce drone est assimilé à un corps rigide à masse fixe, le modèle générique à six degrés de libertés (6DoF) représente ses trois translations et ses trois rotations spatiales (constituant ainsi sa pose). Il est possible, grâce à ce modèle, de décrire la cinématique de rotation et de translation du drone. Cette section, définit les différents repères et systèmes d'axes utilisés ainsi que les relations composant le modèle cinématique du drone considéré.

4.2.1. Repères et pose d'un drone

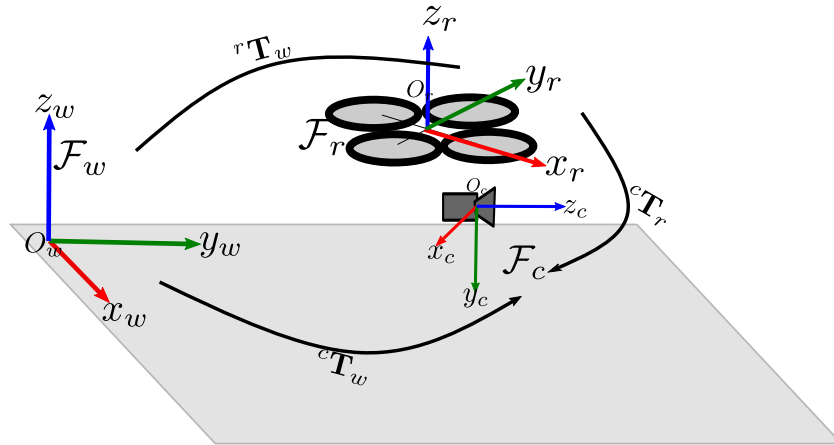


FIGURE 4.1. – Définition des repères utilisés : inertiel \mathcal{F}_w , robot \mathcal{F}_r , caméra \mathcal{F}_c .

La figure 4.1 illustre les différents repères permettant de modéliser le mouvement du drone dans son environnement. Nous avons :

- le repère monde (ou repère inertiel) \mathcal{F}_w lié à la terre et défini par la base $\{x_w, y_w, z_w\}$, relatif à l'origine O_w où la première loi de Newton s'applique.
- le repère drone \mathcal{F}_r ayant pour origine son centre de gravité O_r . La base $\{x_r, y_r, z_r\}$ est associée à \mathcal{F}_r où x_r est l'axe de roulis, y_r l'axe de tangage et z_r est l'axe de lacet du drone.

4. Caractérisation d'un domaine de localisation pour un drone

— le repère caméra \mathcal{F}_c ayant pour origine son centre optique.

Ces repères étant définis, la position du drone est caractérisée par la position de son centre de gravité et l'orientation de ses axes par rapport au repère monde \mathcal{F}_w . La pose du drone par rapport au référentiel \mathcal{F}_w est notée $\mathbf{r} = {}^r\mathbf{r}_w$. On peut de la même manière définir la pose de la caméra par rapport à \mathcal{F}_w , ${}^c\mathbf{r}_w$ et la pose de la caméra par rapport à \mathcal{F}_r , ${}^c\mathbf{r}_r$.

4.2.2. Modèle cinématique

Les paramètres cinématiques utilisés, nécessaires à l'étude du mouvement du drone, sont constitués de la pose du drone, des vitesses linéaires et des vitesses angulaires de son centre de gravité (centre instantané de rotation). Nous avons :

- sa position $\mathbf{p} = (x, y, z)^\top$ exprimée dans le repère inertiel \mathcal{F}_w
- sa position angulaire (équivalente à la position angulaire du repère robot \mathcal{F}_r), est décrite par la matrice de rotation ${}^r\mathbf{R}_w = \mathbf{R}$ pouvant être représentée en utilisant les angle d'Euler autour des axes (x_r, y_r, z_r) appelés $\mathbf{q} = (\phi, \theta, \psi)^\top$. La matrice \mathbf{R} peut être définie de la façon suivante :

$$\mathbf{R} = \begin{pmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{pmatrix} \quad (4.1)$$

- sa vitesse linéaire $\mathbf{v} = (u, v, w)^\top$ par rapport à \mathcal{F}_w et exprimée dans \mathcal{F}_r
- sa rotation instantanée $\mathbf{w} = (p, q, r)^\top$ par rapport à \mathcal{F}_w et exprimée dans \mathcal{F}_r .

Les relations cinématiques décrivant le mouvement du drone dans le repère inertiel sont écrites à partir de ces paramètres. On a :

- la cinématique de translation donnée par :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \mathbf{R} \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (4.2)$$

- la cinématique de rotation donnée par :

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (4.3)$$

Posons $\mathbf{M} = \begin{pmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{pmatrix}$. On peut réécrire $\dot{\mathbf{p}} = \mathbf{R} \cdot \mathbf{v}$ et $\dot{\mathbf{q}} = \mathbf{M} \cdot \mathbf{w}$.

Ce modèle sera utilisé tout au long de cette thèse pour modéliser le mouvement du drone dans son environnement de navigation. Il est principalement utilisé pour effectuer la prédiction du domaine de pose d'un instant t_k à un instant t_{k+1} en intégrant les mesures de vitesses acquises des capteurs embarqués du drone.

4.3. Estimation de pose à partir d'un modèle de mesures à erreurs bornées

Le chapitre 2 a montré comment estimer la pose d'une caméra à partir de la mise en correspondance de mesures 2D dans l'image et des points 3D géo-référencés. Les méthodes classiques sont principalement basées sur le fait de trouver une valeur de la pose minimisant l'erreur de projection des points 3D. Sous l'hypothèse de mesures à erreurs bornées, nous allons utiliser l'approche ensembliste (présentée au chapitres 3) pour déterminer l'ensemble des poses compatibles avec les observations et leurs intervalles d'erreurs associés (incertitude sur les mesures).

Cette section présente comment utiliser l'analyse par intervalle pour fournir un domaine contenant toutes les poses du robot en tenant compte des incertitudes sur les mesures.

La figure 4.2 permet d'illustrer la situation.

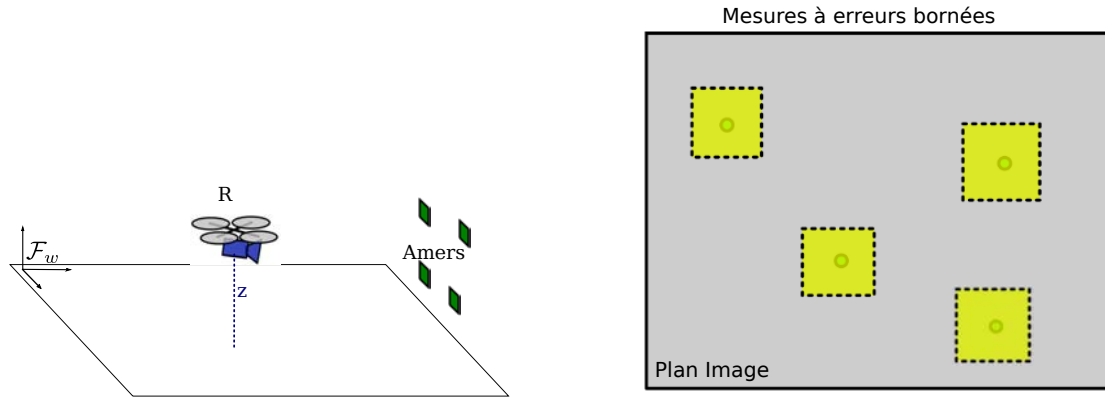


FIGURE 4.2. – Problème d'estimation de pose en présence de mesures à erreurs bornées.

4.3.1. Mesures à erreurs bornées

Les données mesurées (coordonnées des points 2D dans l'image) et les positions des amers (points 3D géolocalisés) sont sujettes à des incertitudes. La présence d'erreurs dans les mesures des coordonnées en pixels des points 2D est principalement due aux imprécisions et erreurs de la méthode de détection et suivi dans l'image. En ce qui concerne les amers, l'incertitude de leur position provient de la précision de la carte et du référentiel associé.

Considérons que l'erreur de mesure dans l'image est bornée; on peut écrire pour les coordonnées de chaque point 2D mesuré dans l'image

$$\bar{\mathbf{x}}_i \in \bar{\mathbf{x}}_i^{\text{meas}} + [\mathbf{e}_{img}] \quad (4.4)$$

avec $[\mathbf{e}_{img}] = ([-\epsilon_{img}, \epsilon_{img}], [-\epsilon_{img}, \epsilon_{img}])^\top$ est l'incertitude sur la mesure et $\bar{\mathbf{x}}_i^{\text{meas}} = (u_i, v_i)^\top$, la valeur mesurée. On peut donc représenter les mesures dans l'image par un vecteur d'intervalles, tel que $[\bar{\mathbf{x}}] = \bar{\mathbf{x}}_i^{\text{meas}} + [\mathbf{e}_{img}]$.

4. Caractérisation d'un domaine de localisation pour un drone

De manière similaire, un point 3D dans le repère global de référence ${}^w\mathbf{X}$ ayant une incertitude $[\mathbf{e}_w]$, sera représenté par un vecteur d'intervalles tel que $[{}^w\mathbf{X}] = {}^w\mathbf{X} + [\mathbf{e}_w]$, avec $[\mathbf{e}_w] = ([-\epsilon_w, \epsilon_w], [-\epsilon_w, \epsilon_w], [-\epsilon_w, \epsilon_w])^\top$.

4.3.2. Ensemble de contraintes liés à l'image

Rappelons qu'estimer la pose de la caméra revient à déterminer la matrice de transformation ${}^c\mathbf{T}_w$ entre le repère inertiel et le repère de la caméra, à partir de l'équation de projection d'un point 3D géolocalisé dans l'image.

$$\bar{\mathbf{x}} = \mathbf{K} \mathbf{\Pi} {}^c\mathbf{T}_w {}^w\mathbf{X} \quad (4.5)$$

Dans ces travaux, pour résoudre le problème d'estimation de pose d'un drone à partir de la vision et en présence de mesures à erreurs bornées, nous considérons connus les paramètres intrinsèques de calibration de la caméra et la matrice rigide de transformation entre le repère drone et le repère caméra ${}^c\mathbf{T}_r$. Ainsi, étant donné que ${}^c\mathbf{T}_w = {}^c\mathbf{T}_r {}^r\mathbf{T}_w(\mathbf{r})$, déterminer la pose du drone revient à déterminer la pose de la caméra. La matrice ${}^r\mathbf{T}_w(\mathbf{r})$ représente la pose du robot dans le repère inertiel.

En considérant les 6 degrés de liberté de la pose du drone inconnus, à partir des relations liés aux mesures images, un ensemble de contraintes est obtenu. L'équation de projection (équation 4.5) et l'utilisation d'une caméra frontale donnent lieu à 3 contraintes associées à chaque paire de mesures.

- deux contraintes C_{proj} exprimant la correspondance entre l'image et les points du monde qui vérifient la relation de projection perspective.

Les coordonnées des amers sont d'abord transformées dans le repère camera

$$\begin{pmatrix} {}^cX \\ {}^cY \\ {}^cZ \\ 1 \end{pmatrix} = {}^c\mathbf{T}_r {}^r\mathbf{T}_w(\mathbf{r}) \begin{pmatrix} {}^wX \\ {}^wY \\ {}^wZ \\ 1 \end{pmatrix} \quad (4.6)$$

puis projetées dans l'image :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K} \mathbf{\Pi} \begin{pmatrix} {}^cX \\ {}^cY \\ {}^cZ \\ 1 \end{pmatrix} \quad (4.7)$$

avec $({}^cX, {}^cY, {}^cZ, 1)^\top$ les coordonnées de l'amer dans le repère caméra.

C_{proj} est exprimée comme suit :

$$C_{proj} = \left\{ \begin{array}{l} u = p_x x_i + u_0, \\ v = p_y y_i + v_0 \end{array} \right\}$$

$$\text{avec } x_i = \frac{{}^cX}{{}^cZ} \text{ et } y_i = \frac{{}^cY}{{}^cZ}.$$

4. Caractérisation d'un domaine de localisation pour un drone

- la contrainte C_{front} exprime le fait qu'ayant une caméra, tous les points dans le repère caméra ont la coordonnée cZ positive (en d'autres termes, les points observés sont situés dans le demi-espace situé devant le plan image). Cela conduit à :

$$C_{front} = \{{}^cZ > 0\}.$$

4.3.3. Méthode de résolution du problème

Le calcul de la pose $\mathbf{r} = (x, y, z, \phi, \theta, \psi)^\top$ consiste à déterminer l'ensemble \mathbb{S} des poses compatibles avec les N points observés. La méthode présentée ici ne consiste donc pas à déterminer la pose minimisant un certain critère, mais plutôt à rechercher le domaine contenant toutes les poses possibles \mathbf{r} telles que $\bar{\mathbf{x}}_i = \mathbf{K} \mathbf{\Pi} {}^c\mathbf{T}_w(\mathbf{r}) {}^w\mathbf{X}_i$ (avec $i = 1 \dots N$) en considérant des correspondances entre mesures des points 2D incertains $\bar{\mathbf{x}}_i$ et positions des points 3D ${}^w\mathbf{X}_i$.

La largeur du domaine calculé quantifiera donc l'incertitude sur la pose par rapport à celle des mesures.

Pour atteindre cet objectif, notre méthode consiste à :

1. écrire notre problème comme un problème de satisfaction de contrainte (CSP),
2. sélectionner un contracteur pour la propagation des contraintes du CSP,
3. puis appliquer l'algorithme SIVIA avec le ou les contracteurs choisi(s) pour caractériser l'ensemble solution \mathbb{S} .

Cette méthode calcule donc un sous-pavage de \mathbb{S} en fonction des données mesurées. La bibliothèque C++ IBEX¹ offre un ensemble d'outils pour les calculs sur les intervalles. Nous avons utilisée cette dernière pour implémenter la méthode proposée.

4.3.4. Définition des CSP et de l'ensemble solution correspondant

À partir des contraintes sur les mesures disponibles, un CSP peut être défini. Dans ce chapitre, nous avons traité deux cas de figure. Le premier consiste à estimer la pose complète du robot, les 6 degrés de liberté (équation 4.8) en considérant uniquement les mesures dans l'image. Le second cas correspond à l'estimation de 3 des 6 degrés de liberté (équation 4.10) du fait que les 3 autres peuvent être mesurés à partir de capteurs proprioceptifs (IMU et altimètre).

1. <http://www.ibex-lib.org/>

4.3.4.1. CSP pour l'estimation des 6DoF

Pour l'estimation des six degrés de liberté avec uniquement comme mesures les points dans l'image et les amers géo-référencés, nous pouvons définir le CSP \mathcal{H}_{6DoF} suivant :

$$\mathcal{H}_{6DoF} : \left(\begin{array}{l} \mathbf{r} \in [\mathbf{r}], \\ \bar{\mathbf{x}}_i \in [\bar{\mathbf{x}}_i], \\ {}^w\mathbf{X}_i \in [{}^w\mathbf{X}_i], \\ \{C_{proj,i}\}, \\ \{C_{front,i}\}, \\ i = 1 \dots N \end{array} \right) \quad (4.8)$$

La boîte $[\mathbf{r}]$ représente la connaissance à priori sur la pose du robot. L'ensemble solution \mathbb{S}_{6DoF} de ce problème est l'ensemble des poses \mathbf{r} contenues dans $[\mathbf{r}]$ satisfaisant toutes les contraintes de \mathcal{H}_{6DoF} et décrit comme suit :

$$\mathbb{S}_{6DoF} = \{\mathbf{r} \in \mathbb{R}^3 \times SO(3) \mid \exists {}^w\mathbf{X}_i \in [{}^w\mathbf{X}_i], i = 1 \dots N, \mathbf{K} \Pi {}^c\mathbf{T}_w(\mathbf{r}) {}^w\mathbf{X}_i \in [\bar{\mathbf{x}}_i], {}^cZ_i > 0\} \quad (4.9)$$

4.3.4.2. CSP enrichi, estimation de 3 DoF

Il est possible à partir des capteurs proprioceptifs du drone d'avoir des mesures donnant une information sur trois degrés de libertés de sa pose. Sous hypothèse que le drone navigue en régime *quasi-stationnaire*, le roulis ϕ et le tangage θ peuvent être mesurés avec une faible incertitude $[\epsilon_{angle}]$ grâce à l'IMU. Cette hypothèse laisse donc un seul degré de liberté parmi les trois rotations à estimer : le lacet ψ . L'altitude z est également mesurée avec une incertitude $[\epsilon_z]$ à partir de l'altimètre. Ces données sont telles que :

- $[z_m] = z_{meas} + [\epsilon_z]$
- $[\phi_m] = \phi_{meas} + [\epsilon_{angle}]$
- $[\theta_m] = \theta_{meas} + [\epsilon_{angle}]$.

Ceci nous permet de décrire le CSP \mathcal{H}_{3DoF} suivant :

$$\mathcal{H}_{3DoF} : \left(\begin{array}{l} x \in [x], y \in [y], \psi \in [\psi], \\ \bar{\mathbf{x}}_i \in [\bar{\mathbf{x}}_i], \\ {}^w\mathbf{X}_i \in [{}^w\mathbf{X}_i], \\ \{C_{proj,i}\}, \\ \{C_{front,i}\}, i = 1 \dots N \\ z \in [z_m], \phi \in [\phi_m], \theta \in [\theta_m] \end{array} \right). \quad (4.10)$$

L'ensemble solution de \mathcal{H}_{3DoF} peut être décrit comme suit :

$$\mathbb{S}_{3DoF} = \{\mathbf{r} \in \mathbb{R}^2 \times [z_m] \times [\phi_m] \times [\theta_m] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \mid \exists {}^w\mathbf{X}_i \in [{}^w\mathbf{X}_i], i = 1 \dots N, \mathbf{K} \Pi {}^c\mathbf{T}_w(\mathbf{r}) {}^w\mathbf{X}_i \in [\bar{\mathbf{x}}_i], {}^cZ_i > 0\} \quad (4.11)$$

4.3.5. Contracteur utilisé

Les CSP et leurs solutions ayant été définis, il est nécessaire de choisir un contracteur à utiliser pour résoudre le problème.

Dans notre cas, nous avons utilisé un point fixe de chacun des contracteurs par découpage C_{3B} , C_{4B} et C_{5B} pour les utiliser dans l'algorithme SIVIA afin de calculer une approximation extérieure de l'ensemble solution. Chacun de ces contracteurs utilise le contracteur par propagation-rétropropagation de contraintes $C_{\downarrow\uparrow}$ comme contracteur atomique car il est rapide pour la contraction de grands domaines.

En effet, l'algorithme SIVIA avec test d'inclusion effectue des bisections sur 6 dimensions en partant de domaines arbitrairement grand, ce qui rend le problème intraitable en terme de complexité en temps de calcul. Comme vu précédemment, l'utilisation des contracteurs permet en partie de pallier à ce problème grâce à une réduction considérable du domaine en éliminant la partie de la pose non consistante avec les mesures.

Ces contracteurs ont été définis et expliqués à la section 3.4 du chapitre 3. Les résultats en simulation retournés par l'algorithme SIVIA avec chacun de ces contracteurs, dans les cas 6DoF et à 3DoF, sont présentés et comparés à la section 4.4.

4.3.6. Calcul d'une estimée ponctuelle

Afin de pouvoir comparer la méthode proposée aux méthodes classiques d'estimation de pose et de pouvoir fournir une valeur de la pose lorsque nécessaire, nous pouvons calculer une estimée ponctuelle. A partir du sous-pavage retourné par l'algorithme, il existe plusieurs moyens de calcul une pose ponctuelle contenue dans le domaine. Cela peut être au travers du calcul du centre de masse du sous pavage, de la médiane généralisée des centres ou alors du centre de la boîte englobante du sous-pavage.

Dans la suite de cette thèse, nous faisons le choix d'utiliser le centre de la boîte englobante (CoB pour *Center of Bounding-box*) comme estimée ponctuelle. C'est l'estimée ponctuelle la plus simple et directe à obtenir. En effet, étant donné un sous-pavage \mathbb{S}_{sol} , formé d'un ensemble de K boîtes de pose $\{[\mathbf{r}]_k, k = 1 \dots K\}$, le calcul de \mathbf{r}_{CoB}^* qui est l'estimée ponctuelle se fait comme suit :

$$\mathbf{r}_{CoB}^* = mid\left(\bigsqcup_{k=1}^K [\mathbf{r}]_k\right).$$

4.4. Validation et comparaison avec un filtre de Kalman étendu

4.4.1. Validation en simulation

La méthode ensembliste d'estimation de pose présentée dans ce chapitre a été d'abord testée en simulation. Nous avons simulé le mouvement d'un drone en utilisant le modèle

4. Caractérisation d'un domaine de localisation pour un drone

cinématique défini à la section 4.2. Six amers ont été choisis, situés à l'avant de la caméra (garantissant leur visibilité). Les mesures dans l'image ont été générées en projetant ces amers (points 3D) dans l'image grâce à l'équation 4.5. L'algorithme SIVIA a ensuite été utilisé sur ce jeu de données, avec trois contracteurs différents (C_{3B} , C_{4B} et C_{5B}).

4.4.1.1. Estimation des 6DoF

Le drone initialement situé à la position $\mathbf{r}_{init} = (-4, -4, 2, 0^\circ, 0^\circ, 0^\circ)^\top$, effectue un déplacement latéral sur l'axe x de moins d'un mètre, d'un déplacement vers l'avant de 4 m (suivant l'axe y) et s'élève à une altitude de 6 m (axe z), la trajectoire de référence est représentée par la courbe en pointillés verts présents sur la figure 4.5. L'essai dure 20 s et les vitesses du robot ont également été simulées. Pour les mesures 2D dans l'image, nous avons considéré des points 2D normalisés dans le repère caméra avec une incertitude de $\epsilon_{img} = 0.01$. 6 amers virtuels dont les positions sont connues avec une incertitude de $\epsilon_{world} = 1\text{mm}$. Les bornes des erreurs de mesures de la vitesse sont définies sur $\pm 0,1$ m/s ($\epsilon_{\mathbf{v}} = 0,1$) pour les vitesses linéaires et $\pm 0,005$ rad/s ($\epsilon_{\mathbf{w}} = 0,005$) pour les vitesses de rotation.

L'algorithme SIVIA a été utilisé pour déterminer le domaine des poses faisables du drone à chaque instant de l'essai.

À l'instant $t = 0$ s, le domaine initial de l'état a été réglé à $[\mathbf{r}]_{init} = ([-\infty, +\infty], [-\infty, +\infty], [-\infty, +\infty], [-\frac{\pi}{2}, +\frac{\pi}{2}], [-\frac{\pi}{2}, +\frac{\pi}{2}], [-\pi, +\pi])^\top$.

Le domaine calculé à l'instant t_k est utilisé pour effectuer une prédiction du domaine à l'instant t_{k+1} de sorte que :

$$[\mathbf{r}](t_{k+1}) = [\mathbf{r}](t_k) + T_e \cdot \begin{pmatrix} [\mathbf{R}] \\ [\mathbf{M}] \end{pmatrix} ([\mathbf{r}](t_k)) \cdot \begin{pmatrix} [\mathbf{v}](t_k) \\ [\mathbf{w}](t_k) \end{pmatrix} \quad (4.12)$$

4. Caractérisation d'un domaine de localisation pour un drone

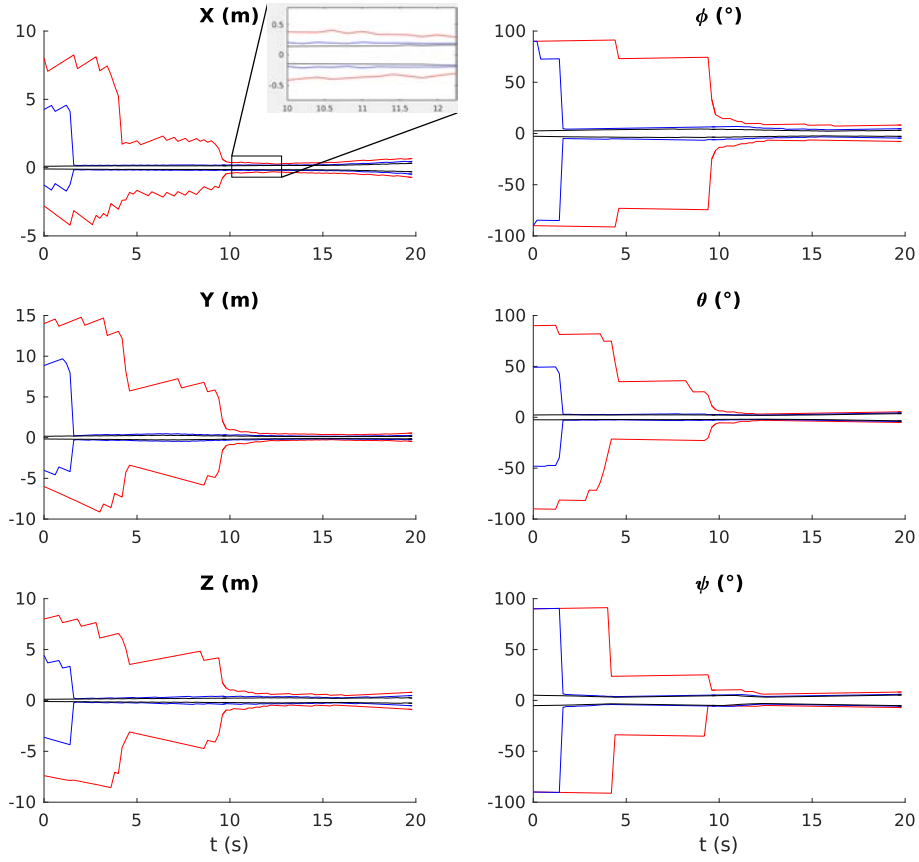


FIGURE 4.3. – Bornes inférieures et supérieures de l’erreur d’estimation par rapport à la vérité terrain, centrée sur 0. En *rouge* le résultat en utilisant le contracteur C_{3B} dans l’algorithme SIVIA. En *bleu* celui utilisant la C_{4B} et en *noir* le résultat utilisant le contracteur C_{5B} .

La figure 4.3 présente les bornes de l’erreur d’estimation pour le problème à 6DoF, \mathbb{S}_{6DoF} (équation 4.9) par rapport à la vérité terrain (centrée en zéro) pour une valeur du paramètre ϵ égale à 1 cm pour la position et à 0.57° pour l’angle.

La première observation est que, malgré la largeur arbitrairement grande du domaine initial, la méthode converge vers un domaine assez réduit dès la première estimation. Par exemple, la largeur maximale du domaine de position est de 12.29m en x , 23.65m en y et 16.04m en z à l’instant $t = 0$ s avec le contracteur C_{3B} ; ce qui montre que la $3B$ a du mal à contracter le domaine. Les résultats obtenus avec C_{4B} au même instant sont de 6.28m en x , de 13.92m en y et de 8.05m en z . Dans le cas du contracteur C_{5B} , nous avons 0.64m en x , 0.56m en y et 0.54m en z . Ces résultats montrent que la taille du domaine obtenu varie en fonction du contracteur utilisé dans l’algorithme SIVIA. En *noir* (figure 4.3), le

4. Caractérisation d'un domaine de localisation pour un drone

contracteur par découpage C_{5B} fournit un meilleur résultat que le contracteur C_{4B} qui lui à son tour est meilleur que le contracteur C_{3B} . Ce qui est bien cohérent avec le fait que, le contracteur C_{5B} a pour sous contracteur le C_{4B} élevant ainsi la performance du test de consistance.

Largeurs des boîtes englobantes des sous-pavages												
	Moyenne			Médiane			95%			max		
SIVIA	C_{3B}	C_{4B}	C_{5B}	C_{3B}	C_{4B}	C_{5B}	C_{3B}	C_{4B}	C_{5B}	C_{3B}	C_{4B}	C_{5B}
$x(m)$	3.75	0.86	0.31	1.34	0.39	0.27	11.58	5.57	0.54	12.29	6.28	0.64
$y(m)$	8.08	1.58	0.37	1.96	0.58	0.36	22.65	13.13	0.55	23.65	13.92	0.56
$z(m)$	6.12	1.19	0.39	2.06	0.66	0.42	15.74	7.55	0.51	16.04	8.05	0.54
$\phi(^{\circ})$	46.60	20.36	6.49	15.66	10.07	6.48	148.34	157.51	8.17	148.85	174.71	8.30
$\theta(^{\circ})$	41.63	13.06	4.75	10.29	5.82	4.61	163.0	96.83	6.60	163.52	97.52	7.11
$\psi(^{\circ})$	23.50	8.95	8.32	14.09	9.53	8.31	59.84	11.54	10.09	114.87	13.21	10.44

TABLE 4.1. – Largeur des boîtes englobantes des sous-pavages solutions donnés par SIVIA en utilisant les trois contracteurs.

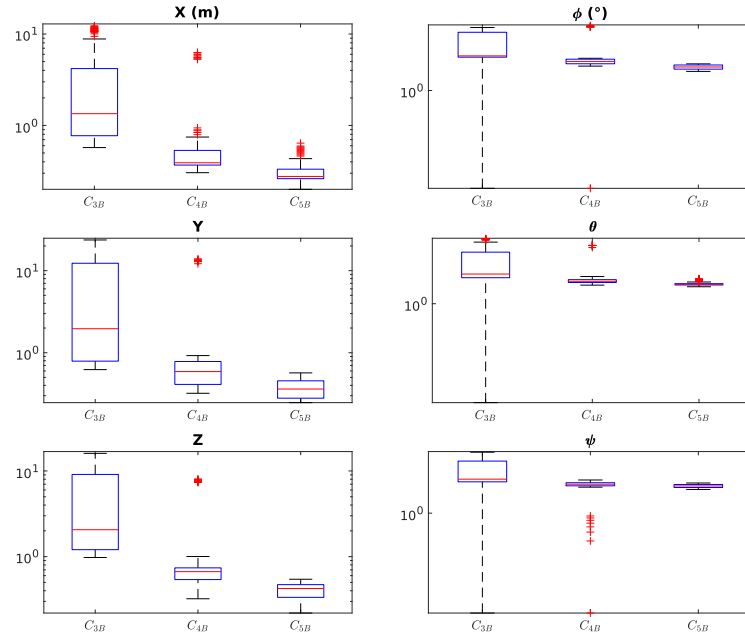


FIGURE 4.4. – Diagramme représentant les largeurs des domaines des 6 dimensions de la pose en utilisant SIVIA avec chaque contracteur.

Le tableau 4.1 donne plus de statistiques sur les tailles des domaines de pose pour les trois contracteurs.

4. Caractérisation d'un domaine de localisation pour un drone

La deuxième chose à remarquer est que la prédiction permet de raffiner les bornes de sorte que à $t \approx 10$ s, les domaines obtenus sont considérablement proches de la solution exacte comme le montre les courbes des figures 4.3 et 4.5.

En prenant le CoB comme estimée ponctuelle, il est possible de comparer le résultat donné par notre méthode ensembliste à la vérité terrain. La figure 4.5 montre les courbes de l'estimée ponctuelle par rapport à la trajectoire du drone. La moyenne de l'erreur horizontale est de 1.1m pour le contracteur C_{3B} , de 0.25m pour le contracteur C_{4B} et de 9mm pour le contracteur C_{5B} . Globalement, nous observons que les bornes d'erreurs sont meilleures dans les cas où les contracteurs C_{5B} et C_{4B} sont utilisés, ils sont tout de même plus coûteux en temps de calcul en raison de leur complexité.

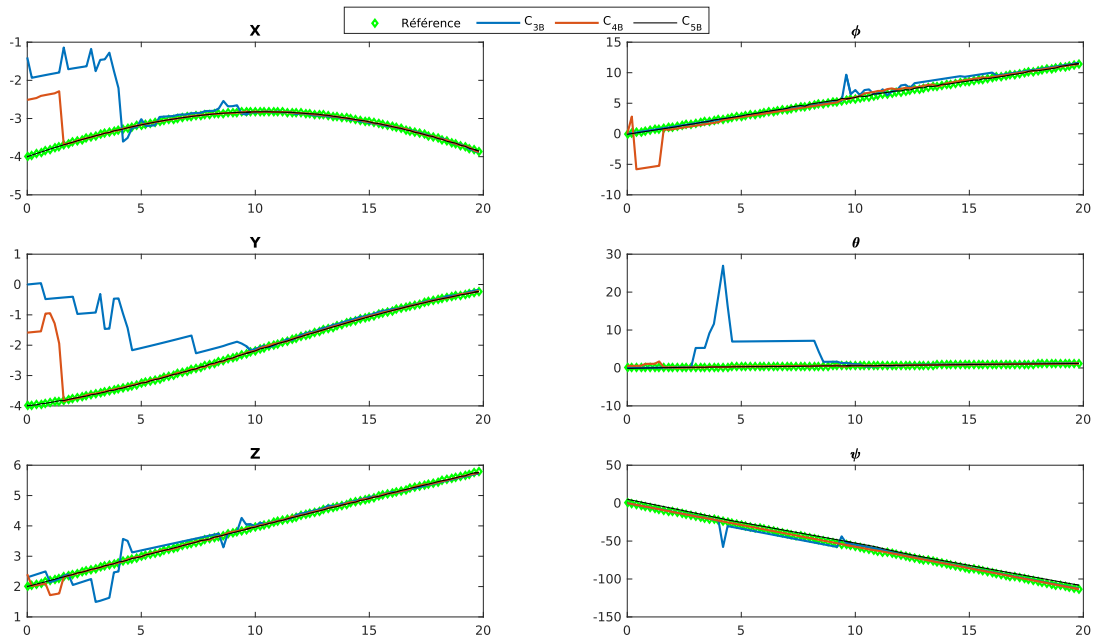


FIGURE 4.5. – Trajectoire de référence du drone et tracé du centre de la boîte englobante du domaine estimé pour chacun des 3 contracteurs. La ligne diamantée en vert représente le tracé de la vérité terrain. En bleu le résultat en utilisant le contracteur C_{3B} dans l'algorithme SIVIA. En rouge celui utilisant la C_{4B} et en noir le résultat utilisant le contracteur C_{5B} .

Le tableau 4.2 en fait une étude comparative dans le cas à 6 degrés de libertés. SIVIA avec le contracteur C_{3B} met un temps maximal de 0,59 s pour l'estimation à un instant ; ce qui est de loin inférieur au temps moyen mis lorsque les contracteurs C_{4B} (6,37 s) et C_{5B} (138 s) sont utilisés. En comparant ces trois méthodes, nous nous rendons bien compte qu'il est nécessaire de faire un compromis entre temps de calcul et précision de l'estimation. L'utilisation du contracteur C_{5B} n'est pas réaliste en raison du temps

4. Caractérisation d'un domaine de localisation pour un drone

de calcul prohibitif. Le temps de calcul de SIVIA avec contracteur C_{4B} le limite à une utilisation du type localisation initiale, avec le robot immobile.

Caméra seule		
Temps de calcul	moyenne(s)	max(s)
SIVIA+ C_{3B}	0,26	0,59
SIVIA+ C_{4B}	6,37	14,4
SIVIA+ C_{5B}	137,88	223,34

TABLE 4.2. – Temps de calcul pour l'estimation du domaine de la pose lorsque l'algorithme SIVIA utilise chacun des trois contracteurs pour la pose à 6DOF.

4.4.1.2. Estimation de 3DoF

Il est possible de mesurer 3 des 6 degrés de liberté de la pose du robot à partir de capteurs embarqués. Ceci correspond à la formulation du CSP enrichi où 3DoF sont connus avec une légère incertitude et les 3 autres sont à estimer (exprimé à l'équation 4.10 et ayant pour solution 4.11).

Dans ce cas, le domaine initial est réglé à $[\mathbf{r}]_{init} = ([-\infty, +\infty], [-\infty, +\infty], [y_{mes} \pm \epsilon_y], [\phi_{mes} \pm \epsilon_\phi], [\theta_{mes} \pm \epsilon_\theta], [-\pi, +\pi])^\top$ avec $\epsilon_y = 0.1 m$, $\epsilon_\phi = \epsilon_\theta = 0.1 \text{ rad} \approx 5^\circ$ et les autres paramètres réglés comme précédemment. L'algorithme SIVIA appliqué avec les différents contracteurs a des performances bien meilleures.

Nous observons que pour les trois contracteurs, dès l'instant $t = 0$ s, le domaine obtenu est considérablement proche de la pose de référence comme le montre la figure 4.6. Les trois courbes sont superposées et presque identiques. Toutefois étant donné que dans le cas à 3DoF l'algorithme SIVIA effectue des bisections uniquement sur 3 dimensions, le temps de calcul est bien meilleur que dans le cas à 6DoF.

SIVIA avec le contracteur C_{3B} met en moyenne 0,15 s pour l'estimation à un instant t donné, tandis que SIVIA avec les contracteurs C_{4B} et C_{5B} mettent respectivement 0,28 s et 0,73 s (voir tableau 4.3).

Caméra + Altimètre + IMU		
temps de calcul	moyenne(s)	max(s)
SIVIA+ C_{3B}	0,15	0,25
SIVIA+ C_{4B}	0,28	0,38
SIVIA+ C_{5B}	0,73	52,76

TABLE 4.3. – Temps de calcul d'une instant d'estimation de la pose à 3DOF lorsque l'algorithme SIVIA utilise chacun des trois contracteurs.

Ces résultats nous montrent que, bien que partant de domaines assez larges, l'algorithme SIVIA fournit un domaine de pose considérablement réduit (à partir de l'instant $t \approx 10$ s

4. Caractérisation d'un domaine de localisation pour un drone

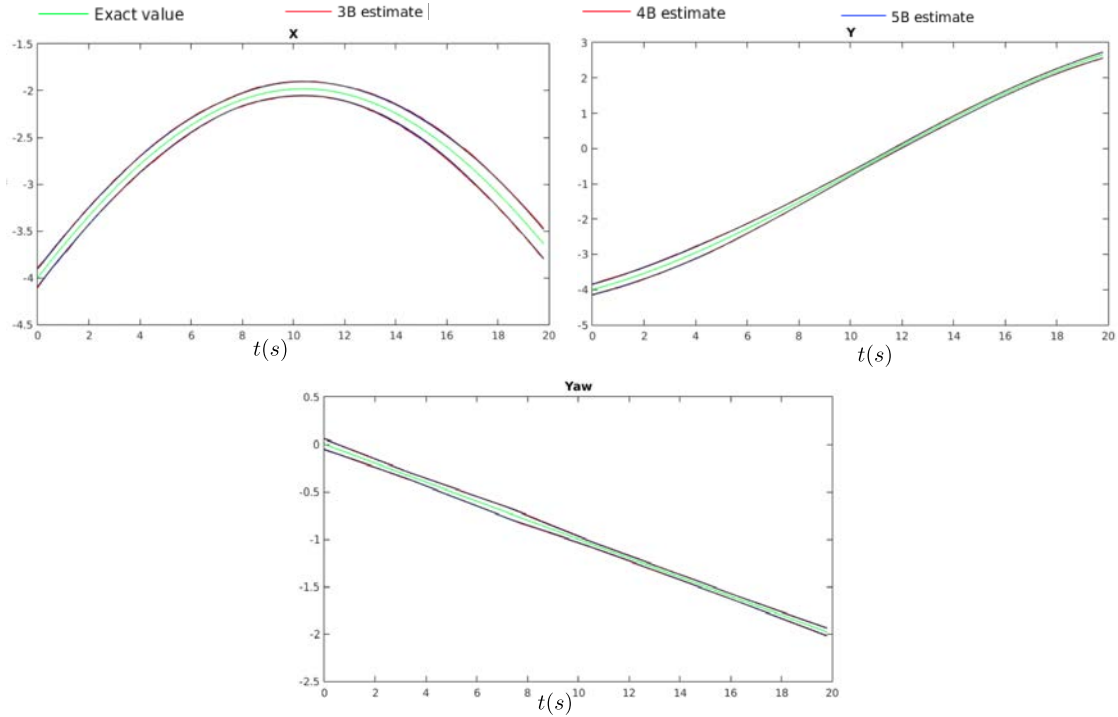


FIGURE 4.6. – Résultats d'estimation pour la pose à 3degrés de libertés pour différents contracteurs.

dans le cas d'une estimation à 6DoF. Toutefois le choix de l'algorithme est contrôlé par le nombre de dimensions de bisection influant directement sur le temps de calcul de la méthode. La fusion de données est assez directe en présence de mesures proprioceptives, qui se traduisent simplement par l'ajout de contraintes sur les variables du CSP. Ceci rend possible une meilleure estimation (domaine solution plus fin dès l'instant $t = 0$ s, voir figure 4.6) moins coûteuse en temps de calcul.

Dans la suite, nous faisons donc le choix d'utiliser la méthode SIVIA avec le contracteur C_{3B} car elle présente le meilleur rapport temps/précision dans le cas à 3DoF.

4.4.2. Validation expérimentale

Après cette phase de simulation, des essais expérimentaux ont également été menés, afin d'éprouver la méthode sur des données réelles. Cette section décrit le dispositif expérimental utilisé ainsi que les résultats ensemblistes obtenus.

4.4.2.1. Dispositif expérimental

Le dispositif expérimental utilisé est constitué :

4. Caractérisation d'un domaine de localisation pour un drone

- d'un drone de type quadrirotor, de modèle MK-Quadro de MikroKopter (image à gauche de la figure 4.7) dont l'architecture est décrite à l'annexe B. Ce quadrirotor est équipé d'une caméra embarquée de type Point Grey Flea3 pour l'acquisition d'images et d'une carte ODROID-XU4 sur lequel tourne le middleware ROS² (Robot Operating System).



FIGURE 4.7. – Système complet : Quadrirotor utilisé avec caméra (*gauche*) et Caméra de type Point Grey Flea3 (*droite*)

- d'une salle équipée d'un système de capture de mouvement Vicon (également décrit en Annexe B) permettant de suivre avec précision la pose du quadrirotor et la position des amers présents dans la salle. Ceci est visible sur l'image à droite de la figure 4.8.
- d'une station au sol, qui permet de réceptionner les images acquises par la caméra du drone et de faire le tracking dans l'image des amers observés par le drone. Tous les algorithmes de tracking utilisés sont ceux de la bibliothèque open-source *VISP*³ (MARCHAND et al. 2005).
- Six cubes, dont la position est connue dans le repère de référence, placés dans la salle pour servir d'amers. Chacun d'eux présente un carré blanc de 20cm sur une face pour le suivi visuel (voir figure 4.8). Le centre du carré est considéré comme la coordonnée géographique de l'amer représenté par ce carré.

Ainsi, le drone en vol observe les amers à partir de sa caméra à une fréquence de 30 Hz, rééchantillonnée à 7 Hz pour être compatible avec les temps de calculs présentés dans

2. <http://www.ros.org/>

3. <https://visp.inria.fr/>

4. Caractérisation d'un domaine de localisation pour un drone

le tableau 4.3. Les images obtenues sont ensuite envoyées à la station au sol. À une fréquence de 100 Hz, la position du drone est localisé par le système Vicon et accessible grâce à des *topics* ROS. Nous avons effectué plusieurs séquences de vols de différentes durées et les données (pose Vicon du drone, position des amers dans la salle, retour visuel des amers et coordonnées dans l'image des centres de boîtes issues du tracking) ont été enregistrées.

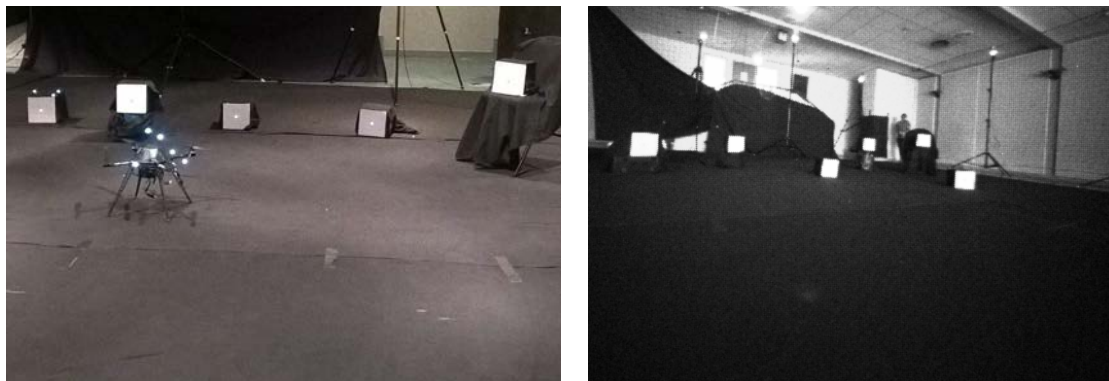


FIGURE 4.8. – *Gauche* : UAV dans la salle d'expérimentation. *Droite* : Vue interne de la caméra embarquée

Les résultats qui suivent sont ceux issues d'un vol de 77 secondes du quadrirotor. Cet essai comprend :

- un décollage,
- un déplacement latéral de 3 m et une rotation de 50° ,
- un retour au point de décollage ainsi qu'un atterrissage.

Le jeu de données obtenu comporte 545 images, ainsi que la vérité terrain de la pose du drone mesurée par le système de motion capture. Les centres des amers sont suivis à l'aide d'un algorithme de tracking *blobTracking*⁴ de *VISP* qui fournit ici des résultats plutôt bruités. L'erreur de tracking est calculée par rapport à la re-projection dans l'image des points 3D géo-référencés (point au centre des boîtes) à partir de la pose Vicon du drone (prise ici comme vérité terrain). Elle est inférieure à 4,5 px pour 98% des mesures. Ces erreurs sont tracées dans la figure 4.9. Nous remarquons également la présence de mesures aberrantes de quelques dizaines de pixels clairement visibles pendant l'essai (voir les pics sur la figure 4.9, par exemple à $t = 16$ s et $t = 20$ s). Celles-ci correspondent aux défaillances de l'algorithme de tracking des amers. De plus petites erreurs de tracking de quelques pixels sont également présentes, elles sont principalement dues à des configurations plus complexes lorsque les amers ne se distinguent pas de l'arrière-plan de la salle.

4. <http://visp-doc.inria.fr/doxygen/visp-daily/tutorial-tracking-blob.html>

4. Caractérisation d'un domaine de localisation pour un drone

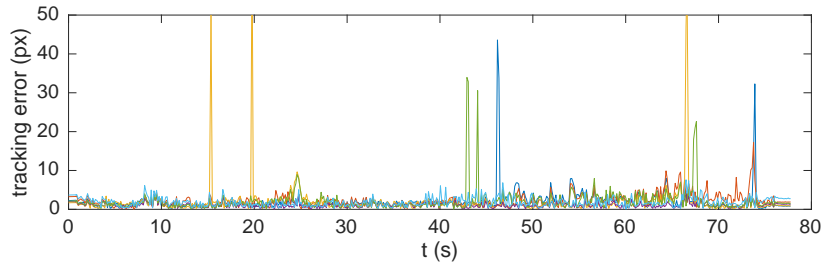


FIGURE 4.9. – Erreur de tracking

4.4.2.2. Calcul d'un domaine de la pose

Le calcul du domaine de pose est exécuté sur l'ensemble de l'essai. Nous avons réglé les bornes ϵ_{img} des erreurs dans l'image à $\pm 4,5$ pixels (ce qui correspond à un bruit très significatif dans un problème de calcul de pose). Ceci est cohérent avec 98% des mesures comme expliqué précédemment. Les bornes des erreurs de mesures de la vitesse sont définies sur $\pm 0,1$ m/s ($\epsilon_v = 0,1$) pour les vitesses linéaires et $\pm 0,005$ rad/s ($\epsilon_w \pm 0,005$) pour les vitesses de rotation. Les positions des amers géo-référencés sont connues à ± 1 mm ($\epsilon_w = 1$ mm) près (précision donnée par le Vicon).

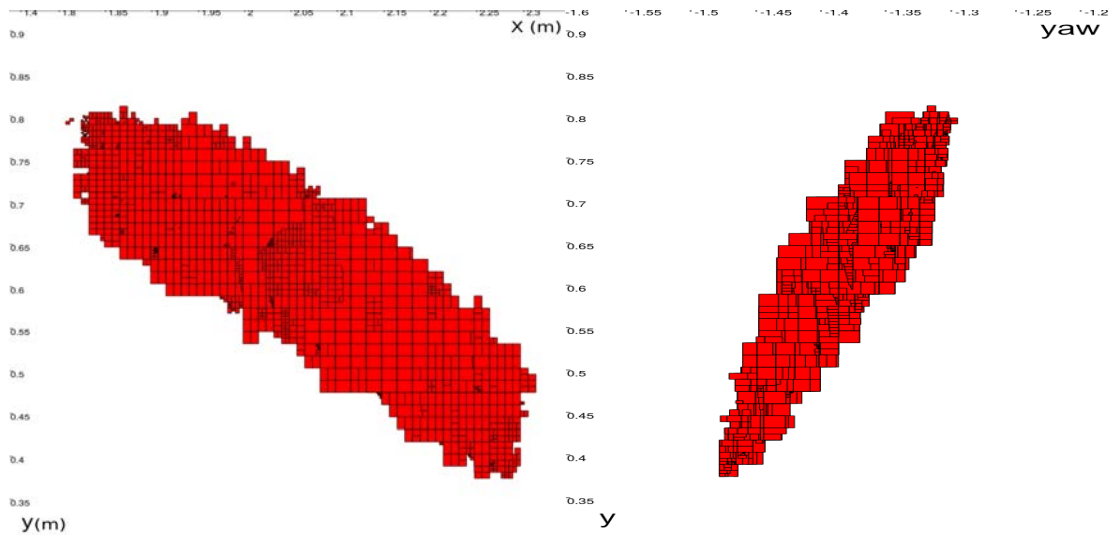


FIGURE 4.10. – Vue du sous-pavage de la pose calculée à $t = 0$ s. *Gauche* : projection sur l'axe x, y . *Droite* : projection sur l'axe ψ, y .

Le sous-pavage du domaine d'incertitude de pose calculé à partir des mesures d'image à l'instant $t = 0$ s est représenté à la figure 4.10. La figure présente les projections sur les plans (x, y) et (ψ, y) du domaine à 6 dimensions de la pose $(x, y, z, \phi, \theta, \psi)$. Les composantes z, ϕ, θ étant connues avec une faible incertitude, notre attention sera principalement portée sur les trois autres. En effet, l'information a priori que nous avons de

4. Caractérisation d'un domaine de localisation pour un drone

l'attitude et de l'altitude agit comme une contrainte sur les composantes x, y et ψ comme explicité dans l'équation 4.10, \mathcal{H}_{3DoF} .

La figure 4.11 présente les résultats de l'estimation sur l'ensemble de l'essai, pour les composantes x, y et ψ , à chaque instant d'acquisition d'une mesure image. Les valeurs des erreurs sont relatives à la vérité terrain fournie par le système de motion capture, ce qui signifie que 0 équivaut à la vérité terrain. Les lignes bleues sont les bornes supérieure et inférieure du domaine. L'aspect en rampes des bornes du lacet (entre $t = 30$ s et $t = 40$ s) correspond à des instants où la méthode proposée n'est pas capable de réduire le domaine initial du lacet. Le domaine est donc seulement prédit à partir du domaine précédent et de la vitesse angulaire. On peut observer que le domaine d'incertitude calculé pour x se situe dans un rayon de ± 30 cm, pour y il est égale à approximativement ± 20 cm et le rayon du domaine de lacet ψ correspond à $\pm 5^\circ$.

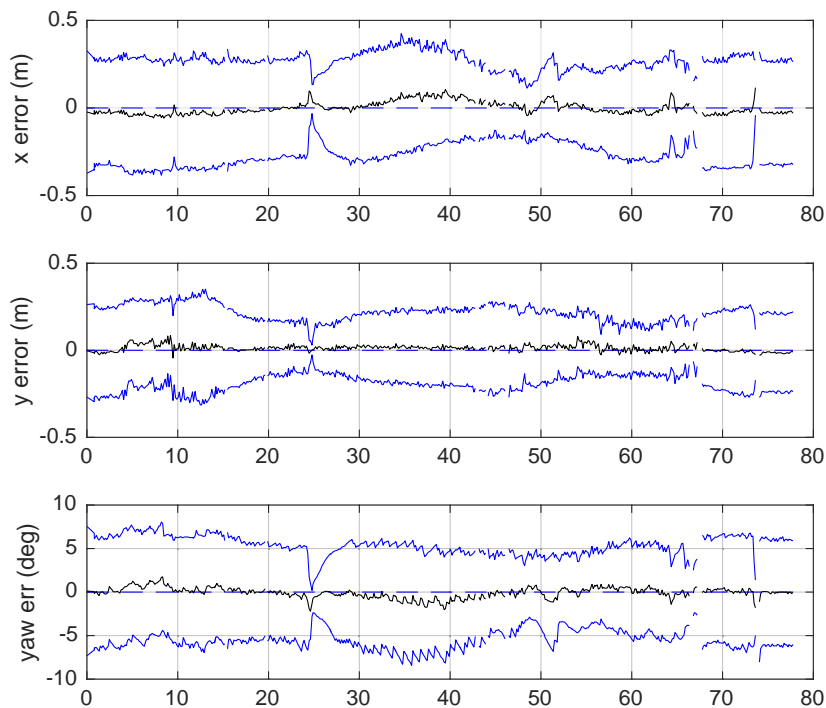


FIGURE 4.11. – *Bleu* : bornes d'erreur des composantes $[x], [y]$ et $[\psi]$ du domaine de pose calculé. Celles-ci sont par rapport à la vérité terrain, c'est-à-dire que la vérité terrain est à 0. *Noir* : erreur d'estimation, en utilisant le centre du domaine calculé

4.4.2.3. Détection d'inconsistances

Un problème de satisfaction de contrainte peut ne pas avoir de solution. Dans le cas de l'estimation de la pose, cela se produit lorsque les mesures sont incohérentes, c'est-à-dire

4. Caractérisation d'un domaine de localisation pour un drone

qu'il n'y a pas de pose possible dans le domaine initial à partir de laquelle l'ensemble actuel de mesures pourrait être obtenu. Le résultat de SIVIA est alors un ensemble vide. Une telle situation résulte soit des erreurs de modélisation, de la sous-estimation des bornes des erreurs de mesure, soit de la présence de mesures aberrantes. En situation réelle, il existe toujours un compromis lors du choix des bornes d'erreur, car il n'est généralement pas possible de fournir une borne d'erreur qui couvrira même des événements rares.

Dans l'approche que nous avons développée, si l'inversion ensembliste renvoie un domaine de pose vide, un indicateur « *erreur détectée* » (*fault detected*) est déclenché et l'estimation de pose suivante redémarre à partir du domaine initial $[\mathbf{r}_0]$. En effet, il est plus sûr de relancer complètement l'estimation après la détection d'un défaut pour éviter de propager des erreurs éventuelles non détectées auparavant dans la prédiction.

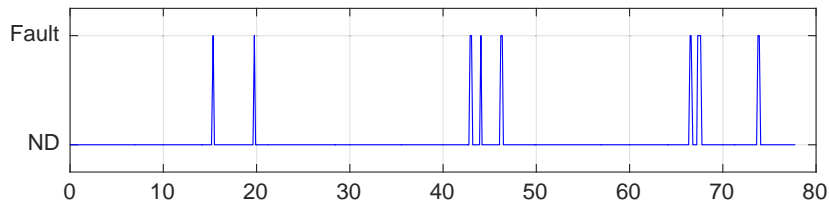


FIGURE 4.12. – Résultats de détection de fautes

Deux points sont à observer dans ces résultats. Des instants vides sont remarquables sur les courbes de la figure 4.11, ils correspondent aux ensembles solutions vides retournés par l'algorithme. Une solution vide indique une inconsistance entre les mesures, les bornes d'erreurs ou le modèle. Ces solutions vides se produisent lorsque l'erreur de tracking dans l'image est si importante que la position réelle de l'amer dans l'image n'est plus dans le domaine $[\bar{\mathbf{x}}_i]$. Les instants où les fautes sont détectées correspondent aux traits bleus sur la figure 4.11 (matérialisant les erreurs de tracking importantes sur la figure 4.11). Sur l'ensemble de l'essai, 2,6% des instants correspondent aux moments où des fautes ont été détectées.

Le second point concerne les inconsistances non détectées (*missed detection*). Ces détections manquées se produisent lorsque les erreurs de mesure sont supérieures aux bornes d'erreurs spécifiées, mais insuffisantes pour entraîner une inconsistance avec les autres mesures. Cela peut conduire à des informations trompeuses, car le domaine d'incertitude calculé n'est plus garanti de contenir la véritable pose du robot dans le cas de détection manquée. Nous avons effectué une étude pour déterminer à partir de quel seuil le domaine retourné ne contient plus l'ensemble solution et à partir de quel seuil la faute peut être détectée.

La figure 4.13 présente les bornes du domaine lorsqu'un biais croissant est appliqué à chaque intervalle de toutes les mesures dans l'image. À chaque instant, un biais δ est rajouté à $[\bar{\mathbf{x}}_i]$. Ainsi après plusieurs itérations, le domaine de chaque mesure dans l'image est $[\bar{\mathbf{x}}_i](t) = [\bar{\mathbf{x}}_i](t) \pm k \cdot \delta$.

4. Caractérisation d'un domaine de localisation pour un drone

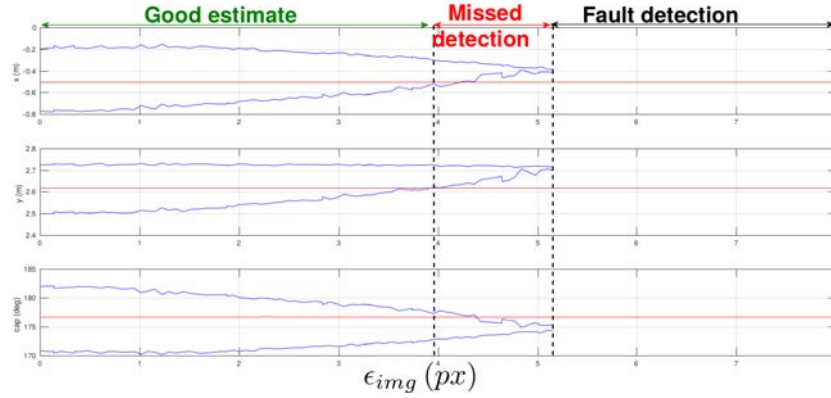


FIGURE 4.13. – Intervalle englobant du domaine de pose par rapport à la vérité terrain pour un biais croissant sur les mesures dans l'image. $\epsilon_w = 1\text{cm}$, ϵ_{img} (variant)

Sur la figure 4.13, la zone de *missed detection* correspond à une valeur de biais comprise entre 3 pixels et 4 pixels lorsque la position de l'amer est parfaitement connue et $\epsilon_{img} = 0.25\text{ px}$ (entre 5.8 et 7.6 pixels pour une incertitude de 1 cm sur les positions des amers).

Valeur de biais	$\epsilon_{world} = 0$	$\epsilon_{world} = 1\text{cm}$
<i>Missed detection</i>	3	5.8
<i>Fault detection</i>	4.1	7.6

TABLE 4.4. – Valeurs en pixel des biais de détection de fautes pour des valeurs différentes de l'incertitude sur la position des amers. $\epsilon_{img} = 0.25\text{ px}$.

Pour un biais supérieur à 4.1 pixels (voir tableau 4.4), les inconsistances peuvent être détectées car le biais est suffisamment grand pour que l'ensemble solution soit vide.

Ces résultats montrent qu'à partir de l'inversion ensembliste, il est possible de détecter les mesures aberrantes, grâce au seuil de *fault detection*, et de les extraire du jeu de données afin de fournir des bornes de mesures garanties pour d'autres méthodes.

En cas de faute, il est également possible de caractériser un domaine solution qui prend en compte la présence de mesures aberrantes. Ceci est possible en effectuant l'inversion ensembliste q -relaxée à partir des mesures dans l'image. L'annexe C présente quelques résultats de l'application de cette méthode.

4.4.3. Comparaison à un filtre de Kalman étendu

Il est parfois nécessaire de fournir une estimée ponctuelle de la pose, dans de tels cas, le centre de boîte peut être utilisé (tel que expliqué à la section 4.3.6). Pour analyser les performances de la méthode d'estimation ensembliste proposée, nous avons comparé le CoB aux résultats d'un filtre de Kalman étendu (EKF).

4. Caractérisation d'un domaine de localisation pour un drone

Le modèle d'évolution utilisé pour implémenter l'EKF est donné par :

$$\dot{\mathbf{r}} = f(\mathbf{r}, \mathbf{u}) \quad (4.13)$$

où $\mathbf{u} = (\mathbf{v}, \mathbf{w})^\top$ est le vecteur des vitesses linéaires et des vitesses de rotation du drone définies à la section 4.2.2.

$$f(\mathbf{r}, \mathbf{u}) = \begin{pmatrix} \mathbf{R} \cdot \mathbf{v} \\ \mathbf{M} \cdot \mathbf{w} \end{pmatrix} \quad (4.14)$$

Le modèle d'observation, en supposant connue la matrice \mathbf{K} , est donné par :

$$g(\mathbf{r}, {}^w \mathbf{X}) = \begin{pmatrix} \mathbf{K} \Pi {}^c \mathbf{T}_w(\mathbf{r}) {}^w \mathbf{X}_1 \\ \vdots \\ \mathbf{K} \Pi {}^c \mathbf{T}_w(\mathbf{r}) {}^w \mathbf{X}_6 \end{pmatrix} \quad (4.15)$$

La courbe noire de la figure 4.11 représente le centre du domaine de pose calculé à chaque instant et le tableau 4.11 présente l'erreur de position horizontale et l'erreur de lacet en utilisant le CoB comme estimée ponctuelle. Ce dernier est comparé au résultat d'un EKF sur les mêmes données.

Les covariances d'erreurs d'observation du filtre ont été initialement réglées de manière à être cohérentes avec les écarts-types correspondant aux largeurs des bornes des intervalles de mesures. Par exemple, pour une valeur des bornes d'erreur des mesures dans l'image $\epsilon_{img} = 0.5$ px, le gain initial d'observation pour chaque image est réglé de manière à être cohérent avec l'intervalle de l'erreur donc $\pm 3\sigma \approx [-\epsilon_{img}, +\epsilon_{img}]$. Le réglage du filtre a ensuite été optimisé (« *fine tuning* ») pour le jeu de données.

Erreur	moyenne	médiane	95%	max
CoB : lacet (°)	0.45	0.34	1.26	2.21
CoB 2D horizontale (cm)	3.65	3.15	7.76	11.89
EKF : lacet (°)	0.64	0.51	1.60	5.03
EKF : 2D horizontale (cm)	5.10	4.33	11.84	28.33

TABLE 4.5. – Comparaison de notre méthode à l'EKF : erreurs absolues de la position horizontale et du lacet.

Dans 95% des cas, l'erreur de lacet de l'estimée CoB est inférieure à 1.26° tandis que celle de l'EKF est de 1.6°. L'erreur de positionnement est également significativement plus importante avec l'EKF par rapport à l'estimée CoB (11.89 cm d'erreur maximale pour le CoB contre 28.33 cm pour l'EKF). Ceci peut s'expliquer par le fait que, l'EKF implémenté n'est pas robuste à la présence de mesures aberrantes tandis que la méthode développée permet de gérer la détection des inconsistances par une ré-initialisation du domaine de la pose.

Le tableau 4.5, donne plus de détails statistiques sur les erreurs de positionnement horizontale et en lacet des deux méthodes. Afin d'être équitable, les instants où les fautes

4. Caractérisation d'un domaine de localisation pour un drone

ont été détectées ont été enlevés avant la génération de ces chiffres. Malgré cela, nous pouvons conclure que l'estimée ponctuelle de notre méthode, le CoB, est donc meilleure que l'EKF.

4.4.4. Évolution du domaine d'incertitude sur la pose en fonction de l'incertitude sur les mesures dans l'image

Nous pouvons observer l'évolution du domaine de pose en fonction des bornes d'erreurs croissantes pour les mesures dans l'image. La figure 4.14 présente la projection sur le plan (x, y) des sous-pavages donné par SIVIA pour une incertitude de mesure dans l'image ϵ_{img} variant de ± 0.25 px à ± 2 px. Nous observons que la taille du domaine s'accroît

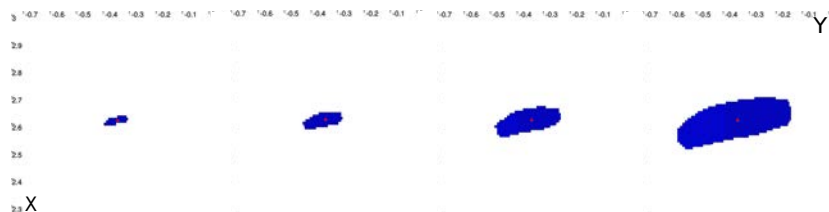


FIGURE 4.14. – Plan x, y du sous-pavage du domaine de la pose pour une valeur croissante des bornes d'erreur des mesures dans l'image : $\pm 0.25, \pm 0.5, \pm 1, \pm 2$ px

proportionnellement à la largeur des intervalles de mesures dans l'image. Le domaine de pose passe d'une largeur en x ($w(x)$) inférieure à 5 cm à une largeur supérieure à 15 cm, celle pour la composante y initialement inférieure à 10 cm devient approximativement égale à 50 cm.

4.4.5. Influence de l'incertitude de position des amers sur la taille du domaine

Il existe différentes manières de mesurer les positions des amers. La connaissance précise des positions des amers géo-référencés dans le repère global est fortement dépendante de la méthode utilisée lors de la campagne de cartographie des amers. Dans notre cas, la précision du système de capture Vicon est de ± 1 mm.

La méthode proposée a été testée avec différentes valeurs d'incertitudes de position des amers 3D, ± 1 mm, ± 1 cm, ± 5 cm et ± 10 cm.

Les moyennes des erreurs de position horizontale et de lacet obtenues (deux premières lignes), ainsi que la largeur moyenne du domaine (trois dernières lignes) des composantes x, y et ψ sont indiquées dans le tableau 5. L'erreur moyenne de position horizontale est de 3.65cm lorsque $\epsilon_w = 1$ mm et de 4.43cm pour $\epsilon_w = 10$ cm. La différence d'erreur est remarquable tandis que celle de l'angle n'est pas très influencée en ce qui concerne l'erreur moyenne en prenant le CoB comme estimée.

4. Caractérisation d'un domaine de localisation pour un drone

TABLE 4.6. – Erreur moyenne et largeur du domaine par rapport à l'incertitude sur la position des amers géo-référencés.

Incertitude	± 1 mm	± 1 cm	± 5 cm	± 10 cm
moyenne erreur 2D (cm)	3.65	3.75	4.14	4.43
moyenne erreur lacet ($^\circ$)	0.45	0.45	0.46	0.50
largeur domaine en x (cm)	54.2	55.9	59.8	61.8
largeur domaine en y (cm)	40.8	43.0	47.6	49.8
largeur domaine ψ ($^\circ$)	10.9	11.3	12.1	12.6

Toutefois, les largeurs de domaines quant-à elles sont remarquables; nous avons par exemple, une augmentation de 1.7° lorsqu'on passe de $\epsilon_w = 1\text{mm}$ à $\epsilon_w = 10\text{cm}$.

Ces résultats montrent que l'augmentation de l'incertitude sur les positions des amers influence la taille du domaine de pose calculé de manière significative. Ceci conduit à une grande zone indéterminée et ainsi entraîne de nombreuses bisections supplémentaires. Nous allons donc au chapitre 5, étudier ce problème et fournir une solution pour le résoudre.

4.5. Conclusion

Ce chapitre a permis de présenter la méthode ensembliste par intervalles que nous avons implémentée dans cette thèse afin de caractériser un domaine de localisation d'un drone. À partir de l'observation d'amers de positions connues, l'approche proposée calcule un domaine contenant la pose réelle du robot, en supposant des mesures dans l'image à erreurs bornées. Nous nous appuyons sur des techniques de l'analyse par intervalles et de propagation de contraintes pour propager rigoureusement les erreurs à travers un modèle d'observation non-linéaire. Les informations d'attitude des capteurs embarqués sont fusionnées avec les observations dans l'image pour réduire le domaine d'incertitude de pose et la complexité des calculs.

Les résultats obtenus en simulation montrent qu'il est possible, partant d'un domaine arbitrairement grand pour la pose, d'arriver à un domaine pour lequel le centre de boîte englobante (CoB) fournit une estimation ponctuelle précise de la pose. Les résultats expérimentaux quant-à eux ont permis de confirmer les performances de notre approche sur des données réelles. Aussi, un système de détection de fautes a été implémenté dans la boucle d'estimation afin de faire face aux inconsistances dues aux mesures aberrantes (qui ne respectent pas le modèle d'erreur borné). La détection des erreurs permet de gérer les erreurs de tracking dans l'image sans renvoyer un domaine de pose erroné.

Une comparaison à l'EKF a également été menée, montrant que le centre des domaines calculés est une meilleure estimation ponctuelle de la pose que l'EKF, avec une erreur de position horizontale moyenne inférieure à 4 cm sur les données expérimentales.

4. Caractérisation d'un domaine de localisation pour un drone

La présence d'incertitudes sur la position des amers entraîne de nombreuses boîtes indéterminées dans la solution du problème ensembliste. Le chapitre suivant est consacré à la prise en compte de ces incertitudes.

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

5.1. Introduction

Nous avons, au chapitre précédent, considéré l'estimation de pose en présence d'erreurs de mesures bornées dans l'image et des amers parfaitement localisés. En réalité, les positions géographiques des amers ne sont pas connues précisément, la précision de la mesure étant étroitement liée au système d'acquisition. Cette incertitude sur les mesures de position des amers est source de pessimisme. Le problème ainsi obtenu est un problème d'inversion ensembliste quantifié tel que présenté dans JAULIN et al. 2010.

Le but de ce chapitre est de résoudre ce problème d'inversion ensembliste quantifié, tout en réduisant le pessimisme induit par la présence d'incertitude sur les positions des amers. Pour ce faire, nous considérons le problème de localisation dans le plan. Cela correspond au cas où le drone et les amers sont situés à la même altitude comme illustré à la figure 5.1. Il s'agira donc de calculer l'ensemble des poses (position et orientation) $\mathbf{r} = (x, y, \theta)^\top$ du robot telles qu'il existe une position d'un amer 2D les rendant compatibles avec les mesures d'erreurs bornées dans l'image (ensemble dans lequel nous sommes sûrs que le robot est situé (SPLETZER et al. 2003)).

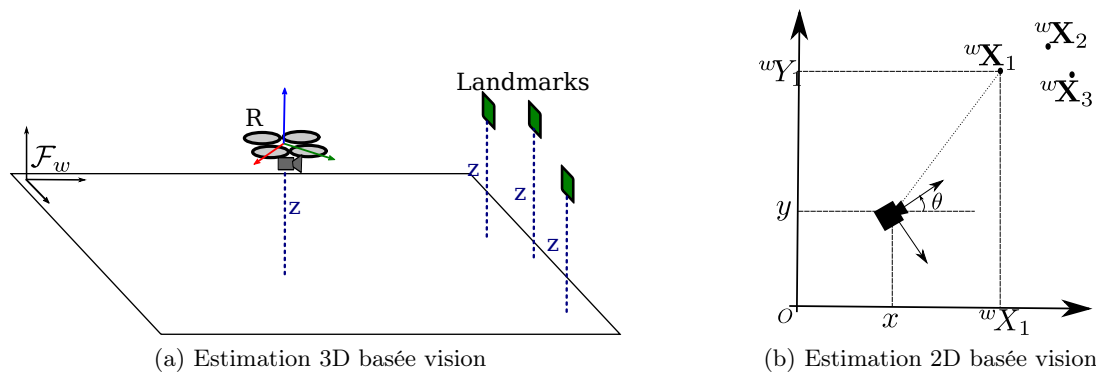


FIGURE 5.1. – Localisation dans le plan. Cas particulier où le drone et les amers sont situés à la même altitude ${}^wZ = z$.

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

Ce chapitre est organisé comme suit : la section 5.2 présente la solution classique du problème d'estimation en 2D sans et avec incertitude sur les amers afin de mieux observer l'influence de cette incertitude sur la caractérisation du domaine de pose. Par la suite, après une brève étude bibliographique sur la résolution des problèmes quantifiés, nous présentons la réécriture de notre problème sous une forme non-quantifiée. Nous terminons ce chapitre en présentant quelques résultats en simulation à la section 5.4.

5.2. Problème d'inversion ensembliste quantifié

5.2.1. Problème d'inversion ensembliste 2D

Considérons le problème tel que présenté à la figure 5.1b. Le robot connaît la position de chaque amer ${}^w\mathbf{X} = ({}^wX, {}^wY)^\top$ dans un repère global de référence et la mesure de sa projection u dans l'image représentée en coordonnées normalisées, en supposant connus les paramètres intrinsèques de la caméra (voir figure 5.2a). La projection perspective d'un point du repère global dans le repère de la caméra est donnée par :

$$\begin{cases} u &= f(\mathbf{r}, {}^w\mathbf{X}) = \frac{1}{z_c} (({}^wX - x) \cos \theta - ({}^wY - y) \sin \theta) \\ {}^cZ &= ({}^wX - x) \sin \theta + ({}^wY - y) \cos \theta \end{cases} \quad (5.1)$$

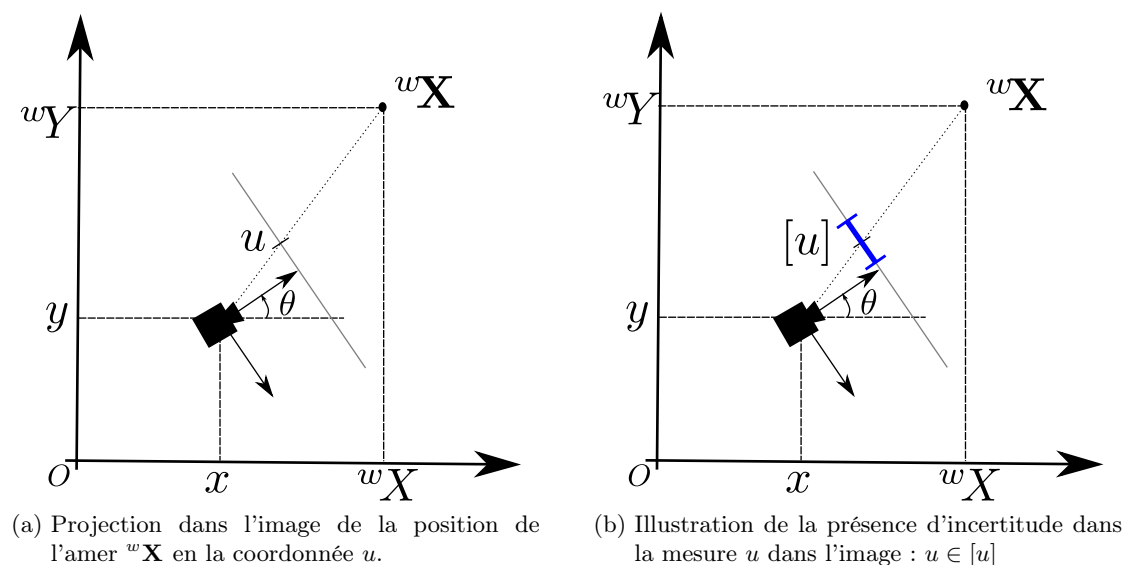


FIGURE 5.2. – Robot mobile sur lequel est montée une caméra fixe observant un amer incertain.

Sous hypothèse de mesures à erreurs bornées, l'incertitude sur la mesure dans l'image est représentée par l'intervalle $[u]$, visible en bleu dans la figure 5.2b. Déterminer la pose revient donc à calculer la solution d'un problème d'inversion ensembliste qui équivaut

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

à calculer la préimage de l'ensemble des observations dans l'image. Le domaine de pose recherché est l'ensemble contenant toutes les poses compatibles avec les mesures à erreurs bornées $[u]$ et les positions des amers parfaitement connues. Il est décrit comme suit :

$$\mathbb{S} = \{\mathbf{r} \in \mathbb{R}^2 \times [0, 2\pi] \mid f(\mathbf{r}, {}^w\mathbf{X}) \in [u], {}^cZ > 0\} \quad (5.2)$$

où ${}^cZ > 0$ exprime le fait que la caméra regarde vers l'avant, donc tout point observé ne peut être situé derrière le centre optique de la caméra.

5.2.1.1. Re-formulation de l'ensemble solution

L'expression de la fonction f de l'équation 5.2 n'étant pas bien posée lorsque cZ tend vers 0, les contraintes $f(\mathbf{r}, {}^w\mathbf{X}) \in [u]$, ${}^cZ > 0$ peuvent être réécrites comme suit :

$$\begin{cases} \frac{{}^wX-x}{({}^wX-x)\sin\theta+({}^wY-y)\cos\theta} \frac{({}^wX-x)\cos\theta-({}^wY-y)\sin\theta}{{}^wX-x} \in [u] \\ ({}^wX-x)\sin\theta+({}^wY-y)\cos\theta > 0 \end{cases} \quad (5.3)$$

$$\iff \begin{cases} \underline{u} \leq \frac{({}^wX-x)\cos\theta-({}^wY-y)\sin\theta}{{}^wX-x} \leq \bar{u} \\ ({}^wX-x)\sin\theta+({}^wY-y)\cos\theta > 0 \end{cases} \quad (5.4)$$

$$\iff \begin{cases} ({}^wX-x)\cos\theta-({}^wY-y)\sin\theta-\underline{u}({}^wX-x)\sin\theta+({}^wY-y)\cos\theta \geq 0 \\ ({}^wX-x)\cos\theta-({}^wY-y)\sin\theta-\bar{u}({}^wX-x)\sin\theta+({}^wY-y)\cos\theta \leq 0 \\ ({}^wX-x)\sin\theta+({}^wY-y)\cos\theta > 0 \end{cases} \quad (5.5)$$

$$\iff \begin{cases} ({}^wX-x)(\cos\theta-\underline{u}\sin\theta)-({}^wY-y)(\sin\theta+\underline{u}\cos\theta) \geq 0 \\ ({}^wX-x)(\cos\theta-\bar{u}\sin\theta)-({}^wY-y)(\sin\theta+\bar{u}\cos\theta) \leq 0 \\ ({}^wX-x)\sin\theta+({}^wY-y)\cos\theta > 0 \end{cases} \quad (5.6)$$

Géométriquement, lorsque θ est connu, ceci est interprété comme l'ensemble des points appartenant au secteur ayant pour sommet le point ${}^w\mathbf{X}$ et situé entre deux droites g_1 et g_2 . Lorsque θ est inconnu, les contraintes définissent l'espace entre deux surfaces. Ceci est illustré par la figure 5.3 et représenté par les inégalités suivantes :

$$\begin{cases} C_{g_1} : ({}^wX-x)(\cos\theta-\underline{u}\sin\theta)-({}^wY-y)(\sin\theta+\underline{u}\cos\theta) \geq 0 \\ C_{g_2} : ({}^wX-x)(\cos\theta-\bar{u}\sin\theta)-({}^wY-y)(\sin\theta+\bar{u}\cos\theta) \leq 0 \end{cases} \quad (5.7)$$

Un point est donc solution si il vérifie les contraintes C_{g_1} , C_{g_2} et ${}^cZ > 0$.

L'ensemble solution \mathbb{S} est donc reformulé comme suit :

$$\mathbb{S} = \{\mathbf{r} \in \mathbb{R}^2 \times [0, 2\pi] \mid C_{g_1}(\mathbf{r}, {}^w\mathbf{X}, \underline{u}), C_{g_2}(\mathbf{r}, {}^w\mathbf{X}, \bar{u}), {}^cZ > 0\} \quad (5.8)$$

Pour résoudre ce problème, nous utilisons l'algorithme SIVIA tel que vu au chapitre précédent. Le résultat de l'algorithme est un sous-pavage permettant de caractériser un intérieur (s'il existe) et un extérieur de l'ensemble solution \mathbb{S} .

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

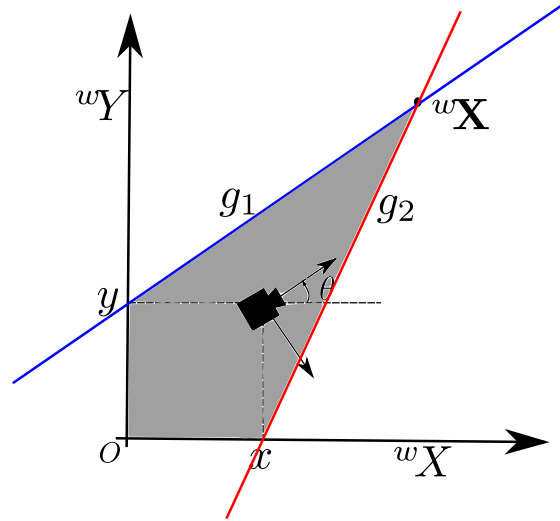


FIGURE 5.3. – Robot mobile sur lequel est monté une caméra fixe observant un amer incertain.

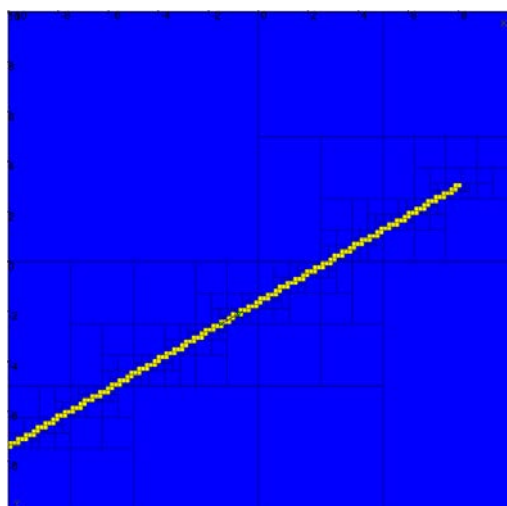
5.2.1.2. Résultats du problème d'inversion avec $N=1$ amer

Nous avons simulé la pose d'un robot situé à $\mathbf{r} = (-1, 5; -2, 5; \frac{\pi}{6})^\top$, capable d'observer un amer situé à ${}^w\mathbf{X}_1 = (8; 3)^\top$ dont la projection sur le plan image donne le point $u = 0,57$ en coordonnée normalisée. L'algorithme SIVIA a été employé pour calculer une valeur de l'ensemble solution \mathbb{S} pour différentes bornes d'erreurs de la mesure u dans deux cas de figure : *angle θ parfaitement connu* (figures 5.4a et 5.4c) et *angle θ incertain* (figures 5.4b et 5.4d). Le seuil ϵ représentant le plus petite taille des boites pour laquelle la bissection doit être arrêtée a été fixé à $\epsilon = 0,3$ m.

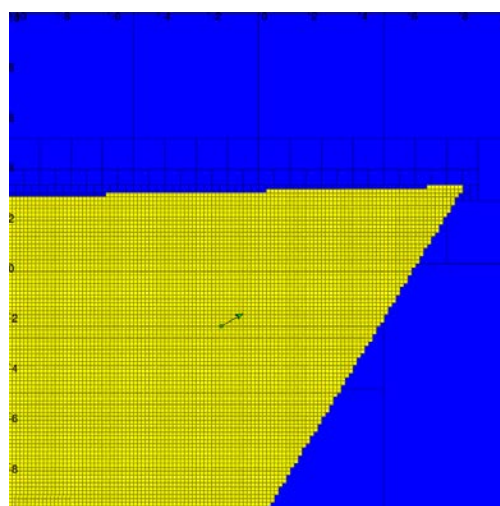
Lorsqu'aucune incertitude n'est présente sur la mesure bornée u ($\epsilon_u = 0$), il est impossible de caractériser un intérieur de l'ensemble solution \mathbb{S} car c'est une demi-droite. Ceci conduit à ne caractériser qu'un domaine frontière, représenté par la zone indéterminée visible sur les figures 5.4a et 5.4b. Dans le cas où l'angle est inconnu $[\theta] = \theta_{ref} + [-\epsilon_\theta, \epsilon_\theta] = [-\frac{\pi}{3}, \frac{\pi}{3}]$, l'ensemble solution est un secteur de sommet l'amer ${}^w\mathbf{X}$, ceci correspond à l'ensemble des boites jaunes de la figure 5.4b, sa taille est de 18789 boites jaunes contre 14182 boites bleues. Le cas où l'angle est connu $\epsilon_\theta = 0$ représente une tranche de l'ensemble solution précédent et est illustré à la figure 5.4a (taille en nombre de boites jaunes 183 et 303 boites bleues).

Dans le cas où l'erreur de mesure dans l'image n'est pas nulle $\epsilon_u \neq 0$, il est possible de caractériser un intérieur de l'ensemble solution \mathbb{S} . Par exemple, les figures 5.4c et 5.4d présentent les résultats pour les différentes valeurs de ϵ_θ lorsque $\epsilon_u = 0.1$. Dans ce cas, le volume de boites rouges est de $19,84\text{m}^2\text{rad}$ lorsque θ n'est pas connu parfaitement. Le nombre de boites jaunes dans ce cas est 100 fois plus grand que lorsque l'angle est connu ; ceci est dû au pessimisme dans la fonction d'inclusion naturelle des fonctions constituant les contraintes de notre problème.

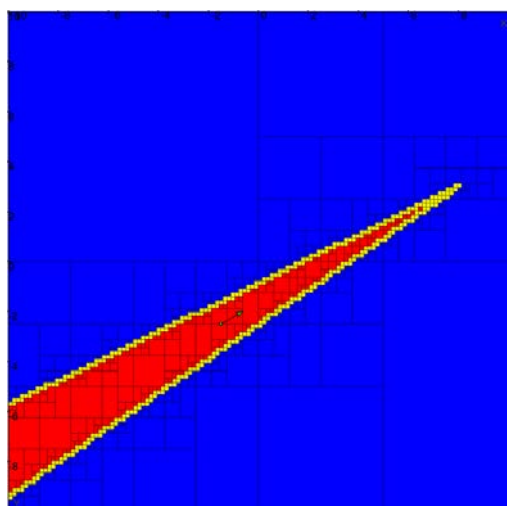
5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs



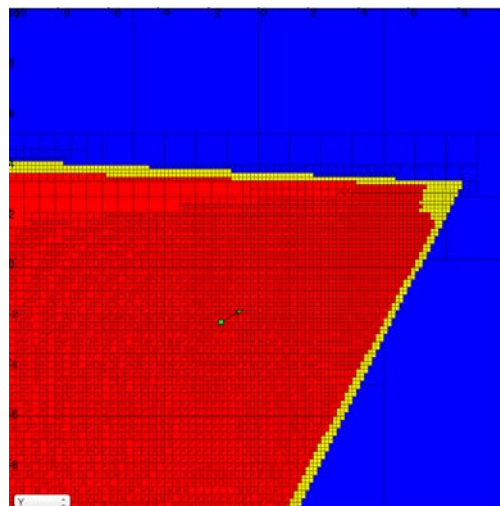
(a) Sous-pavage solution lorsque $\epsilon_u = 0$ et l'angle θ est parfaitement connu ($\epsilon_\theta = 0$)



(b) Sous-pavage solution lorsque $\epsilon_u = 0$ et l'angle θ a une incertitude $\epsilon_\theta = \frac{\pi}{6}$



(c) Sous-pavage solution lorsque $\epsilon_u = 0.1$ et l'angle θ est parfaitement connu ($\epsilon_\theta = 0$)

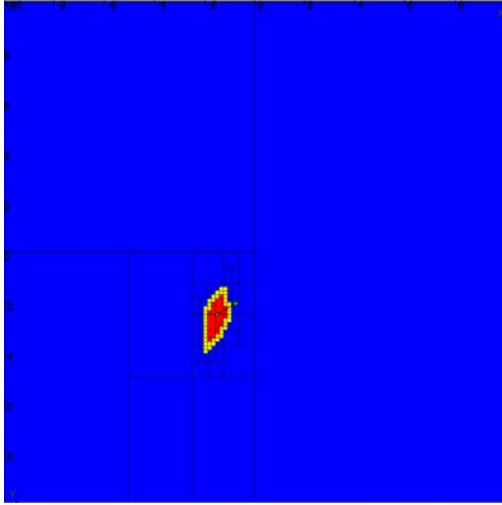


(d) Sous-pavage solution lorsque $\epsilon_u = 0.1$ et l'angle θ a une incertitude $\epsilon_\theta = \frac{\pi}{6}$

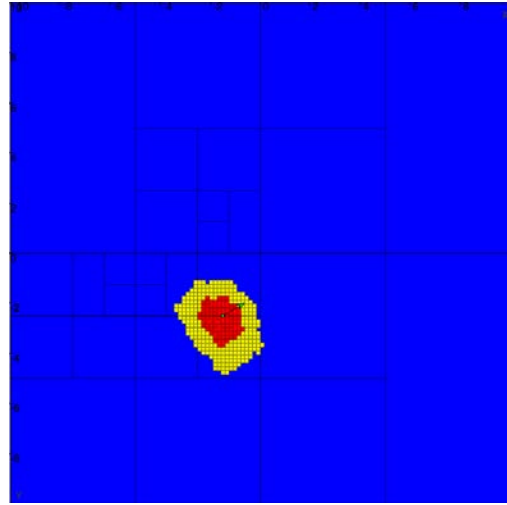
FIGURE 5.4. – Projection sur le plan (x, y) des sous-pavages de l'ensemble solution \mathbb{S} pour différentes valeurs de l'erreur de mesure dans l'image pour des cas de figure où l'angle θ est parfaitement connu (à gauche) ou pas (à droite). Un seul amer de position parfaitement connue est observé. En *bleu*, l'ensemble des boîtes qui ne sont pas solution. En *rouge* l'intérieur de l'ensemble solution $\underline{\mathbb{S}}$. Et en *jaune*, la zone indéterminée représentant la frontière de l'ensemble $\Delta\mathbb{S}$.

5.2.1.3. Résultats du problème d'inversion avec N=3 amers

Lorsque le robot est capable d'observer plusieurs amers ($N = 3$), l'ensemble solution est l'intersection des ensembles solutions correspondant à chaque amer. Deux amers parfaitement localisés (${}^w\mathbf{X}_2 = (1; 1)$ et ${}^w\mathbf{X}_3 = (-2; 4.5)^\top$) et leurs projections dans l'image ($u_2 = -0.67$ et $u_3 = 0.57$) ont été rajoutés à la simulation précédente. La figure 5.5 présente les résultats de l'algorithme SIVIA lorsque le robot observe trois amers. La figure 5.5a est une tranche du résultat de la figure 5.5b lorsque l'angle θ est connu. La présence de plusieurs observations permet de raffiner le résultat et la zone en jaune épaisse de la figure 5.5b est due au pessimisme de la fonction d'inclusion naturelle.



(a) Sous pavage de l'ensemble solution lorsque $\epsilon_u = 0.1$ et le cap parfaitement connu ($\epsilon_\theta = 0$).



(b) Sous-pavage de l'ensemble solution lorsque $\epsilon_u = 0.1$ et le cap connu avec une incertitude $\epsilon_\theta = \frac{\pi}{6}$.

FIGURE 5.5. – Projection sur le plan (x, y) des sous-pavages de l'ensemble solution \mathbb{S} pour différentes valeurs de l'erreur de mesure dans l'image pour des cas de figure où le cap θ est connu ou pas en présence de trois amers dont les positions sont parfaitement connues. En *bleu*, l'ensemble des boites qui ne sont pas solution. En *rouge* l'intérieur de l'ensemble solution $\underline{\mathbb{S}}$. Et en *jaune*, la zone indéterminée représentant la frontière de l'ensemble $\Delta\mathbb{S}$.

Le tableau 5.1 explicite clairement le nombre de boites et les volumes des sous-pavages pour les cas à angle θ connu et non connu. Pour une incertitude sur les points images $\epsilon_u = 0.1$ et pour le paramètre de bisection $\epsilon = 0.3$, nous avons observé que la caractérisation de l'intérieur de l'ensemble solution $\underline{\mathbb{S}}$ est possible.

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

	Nombre de boîtes			Volumes (m ² rad)	
	Rouge	Jaune	Total	Rouge	Jaune
$\epsilon_\theta = 0$	14	43	106	0.9	1.04
$\epsilon_\theta = \frac{\pi}{6}$	136	1341	2645	0.1	0.51

TABLE 5.1. – Statistiques des tailles de sous-pavages pour trois amers observés et différentes valeurs d'angle.

5.2.2. Problème d'inversion ensembliste avec incertitude sur la position des amers

En situation réelle, la position des amers géo-localisés est connue avec une incertitude (voir figure 5.6). Nous supposons dans la suite que cette incertitude est bornée. Ainsi, nommons ϵ_w l'erreur sur la position des amers, chaque amer sera donc représenté par une boîte $[{}^w\mathbf{X}]$ tel que $[{}^w\mathbf{X}] = {}^w\mathbf{X} + [-\epsilon_w, \epsilon_w]$.

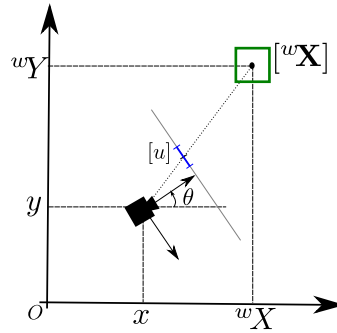


FIGURE 5.6. – Robot mobile sur lequel est montée une caméra fixe observant un amer de position incertaine.

Dans ce cas, il s'agit de calculer la zone comprise entre deux surfaces épaisses. L'ensemble solution peut être représenté comme suit :

$$\mathbb{S}_{\text{thick}} = \{\mathbf{r} \in \mathbb{R}^2 \times [0, 2\pi] \mid C_{g_1}(\mathbf{r}, [{}^w\mathbf{X}], \underline{u}), C_{g_2}(\mathbf{r}, [{}^w\mathbf{X}], \bar{u}), {}^cZ > 0\} \quad (5.9)$$

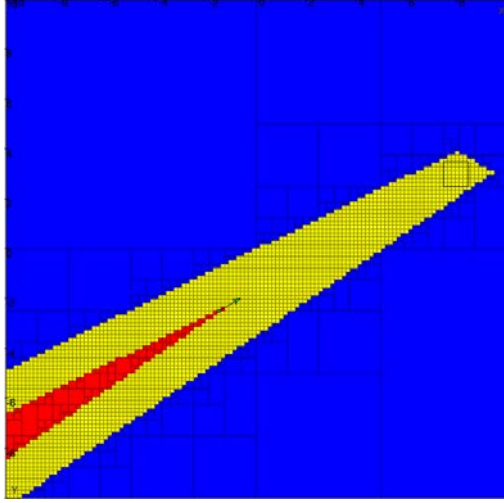
Nous avons utilisé l'algorithme SIVIA identique à celui de la section précédente et implémentant le même test d'inclusion pour résoudre le problème d'estimation ayant à la fois une incertitude sur les mesures dans l'image et les mesures de position des amers.

La figure 5.7 présente les sous-pavages obtenus pour un amer de position incertaine $\epsilon_w = 50$ cm. En utilisant dans le test d'inclusion de SIVIA une boîte à la place d'un vecteur ponctuel pour la position de l'amer, nous remarquons que le nombre de boîtes de l'approximation intérieure $\underline{\mathbb{S}}_{\text{thick}}$ (boîtes rouges) diminue (de 216 à 103 boîtes). Il en est de même pour le nombre de boîtes de l'approximation extérieure $\bar{\mathbb{S}}_{\text{thick}}$ (boîtes bleues, de 309 à 304). Par contre, la zone indéterminée (ou frontière en *jaune*) passe de 362 boîtes (figure 5.4c) à 2419 boîtes. Ceci montre qu'il est impossible pour l'algorithme de donner

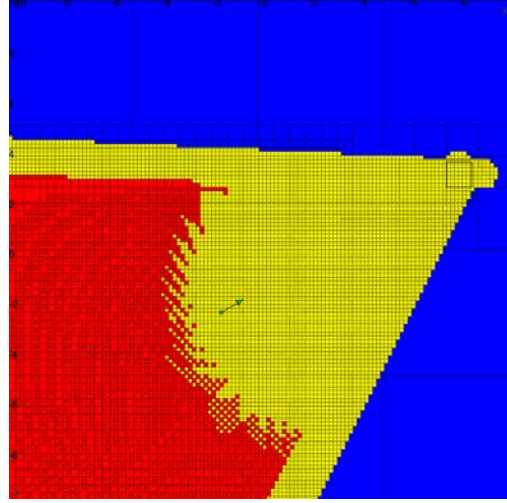
5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

une décision pour la plupart des boîtes, impliquant ainsi plus de bisections et le nombre grandissant de boîtes jaunes.

De façon similaire, en comparant les figures 5.4c et 5.7b, nous observons clairement que le problème devient encore plus complexe lorsque le cap θ est incertain.



(a) Sous-pavage de l'ensemble solution de $\mathbb{S}_{\text{thick}}$ lorsque $\epsilon_u = 0.1$ et le cap parfaitement connu ($\epsilon_\theta = 0$)



(b) Sous-pavage de l'ensemble solution de $\mathbb{S}_{\text{thick}}$ lorsque $\epsilon_u = 0.1$ et l'angle de cap incertain ($\epsilon_\theta = \frac{\pi}{6}$).

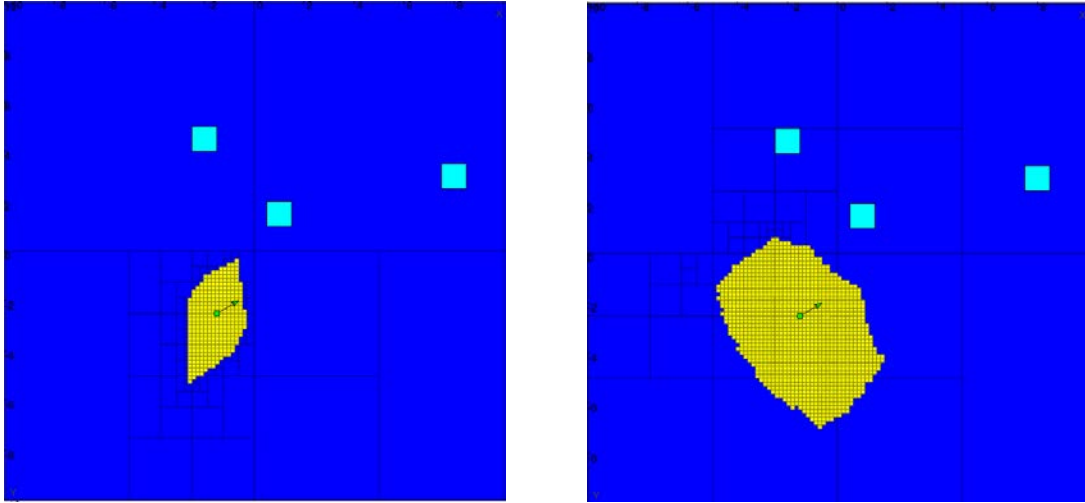
FIGURE 5.7. – Plan X-Y des sous-pavages de l'ensemble solution $\mathbb{S}_{\text{thick}}$ lorsque le robot observe un amer incertain.

Le cas où le robot observe 3 amers a également été étudié et les résultats présentés à la figure 5.8. Nous observons qu'il n'est pas possible de caractériser un intérieur à cause du pessimisme induit par le test d'inclusion qui n'est pas adéquat en cas d'incertitude sur les amers. Le sous-pavage obtenu est constitué entièrement de boîtes indéterminées.

Conclusion Les résultats obtenus en utilisant SIVIA pour résoudre $\mathbb{S}_{\text{thick}}$ montrent que la présence d'incertitudes sur les positions des amers est source d'indécision sur l'ensemble solution. Ceci tout à fait normal car lorsque les amers sont incertains, le problème n'est plus le même que celui exprimé par \mathbb{S} , il ne s'agit plus de *déterminer l'ensemble des pose compatibles avec les mesures dans l'image à erreurs bornées* ; mais plutôt de *déterminer l'ensembles des poses telles qu'il existe une configuration possible des amers qui les rend compatibles avec les mesures dans l'image à erreurs bornées*. L'ensemble solution correspondant est décrit par :

$$\mathbb{S}_{\text{Quant}} = \{\mathbf{r} \in \mathbb{R}^2 \times [0, 2\pi] \mid \exists^w \mathbf{X} \in [{}^w\mathbf{X}], C_{g_1}(\mathbf{r}, {}^w\mathbf{X}, \underline{u}), C_{g_2}(\mathbf{r}, {}^w\mathbf{X}, \bar{u}), {}^cZ > 0\} \quad (5.10)$$

Le problème ainsi obtenu est *un problème d'inversion ensembliste quantifié* du fait de la présence du quantificateur \exists dans $\mathbb{S}_{\text{Quant}}$. Pour le résoudre en utilisant un algorithme



(a) Sous-pavage de l'ensemble solution lorsque $\epsilon_u = 0.1$ et l'angle de cap est parfaitement connu ($\epsilon_\theta = 0$).

(b) Sous-pavage de l'ensemble solution lorsque $\epsilon_u = 0.1$ et l'angle de cap est incertain ($\epsilon = \frac{\pi}{6}$)

FIGURE 5.8. – Projection sur le plan (x, y) des sous-pavages de l'ensemble solution $\mathbb{S}_{\text{thick}}$ pour $\epsilon_u = 0.1$ et différentes valeurs de ϵ_θ en présence de trois amers incertains. En *bleu*, l'ensemble des boites qui ne sont pas solution. Et en *jaune*, la zone indéterminée représentant la frontière de l'ensemble $\Delta\mathbb{S}_{\text{thick}}$.

de type *branch and bound* tel que SIVIA, il est nécessaire d'effectuer, au niveau du test d'inclusion, des bisections sur les variables quantifiées. Si les bisections sur les variables quantifiées ne sont pas effectuées (ce qui est le cas dans les résultats obtenus dans cette section), certaines parties du domaine peuvent rester indéterminées.

Les bisections sur les variables quantifiées augmentent la complexité de l'algorithme et rendent ainsi le problème non traitable dans un temps compatible avec l'application de localisation.

Dans la suite, nous faisons le choix d'éliminer le quantificateur \exists afin de donner une meilleure approximation de l'ensemble solution.

5.3. Résolution par élimination du quantificateur existentiel

Pour permettre une meilleur approximation de l'intérieur de l'ensemble solution $\mathbb{S}_{\text{Quant}}$ et pour minimiser la taille de la zone indéterminée, nous allons éliminer le quantificateur \exists pour éviter de bissecter sur le domaine de l'amer $[{}^w\mathbf{X}]$. Dans cette section, nous présentons tout d'abord un bref état de l'art sur les solutions existantes pour la résolution de problèmes quantifiés. Nous décrivons ensuite une expression sans quantificateur du problème de localisation avec des amers incertains.

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

5.3.1. Résolution d'un problème quantifié

5.3.1.1. Définition

Un problème de satisfaction de contraintes quantifié (ou QCSP) $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{Q})$ est défini comme :

- un ensemble de n variables $\mathcal{X} = x_1, \dots, x_n$,
- un ensemble de domaines $\mathcal{D} = D_1, \dots, D_n$ où $D_i \subseteq \mathbb{R}$ est l'ensemble de toutes les valeurs potentielles de x_i ,
- une conjonction $\mathcal{C} = C_1 \wedge C_2 \wedge \dots \wedge C_e$ de contraintes (avec e le nombre de contraintes),
- et une séquence de quantificateurs $\mathcal{Q} = Q_{1x_1}, \dots, Q_{nx_n}$ où chaque Q_i est un quantificateur, \exists (existentiel, «il existe») ou \forall (universel, «pour tout »).

5.3.1.2. Méthodes existantes

Dans la littérature, plusieurs méthodes existent pour résoudre les QCSP.

Considérée comme un outil plus général permettant de traiter des sous-ensembles de \mathbb{R}^n pouvant être décrits par des équations polynomiales et des inégalités, la décomposition algébrique cylindrique (ou CAD) (COLLINS 1975 ; HONG 1992) est la méthode la plus célèbre pour résoudre les QCSP.

Son principe est le suivant. Étant donnée une formule quantifiée, l'algorithme CAD trouve une formule sans quantificateurs équivalente sur les réels. La construction basique de CAD comprend deux étapes : la projection et la remontée. Une troisième étape facultative peut être effectuée, si la construction de la formule non quantifiée est souhaitée.

La *projection* consiste à calculer des ensembles successifs de polynômes à $n - 1, n - 2, \dots, 1$ variables. Étant donné un ensemble de polynômes en entrée, l'idée principale est de calculer à chaque étape un nouvel ensemble de polynômes obtenus en éliminant le quantificateur sur une variable à la fois. En général, l'ordre d'élimination est important et un bon choix réduit la complexité de calcul.

Partant du niveau de projection le plus bas, la *remontée* (*lifting*) consiste à décomposer le polynôme de \mathbb{R} après élimination de toutes les variables sauf une. Cette décomposition de \mathbb{R} est étendue successivement (de manière incrémentale) à une décomposition de \mathbb{R}^n .

La CAD est une méthode puissante car elle permet de gérer plusieurs variables quantifiées pour les disjonctions/conjonctions de contraintes. Toutefois, une exigence forte se trouve au niveau de la forme des contraintes traitées, qui se limite aux contraintes polynomiales. De plus, CAD est plus adaptée aux problèmes de petite ou moyenne taille car le nombre d'opérations varie généralement de manière doublement exponentielle avec le nombre de variables.

Dans BORDEAUX et al. (2002) et BORDEAUX (2005), une méthode plus générale de propagation des contraintes numériques et logiques quantifiées est proposée. Elle consiste

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

dans un premier temps à décomposer les expressions longues en contraintes primitives ternaires et binaires quantifiées (ce qui rend la méthode extensible avec ajout de nouvelles contraintes primitives) et par la suite à appliquer un algorithme de propagation (sous forme de règles) afin d'éliminer la partie inconsistante des domaines des variables. L'un des avantages de cette méthode c'est qu'elle est extensible. Toutefois, l'étape de décomposition peut donner lieu à un nombre important de variables intermédiaires et de contraintes secondaires ce qui peut conduire à un problème d'efficacité. Aussi, le nombre de règles de propagation peut devenir considérable et pendant le processus de propagation local, l'interaction (dépendance) entre les variables peut être perdue. Ainsi, mettre l'emphase sur la consistance des primitives secondaires (ce qui est fait dans cette méthode) peut s'avérer être moins efficace que si l'expression initiale avait été maintenue.

BENEDETTI et al. (2006) proposent de réutiliser les algorithmes de propagation pour CSPs classiques en les adaptant de quatre manières. Étant donnée une contrainte C_k , un algorithme quelconque de propagation, et un ensemble de variables x_{k_1}, \dots, x_{k_r} (ordonné suivant l'ordre d'application des quantificateurs associés à chaque variable), le principe consiste à :

- étape 1 : une analyse existentielle en appliquant l'algorithme de propagation aux variables de C_k indépendamment des quantificateurs, ceci permet de réduire les domaines de toutes les variables associées au quantificateur existentiel
- étape 2 : une analyse fonctionnelle du domaine, qui permet de réduire le domaine des variables auxquelles est attaché un quantificateur universel
- étape 3 : une analyse anticipée consistant à appliquer un algorithme de propagation de la contrainte sur chaque variable x_{k_i} et ensuite effectuer une fusion des domaines résultants (des intersections pour les variables universelles et des unions pour celles existentielles).
- étape 4 : une analyse du dual consistant à construire une contrainte duale à C_k et y effectuer une analyse existentielle. A l'issue de cela, les variables rejetées par le dual sont ignorées dans la prochaine analyse de C_k , rendant les étapes 1 et 3 plus efficaces.

Ces quatre étapes d'adaptation permettent de fournir une solution au QCSP.

Les méthodes ainsi présentées sont des formes générales pour résoudre des QCSPs, toutefois il existe plusieurs méthodes adaptées à des formes spécifiques de problèmes de satisfaction de contraintes quantifiés.

YORKE-SMITH et al. (2005), par exemple, ont abordé la résolution des CSP incertains (ou UCSP), qui sont une restriction des QCSPs si l'on considère les paramètres comme des variables quantifiées existentielles. Ils proposent une méthode basée sur le calcul de plusieurs fermetures pour afin de caractériser l'ensemble solution au problème.

BENHAMOU et al. (2000) ont introduit la résolution de contraintes quantifiées sur les intervalles. Ils relient le traitement des quantificateurs \forall sur des variables intervalles, au calcul d'une approximation intérieure de la relation réelle quantifiée. Le principe est de

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

calculer une approximation extérieure du problème inverse constitué de la négation des contraintes et de prendre le complémentaire de la solution obtenue. La viabilité de la méthode a été prouvée par une application de calcul de mouvement de caméra.

Plusieurs exemples d'application de l'élimination de quantificateur pour l'inversion ensembliste ont été proposés, comme par exemple, GOLDSZTEJN et al. 2006 (quantificateur existentiel \exists), GOLDSZTEJN et al. 2009 (quantificateur universel \forall), JAULIN et al. 2010 (élimination du quantificateur \exists), etc.

Dans la suite, nous nous sommes inspiré de ce procédé pour transformer notre problème et l'exprimer en fonction des bornes de la variable quantifiée universellement.

5.3.1.3. QCSP et l'inversion ensembliste par intervalles

Dans le domaine de l'inversion ensembliste par intervalles, qui recouvre plusieurs cas restreints de QCSP, plusieurs travaux comme les nôtres, ont proposé des solutions pour résoudre des problèmes d'inversion ensembliste quantifiés (QSI). Les méthodes classiques d'inversion ensembliste, comme démontré par les résultats de la section 5.2.2, ne sont pas efficace pour résoudre le QCSP.

Des méthodes reposant sur l'utilisation des intervalles modaux (QSIMIA, pour quantified set inversion via Modal Interval Analysis) ont été proposées dans GOLDSZTEJN et al. 2005; VIÑAS et al. 2006.

JAULIN et al. (2010) ont opté pour l'élimination des quantificateurs (méthode proposée plus tôt par TARSKI (1998) pour des contraintes réelles). Ils ont proposé de transformer les contraintes sur les variables intervalles en contraintes sur les bornes de ces variables en utilisant l'arithmétique sur les intervalles symboliques. Après quoi une méthode classique d'inversion ensembliste est utilisée pour résoudre le problème non quantifié (*quantifier free*) obtenu.

Déterminer un procédé permettant d'élimination les quantificateurs sur les variables intervalles quantifiées, s'est avéré être l'une des solutions rapidement choisie en pratique.

5.3.2. Réécriture du problème d'estimation de pose avec amers incertains sous forme non quantifiée

Nous avons vu à la section 5.2.2, que le problème d'estimation de pose en présence de mesures dans l'image et de positions des amers incertaines peut être modélisé comme un QCSP.

Pour le résoudre, notre méthode consiste à éliminer le quantificateur \exists , sur la variable de position des amers ${}^w\mathbf{X}$, en exprimant l'ensemble solution $\mathbb{S}_{\text{Quant}}$ en fonction des bornes de la variable quantifiée. Pour cela, rappelons que l'ensemble solution $\mathbb{S}_{\text{Quant}}$ est défini par :

$$\mathbb{S}_{\text{Quant}} = \{\mathbf{r} \in \mathbb{R}^2 \times [0, 2\pi] \mid \exists {}^w\mathbf{X} \in [{}^w\mathbf{X}], C_{g_1}(\mathbf{r}, {}^w\mathbf{X}, \underline{u}), C_{g_2}(\mathbf{r}, {}^w\mathbf{X}, \bar{u}), {}^cZ > 0\} \quad (5.11)$$

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

Nous allons dans la suite dériver une expression non quantifiée du domaine solution.

Contraintes C_{g_1} et C_{g_2}

$$\exists {}^w \mathbf{X} \in [{}^w \mathbf{X}], C_{g_1}(\mathbf{r}, {}^w \mathbf{X}, \underline{u}), C_{g_2}(\mathbf{r}, {}^w \mathbf{X}, \bar{u}) \quad (5.12)$$

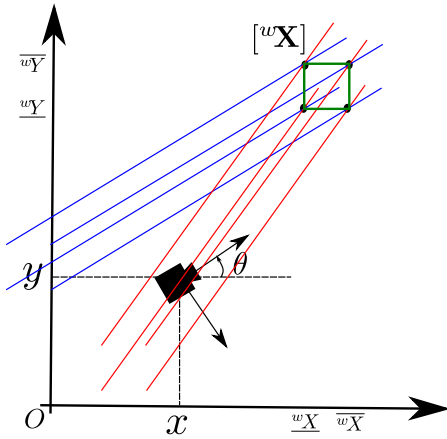
$$\iff \exists {}^w \mathbf{X} \in [{}^w \mathbf{X}] \begin{cases} ({}^w X - x)(\cos \theta - \underline{u} \sin \theta) - ({}^w Y - y)(\sin \theta + \underline{u} \cos \theta) \geq 0 \\ ({}^w X - x)(\cos \theta - \bar{u} \sin \theta) - ({}^w Y - y)(\sin \theta + \bar{u} \cos \theta) \leq 0 \end{cases} \quad (5.13)$$

$$\implies \begin{cases} \max_{{}^w \mathbf{X} \in [{}^w \mathbf{X}]} (({}^w X - x)(\cos \theta - \underline{u} \sin \theta) - ({}^w Y - y)(\sin \theta + \underline{u} \cos \theta)) \geq 0 \\ \min_{{}^w \mathbf{X} \in [{}^w \mathbf{X}]} (({}^w X - x)(\cos \theta - \bar{u} \sin \theta) - ({}^w Y - y)(\sin \theta + \bar{u} \cos \theta)) \leq 0 \end{cases} \quad (5.14)$$

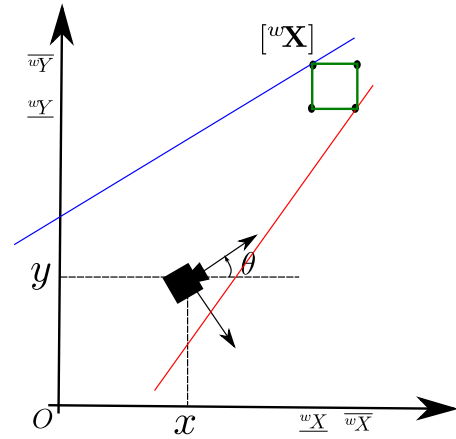
Les expressions des contraintes C_{g_1} et C_{g_2} étant linéaires en ${}^w \mathbf{X}$, les extremums exprimés dans (5.14) sont fonction des extremums de la variable ${}^w \mathbf{X}$. Ces extremums correspondent aux coins de la boîte $[{}^w \mathbf{X}]$.

Soit $C = \{(\underline{{}^w X}, \underline{{}^w Y}), (\underline{{}^w X}, \overline{{}^w Y}), (\overline{{}^w X}, \overline{{}^w Y}), (\overline{{}^w X}, \underline{{}^w Y})\}$, l'ensemble des coins qui constituent la boîte $[{}^w \mathbf{X}]$. L'équation 5.14 est donc équivalente à :

$$\begin{cases} \max_{{}^w \mathbf{X} \in C} (({}^w X - x)(\cos \theta - \underline{u} \sin \theta) - ({}^w Y - y)(\sin \theta + \underline{u} \cos \theta)) \geq 0 \\ \min_{{}^w \mathbf{X} \in C} (({}^w X - x)(\cos \theta - \bar{u} \sin \theta) - ({}^w Y - y)(\sin \theta + \bar{u} \cos \theta)) \leq 0 \end{cases} \quad (5.15)$$



(a) Application des contraintes C_{g_1} et C_{g_2} à chacun des coins de $[{}^w \mathbf{X}]$. En *bleu* la droite $g_1(\mathbf{r}, {}^w \mathbf{X}_i, \underline{u})$ et en *rouge* la droite $g_2(\mathbf{r}, {}^w \mathbf{X}_i, \underline{u})$ avec $i = 1 \dots 4$.



(b) Ensemble solution rattaché à l'application de C_{g_1} et C_{g_2} .

FIGURE 5.9. – Explication graphique des 3 conditions de consistance d'une pose.

Géométriquement, cela revient à appliquer les contraintes C_{g_1} et C_{g_2} à chacun des coins de C (visible à la figure 5.9a) et d'en prendre les extremums. Les points de l'ensemble

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

solution sont en dessous du maximum des droites g_1 pour chacun des quatre coins et au dessus du minimum des g_2 illustré à la figure 5.9.

Une pose est donc validée comme étant consistante si :

- C_{g_1} est vérifiée par au moins un coin de C ,
- C_{g_2} est également vérifiée par au moins un coin,
- et la boîte $[{}^w\mathbf{X}]$ n'est pas situé dans le demi plan arrière de la caméra (${}^cZ > 0$).

Contrainte landmark_visible($\mathbf{r}, [{}^w\mathbf{X}], u$) Cette contrainte exprime le fait que le point ${}^w\mathbf{X}$ est situé dans le demi plan avant de la caméra, donc qu'il est visible. Ainsi $\exists {}^w\mathbf{X} \in [{}^w\mathbf{X}], {}^cZ > 0$ est traduit dans cette section géométriquement comme déterminer l'ensemble des positions à partir desquelles l'amer est visible. Les différents cas pour lesquels ${}^cZ > 0$ sont représentés à la figure 5.10.

Une pose est donc dite consistante par rapport à la contrainte de visibilité de l'amer landmark_visible($\mathbf{r}, [{}^w\mathbf{X}], u$) si elle est dans l'une des quatre situations illustrées à la figure 5.10. Dans tous les autres cas de figure l'amer serait situé à l'arrière de la camera, donc invisible. Soit α l'angle de vue de la caméra dans le repère globale de référence, et soit $\vec{d} = (d_x, d_y)$ le vecteur unitaire représentant la ligne de vue exprimée dans le repère global. Ces quatre cas se traduisent comme suit :

$$\begin{cases} d_x \geq 0 \implies x < \overline{{}^wX} \\ d_x \leq 0 \implies x > \underline{{}^wX} \\ d_y \geq 0 \implies y < \overline{{}^wY} \\ d_y \leq 0 \implies y > \underline{{}^wY} \end{cases} \quad (5.16)$$

$$\iff \begin{cases} \cos \alpha \geq 0 \implies x < \overline{{}^wX} \\ \cos \alpha \leq 0 \implies x > \underline{{}^wX} \\ \sin \alpha \geq 0 \implies y < \overline{{}^wY} \\ \sin \alpha \leq 0 \implies y > \underline{{}^wY} \end{cases} \quad (5.17)$$

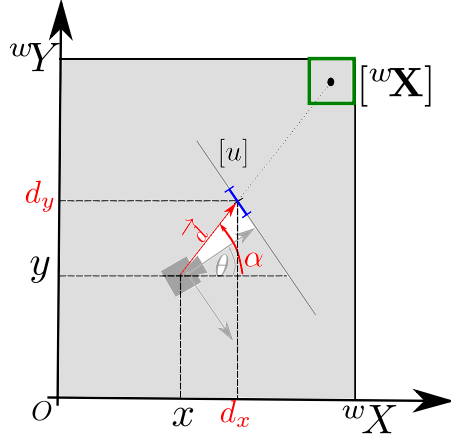
En appliquant la loi de De Morgan pour la formule logique $A \implies B \equiv \neg A \vee B$ la contrainte landmark_visible s'écrit donc comme suit :

$$\text{landmark_visible}(\mathbf{r}, [{}^w\mathbf{X}], u) : \begin{cases} \cos \alpha < 0 \vee x < \overline{{}^wX} \\ \cos \alpha > 0 \vee x > \underline{{}^wX} \\ \sin \alpha > 0 \vee y > \underline{{}^wY} \\ \sin \alpha < 0 \vee y < \overline{{}^wY} \end{cases} \quad (5.18)$$

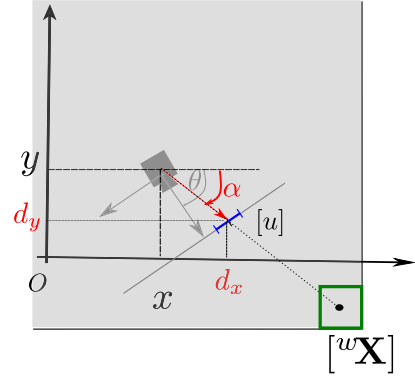
$\cos \alpha$ et $\sin \alpha$ peuvent être écrits en fonction de u et θ de la manière suivante :

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} = \frac{1}{\sqrt{1+u^2}} \begin{pmatrix} \cos \theta - u \sin \theta \\ \sin \theta + u \cos \theta \end{pmatrix} \quad (5.19)$$

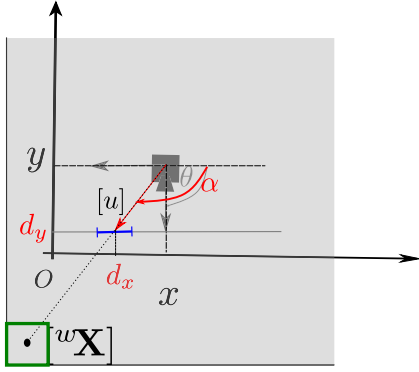
5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs



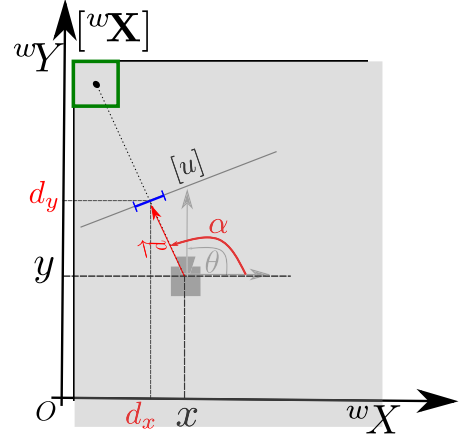
(a) cas 1 : $d_x \geq 0 \wedge d_y > 0$, le robot est situé à l'abscisse $x < \overline{wX}$ et à l'ordonnée $y < \overline{wY}$



(b) cas 2 : $d_x \geq 0 \wedge d_y \leq 0$, le robot est situé à l'abscisse $x > \overline{wX}$ et à l'ordonnée $y > \overline{wY}$



(c) cas 3 : $d_x \leq 0 \wedge d_y \leq 0$, le robot est situé à l'abscisse $x < \overline{wX}$ et à l'ordonnée $y > \overline{wY}$



(d) cas 4 : $d_x \leq 0 \wedge d_y \geq 0$, le robot est situé à l'abscisse $x > \overline{wX}$ et à l'ordonnée $y < \overline{wY}$

FIGURE 5.10. – Illustration graphique de la contrainte landmark_visible en fonction des bornes de $[w\mathbf{X}]$ et de l'angle de vue de l'amers en fonction de la position du robot.

On obtient donc la contrainte landmark_visible décrite par :

$$\text{landmark_visible}(\mathbf{r}, [w\mathbf{X}], u) : \begin{cases} \cos \theta - u \sin \theta < 0 & \vee & x < \overline{wX} \\ \cos \theta - u \sin \theta > 0 & \vee & x > \overline{wX} \\ \sin \theta + u \cos \theta > 0 & \vee & y > \overline{wY} \\ \sin \theta + u \cos \theta < 0 & \vee & y < \overline{wY} \end{cases} \quad (5.20)$$

Après avoir réécrit toutes les contraintes de S_{Quant} , l'ensemble solution après élimination

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

du quantificateur \exists peut donc être écrit comme suit :

$$\mathbb{S}_{\text{Qfree}} = \left\{ \mathbf{r} \in \mathbb{R}^2 \times [0, 2\pi], \bigvee_{w\mathbf{X} \in \mathcal{C}} C_{g_1}(\mathbf{r}, w\mathbf{X}, \underline{u}), \bigvee_{w\mathbf{X} \in \mathcal{C}} C_{g_2}(\mathbf{r}, w\mathbf{X}, \bar{u}), \text{landmark_visible}(\mathbf{r}, [w\mathbf{X}], [u]) \right\}$$

Il est également illustré par la partie grise de la figure 5.11.

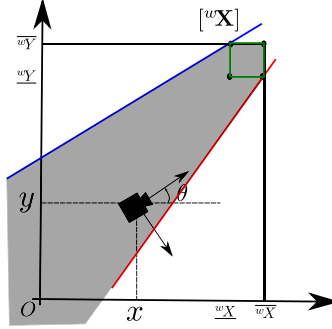


FIGURE 5.11. – Illustration de l'ensemble solution final $\mathbb{S}_{\text{Qfree}}$ en fonction des bornes de $[w\mathbf{X}]$ après élimination des quantificateurs (contraintes C_{g_1} , C_{g_2} et landmark_visible).

La solution ainsi obtenue nous ramène à un CSP que l'on peut résoudre par l'algorithme d'inversion ensembliste SIVIA.

5.4. Résultats de simulation

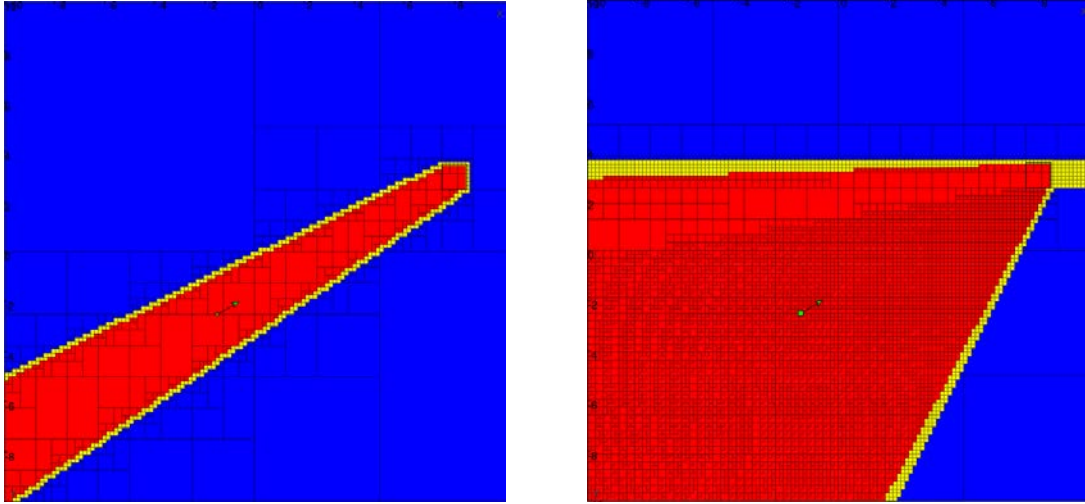
Nous avons utilisé l'algorithme SIVIA pour calculer une approximation $\mathbb{S}_{\text{Qfree}}$ en utilisant comme test d'inclusion le test avec les contraintes modifiées de notre QCSP après élimination du quantificateur. Les situations présentées à la section 5.2.2 ont été reprises et le problème sans quantificateur a été résolu en utilisant les mêmes paramètres que précédemment.

5.4.1. Résultats pour un amer observé

Lorsque le robot observe un amer incertain, nous observons qu'après élimination du quantificateur sur la position de l'amer, la frontière ($\Delta\mathbb{S}_{\text{Qfree}}$) est plus fine que celle précédemment calculée (résultat de la figure 5.7). Pour une tranche de l'angle θ , c-à-d $\epsilon_\theta = 0$ nous avons une diminution de 66.2% sur le nombre total de boîtes traitées.

Le tableau 5.2 donne une comparaison des tailles des sous-pavages avant et après élimination du quantificateur. Nous pouvons observer que la taille de la frontière, qui est la partie indéterminée (boîtes jaunes) passe de 2419 à 293 boîtes.

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs



(a) Sous-pavage de l'ensemble solution $\mathbb{S}_{\text{Qfree}}$ après élimination du quantificateur, θ parfaitement connu ($\epsilon_\theta = 0$).
 (b) Sous-pavage représentant l'ensemble solution $\mathbb{S}_{\text{Qfree}}$ après élimination du quantificateur, avec θ incertain ($\epsilon_\theta = \frac{\pi}{6}$).

FIGURE 5.12. – Domaine de la pose avec élimination de quantificateurs en présence d'un seul amer incertain.

	Taille en nombre de boites			
	$\underline{\mathbb{S}}$ (rouge)	$\Delta\mathbb{S}$ (jaune)	$\overline{\mathbb{S}}$ (bleu)	Total
$\mathbb{S}_{\text{thick}}$	103	2419	304	2826
$\mathbb{S}_{\text{Qfree}}$	284	376	293	953

TABLE 5.2. – Taille des sous-pavages obtenus avec et sans élimination du quantificateur lorsque l'angle est connu.

Dans le cas où l'angle est incertain (figure 5.12b), nous observons une diminution de 56.9% de boites traitées par SIVIA. Le domaine indéterminé est considérablement réduit tel que le montre le tableau comparatif 5.3. Le volume de l'ensemble solution représenté par $\Delta\mathbb{S}_{\text{Qfree}} \cup \underline{\mathbb{S}}_{\text{Qfree}} = \overline{\mathbb{S}}_{\text{Qfree}}$ passe de 59.68 m²rad avant élimination à 58.94 m²rad après élimination, avec une diminution de 70% du volume du domaine indéterminé.

	Taille en nombre de boites				Volumes (m ² rad)		
	$\underline{\mathbb{S}}$ (rouge)	$\Delta\mathbb{S}$ (jaune)	$\overline{\mathbb{S}}$ (bleu)	Total	$\underline{\mathbb{S}}$	$\Delta\mathbb{S}$	$\overline{\mathbb{S}}$
$\mathbb{S}_{\text{thick}}$	4168	146598	14840	165606	3.76	55.92	59.68
$\mathbb{S}_{\text{Qfree}}$	12810	43890	14673	71373	42.24	16.7	58.94

TABLE 5.3. – Taille des sous-pavages obtenus avec et sans élimination du quantificateur lorsque θ est incertain ($\epsilon_\theta = \frac{\pi}{6}$).

5.4.2. Résultats avec trois amers

Dans le cas où le robot observe trois amers incertains (3 boîtes de couleur cyan sur la figure 5.13), nous avons vu à la figure 5.8 que le résultat obtenu était une zone totalement indéterminée. Après élimination du quantificateur, un intérieur peut être caractérisé visible dans les deux cas de la figure 5.13. La pose réelle (point fléché vert sur la figure) est bien située dans $\underline{\mathbb{S}}_{\text{Qfree}}$. Nous avons une diminution de 48.15% du nombre total de boîtes traitées lorsque l'angle est connu (25.52% pour l'angle inconnu) et une diminution 43.98% du volume du domaine indéterminé lorsque l'angle n'est pas connu.

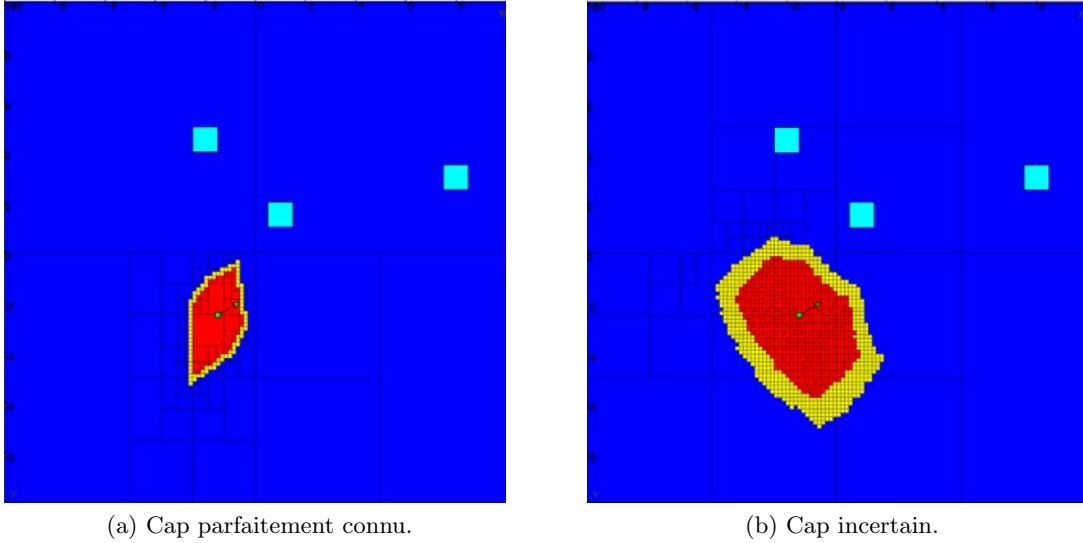


FIGURE 5.13. – Domaine de la pose avec élimination de quantificateurs en présence de trois amers de positions incertaines.

Les chiffres des tableaux 5.4 et 5.5 nous montrent également que malgré la diminution en surface (respectivement en volume) de la taille de la frontière, la taille totale de la solution $\bar{\mathbb{S}}$ reste la même (surface($\mathbb{S}_{\text{thick}}$) \approx surface($\mathbb{S}_{\text{Qfree}}$) = 7.93 m² et volume($\mathbb{S}_{\text{thick}}$) \approx volume($\mathbb{S}_{\text{Qfree}}$) = 3.91m²rad).

	Taille en nombre de boîtes				Surface (m ²)		
	$\underline{\mathbb{S}}$ (rouge)	$\Delta\mathbb{S}$ (jaune)	$\mathbb{C}\bar{\mathbb{S}}$ (bleu)	Total	$\underline{\mathbb{S}}$	$\Delta\mathbb{S}$	$\bar{\mathbb{S}}$
$\mathbb{S}_{\text{thick}}$	0	325	82	407	0	7.93	7.93
$\mathbb{S}_{\text{Qfree}}$	41	88	82	211	5.78	2.14	7.92

TABLE 5.4. – Taille des sous-pavages obtenus pour trois amers incertains observés, avec l'angle θ connu ($\epsilon_\theta = 0$).

Il est également à noter que dans le cas de multiples amers, il n'est généralement pas nécessaire de recourir à la contrainte landmark_visible, car les positions correspondant

5. Prise en compte de l'incertitude de position des amers par élimination de quantificateurs

	Taille en nombre de boites				Volume (m ² rad)		
	$\underline{\mathbb{S}}$ (rouge)	$\Delta\mathbb{S}$ (jaune)	$\mathbb{C}\mathbb{S}$ (bleu)	Total	$\underline{\mathbb{S}}$	$\Delta\mathbb{S}$	$\overline{\mathbb{S}}$
$\mathbb{S}_{\text{thick}}$	0	10252	3088	13340	0	3.91	3.91
$\mathbb{S}_{\text{Qfree}}$	1083	5764	3088	9935	1.712	2.198	3.91

TABLE 5.5. – Taille des sous-pavages obtenus pour trois amers incertains observés, lorsque l'angle θ est incertain ($\epsilon_\theta = \frac{\pi}{6}$).

à une projection depuis l'arrière ne sont généralement pas consistantes avec les mesures.

5.4.3. Discussion

Ces résultats nous laissent clairement voir qu'éliminer le quantificateur permet de mieux caractériser l'intérieur $\underline{\mathbb{S}}_{\text{Qfree}}$ de l'ensemble solution (voir partie en rouge des figures 5.12a, 5.12b et 5.12b) en présence d'incertitudes sur les positions des amers. Nous pouvons également dire qu'elle conduit l'algorithme à passer moins de temps à bissecter sur la frontière de $\underline{\mathbb{S}}_{\text{Qfree}}$ permettant ainsi d'affiner $\Delta\underline{\mathbb{S}}_{\text{Qfree}}$ et d'améliorer le temps de calcul. Toutefois, sur la figure 5.12b nous observons un prolongement de la zone incertaine à l'arrière de l'amer incertain, ceci est dû au pessimisme de la fonction d'inclusion naturelle dans laquelle une somme de cosinus et de sinus est présente. Cela explique également la faible diminution du volume de la zone incertaine dans le cas à 3 amers comparativement à celui à un amer (43.98% contre 70%).

En définitive, l'élimination du quantificateur a permis de réduire, comme prévu, la zone indéterminée en permettant de caractériser un intérieur de \mathbb{S} et de d'améliorer le temps de calcul en permettant à SIVIA de perdre moins de temps sur les bisections. Néanmoins avoir une bonne fonction d'inclusion est un impératif pour un résultat moins pessimiste.

5.5. Conclusion

Dans ce chapitre, nous avons traité le problème d'estimation en présence de mesures à erreurs bornées à la fois dans l'image et sur les positions des amers. S'agissant d'un problème d'inversion ensembliste quantifié, les méthodes classiques de résolution de problèmes d'inversion ensembliste ne sont plus adaptés pour le résoudre.

Nous avons donc proposé une méthode consistant à éliminer le quantificateur existentiel en réécrivant les équations de la fonction d'observation à partir des bornes de la variable quantifiée (mesure de position de l'amer) en utilisant la fonction d'inclusion naturelle de la fonction d'observation. Les résultats obtenus pour un modèle 2D d'estimation ont montré que l'élimination de quantificateur a permis de caractériser un intérieur de l'ensemble solution. Il a également été montré que la présence de plusieurs amers permet de réduire l'incertitude induite par le pessimisme de la fonction d'inclusion naturelle utilisée. De plus, nous avons montré que l'élimination de quantificateur est un moyen efficace pour

5. *Prise en compte de l'incertitude de position des amers par élimination de quantificateurs*

réduire le nombre de dimensions de bisections des algorithmes d'inversion ensemblistes classiques et par conséquent réduire le temps de calcul.

En perspective à ces travaux, une comparaison de plusieurs formes de fonctions d'inclusion permettrait de davantage réduire la zone indéterminée. Le passage au cas 3D peut être rendu possible par le même procédé que celui vu dans ce chapitre mais avec un niveau supérieur de complexité de la fonction d'inclusion qui est fortement non linéaire. Plus de détails à ce propos sont donnés dans les perspectives de ce manuscrit (chapitre 7).

6. Localisation coopérative ensembliste de drones

6.1. Introduction

Ces dernières décennies, les véhicules aériens sans pilote ont attiré l'attention dans plusieurs domaines impliquant des tâches multi-robots tels que, les missions de recherche et de couverture (MAZA et al. 2007 ; BEVACQUA et al. 2015), la manipulation coopérative (MICHAEL et al. 2011 ; GIOIOSO et al. 2014) et la commande (ROBUFFO GIORDANO 2012 ; JIANG et al. 2015 ; SCHIANO et al. 2017). Les tâches fastidieuses comme celle d'exploration (FOX et al. 2006) peuvent également être réalisées par ces véhicules.

Dans chacun de ces cas de figure, l'un des problèmes à résoudre avec un groupe de robots est la localisation coopérative (ROUMELIOTIS et al. 2002), c'est-à-dire l'amélioration de la capacité de positionnement de chaque robot grâce à l'échange d'informations (par exemple des mesures de capteurs) avec d'autres robots du groupes.

Plusieurs méthodes sont disponibles pour la localisation coopérative (CL), comme le filtre de Kalman étendu (EKF) pour un système centralisé (HUANG et al. 2011 ; QU et al. 2011 ; DAVIS 2014), ou, si le calcul est décentralisé et la communication peu fiable, d'autres techniques comme l'intersection de covariance ou la mise à jour entrelacée (BAHR et al. 2009). Des approches supposant des erreurs bornées qui utilisent des polytopes et des algorithmes de programmation linéaires ont également été proposées par SPLETZER et al. (2003) et CAMILLO JOSE TAYLOR (2007).

Sous hypothèse de mesures à erreurs bornées, nous avons montré au chapitre 4, que les méthodes ensemblistes par intervalles sont un outil efficace pour la caractérisation d'un domaine de pose pour un drone en effectuant la propagation garantie des incertitudes sur les mesures. Plusieurs applications de ces méthodes ensemblistes à la localisation coopérative sont également présentes dans la littérature : pour des robots sous-marins avec sonar (BETHENCOURT et al. 2013), des réseaux de véhicules avec GNSS (LASSOUED et al. 2017), etc.

Ainsi, dans ce chapitre, nous proposons d'utiliser les méthodes ensemblistes par intervalles comme outil pour résoudre le problème de localisation coopérative de drones dans un groupe. Pour cela, considérons donc une flottille de drones communicants qui ont chacun besoin de se localiser afin d'accomplir une tâche de navigation précise. Chaque drone est équipé d'une caméra capable de suivre des amers géo-référencés et d'un système de communication fournissant à chaque robot un moyen d'échanger des données et de

6. Localisation coopérative ensembliste de drones

mesurer des distances avec ses voisins et avec une station de base dont la position est connue.

Notre but est de calculer un domaine de pose pour chaque robot à partir des techniques de propagation de contraintes et d'inversion ensemblistes. La méthode que nous proposons dans ce chapitre détermine, pour chaque robot, un ensemble contenant toutes les poses possibles de ce dernier, en prenant en compte les mesures dans image, les mesures de distance et les bornes d'erreur des amers géo-référencés ; toutes étant à erreurs bornées. La pose réelle de chaque robot est garantie d'être contenue dans l'ensemble solution calculé tant que les bornes des erreurs de mesures ne sont pas violés.

La section 6.2 présente la formulation du problème de localisation multi-robots. Une première approche de résolution par échange de mesures entre les drones est présentée à la section 6.3. La solution précédente ne permettant pas de gérer un nombre considérable de robots, la section 6.4 présente la solution distribuée que nous avons développé permettant le passage à l'échelle du problème. Pour chacune de ces approches, des validations en simulation et expérimentale sont effectuées et rendent possible la comparaison par rapport aux méthodes existantes.

6.2. Modélisation d'un groupe de drones communicants

Considérons le repère global de référence \mathcal{F}_w , dans lequel se déplacent n drones R_k ($k \in \{1 \dots n\}$), au centre de gravité desquels sont attachés n repères robots \mathcal{F}_{r_k} . La pose d'un drone R_k est donnée par $\mathbf{r}_k = (x_k, y_k, z_k, \phi_k, \theta_k, \psi_k)$ dans le repère global de référence.

Dans la suite, nous considérons que les drones sont identiques dans leur architecture, par conséquent ils ont le même type de caméra. Toutes les caméras sont calibrées (les paramètres intrinsèques sont connus) et les transformations rigides entre le repère caméra et le repère robot (${}^c\mathbf{T}_r$) sont connues par calibration. Nous supposons également, comme dans SCHIANO et al. 2017, que chaque robot se déplace en régime *quasi-stationnaire* permettant ainsi la mesure précise du roulis et du tangage par la centrale inertielle.

6.2.1. Scénario

L'environnement de navigation est constitué d'une station de base B capable de communiquer avec les drones, et d'amers géolocalisés (voir figure 6.1). Afin d'accomplir sa tâche de navigation, chaque robot du groupe doit se localiser par rapport au repère \mathcal{F}_w . Pour cela, chaque drone R_k est équipé de capteurs embarqués (une camera, un altimètre, une unité de mesure inertielle) lui permettant d'effectuer les mesures :

- de son altitude z_k
- de ses angles de roulis ϕ_k et de tangage θ_k de manière précise
- d'une estimation approximative de l'angle de lacet ψ_k

6. Localisation coopérative ensembliste de drones

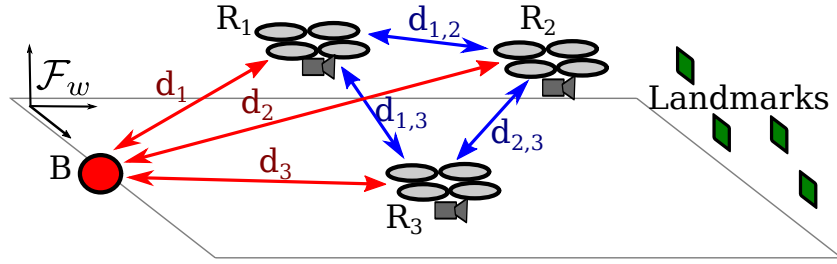


FIGURE 6.1. – Localisation coopérative avec des mesures de caméra et de distance

- des points 2D dans l'image $\bar{\mathbf{x}}_1^k, \dots, \bar{\mathbf{x}}_N^k$ correspondant aux amers 3D observables ${}^w\mathbf{X}_1^k, \dots, {}^w\mathbf{X}_N^k$ dans l'environnement.

Outre ces mesures de capteurs embarqués, chaque robot R_k est équipé d'un système de communication lui permettant de :

- mesurer sa distance $d_{k,j}$ à un autre robot R_j ($j = 1 \dots n, j \neq k$) appartenant à l'ensemble de ses voisins $\mathcal{N}(R_k)$
- mesurer sa distance d_k à la station de base B de position $\mathbf{b} = (x_B, y_B, z_B)^\top$ dans le repère \mathcal{F}_w
- communiquer et échanger des informations avec chaque robot dans son voisinage.

Cela peut être fait en utilisant un système de communication Ultra Wideband (UWB) IEEE 802.15.4a (SAHINOGLU et al. 2006) qui permet de mesurer via des mesures de temps de vol, ou d'autres capteurs externes.

Sous hypothèses de mesures à erreurs bornées, nous souhaitons calculer les domaines des poses 6-DOF réalisables pour chaque drone dans le repère global de référence \mathcal{F}_w , c'est-à-dire l'ensemble des positions $\mathbf{p}_k = (x_k, y_k, z_k)$ et des attitudes (roulis, tangage et lacet) $\mathbf{q}_k = (\phi_k, \theta_k, \psi_k)$ consistantes avec à la fois les mesures proprioceptives et les mesures extéroceptives de chaque drone. Étant donné que trois des six degrés de libertés sont mesurés avec une faible incertitude (z_k, ϕ_k, θ_k), notre attention sera principalement portée sur l'estimation de la position horizontale x_k, y_k et de l'angle de lacet ψ_k de chaque robot. La connaissance à priori sur l'altitude, le roulis et le tangage mesurés pour chaque robot sera intégrée dans le problème sous forme de contraintes. L'ensemble des équations d'observation des mesures nous permet de définir, pour chaque robot, un ensemble de contraintes décrites dans la section suivante.

6.2.2. Ensemble de contraintes liées aux mesures

Pour résoudre le problème décrit dans par le scénario précédent, nous l'exprimons sous forme d'un problème de satisfaction de contraintes (CSP), puis nous utilisons l'algorithme SIVIA avec contracteurs pour caractériser le domaine de poses réalisables pour chaque robot. Pour cela, chaque observation est liée à l'état du robot par une contrainte qui intervient dans le CSP décrivant le problème. Ainsi, dans cette section nous présentons l'ensemble de contraintes rattachées aux observations.

6.2.2.1. Contraintes liées aux mesures dans l'image

La prise en compte des mesures dans l'image suit l'approche présentée au chapitre 4.

Rappelons que, grâce au modèle sténopé de projection d'un point 3D dans l'image (équation 4.5), nous pouvons définir l'ensemble de contraintes liées aux mesures dans l'image.

Nous avons donc :

- C_{proj}^k projection 2D d'un point dans l'image

$$\bar{\mathbf{x}}_i^k = \mathbf{K} \boldsymbol{\Pi} {}^c \mathbf{T}_w {}^w \mathbf{X}_i^k = \mathbf{f}(\mathbf{r}, {}^w \mathbf{X}_i^k) \quad (6.1)$$

$$\text{avec, } ({}^c X^k, {}^c Y^k, {}^c Z^k, 1)^\top = {}^c \mathbf{T}_r {}^r \mathbf{T}_w(\mathbf{r}) ({}^w X, {}^w Y, {}^w Z, 1)^\top$$

$$C_{proj,i}^k = \left\{ \begin{array}{l} u = p_x x_i^k + u_0, \\ v = p_y y_i^k + v_0 \end{array} \right\} \quad (6.2)$$

$$\text{avec, } x_i^k = \frac{{}^c X_i^k}{{}^c Z_i^k}, y_i^k = \frac{{}^c Y_i^k}{{}^c Z_i^k}.$$

- $C_{front,i}^k = \{{}^c Z_i^k > 0\}$, l'amer est situé à l'avant de la caméra, avec $i = 1 \dots N$

Chaque drone R_k devra donc vérifier l'ensemble des contraintes C_{proj}^k et C_{front}^k .

6.2.2.2. Contrainte liée à la mesure de distance à la base

En supposant que l'antenne du capteur de mesure coïncide avec le centre de drone, les mesures de distance sont indépendantes de l'attitude du robot.

La mesure d_k de distance à la base donne la contrainte C_{base}^k sur la distance euclidienne entre le robot R_k et la station de base B . Rappelons que la partie position de la pose du robot est $\mathbf{p}_k = (x_k, y_k, z_k)^\top$, nous avons donc :

$$C_{base}^k : \{d_k = \|\mathbf{p}_k - \mathbf{b}\|_2\} \quad (6.3)$$

Comme la position de la station de base \mathbf{b} est connue, la connaissance de d_k contraint la position de \mathbf{r}_k du drone à être située sur la sphère $S^2(\mathbf{b}, d_k)$ tel que l'illustre la figure 6.2.

6.2.2.3. Contraintes liées aux mesures d'inter-distances

Les contraintes liées aux mesures de l'inter-distance entre un drone R_k et ses voisins R_j avec $R_j \in \mathcal{N}(R_k)$ sont définies par :

$$d_{k,j} = \|\mathbf{p}_k - \mathbf{p}_j\|_2 \quad \text{avec } (k, j) \text{ dans } \{1 \dots n\}^2 \text{ et } k \neq j \quad (6.4)$$

$$C_{r_k, r_j} : \{d_{k,j} = \|\mathbf{p}_k - \mathbf{p}_j\|_2\} \quad (6.5)$$

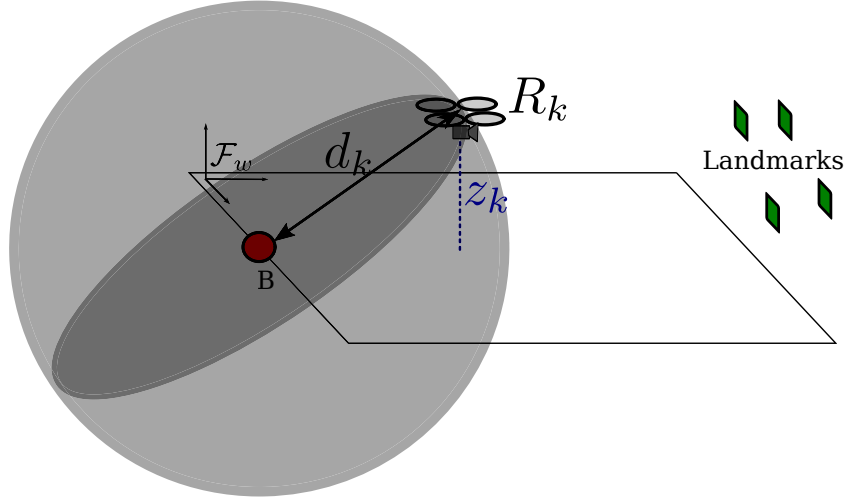


FIGURE 6.2. – Contrainte de distance à la base

La connaissance supplémentaire de $d_{k,j}$ par le robot R_k restreint les configurations possibles. Pour une position candidate donnée d'un voisin R_j , les positions admissibles de R_k sont limitées à un cercle (intersection des deux sphères $S^2(\mathbf{b}, d_k)$ et $S^2(\mathbf{p}_k, d_{k,j})$).

Les mesures des distances d_k et $d_{k,j}$ fournissent des contraintes sur les positions des robots, mais il reste 3 degrés de liberté (sur 6DoF) correspondant à toutes les rotations de la formation autour de la station de base.

6.2.2.4. Contrainte liée à la mesure de l'altitude

Considérons maintenant l'altitude z_k de chaque robot fournie par l'altimètre, nous avons la contrainte $z_k = z_k^{\text{meas}}$ qui situe R_k sur un plan.

Elle permet de contraindre la sphère $S^2(\mathbf{b}, d_k)$ formée par la distance à la station de base B et le robot R_k à un cercle horizontal, résultat de l'intersection de $S^2(\mathbf{b}, d_k)$ et du plan.

Elle permet également de contraindre la sphère $S^2(\mathbf{p}_k, d_{k,j})$ solution de la mesure de l'inter-distances $d_{k,j}$. Pour une position de R_j connue, R_k n'a généralement que deux configurations possibles¹ correspondant à l'intersection du cercle de position horizontal de R_k avec la sphère $S^2(\mathbf{p}_k, d_{k,j})$.

Ces trois mesures conduisent à une ambiguïté sur la pose du robot dépendante de l'angle et correspondant aux rotations autour de l'axe vertical de la station de base B . La fusion de ces mesures avec celles extraites dans l'image permet de lever l'ambiguïté.

1. Pour une position de R_j connue, la position de R_k est généralement contrainte à deux points, sauf si R_j est à la verticale de la station de base. Dans ce dernier cas, l'intersection des deux sphères se trouve dans le plan de la contrainte d'altitude, et la position de R_k reste donc contrainte sur un cercle.

6.3. Localisation coopérative par partage de mesures

A partir de l'ensemble des contraintes liées aux mesures définies à la section 6.2.2, il est possible de définir un CSP qui permettra le calcul effectif de la pose tout en propageant les incertitudes sur les mesures. La première approche consiste à effectuer une fusion de données basée sur le partage de mesures entre les robots. Un robot R_k partage donc toutes ses mesures avec l'ensemble de ses voisins $\mathcal{N}(R_k)$ et reçoit également les données venant de l'ensemble de ses voisins.

6.3.1. Ensemble solution avec partage de mesures

Étant donné que tous les drones échangent leurs mesures d'image, leurs distances par rapport à la station de base et la distance qui les sépare, nous pouvons formuler le problème sous la forme d'un système de fusion centralisé, où chaque drone est capable d'estimer de façon jointe sa position et celle des autres.

Les boîtes $[\mathbf{r}]_k$ représentent une connaissance préalable du domaine des poses des drones, $[\bar{\mathbf{x}}_i^k]$ représentent les mesures d'erreur bornées dans l'image vue de R_k , et la largeur de la boîte $[{}^w\mathbf{X}_i^k]$ représente l'incertitude sur les coordonnées de l'amer i .

Pour caractériser l'ensemble des solutions du drone, nous utilisons l'algorithme SIVIA avec contracteur pour calculer le sous-pavage de toutes les poses possibles dans $[\mathbf{r}]_k$ compatibles avec les mesures utilisant le CSP $\mathcal{H}_{\text{meta}}$ défini par :

$$\mathcal{H}_{\text{meta}} : \left(\begin{array}{l} \mathbf{r}_k \in [\mathbf{r}]_k, \\ \bar{\mathbf{x}}_i^k \in [\bar{\mathbf{x}}_i^k], \\ {}^w\mathbf{X}_i^k \in [{}^w\mathbf{X}_i^k], \\ \{C_{proj,i}^k, C_{front,i}^k, \\ C_{base}^k, \\ C_{r_k,r_j}, r_j \in \mathcal{N}(R_k)\} \\ i \in 1 \dots N, k \in 1 \dots n \end{array} \right) \quad (6.6)$$

Étant donné que les composantes ϕ_k, θ_k , et z_k de la pose \mathbf{r}_k de chaque drone sont directement mesurés grâce aux capteurs, les domaines initiaux de ces composantes de l'état sont connus avec une faible incertitude liée à la précision des capteurs. Ainsi, les domaines initiaux sont réglés de sorte que $[\phi_k] = \phi_k^{\text{meas}} + [-\epsilon_\phi, \epsilon_\phi], [\theta_k] = \theta_k^{\text{meas}} + [-\epsilon_\theta, \epsilon_\theta], [z_k] = z_k^{\text{meas}} + [-\epsilon_z, \epsilon_z]$. Pour les degrés de liberté inconnus, leurs domaines initiaux dans $[\mathbf{r}_k]$ sont définis en fonction de la connaissance a priori sur l'environnement de navigation (par exemple les dimensions de la salle expérimentale présentée en Annexe B).

Après avoir défini le domaine initial pour chaque robot, un contracteur (HC4) est créé et appliqué dans l'algorithme SIVIA avec des bisections appliquées uniquement sur les composantes inconnues de la pose (x, y, ψ) . Ceci veut dire que, si le robot R_k a m voisins, alors l'algorithme SIVIA effectuera des bisections sur les 3 composantes de R_k et sur chacune des 3 composantes de ses m voisins, car dans le CSP $\mathcal{H}_{\text{meta}}$, R_k calcule également

6. Localisation coopérative ensembliste de drones

la pose de tous ses voisins. Ce qui fait de notre problème un problème de dimensions $3(m+1)$. Les bissections/contractions permettront de réduire récursivement le domaine initial en rejetant les parties non consistantes avec les mesures lors de l'application des contraintes de $\mathcal{H}_{\text{meta}}$.

L'ensemble solution est donc :

$$\begin{aligned} \mathbb{S}_{\mathcal{H}_{\text{meta}}} = \{ & \mathbf{r}_k \in \mathbb{R}^2 \times [z_k^{\text{mes}} \pm \epsilon_z] \times [\phi_k^{\text{mes}} \pm \epsilon_\phi] \times [\theta_k^{\text{mes}} \pm \epsilon_\theta] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \mid \\ & \mathbf{f}(\mathbf{r}_k, [{}^w\mathbf{X}_i^k]) \subset [\bar{\mathbf{x}}_i^k], \\ & \|\mathbf{p}_k, \mathbf{b}\|_2 \in [d_k], \|\mathbf{p}_k, \mathbf{p}_j\|_2 \in [d_{k,j}], i \in 1 \dots N, k \in 1 \dots n \} \end{aligned}$$

Avec SIVIA, des domaines bornés pour les poses $\mathbf{r}_1 \dots \mathbf{r}_n$ sont obtenus sous formes de sous-pavages extérieurs $\bar{\mathbb{X}}_{\mathbf{r}_1} \dots \bar{\mathbb{X}}_{\mathbf{r}_n}$. En supposant que les bornes des erreurs de mesures n'ont pas été violées, il est garanti que les domaines calculés contiennent les vraies poses des robots $\mathbf{r}_1^* \dots \mathbf{r}_n^*$, c'est-à-dire $\mathbf{r}_1^* \in \bar{\mathbb{X}}_{\mathbf{r}_1} \dots \mathbf{r}_n^* \in \bar{\mathbb{X}}_{\mathbf{r}_n}$.

6.3.2. Résultats en simulation et comparaison à la méthode des moindres carrés

Dans cette section, nous présentons les résultats de simulations et d'expérimentations obtenus pour $n = 2$ drones, la figure 6.3 présente le scénario considéré.

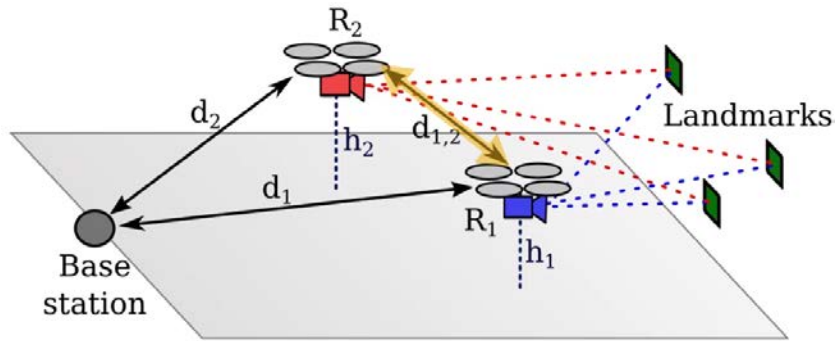


FIGURE 6.3. – Scénario illustrant l'ensemble des mesures pour deux robots.

6.3.2.1. Environnement de simulation

L'environnement de simulation est une pièce constituée d'une station de base située aux coordonnées $\mathbf{b} = (0, 0, 0)^\top$ et trois amers de positions connues situés vers le bas de la pièce tel que l'illustre la vue de dessus de la figure 6.4. Un échantillon de 50 positions de test a été généré pour valider notre approche. Cela conduit à un ensemble de 2450 combinaisons de positions à deux robots (pour chaque position fixe $i \in 1 \dots 50$ du robot R_1 , toutes les positions $j \neq i, j \in 1 \dots 50$ de R_2 sont considérées).

6. Localisation coopérative ensembliste de drones

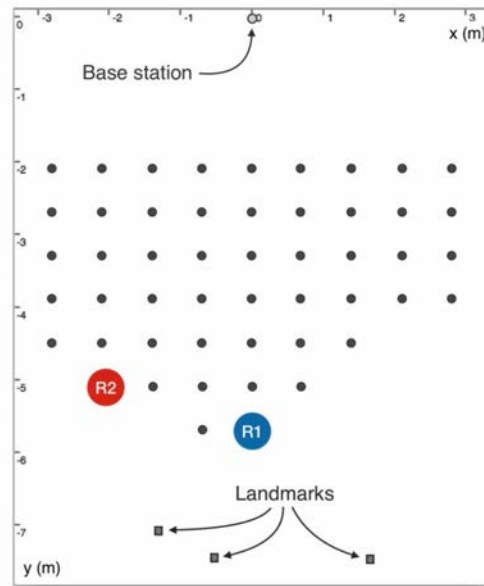


FIGURE 6.4. – Vue de dessus de l’environnement de simulation. Les rond *noirs* sont les 50 positions simulées pour le déplacement du drone. En *bleu* les robot R_1 et en *rouge* R_2 .

L’altitude, l’angle de tangage et l’angle de roulis sont supposés être connus. En supposant des mesures de boussole peu fiables, le cap du robot est supposé être connu très grossièrement (dans un intervalle de $\pm 20^\circ$). Les erreurs de mesure de distance (de la station de base et de l’inter-distance) sont supposées être comprises entre ± 5 cm. Le suivi des amers dans l’image est supposé être précis à $\pm 0,5$ pixel.

6.3.2.2. Sous-pavages résultats dans le cas d’un robot isolé

Les résultats du positionnement d’un seul robot, à partir des mesures de la station de base et de la caméra de 3 amers, sont illustrés à la figure 6.5. La figure représente en bleu la projection sur le plan (x, y) des sous-pavages calculés pour les 50 positions simulées. Les statistiques de la largeur des domaines calculés sont reportées dans la partie gauche du tableau 6.1. Nous observons clairement sur la figure que les domaines de pose s’agrandissent à mesure que le robot s’éloigne des amers. Il est également intéressant de noter que de plus grandes incertitudes sont observées près de l’axe $x = 0$. Cela est dû à la mauvaise intersection de la sphère par rapport à la distance de la station de base $S^2(\mathbf{b}, d_1)$ avec les contraintes de projection des amers dans l’image (voir équation 6.2) .

6. Localisation coopérative ensembliste de drones

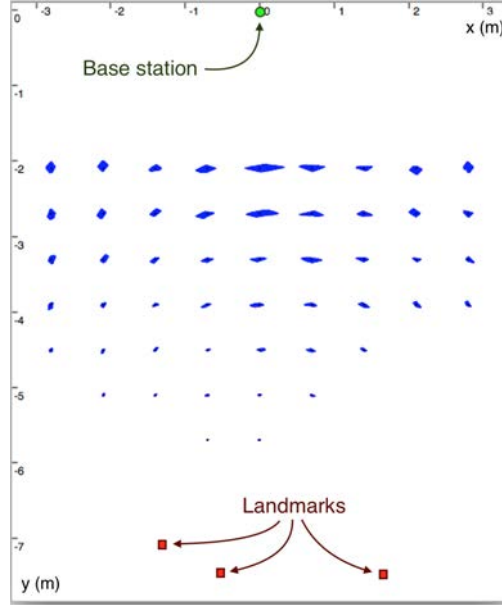


FIGURE 6.5. – Vue de dessus des domaines de pose pour un seul robot, pour un échantillon de 50 positions différentes, avec mesure de distance à la station de base ($\epsilon_d = \pm 5\text{cm}$) et 3 amers connus.

6.3.2.3. Localisation coopérative avec visibilité totale des amers

L'algorithme SIVIA avec le contracteur HC4 résolvant le problème d'estimation avec coopération de l'équation 6.6 a été implémenté sur deux robots communicants R_1 et R_2 . Ces deux robots sont supposés identiques et observent tous deux les trois amers. La figure 6.7a présente les résultats obtenus de $\mathbb{S}_{\mathcal{H}_{\text{meta}}}^1$ et $\mathbb{S}_{\mathcal{H}_{\text{meta}}}^2$ pour R_1 statique et 49 configurations de R_2 .

Un robot vs deux robots Nous observons une diminution globale de l'incertitude de position par rapport au cas à un robot sur la coordonnée x ; visible par de meilleures valeurs du minimum et de la moyenne dans le tableau 6.1. Ces chiffres montrent que la coopération n'apporte ici un gain que lorsqu'un des deux robots est situé à une position mal configurée géométriquement. Elle profite de la contrainte de distance au second robot pour améliorer sa position.

La figure 6.7a montre un exemple de configuration pour laquelle la coopération réduit sensiblement l'incertitude de positionnement; ceci est dû à une bonne configuration géométrique de la position des deux robots l'un par rapport à l'autre et les deux par rapport au centre de la salle. Par exemple, si l'on compare la colonne $x = 0$ avec les sous-pavages de la figure 6.5, nous remarquons que ces derniers sont moins volumineux.

En effectuant une implémentation mono-thread de l'algorithme sur un Core i7 à 2.8 GHz 2014, le temps de calcul moyen est de 200 ms pour le paramètre de précision $\epsilon = 30$ mm

6. Localisation coopérative ensembliste de drones

Dimensions du domaine	1-drone			2-drones coop.		
	min	moyenne	max	min	moyenne	max
x (cm)	3.62	15.8	53.8	2.8	15.4	60
y (cm)	2.0	8.0	16.53	1.1	8.1	18.6
ψ (°)	0.75	2.1	5.6	0.67	2.1	6.2

TABLE 6.1. – Largeur des domaines de poses dans le cas d’un drone et dans le cas deux drones coopératifs lorsque la visibilité est totale (chaque drone observe 3 amers).

(résultats présentés à la figure 6.7) et de 32 ms pour $\epsilon = 60$ mm.

L’augmentation de ϵ réduit le temps de calcul, mais fournit une estimation plus grossière du domaine de poses réalisable. Ainsi, lorsque ϵ est fixe, le temps de calcul varie avec la taille du domaine calculé; elle est linéaire avec le nombre de d’amers observés par un robot. L’implémentation en temps réel de notre méthode est possible en utilisant un nombre fixe d’itérations de l’algorithme SIVIA, ou une implémentation interruptible à tout moment comme dans DREVELLE et al. 2016.

Comparaison avec la méthode des moindres carrés itérés Nous avons comparé les domaines obtenus aux résultats de la méthode des moindres carrés pondérés non-linéaires (WLS) calculées avec l’algorithme de Levenberg-Marquardt (LM). Les variances du bruit d’observation ont été réglées de manière à correspondre aux largeurs des domaines intervalles.

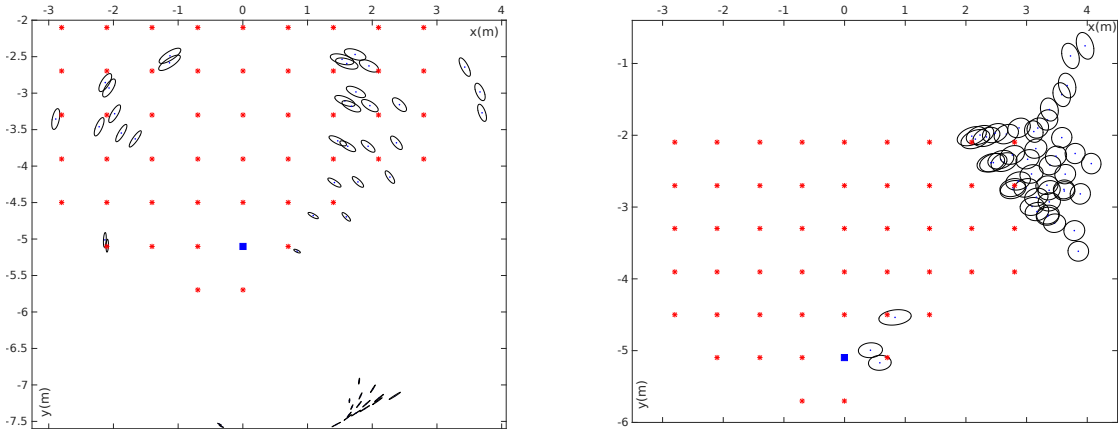


FIGURE 6.6. – Ellipsoïde d’incertitude pour l’algorithme LM avec comme valeur initiale le point situé au centre de la salle de simulation.

Tout d’abord, l’algorithme LM est connu pour converger lentement (ou pas du tout) lorsqu’il est initialisé loin de la solution exacte (ou d’un extremum) (LEVENBERG 1944),

6. Localisation coopérative ensembliste de drones

nous pouvons l’observer sur la figure 6.6. Sur cette figure, la valeur initiale de la pose à chaque instant a été fixée au centre de la salle d’acquisition, ce qui correspond à une configuration assez éloignée de la solution exacte. La figure présente les valeurs exactes (carré bleu pour R_1 et étoiles en rouge pour R_2) et les ellipsoïdes résultats de l’estimation lorsque R_1 est fixe (*en bleu*) et R_2 se déplace sur 49 positions différentes (*en noir*). Il est clairement visible, par les positions des ellipsoïdes sur la figure, que LM mal initialisé (*LM bad guess*) ne converge pas vers une solution proche de la solution exacte.

Ainsi, afin d’avoir une meilleure base de comparaison, nous avons calculé l’erreur relative de l’estimée de l’algorithme LM en utilisant à chaque instant la *vérité terrain* comme valeur initiale de la pose (*LM GT guess*).

Le tableau 6.2 reporte les erreurs de position horizontale et de cap dans le cas où tous les amers sont visibles. La quatrième ligne du tableau 6.2 confirme bien le fait que *LM bad guess* ne parvient souvent pas à atteindre la vraie pose. Les erreurs moyennes de l’estimée ponctuelle CoB sont de 1,51 cm en horizontal et de 0,16° en cap, contre 1,43 cm et 0,14° avec LM bien initialisé. Cela montre que le centre de la boîte englobante est une bonne estimation ponctuelle, avec des chiffres d’erreur très proches de l’estimation WLS lorsque les domaines ne sont pas trop grands. Cependant, lorsque les domaines calculés deviennent plus grands, le CoB devient moins précis (avec une erreur horizontale maximale de 17.7cm soit 7.4cm de plus que WLS).

Méthode	Erreur de position horiz. (cm)			Erreur de lacet (deg)		
	moyenne	95 %	max	moyenne	95 %	max
<i>Interval (CoB)</i>	1.51	3.52	17.5	0.16	0.41	1.99
<i>Interval + LM</i>	1.43	3.36	10.1	0.14	0.36	1.06
<i>LM GT guess</i>	1.43	3.36	10.1	0.14	0.36	1.06
<i>LM bad guess</i>	351.7	645.2	721.9	39.1	105.7	179.3

TABLE 6.2. – Statistiques d’erreurs de position horizontale 2D et de cap dans le cas coopératif 2-UAV avec visibilité totale.

Une solution serait de combiner le domaine calculé avec la méthode par intervalle (en tant qu’indicateur d’incertitude) avec LM pour fournir une estimation ponctuelle précise. La figure 6.7a, présente les domaines obtenus en appliquant SIVIA+HC4 et la figure 6.7b présente les ellipsoïdes résultats obtenus lorsque nous utilisons le CoB comme hypothèse initiale pour LM.

Une meilleure estimation ponctuelle (*Interval + LM*) est donc obtenue. À partir du CoB, *Interval + LM* a une erreur horizontale moyenne de 1.43 cm et 0.14° comme erreur de lacet. Ce résultat est similaire à *LM GT guess* initialisé à partir de la vérité terrain (deuxième ligne du tableau 6.2 et appuyé par des ellipsoïdes d’incertitudes centrées sur la pose exacte à la figure 6.7b). Cela montre que le CoB est un bon point de départ pour une convergence efficace vers la solution et permet d’éviter les minimums locaux.

6. Localisation coopérative ensembliste de drones

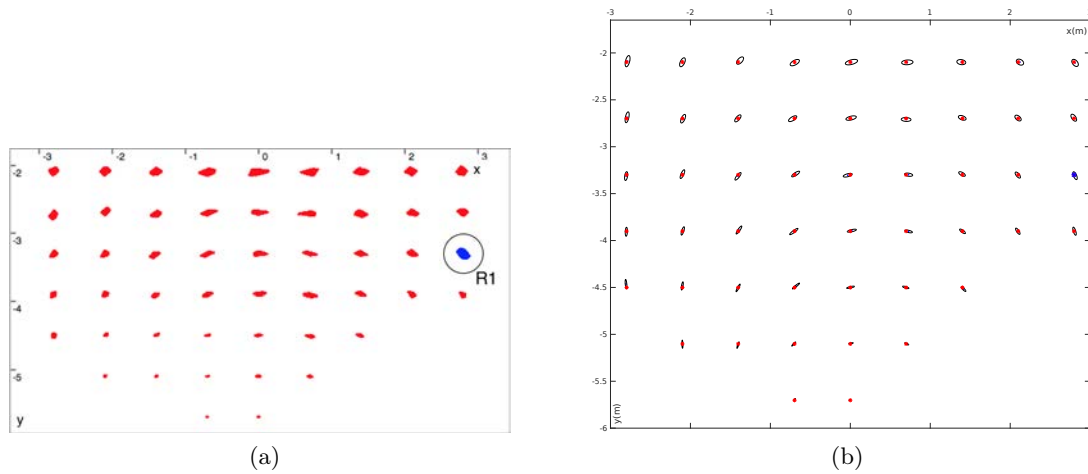


FIGURE 6.7. – À gauche : Domaines de pose pour la localisation coopérative. R_1 est fixé (sous-pavage en bleu, encadré), R_2 à 49 positions différentes (sous-pavages en rouge). À droite : Ellipsoïdes d’incertitude du LM initialisé avec le centre de la boîte englobante du sous-pavage calculé.

6.3.2.4. Localisation coopérative avec visibilité réduite

Afin de mettre en avant les avantages de la méthode de localisation coopérative proposée, nous supposons maintenant que le robot R_1 voit tous les 3 amers, tandis que le robot R_2 n’est capable de voir qu’un seul des trois amers (la figure 6.8).

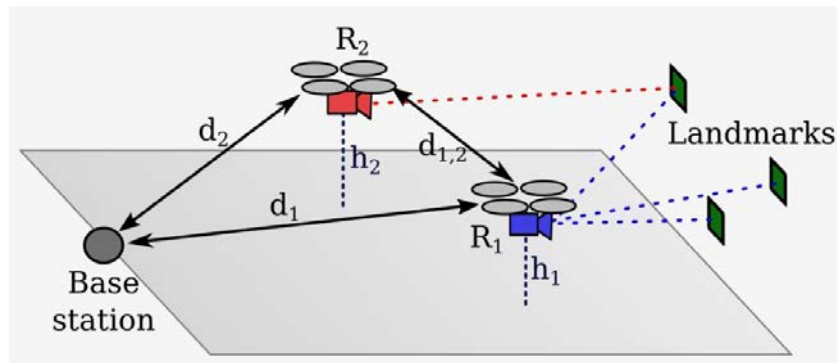


FIGURE 6.8. – Situation de visibilité réduite. les lignes en pointillés bleu et rouge représentent la visibilité d’un amer.

Cette situation peut survenir en raison d’occlusion par des obstacles, ou lorsqu’un opérateur souhaite faire pivoter le drone afin d’observer une autre partie de l’environnement. La localisation avec un seul amer n’est pas possible pour un drone seul car sa position horizontale n’est contrainte que par la distance à la station de base, ce qui, compte tenu des incertitudes et du cap mal connu, conduit à un ensemble de solutions en forme

6. Localisation coopérative ensembliste de drones

d’arc. Ainsi, connaître la distance à un autre drone (qui lui est mieux localisé) permet de calculer un domaine de position utilisable.

Les domaines d’incertitude obtenus, pour les 2450 configurations de simulations possibles, sont plus larges que dans le cas à visibilité totale, en raison du manque de mesures de la caméra (voir le tableau 6.3). Les très grandes valeurs maximales des largeurs de domaines obtenues en x (égale à 3.5 m), en y (égale 0.83 m) et en ψ (égale à 40°) correspondent à quelques cas où l’incertitude de mesure de cap empêche de résoudre l’ambiguïté de position de R_2 , laissant un ensemble de deux domaines de pose disjoints.

Largeur de domaine	min	moyenne	max
x (m)	0.110	0.325	3.573
y (m)	0.029	0.144	0.833
ψ (°)	2.089	5.62	40

TABLE 6.3. – Largeurs de domaines de pose pour R_2 dans le cas de la localisation coopérative à visibilité réduite. R_1 voit 3 amers et R_2 n’en voit qu’un seul.

Comparaison à la méthode des moindres carrés itérés Ces résultats ont été comparés à ceux de l’algorithme LM partant de différentes hypothèses initiales comme dans le cas de visibilité totale.

Concernant les tailles des domaines solutions, la figure 6.9 présente les sous-pavages du domaine de pose (à gauche) et les ellipsoïdes d’incertitudes du LM à 99% (à droite) obtenus pour le robot R_2 (en *rouge*), pour 4 positions différentes du robot R_1 (dont le domaine de position est de couleur *bleue* et entouré). Lorsque la configuration géométrique des 2 robots est bonne, figures 6.9a et 6.9c, les domaines de poses calculés sont bien sûr plus larges qu’en visibilité totale, mais permettent une localisation sans ambiguïté. Nous remarquons que les ellipsoïdes du *LM+Interval* (figures à droite) sont plus aplatis que les domaines calculés. Ce phénomène est pire lorsque les positions sont mal configurées aux environs de $x = 0$ (figures 6.9b et 6.9d). Ceci s’explique par le fait que la distance à la base et l’inter-distance entre R_1 et R_2 contraignent la position de R_2 à se situer à l’intersection de cercles presque tangents.

Concernant la précision de localisation, le tableau 6.4 rapporte les chiffres permettant de comparer les erreurs de position horizontale et de lacet pour les différentes estimées ponctuelles pour le robot R_2 . La première ligne montre clairement que l’estimée CoB devient imprécise en cas de visibilité réduite. Cette imprécision est principalement due à l’asymétrie des domaines solutions calculés ; elle est également illustrée par l’éloignement considérable de la courbe de distribution cumulée de l’erreur de la CoB de R_2 (lignes en pointillés bleues) par rapport à celle de *LM GT guess* (lignes en tirets rouges) sur la figure 6.10.

La deuxième ligne représente les erreurs de la méthode *Interval + LM*. Elle montre une amélioration considérable des estimations ponctuelles en particulier dans le cas du robot

6. Localisation coopérative ensembliste de drones

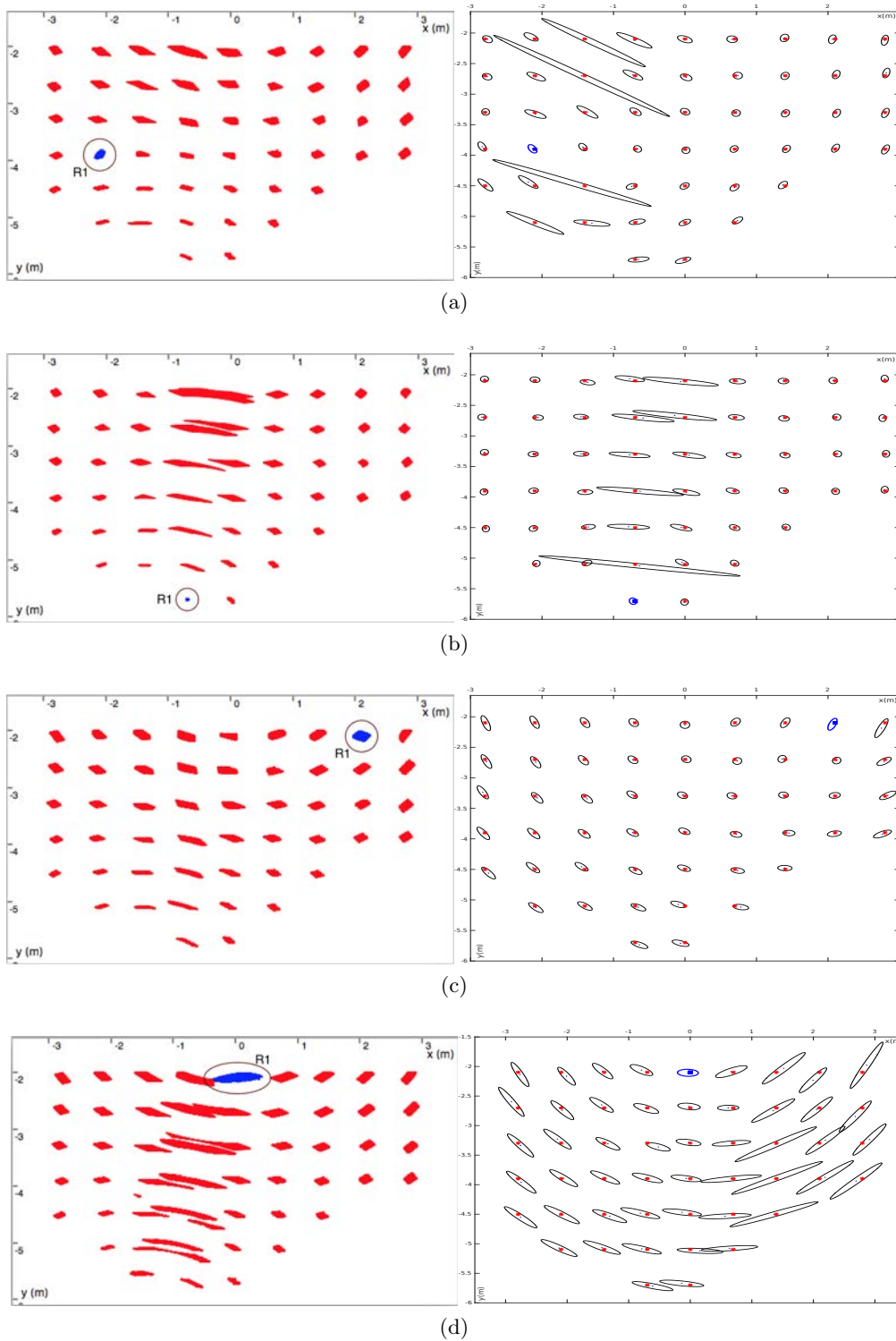


FIGURE 6.9. – Résultats de localisation coopérative R_1 voit 3 amers et R_2 n'en voit qu'un. Chaque sous-figure correspond à une position fixe différente de R_1 (domaine *bleu* encerclé). Les domaines (x_2, y_2) situés à 49 positions de R_2 sont indiqués en *rouge*.

6. Localisation coopérative ensembliste de drones

Méthode	Erreur de position horiz. (cm)			Erreur de lacet (°)		
	moyenne	95 %	max	moyenne	95 %	max
<i>Interval (CoB)</i>	5.39	13.61	167.7	0.80	2.59	19.4
<i>Interval + LM</i>	4.11	7.00	328.4	0.56	1.19	39.4
<i>LM GT guess</i>	2.90	6.68	34.1	0.36	1.08	5.79
<i>LM bad guess</i>	391.9	760.8	1074	140.5	179.0	180.0

TABLE 6.4. – Erreur de position et de cap du drone R_2 dans le cas d’une coopération avec visibilité réduite. R_1 voit tous les amers et R_2 n’en voit qu’un seul.

R_2 . Ce résultat est également illustré par la superposition des courbes noires de la figure 6.10 à celles du *LM GT guess* (en tirets rouges).

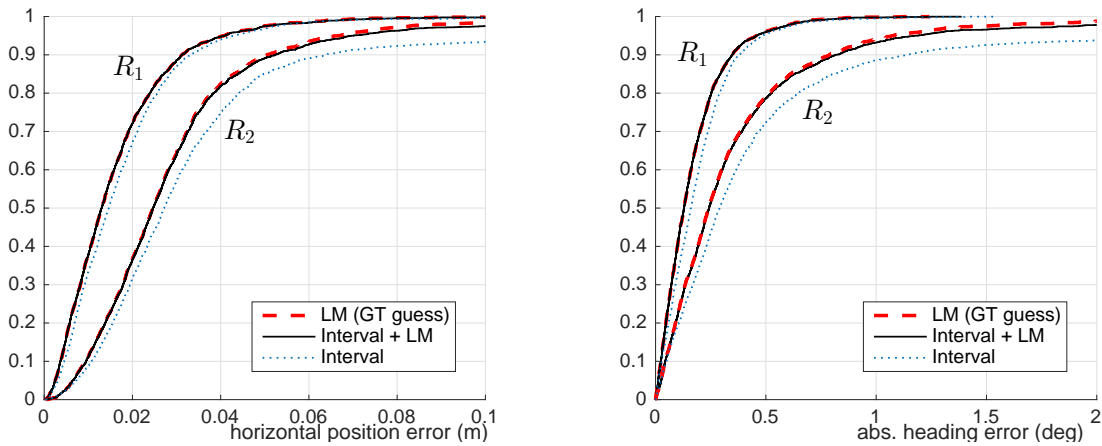


FIGURE 6.10. – Distribution cumulative empirique de l’erreur de position horizontale (à gauche) et de l’erreur absolue de cap (à droite), pour le robot R_1 voyant 3 amers et le robot R_2 n’en voyant qu’un.

Ces résultats montrent que le domaine calculé par la méthode ensembliste, en fournissant un point de départ (CoB) assez proche de la solution exacte, permet de résoudre le problème de localisation initiale dont souffre la méthode des moindres carrés (méthode *Interval+LM*). Contrairement aux ellipsoïdes calculées avec la méthode FOSM, les domaines d’incertitude calculés reflètent fidèlement la géométrie du problème.

6.3.3. Résultats expérimentaux et comparaison

Nous appliquons dans cette section la méthode de localisation coopérative par partage de mesures sur des données réelles.

6.3.3.1. Scénario expérimental

L'environnement expérimental est constitué d'un système de capture de mouvement Vicon qui est capable de suivre la pose d'un drone quadrirotor (modèle MK-Quadro de MikroKopter équipé d'une caméra) afin de fournir la vérité terrain (pour les 2 robots) et les positions des amers dans le repère global de référence. Les amers sont des cubes dont les positions sont connues dans le repère. Outre les incertitudes décrites dans la section précédente sur la simulation, cette expérience prend également en compte les petites incertitudes relatives aux angles de roulis et de tangage ($\pm 0.6^\circ$), aux mesures d'altitude ($\pm 1\text{cm}$) et aux positions amers ($\pm 1\text{mm}$). L'expérience consiste principalement à déplacer les drones le long de l'axe x et légèrement vers l'avant (l'axe y) avec de très légers mouvements de lacet.

6.3.3.2. Résultats et comparaison avec un filtre de Kalman étendu

Nous comparons ici les résultats expérimentaux à deux robots de la méthode par intervalles avec l'estimée d'un filtre de Kalman étendu. Le modèle d'observation du filtre intègre les mesures dans l'image, les mesures de distance, l'altitude et les mesures d'orientation. Le modèle d'évolution considéré pour ces résultats n'intègre pas la mesure de déplacement du robot : le mouvement du robot est considéré comme une perturbation.

Les bornes des domaines de poses calculés pour les deux robots sont reportées à la figure 6.11 (courbes fines en *rouge*), ainsi que l'estimée ponctuelle de l'EKF (*en bleu*). Comme dans la simulation, l'incertitude est plus grande sur la composante x de la position (largeur moyenne de 36 cm) que sur y (largeur moyenne de 17 cm), ce qui est dû à la géométrie du problème. Ceci peut être observé sur la figure 6.12, où les domaines de position calculés pour les deux robots R_1 (en bleu à droite) et R_2 (en rouge à gauche) sont représentés à l'instant $t=0$ s.

La première ligne du tableau 6.5 rapporte les erreurs de position et de cap obtenues pendant l'expérience, lors de l'utilisation du CoB comme estimation ponctuelle. L'erreur horizontale moyenne est de 3,03 cm et l'erreur de cap moyenne est inférieure à 0,6 degré. Un EKF a été appliqué aux mêmes données, la matrice de covariance de l'erreur d'observation ayant été définie de manière identique que l'algorithme LM.

Les résultats de l'EKF (deuxième ligne du tableau 6.5 et courbe *en bleu* de la figure 6.11) montrent que la méthode par inversion ensembliste proposée surpasse l'EKF en termes de précision de la position. La valeur initiale de la pose de l'EKF à $t=0$ s a été définie à partir de la position réelle afin d'éviter tout problème de localisation initiale et de maintenir la comparaison équitable. Les mauvais résultats de l'EKF sur l'estimation de la position sont principalement dus à l'hypothèse d'un bruit centré, blanc et indépendant sur les mesures, ce qui n'est pas le cas en pratique en raison de la corrélation entre les erreurs sur la position des amers (par exemple, dues au flou dans l'image suite au mouvement) et les biais. Cependant, l'EKF donne une meilleure estimation du cap que la méthode par intervalles. Une explication vient du fait que le paramètre de bisection

6. Localisation coopérative ensembliste de drones

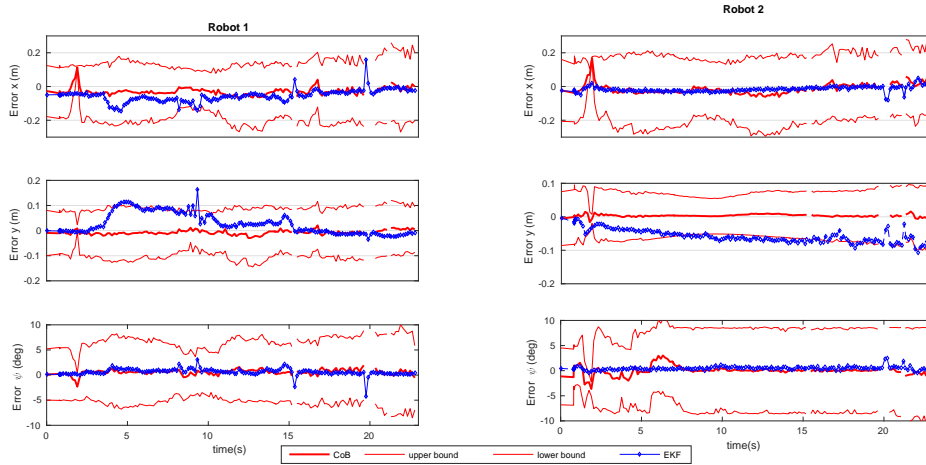


FIGURE 6.11. – Bornes d’erreurs et estimation ponctuelle dans le cas de la localisation coopérative à 2 robots. Zéro correspond à la vérité terrain.

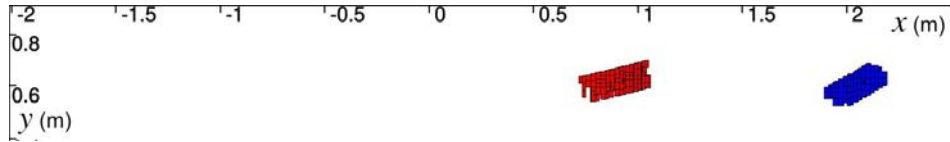


FIGURE 6.12. – Domaines de position des deux UAVs au début de l’expérience : R_1 en bleu (à droite), R_2 en rouge (à gauche).

ϵ de l’algorithme SIVIA a été réglé de manière à effectuer très peu de bisections sur l’angle. Cette limitation de la précision de l’approximation du domaine induit de plus larges domaines sur l’intervalle d’angle.

Méthode	Erreur de position hor. (cm)			Erreur de cap ($^\circ$)		
	moyenne	95 %	max	moyenne	95 %	max
Interval	3.03	6.01	7.22	0.58	1.63	3.02
EKF	4.2	9.47	20.6	0.26	0.66	5.25

TABLE 6.5. – Erreur de position et de cap dans l’expérience coopérative à 2 robots, pour les instants sans faute détectée (méthode par intervalles et EKF).

Plusieurs fois durant l’expérience, le suivi des amers dans l’image a échoué entraînant de ce fait un violation des bornes d’erreurs des points 2D dans l’image. Aux instants $t \approx 16s$ et $t = 20s$, le résultat de la méthode par intervalles est l’ensemble vide, matérialisé par les espaces blancs de la figure 6.11. Ces ensembles vides sont la conséquence directe de de l’inconsistance des mesures, c’est-à-dire qu’aucune pose ne satisfait à toutes les contraintes de l’équation 6.6. Cette inconsistance permet de détecter la présence de

6. Localisation coopérative ensembliste de drones

mesures aberrantes. Aux environs de $t \approx 2s$, nous observons que les domaines de pose calculés sont très petits et ne contiennent pas les vraies poses des robots. Cela correspond toujours à une violation des bornes d'erreur, mais l'écart n'est pas assez important pour générer un ensemble vide. On peut noter que, pendant ces instants de défaillance du système de tracking, la précision de l'EKF est également affectée de manière négative, avec des erreurs moyennes de 6.4 cm en position et de 1° pour le cap (contre 4.2 cm et 0.2° pour les instants où aucune faute n'est rencontrée).

6.3.4. Problème de fusion centralisée (non passage à l'échelle du problème)

Dans cette partie, chacun des robots a effectué la totalité du calcul des deux poses jointes grâce aux mesures échangées entre robots.

Les résultats obtenus pour un nombre $n = 2$ robots tant en simulation que dans des situations expérimentales réelles ont montré que le problème d'estimation coopérative peut être résolu par la méthode ensembliste par intervalles. Il a été possible de quantifier l'incertitude sur les poses de deux drones communicants en calculant un domaine de pose avec l'algorithme SIVIA+HC4.

Pour un nombre croissant de robots ($n \geq 3$) la taille des dimensions de bissection de SIVIA passe de 6 (nombre de drones \times dimensions de bisections=3) à $n \times 3$. Cette croissance rend le problème intraitable. Afin de permettre un passage à l'échelle du problème nous proposons dans la section suivante une solution distribuée basée sur l'échange de domaines de positions.

6.4. Estimation coopérative par partage de domaines de position

Nous proposons dans cette section que les robots s'échangent des domaines de position, chaque robot calculant uniquement sa propre pose. De cette manière, quelque soit le nombre n de robots, le nombre de dimensions de bisections reste inchangé (égal à 3). Nous pouvons donc à partir de cela définir une méthode d'estimation en deux étapes représentées par la figure 6.13.

A chaque instant, chaque robot $R_k (k \in 1 \dots n)$ du groupe estime le domaine de sa pose en utilisant les mesures dans l'image et de distance à la base (mesures dépendantes du robot R_k uniquement). Ensuite, chacun échange son sous-pavage obtenu avec l'ensemble de ses voisins et raffine sa position grâce aux contraintes de distances entre chaque robot R_k et ses voisins $R_j, j \in \mathcal{N}(k)$.

Pour résoudre chaque étape, nous définissons deux problèmes de satisfaction de contraintes pour lesquels nous caractérisons un ensemble solution à partir de l'algorithme SIVIA+HC4.

6. Localisation coopérative ensembliste de drones

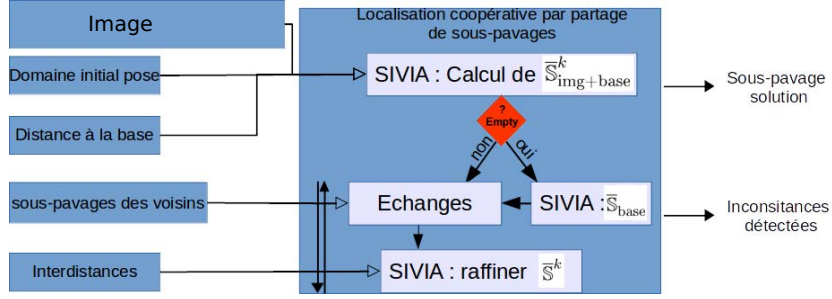
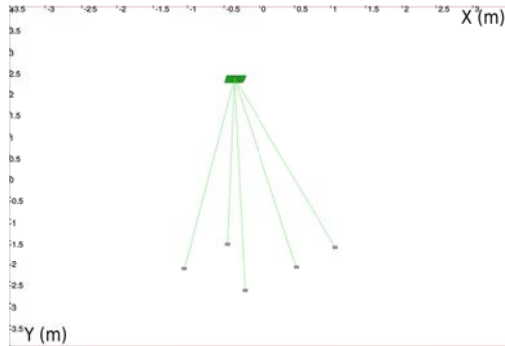


FIGURE 6.13. – Méthode d'estimation de pose ensembliste distribuée.

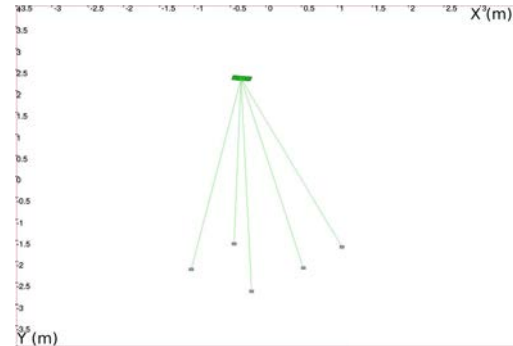
6.4.1. CSP lié aux mesures propres d'un robot (vision et distance à la base)

A chaque instant, R_k calcule tout d'abord une approximation extérieure $\bar{S}_{\text{img+base}}^k$, de l'ensemble solution du CSP $\mathcal{H}_{\text{img+base}}$ défini à partir des contraintes de la section 6.2.2 et donné par :

$$\mathcal{H}_{\text{img+base}} : \left(\begin{array}{l} \mathbf{r}_k \in [\mathbf{r}_k], \\ \bar{\mathbf{x}}_i^k \in [\bar{\mathbf{x}}_i^k], \\ w \mathbf{X}_i^k \in [w \mathbf{X}_i^k], \\ d_k \in [d_k], \\ \{C_{proj,i}^k, C_{front,i}^k, C_{base}^k\}, i \in 1 \dots N \end{array} \right) \quad (6.7)$$



(a) Plan (x, y) du domaine de pose 1 robot avec caméra seule.



(b) Plan (x, y) du domaine de pose 1 robot avec caméra et mesure de distance à la station de base B .

FIGURE 6.14. – Calcul du domaine de pose pour un seul drone (projection horizontale).

L'ajout de la mesure de distance à la base permet de raffiner la position du robot. Nous pouvons observer ce resserrement des bornes par le résultat de la figure 6.14 où un drone observe cinq amers et dispose d'une mesure de sa distance par rapport à la base. La sous-figure 6.14a présente le sous-pavage résultat pour les mesures dans l'image uniquement. La

6. Localisation coopérative ensembliste de drones

sous-figure 6.14b quant-à elle montre le résultat avec mesure dans l'image et de distance à la base.

6.4.2. CSP lié aux mesures d'inter-distances pour la coopération

Après avoir calculé l'ensemble $\bar{\mathbb{S}}_{\text{img+base}}^k$ des poses compatibles avec la mesure de distance à la base et les mesures dans l'image, R_k calcule la boîte englobante $[\mathbf{p}_k]$ de son domaine de position à partir de la formule :

$$[\mathbf{p}_k] = \left[\text{proj}_{\mathbf{p}}(\bar{\mathbb{S}}_{\text{img+base}}^k) \right]$$

où $[\cdot]$ est l'opérateur de boîte englobante et $\text{proj}_{\mathbf{p}}$ la projection de la pose dans l'espace position.

$$\text{proj}_{\mathbf{p}} : \mathbb{R}^6 \rightarrow \mathbb{R}^3$$

$$\begin{pmatrix} [x] \\ [y] \\ [z] \\ [\phi] \\ [\theta] \\ [\psi] \end{pmatrix} \rightarrow \begin{pmatrix} [x] \\ [y] \\ [z] \end{pmatrix}$$

Le domaine $[\mathbf{p}_k]$ est ensuite transmis à tous les voisins $R_j, j \in \mathcal{N}(k)$ du robot R_k , et les distances $d_{k,j}$ entre R_k et R_j sont simultanément mesurées.

6.4.2.1. Contraction de la pose

Le robot R_k peut, à la réception des informations venant de ses voisins (boîtes de positions $[\mathbf{p}_j]$ et mesures de distances à erreurs bornées $[d_{k,j}]$), raffiner son domaine de pose $\bar{\mathbb{S}}_{\text{img+base}}^k$. Pour ce faire, on propage l'information sur les nouvelles mesures d'inter-distance en contractant localement le CSP $\mathcal{H}_{\text{inter}}$ défini comme suit :

$$\mathcal{H}_{\text{inter}} : \begin{pmatrix} \mathbf{r}_k \in \bar{\mathbb{S}}_{\text{img+base}}^k \\ \mathbf{p}_k = \text{proj}_{\mathbf{p}}(\mathbf{r}_k), \\ \mathbf{p}_j \in [\mathbf{p}_j], j \in \mathcal{N}(k) \\ d_{k,j} \in [d_{k,j}], j \in \mathcal{N}(k) \\ d_{k,j} = \|\mathbf{p}_k - \mathbf{p}_j\|_2, j \in \mathcal{N}(k) \end{pmatrix} \quad (6.8)$$

où $\mathbf{p}_k = (x_k, y_k, z_k)^\top$ est la position de R_k . Le domaine de pose $\bar{\mathbb{S}}_{\text{img+base}}^k$ est donc réduit en utilisant la propagation de contraintes par intervalles pour résoudre $\mathcal{H}_{\text{inter}}$ afin de donner l'ensemble solution final $\bar{\mathbb{S}}^k$.

6.4.2.2. Atteinte d'un point fixe

Le réseau de contraintes formé par le groupe de robots peut contenir des cycles couvrant plusieurs robots. La contraction des CSP locaux $\mathcal{H}_{\text{inter}}$ doit être propagée à nouveau à travers le réseau pour améliorer la réduction du domaine de pose de chaque robot.

Si, après la résolution de $\mathcal{H}_{\text{inter}}$ la boîte englobante du domaine de position du robot $[\mathbf{p}_k]$ est réduite, R_k retransmet la mise à jour de $[\mathbf{p}_k]$ à son voisinage. Ce processus est itéré jusqu'à ce qu'un point fixe soit atteint, c'est à dire jusqu'à ce qu'il n'y ait plus d'amélioration significative des domaines de positions.

6.4.3. Résultats expérimentaux

La méthode proposée a été testée pour un nombre drones $n = 4$. Les données ont été acquises sur des drones de modèle Parrot AR-Drone2, capable d'observer cinq amers sur lesquels sont disposés des marqueurs AprilTag². La figure 6.15 illustre les vues internes des caméras pour chaque drone à l'instant $t = 8$ s de l'essai. Les bornes d'erreurs de mesure dans l'image sont définies égales à ± 0.5 px et l'erreur de mesure de distance est supposée être inférieure à ± 5 cm.

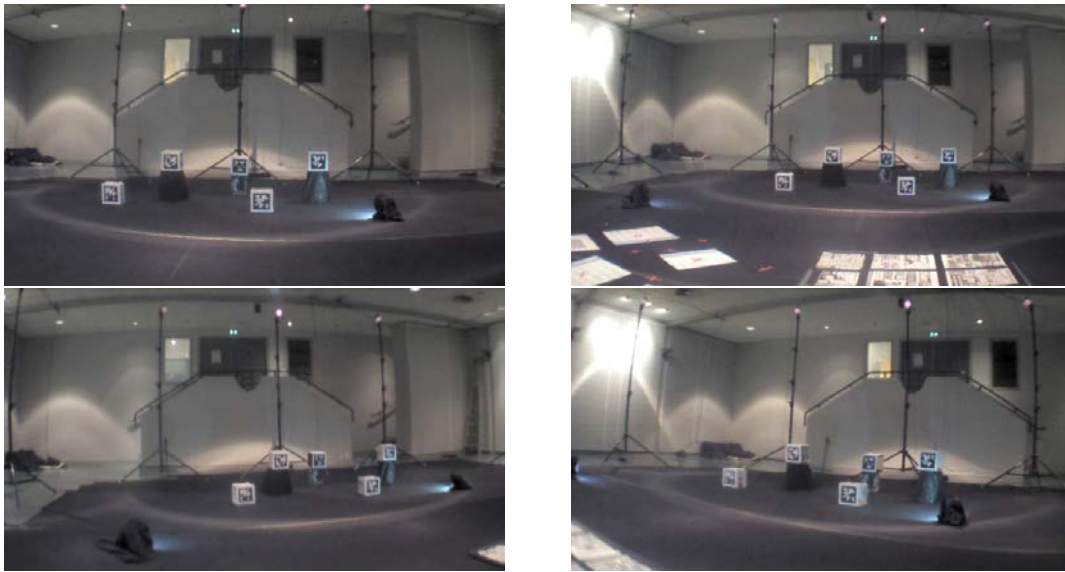


FIGURE 6.15. – Vues intérieures des caméras embarquées pour les 4 drones à l'instant $t = 8$ s. Les amers sont des boîtes avec un marqueur de type AprilTag.

2. <https://april.eecs.umich.edu/software/apriltag.html>

6.4.3.1. Influence de la coopération sur la taille du domaine

Les données acquises ont permis de tester notre méthode pour un nombre croissant de drones (de $n = 1$, pas de coopération ; à $n = 4$). La figure 6.16 présente les sous-pavages obtenus pour le drone R_1 à l'instant $t = 37.8$ s. À cet instant, le suivi des amers dans l'image a complètement échoué pour le drone R_1 ce qui a entraîné une violation des bornes d'erreurs des mesures des points dans l'image. Ainsi, à l'étape 1 de calcul de $\bar{S}_{\text{img}+\text{base}}^1$ la solution est égale à l'ensemble vide. Nous avons implémenté un test de gestion des inconsistances, permettant lorsque la solution avec les données images est vide ($\bar{S}_{\text{img}+\text{base}}^1 = \emptyset$), de calculer un domaine \bar{S}_{base}^1 lié à la mesure de la base uniquement, d'où le domaine de pose de R_1 de la figure 6.16a (un arc épais vert centré sur la position de la base \bar{S}_{base}^1).

La figure 6.16b présente les sous-pavages des drones R_1 et R_2 lorsque $n = 2$, nous remarquons la réduction significative du domaine de position du drone R_1 suite à la coopération avec R_2 bien-qu'il ne soit pas lui-même très bien localisé (capable d'observer uniquement 2 amers). Davantage de contractions sont observées dans les cas où $n = 3$ et $n = 4$ visibles sur les figures 6.16c et 6.16d.

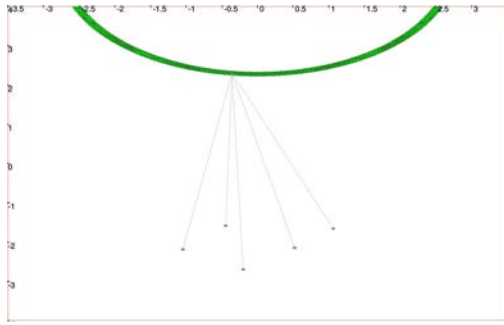
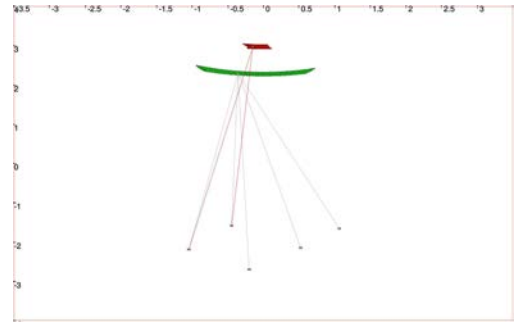
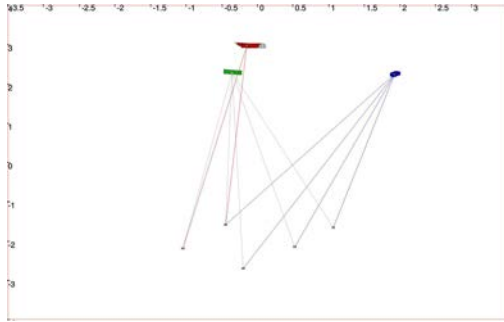
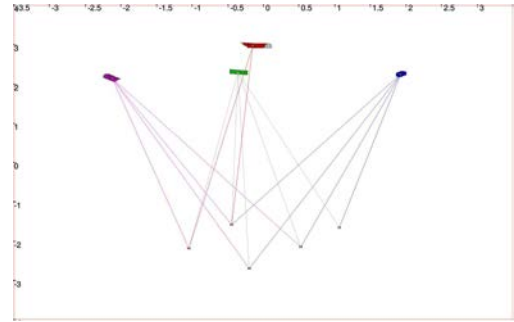
(a) Plan (x, y) du domaine de pose de R_1 (en vert) à l'instant $t = 6.3$ s(b) Plan (x, y) des domaines de poses de R_1 (en vert) et R_2 (en rouge) à l'instant $t = 6.3$ s(c) Plan (x, y) des domaines de poses de R_1 , R_2 et R_3 (en bleu) à l'instant $t = 6.3$ s(d) Plan (x, y) des domaines de poses de R_1 , R_2 , R_3 et R_4 (en rose) à l'instant $t = 6.3$ s

FIGURE 6.16. – Domaines de pose du robot R_1 pour un nombre croissant de robots dans le groupe.

6. Localisation coopérative ensembliste de drones

Ces différentes figures montrent que la coopération permet une meilleure localisation de chaque robot même dans le cas de perte de visibilité complète (cas du robot R_1). Ceci est confirmé par les statistiques des largeurs de domaines et des erreurs de position des 4 robots pour des groupes de tailles différents présentés dans le tableau 6.6. La largeur moyenne du domaine position de R_1 est de 1.29 m lorsque ce dernier se localise sans coopération. Grâce à la coopération, la largeur de son domaine peut être réduite et atteindre une valeur de 33 cm (dans le cas d'un groupe de 4 drones) soit 74% de diminution. L'erreur de position horizontale moyenne pour R_1 en utilisant le CoB comme estimée ponctuelle s'améliore de 22.6 cm à moins de 5 cm (première colonne du tableau 6.6b).

	R_1	R_2	R_3	R_4
1 UAV	1.29			
2 UAVs	0.86	0.56		
3 UAVs	0.80	0.52	2.98	
4 UAVs	0.33	0.37	0.78	0.61

(a) Largeurs moyennes des domaines de position horizontale (m)

	R_1	R_2	R_3	R_4
1 UAV	0.226			
2 UAVs	0.159	0.088		
3 UAVs	0.141	0.071	1.220	
4 UAVs	0.040	0.046	0.248	0.192

(b) Erreurs de position horizontale moyenne (m)

TABLE 6.6. – Résultats de la position horizontale, pour des groupes de 1 à 4 drones.

Avec 4 drones, l'erreur de position horizontale moyenne est d'environ 13 cm pour l'ensemble de la flottille.

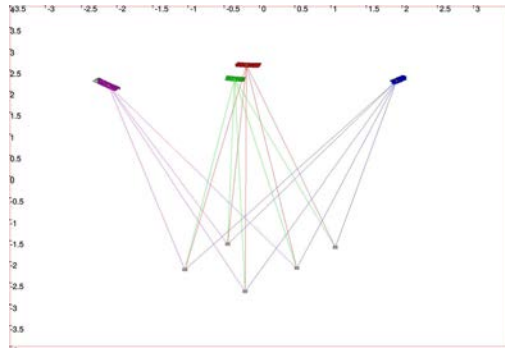
6.4.3.2. Influence de la coopération en cas de visibilité réduite

À certains moments de l'essai, le suivi des amers dans l'image a échoué conduisant ainsi aux violations des bornes d'erreurs.

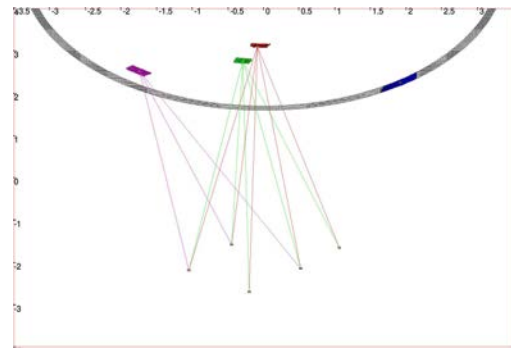
La figure 6.17 présente les sous-pavages résultats de notre approche coopérative pour chacun des quatre drones. La sous-figure 6.17a présente le cas de visibilité complète, où chacun des drones est capable d'observer les 5 amers. Les sous-figures 6.17b, 6.17c et 6.17d correspondent à des cas de visibilité réduite où certains des 4 robots ne sont capables de suivre aucun amer dans l'image. Les contours noirs représentent une inconsistance dans l'image détectée. Dans ces cas, la distance à la base a été utilisée et c'est après calcul du sous-pavage lié à la distance à la base uniquement que la coopération a été effectuée. Les sous-pavages colorés représentent les résultats obtenus pour chaque robot après qu'un point fixe a été atteint à l'étape de coopération.

Ces résultats montrent que si l'un des drones est bien localisé, l'étape de coopération permet de retrouver une pose acceptable pour l'ensemble de ses voisins.

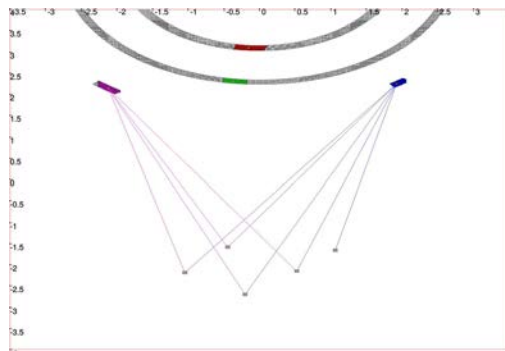
6. Localisation coopérative ensembliste de drones



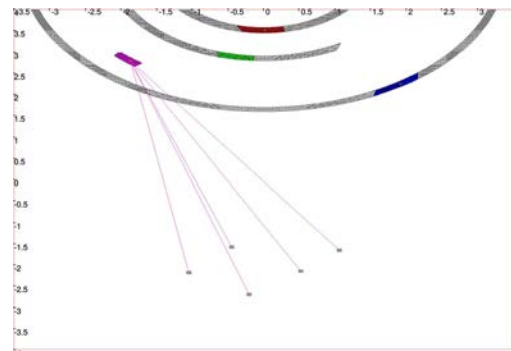
(a) Sous-pavages des 4 drones dans le cas de visibilité complète à $t = 0s$.



(b) Sous-pavages des 4 drones dans le cas de visibilité réduite (R_3 , en *bleu* est aveugle) à $t = 14.8s$.



(c) Sous-pavages des 4 drones dans le cas de visibilité réduite, R_1 (en *vert*) et R_2 (en *rouge*) sont aveugles à $t = 6.4s$.



(d) Sous-pavages des 4 drones dans le cas de visibilité réduite, R_1 (en *vert*), R_2 (en *rouge*) et R_3 (en *bleu*) sont aveugles à $t = 12.2s$.

FIGURE 6.17. – Domaines de position des 4 drones. *Contours noir* : sous-pavage avant la coopération des robots. *Domaines colorés* : sous-pavages obtenus après localisation coopérative.

6.5. Modèle d'estimation/prédiction

Les drones étant équipés de capteurs de vitesse linéaires (par mesure du flot optique) et de vitesse de rotation (gyromètre), le domaine de pose calculé à un instant t donné peut être propagé à l'instant $t + 1$ en intégrant le modèle d'évolution. Se référer à l'équation 4.12 du chapitre 4 pour plus de détails. En supposant des erreurs de mesure bornées, les vitesses mesurées sont représentées par des vecteurs d'intervalles $[\mathbf{v}]$ et $[\mathbf{w}]$.

6.5.1. Réduction du domaine initial à chaque instant

La prédiction d'un instant à l'autre du domaine d'incertitude de la pose permet non seulement un positionnement en continu à une fréquence élevée entre les acquisitions de

6. Localisation coopérative ensembliste de drones

caméras, mais elle permet également de réduire le domaine initial utilisé en entrée de SIVIA.

Cela permet d'augmenter la vitesse de calcul de la méthode du fait qu'à l'instant $t + 1$ la boîte prédite est considérablement plus petite que le domaine initial utilisé dans le cas d'une estimation pure sans prédiction (*snapshot*). La figure 6.18 présente par exemple, les domaines calculés pour les 4 drones entre les instants $t = 3.4$ s et $t = 3.46$ s. La première sous-figure présente l'instant $t = 3.4$ s, à gauche, le sous-pavage résultat dans le cas de l'estimation pure (donc à chaque instant le domaine initial est réglé avec les dimensions de la salle expérimentale pour les paramètres inconnus de l'état). La figure de droite montre le même instant mais avec les domaines initiaux correspondant à la boîte prédite à l'instant précédent. Nous observons que le domaine est considérablement réduit.

A l'instant $t = 3.42$ s nous observons que la taille des domaines est améliorée par rapport au cas *snapshot*.

Pour le drone R_1 la largeur de domaine en *snapshot* est en moyenne 2 fois plus grande que dans le cas estimation/prédiction, tandis que dans le cas du drone R_2 elle est 1.85 fois plus grande (respectivement 1.5 fois pour R_3 et 1.46 fois pour R_4). Ces données montrent que sur l'ensemble de l'essai, quelque soit le drone considéré, la largeur du domaine calculé par l'algorithme d'estimation/prédiction est inférieure à celle dans le cas *snapshot* (estimation seule).

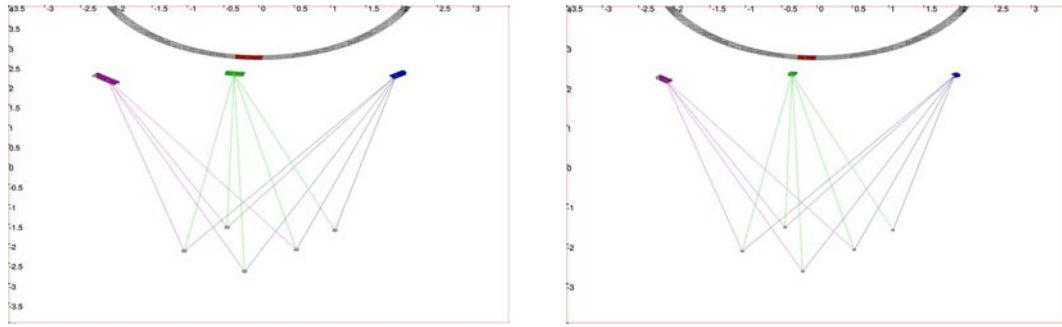
La prédiction permet donc une meilleure caractérisation des domaines de poses sous réserve que les bornes d'erreurs soient respectées. Toutefois des cas de propagation des inconsistances dans le temps ont été relevés et seront détaillés dans la suite.

6.5.2. Présence des inconsistances

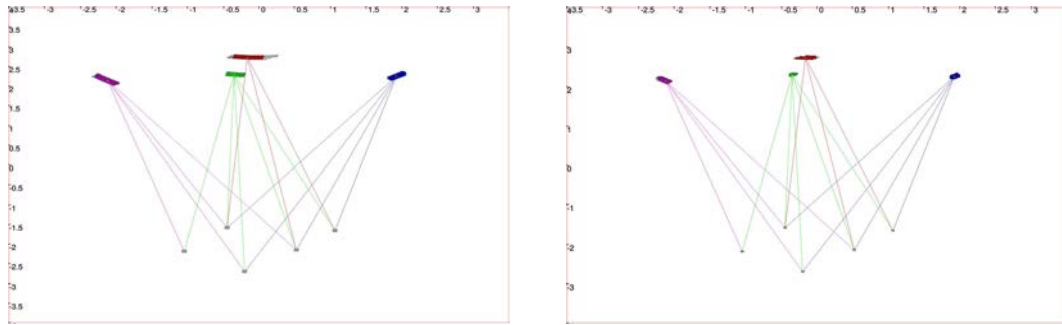
Rappelons que le résultat d'estimation de la pose n'est garanti que lorsque les bornes des erreurs de mesures ne sont pas violées. Nous avons vu au chapitre 4 qu'à cause des problèmes de tracking dans l'image, les bornes d'erreurs peuvent être dépassées. Dans de tels cas, l'ensemble solution retourné est soit l'ensemble vide, soit un domaine qui n'est pas assuré de contenir la vraie pose. Dans le cas coopératif, plusieurs données peuvent être sources de fautes. Sur l'essai complet, des instants de temps ont conduit à des anomalies tel que présenté à la figure 6.19.

- *Missed detection locale* : la sous-figure 6.19a présente les sous-pavages de l'instant $t = 7.14$ s. Pour le drone R_2 , dans le cas estimation/prédiction, le domaine obtenu bien que très réduit et non-vide, ne contient pas la solution. Ce cas correspond au fait que les bornes d'erreurs sont violées mais le décalage n'est pas assez significatif pour conduire à une ensemble solution vide.
- *Missed detection globale* : quelques instants après, à $t = 7.24$ s, tous les autres drones R_1 (en vert), R_3 (en bleu) et R_4 (en rose) expérimentent la même situation (ce qui n'est pas le cas en *snapshot*). L'ensemble solution pour chacun n'est pas

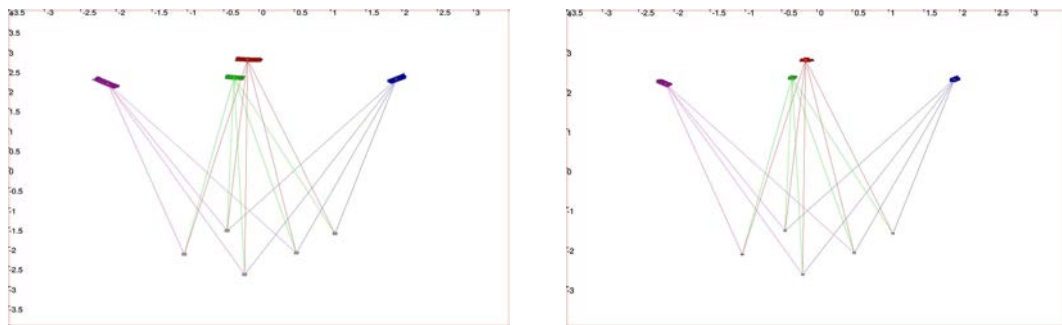
6. Localisation coopérative ensembliste de drones



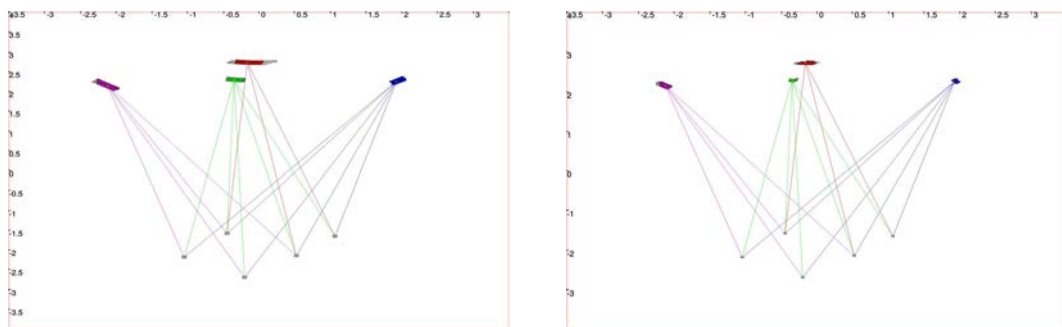
(a) $t = 3.4$ s



(b) $t = 3.42$ s



(c) $t = 3.44$ s



(d) $t = 3.46$ s

FIGURE 6.18. – Résultats d'estimation en *snapshot* (à *gauche*) et avec estimation/prédiction pour des instants successifs (à *droite*).

6. Localisation coopérative ensembliste de drones

vide, mais ils ne contiennent pas la pose solution. D'autres cas peuvent être observé aux instants $t = 7.82s$ et $t = 8.04s$.

Ces inconsistances peuvent provenir :

- des violations des bornes d'erreurs de mesures dans l'image pour chaque robot,
- des violations des bornes d'erreurs de mesures des vitesses
- et/ou du modèle de prédiction erroné (prédiction non garantie)

Il est important de noter que, dans le cas coopératif, les mesures erronées d'un robot peuvent se propager au groupe. De plus, l'utilisation d'une méthode de type estimation/prédiction peut propager une solution erronée au cours du temps.

Il est possible de traiter chacune de ces sources grâce à l'intersection q-relaxée. L'un des solutions consiste à appliquer une intersection q-relaxée à chaque fois où des données (potentiellement sources d'inconsistances) sont fusionnées et, d'effectuer ensuite une intégration garantie au niveau de la prédiction.

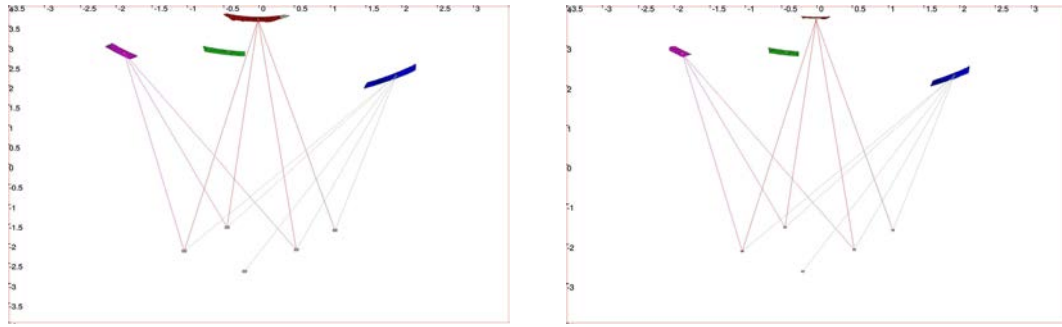
6.6. Conclusion

Dans ce chapitre, nous avons proposé différentes approches basées sur l'analyse par intervalle pour résoudre le problème de localisation coopérative pour un groupe de véhicules aériens sans pilote. La particularité de chacune repose sur le type d'informations échangés entre les drones du groupe.

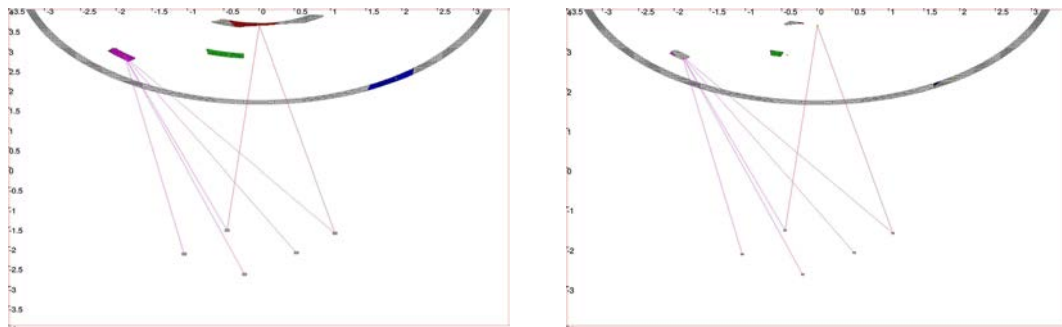
Lorsque ces derniers échangent leurs mesures, chaque drone estime de façon jointe sa position et celles des autres drones de la flottille. Les résultats pour cette méthode ont montré que la coopération permet de réduire le domaine solution calculé, et en cas de manque de mesures dans l'image, la coopération est un atout majeur pour l'obtention d'un domaine exploitable. Non seulement des bornes garanties peuvent être fournies, mais il est également possible d'avoir une estimation ponctuelle de précision satisfaisante en prenant le centre de la boîte englobante (CoB) du domaine calculé. Ceci nous a permis de comparer la méthode à d'autres approches classiques existantes telles qu'un estimateur aux moindres carrés (LM) et un filtre de Kalman étendu. Cette comparaison a montré que le CoB permet de résoudre le problème de localisation initiale dont souffrent ces méthodes classiques qui sont locales et par conséquent nécessitent une initialisation assez proche de la solution exacte pour converger. Bien-que fournissant de bons résultats dans le cas à deux robots, le passage à l'échelle du nombre de robots du groupe était impossible car le domaine des variables inconnues sur lequel l'algorithme SIVIA utilisé effectue des bisections devenait arbitrairement grand, rendant le problème intraitable dans un temps compatible avec la localisation de robots mobiles.

Une deuxième approche, reposant sur l'échange de domaines de position a ensuite été proposée. Extensible et ayant des résultats similaires voir meilleurs que le cas précédent, cette approche a permis d'exprimer le problème et de le résoudre pour un nombre N quelconque de robots. Des résultats jusque 4 robots ont été présentés avec un temps de calcul moyen pour chacun des drones égale à 350 ms sans avoir implémenté d'optimisation

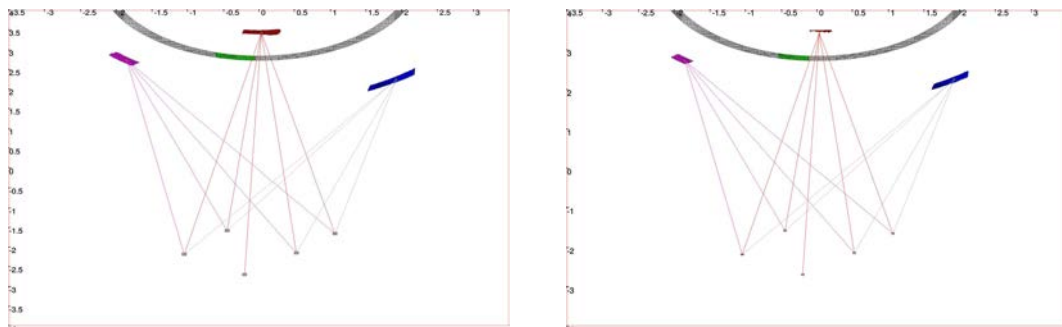
6. Localisation coopérative ensembliste de drones



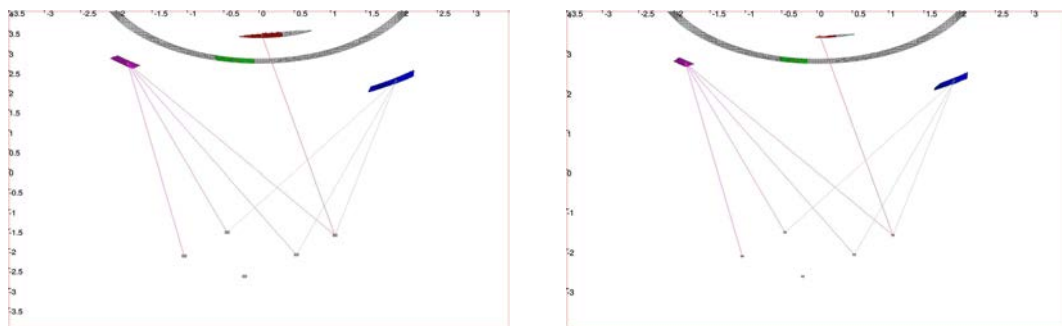
(a) $t = 7.14s$ L'ensemble solution du drone R_2 (en rouge) bien-que réduit ne contient pas l'ensemble solution ce qui n'est pas le cas dans la cas *snapshot*.



(b) $t = 7.24s$



(c) $t = 7.82s$



(d) $t = 8.04s$

FIGURE 6.19. – Cas où la prédiction conduit à une inconsistance, la solution n'est pas contenue dans l'ensemble retourné et (ou) l'ensemble solution est vide. A *gauche* : estimation seule, à *droite* : estimation/prédiction.

6. Localisation coopérative ensembliste de drones

de code. Nous avons également montré qu'il est possible d'effectuer un suivi grâce à l'intégration d'une étape de prédiction dans le procédé de calcul. Les résultats ont montré que la prédiction permet de réduire considérablement le domaine initial des variables conduisant à une diminution du temps de calcul de la méthode entre deux instants consécutifs.

Additionnellement, les résultats ont montré que la coopération entre robots est un atout pour un fonctionnement en groupe lorsque plusieurs des robots ont une visibilité réduite ou sont complètement aveugles.

Quelques cas de fautes peuvent se présenter lorsque les bornes des erreurs de mesures sont violées. Dans ces cas, le résultat n'est plus garanti de contenir la solution exacte. Nous avons présenté quelques exemples de ces cas et une approche de gestion de ces inconsistances est proposée en perspectives de ces travaux.

7. Conclusion et perspectives

7.1. Conclusion

Les travaux présentés dans ce manuscrit ont visé à proposer des stratégies pour l'estimation ensembliste des poses de drones communicants d'une flottille à partir de la vision, de mesures d'inter-distances et de mesures de capteurs proprioceptifs.

Après avoir présenté quelques éléments de bases sur l'estimation de pose basée sur la vision (chapitre 2) et sur l'outil que représente les méthodes ensemblistes par intervalles (chapitre 3), nous avons au chapitre 4, proposé une approche ensembliste d'estimation de pose pour un seul drone. Elle constitue une application directe de l'algorithme d'inversion ensembliste SIVIA pour la caractérisation d'un domaine de pose. Plusieurs contracteurs par propagation de contraintes et par découpage ont été implémentés et leurs efficacités testées pour la contrainte d'appartenance d'un amer au champ de vue de la caméra monté sur le drone. Ceci a permis de caractériser le domaine de la pose du drone en se basant uniquement sur la vision. La détection des mesures aberrantes a également été rendue possible et un procédé de réinitialisation a été implémenté au fil de l'exécution de l'algorithme afin d'éviter la propagation des inconsistances dans le temps.

Par la suite, au chapitre 5, nous avons proposé une méthode d'élimination de quantificateur pour la résolution d'un problème d'inversion ensembliste quantifié. En effet, la présence d'incertitude sur les positions des amers géo-référencées est une source importante de pessimisme. Nous avons donc proposée une méthode géométrique à l'élimination de quantificateur afin de permettre la prise en compte de cette incertitude sur la position des amers.

Pour finir, au chapitre 6 nous avons présenté nos deux approches pour l'estimation coopérative ensembliste de la pose de drones dans un groupe de drones communicants. La première repose sur le partage de mesures entre les drones de la flottille et le problème s'est avéré ne pas passer à l'échelle car chaque drone estime les poses jointes du groupe, induisant un nombre élevé de dimensions de bisections. Une seconde approche consistant à un partage de domaines de positions entre robots, rend le problème moins complexe en nombre de bisections et ainsi permet de passer à une échelle supérieure en terme de nombre de robot de la flotte. Ces deux méthodes ont permis de quantifier l'incertitude de positionnement de chacun des drones. Outre les domaines permettant de quantifier l'incertitude de positionnement, nous avons proposé une estimée ponctuelle qui a rendu possible la comparaison aux méthodes classiques d'estimation (dites ponctuelles). Nous avons également combiné l'analyse par intervalles avec les estimateurs aux

7. Conclusion et perspectives

moindres carrées de type Levenberg-Marquardt et un filtre de Kalman étendu pour résoudre le problème de positionnement initial dont souffrent ces méthodes classiques.

Ces travaux ont permis de calculer de manière garantie le domaine de la pose d'un drone à partir de l'observation dans la caméra d'amers géoréférencés, sous hypothèse de mesures à erreurs bornées. Ceci a conduit à une publication présentée lors de la conférence internationale IEEE IROS 2017 à Vancouver (KENMOGNE et al. 2017).

Une méthode basée sur l'élimination de quantificateur permettant une meilleure prise en compte de l'incertitude sur la position des amers géoréférencés dans le cas 2D a été proposée ; donnant lieu à une communication internationale à SCAN 2018 à Tokyo,

Enfin, deux méthodes ensemblistes rendant possible le calcul collaboratif de domaines de position dans une flottille de drones équipés de caméras et de mesure d'interdistances ont été proposées. La première méthode par échange de mesures entre drones a été présentée à la conférence internationale IROS 2018 à Madrid et la seconde méthode par échange de domaines solutions a été présentée lors de CoProD 2018 à Tokyo.

Ces différentes contributions ont donné lieu à des études en simulations et expérimentales sur des données réelles, avec différents modèles de drones. Les résultats obtenus montrent que les méthodes ensemblistes par intervalles sont bien applicables au problème de localisation collaborative de drones dans un environnement contraint en se servant de la vision par ordinateur.

7.2. Perspectives

Élimination de quantificateur pour la prise en compte de l'incertitude sur la position des amers dans le cas 3D

Au chapitre 5, seul un cas 2D simplifié de prise en compte de l'incertitude sur la position des amers a été traité. Si nous considérons l'équation de projection des position 3D des amers dans le plan image (équation 4.5 du chapitre 4), nous aurons des équations non-linéaires dans lesquelles les coordonnées des points 3D apparaissent plusieurs fois.

A cause de cette multi-occurrence des variables angulaires et de la non-linéarité du problème, nous observerons une zone indéterminée plus conséquente dans le résultat de l'application de l'algorithme SIVIA classique au problème 3D quantifié, prenant en compte l'incertitude sur la position des amers. Ainsi, nous proposons d'étendre la procédure adoptée dans le cas 2D afin de permettre une meilleure caractérisation du domaine solution et aussi de réduire le nombre de dimensions de bisections pour améliorer le temps de calcul.

Utilisation d'une fonction d'inclusion centrée plus fine pour la vision

Nous avons remarqué qu'après élimination du quantificateur existentiel au chapitre 5, la solution du problème d'inversion ensembliste comprenait des zones indéterminées dues au pessimisme de la fonction d'inclusion naturelle des équations des droites exprimant les contraintes aux coins de boîtes de l'équation 5.15. Nous proposons pour résoudre ce problème, en utilisant en plus de la fonction d'inclusion naturelle, une forme centrée de fonction d'inclusion sur l'angle (les angles dans le cas 3D) des équations de ces droites.

En ayant une forme linéaire en l'angle, nous pensons que cela permettra de lever l'indétermination pour certaines valeurs d'angles et permettre de rendre le domaine obtenu moins épais.

Suivi de trajectoire coopératif robuste

Nous avons vu au chapitre 4 que les bornes d'erreur des mesures dans l'image peuvent être violées, conduisant à des inconsistances. Dans le cas coopératif, au chapitre 6, outre les mesures de positions 2D dans l'image, les positions des voisins reçues par chaque drone R_k et les inter-distances mesurées sont également être sources d'inconsistances dans le cas coopératif; ceci est principalement causé par les délais de transmission et de réception des informations. L'intégration des vitesses de translation et rotation dans la boucle de calcul peut également introduire plusieurs sources de pessimisme conduisant ainsi à des inconsistances; car les bornes théoriques des erreurs de vitesses données peuvent parfois être plus petites que dans la réalité.

Afin de permettre aux drones de la flottille de suivre une trajectoire définie et d'effectuer un partage de sous-pavage garanti évitant d'arriver à une inconsistance globale, la solution que nous proposons est constituée de deux étapes.

La première consiste à utiliser l'algorithme SIVIA implémentant une intersection q -relaxée à plusieurs niveaux en fonction des sources d'erreurs. Le principe de ce type de fusion a été présenté à la section 3.5. La deuxième consiste à utiliser un historique de sous-pavages lors de la communication avec les voisins afin d'avoir une estimation plus robuste (cette approche a été proposée dans MOURAD et al. 2012 et est similaire, pour un robot à la méthode de DREVELLE et al. 2011).

SLAM (correction des mesures)

Le principe du SLAM (Simultaneous Localization And Mapping), consiste à résoudre les problèmes de localisation (qui correspond à chercher sa position par rapport à un repère connu) et de cartographie (qui est le calcul de la position des éléments constituant son environnement) en même temps.

Nous proposons d'utiliser la méthode ensembliste du chapitre 4, pour résoudre le problème du SLAM (comme traité par LE BARS (2011)). En effet, la parti localisation ayant

7. Conclusion et perspectives

déjà été abordée dans ces travaux, une solution de cartographie serait d'utiliser le système ainsi conçu pour déterminer la position des amers. Les positions incertaines des amers pourront donc être améliorées/corrigées. Ceci permettrait également de considérer des amers mobiles.

Association de données

Dans ces travaux, nous avons considéré connus les correspondances entre positions 2D dans l'image et positions des amers 3D.

En réalité, ces correspondances ne sont pas toujours connues. Très souvent, on connaît juste le nombre d'amers visibles, leurs positions dans l'image et leur géolocalisation ; mais on ne sait pas quel amer correspond à la mesure dans l'image.

À partir du modèle d'observation et de l'estimation faite de la pose du drone, il sera possible de caractériser un domaine de positions d'un amer compatible avec une mesure dans l'image connue. À partir de ce domaine, un algorithme de recherche locale peut être utilisé pour trouver parmi les positions géographiques des amers connus, laquelle correspond à la mesure dans l'image qui est traitée.

Annexes

A. Types de drones à voilure tournante

Cette annexe présente les types de drones à voilure tournante et un aperçu de leur mode de fonctionnement. Ces drones sont capable d'effectuer un vol stationnaire. Ils possèdent tous une force d'élévation des pales induite par leurs hélices tournant sur l'axe d'un rotor en fonction de l'angle d'attaque des pales et de la vitesse de rotation du (ou des) rotor(s). Ces deux sources de forces fournissent deux moyens de contrôler la poussée.

Suivant le nombre de rotors utilisés pour la conception d'un drone, de la position de ceux-ci et de la présence ou non d'actionneurs, il existe plusieurs types de drones à voilure tournantes NONAMI et al. 2010 ; AUSTIN 2011 ; MARSHALL et al. 2016.

A.1. Drones à un rotor

Les drones à un rotor sont des hélicoptères classiques dont deux exemple sont présentés à la figure A.1. À ce jour, les hélicoptères ont l'architecture des drones VTOL la plus maîtrisée et la plus répandue. Ils permettent principalement la validation d'algorithmes développés pour des hélicoptères de taille normale.

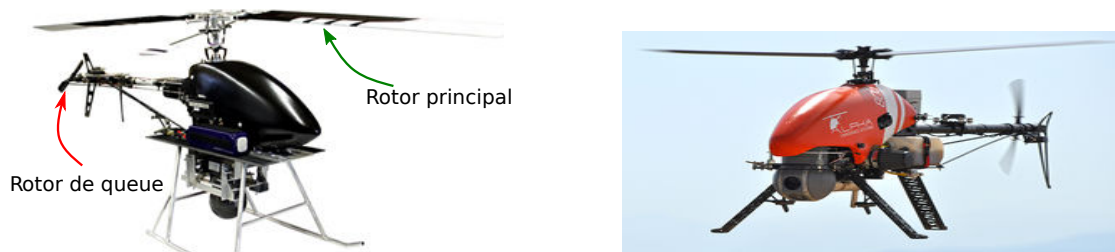


FIGURE A.1. – Hélicoptère à un rotor. À gauche : drone bi-rotor FH-420 (image adaptée de *Drone de surveillance, pour recherche et sauvetage, hélicoptère, bi-rotor FH-420* 2018) pour surveillance, recherche et sauvetage. À droite : drone agricole ALPHA-800 **alpha800**

Ils sont de taille réduite et dotés RIDA 2015 :

- d'un **rotor principal** (monté sur l'axe vertical) qui est le système permettant la propulsion et la sustentation de l'hélicoptère. Constitué d'un mât et d'une tête de rotor sur laquelle sont fixés des pales à travers un composant mécanique appelé plateau cyclique (voir figure A.2). Il est utilisé pour contrôler le tangage et le roulis.

A. Types de drones à voilure tournante

- d'un **rotor de queue** (situé à l'arrière) qui sert d'anti-couple pour contrer la réaction en lacet. Il empêche l'hélicoptère de tourner sur lui-même lorsque le rotor principal tourne.

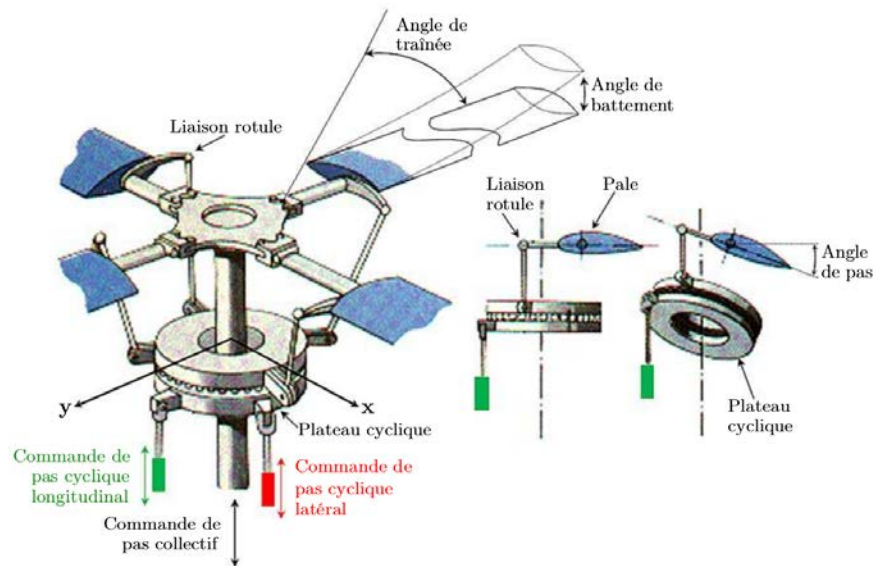


FIGURE A.2. – Fonctionnement d'un rotor principal KOEHL 2012.

Principe de contrôle de l'hélicoptère La propulsion de l'hélicoptère est effectuée à partir des pales du rotor principal. Elles sont entraînées par le moteur à l'aide d'une boîte de transmission. Ces dernières balayent le *disque rotor* pour générer une poussée permettant de compenser le poids de l'appareil afin d'effectuer le contrôle du mouvement vertical de l'hélicoptère. Le disque rotor peut être du type rigide, semi-rigide ou articulé, et permet parfois les variations d'angles de battement et de traînée de l'appareil.

La force de sustentation de l'hélicoptère est contrôlée via le plateau cyclique. Composé de deux demi-plateaux superposés et reliés à l'axe de rotation des rotors par une liaison rotule il permet de produire l'inclinaison des pales et l'inclinaison du plan de rotation. En effet, le demi-plateau inférieur est non tournant et articulé autour de l'axe des rotors par des tiges de commande ; ce qui me rend libre de monter, de descendre ou de s'incliner dans toutes les directions. Le demi-plateau supérieur quand à lui, équipé de biellettes de pas dont chacune est reliée à une pale pour faire varier son angle d'incidence, est tournant et attaché au rotor principal. Le pas de ces deux plateaux (dit « pas collectif ») sert à contrôler l'altitude de l'hélicoptère. La commande du pas collectif est rendue possible par une inclinaison du plateau cyclique par rapport au mât du rotor. Ceci entraîne la modification de l'angle de pas de chacune des pales du rotor principal de sorte qu'un différentiel de traction adéquate est créé pour faire varier l'amplitude de la force de sustentation ; conduisant ainsi la variation de l'attitude de l'appareil.

Si le plateau cyclique est incliné vers l'avant, le disque rotor bascule l'hélicoptère vers

A. Types de drones à voilure tournante

l'avant ce qui lui permet d'avance, respectivement, si l'on incline le plateau cyclique vers l'arrière cela conduira au déplacement vers l'arrière. Pour les mouvements latéraux, il suffira d'effectuer une inclinaison autour de l'axe de roulis.

La puissance créée au niveau du mât rotor peut être compensée à partir du rotor de queue placé à l'extrémité arrière de l'appareil et monté sur l'axe vertical. Ayant un plan de rotation est orthogonal à celui du rotor principal, il sert à contrôler les mouvements de l'hélicoptère autour de l'axe de lacet en produisant une force de poussée latérale qui permet d'éviter une rotation infinie de l'hélicoptère sur lui-même. Le rotor de queue est principalement responsable du bruit et des vibrations qui affectent l'hélicoptère en vol.

Des solutions à ce problème du rotor de queue ont été proposées (l'ajout d'un second rotor principal ou de plusieurs rotors).

A.2. Drones double rotors

Ce sont des drones constitués de deux rotors principaux, sans rotor anti-couple. Les deux rotors principaux peuvent être alignés de manière longitudinale (dit en tandem), alignés de manière latérale (côte à côte ou engrenants) et sous le même axe de rotation (coaxiaux). Ils sont capable d'effectuer un vol stationnaire plus stable grâce aux sens de rotation opposés des deux rotors qui permettent d'annuler leur effet de couple et de contrôler le drone en translation (sauf ceux dit à rotors côte à côte qui tournent dans le même sens). La figure A.3 présente des exemples d'hélicoptères double rotors.



(a) Hélicoptère à rotors en tandem : 2.4Ghz Esky Tandem Rotor 4CH Chinook Helicopter Camo RC Remote Control Radio 2018.



(b) Hélicoptère double rotor coaxial : TRAXXAS DR-1 Double-Rotor coaxial RTF 2018.

FIGURE A.3. – Drones à voilures tournantes à double rotors.

A.3. Drones quadrirotor

Ce sont des hélicoptères équipés rotors principaux. La mécanique de ce type de drone est plus simple que celle des drones vus plus haut. Les rotors avant et arrière ont un sens de rotation contraire à ceux latéraux. Ce qui ramène le contrôle de l'orientation et de la translation de ces derniers à un jeu de vitesses de rotation des différents rotors. Leur principe de fonctionnement est décrit à la section A.5.



FIGURE A.4. – Drones à voilures tournantes à quatre rotors, Quadrirotor agricole Parrot Bluegrass.

Adapté aux vols stationnaires en intérieur, ce type de drones a récemment connu un regain d'intérêt dans les domaines de la robotique KIM et al. 2002 ; STIPANOVIĆ et al. 2004 ; CASTILLO et al. 2006., des transports, de la cartographie, de l'agriculture, du sauvetage, etc.

Fonctionnement du quadrirotor Les mouvements du quadrirotor sont contrôlés par les vitesses de rotation de ses différents rotors. Soient M_1 , M_2 , M_3 et M_4 les quatre rotors du drone (où M_1/M_4 sont les rotors avant et M_2/M_3 les rotors arrière), voir figure A.5a.

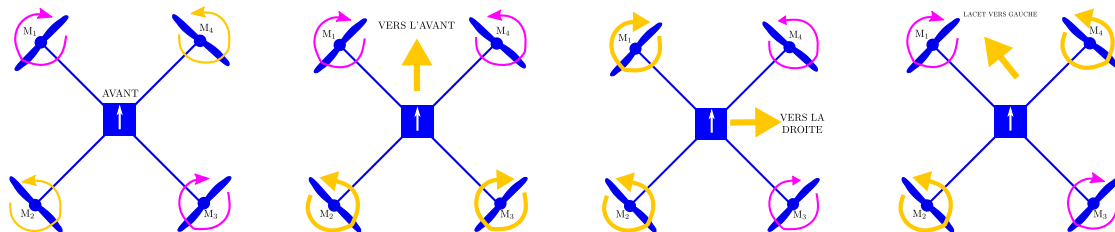


FIGURE A.5. – Fonctionnement d'un quadrirotor. Jeu des couples de vitesses appliqués aux rotors pour les différents mouvements.

Lorsqu'on effectue une rotation des hélices, une force de sustentation est créée (force qui permet de maintenir un corps à courte distance au dessus d'une surface et sans contact avec elle). Cette force va compenser le poids du drone. Lorsqu'elle est supérieure au poids du drone, ce dernier va s'élever dans les airs.

Une fois le drone en l'air, le sens de rotation et la vitesse de rotation des hélices vont permettre de contrôler le déplacement du drone. Les hélices M_1 et M_3 tournent dans le

A. Types de drones à voilure tournante

sens des aiguilles d'une montre, tandis que M_2 et M_4 tournent dans le sens antihoraire ce qui permet d'annuler le couple induit par l'effort sur chaque moteur pour faire tourner les hélices.

Ainsi, pour un déplacement vers l'avant, ou encore une *action de tangage*, il suffira d'augmenter la vitesse de rotation des moteurs arrière (M_2 et M_3) et diminuer la vitesse des rotors avant (M_1 et M_4) voir figure A.5b. Pour l'action inverse (déplacement en translation vers l'arrière, il suffira de faire le mouvement inverse.

Pour un déplacement vers la droite (respectivement vers la gauche), cela correspond à une action en roulis qui consiste à augmenter (resp. diminuer) la vitesse de rotation des moteurs de gauche (M_1 et M_2) et diminuer (resp. augmenter) celle des moteurs de droite (M_3 et M_4) visible à la figure A.5c.

Pour un mouvement du lacet il suffira d'augmenter la vitesse des moteurs situés sur le même. Par exemple pour tourner vers la gauche (respectivement vers la droite) il faut augmenter la vitesse des moteurs (M_2 et M_4) comme illustré à la figure A.5d, (resp. celle de M_1 et M_3).

A.4. Autres

En fonction du nombre de rotors utilisé dans la conception des drones, nous pouvons avoir plusieurs autres types qui en découlent (comme par exemple, les tri-coptères, les hexa-coptères, les octo-coptères, etc.). La figure A.6 présente quelques exemples des autres types de drones présents sur le marché^{1 2 3}.



FIGURE A.6. – Autres types de drones.

1. *Skyborne Technologies - New Tri-rotor UAVs, Project CERBERUS*. Online. Accessed : 2018-10-22. URL : <https://www.qldstartupshowcase.com/uav--unmanned-vehicles-etc/skyborne-technologies-new-tri-rotor-uavs>.

2. *Typhoon H, L'hexacoptère primé avec caméra*. Online. Accessed : 2018-10-22. URL : https://www.yuneec.com/fr_FR/drones-camera/typhoon-h/aperçu.html.

3. *OctoCopter, Drone de série, homologué par la DGAC*. Online. Accessed : 2018-10-22. URL : <https://www.flyingeye.fr/product/octocopter/>.

A.5. Drones utilisés dans ces travaux

Dans cette thèse, nous utiliserons des drones de type quadrirotor.

A.5.1. Quadrirotors utilisés



(a) Drone MK-Quadro de MikroKopter.



(b) PARROT AR.DRONE 2.0 Elite Edition.⁴

FIGURE A.7. – Quadrirotors utilisés pour nos expérimentations.

Le premier drone utilisé dans ces travaux est le quadrirotor de classe Mk-Quadro de MikroKopter⁵. Il est utilisé au chapitre 4 pour la validation expérimentale de notre méthode ensemble de localisation pour un drone. Le drone est également équipé d'une caméra embarquée qui permet l'acquisition d'images de l'environnement de navigation du drone à chaque instant. Grâce à un module de tracking de points de VISP⁶ (*Visual Servoing Platform* développée par l'équipe Rainbow de l'INRIA), il est possible de mesurer les positions 2D d'objets de la scène visibles dans l'image acquise.

Le second drone est l'AR.Drone 2.0 de Parrot. Nous l'utilisons au chapitre 6 pour la validation expérimentale de l'approche de localisation coopérative ensembliste que nous avons proposé.

A.5.2. Composants d'un drone

Un quadrirotor est généralement constitué :

- d'un corps principal sur lequel sont montés quatre bras (disposés en X ou en H) ayant à leurs extrémités un rotor principal. Les 4 moteurs électriques sont alimentés par une batterie Lithium Polymère.

5. <http://wiki.mikrokopter.de/en/MK-Quadro>

6. <https://visp.inria.fr>

A. Types de drones à voilure tournante

- d'une carte électronique embarquée chargée du traitement des données acquises sur les capteurs et définit la commande des 4 variateurs des moteurs en fonction de la consigne à suivre, dénie par l'utilisateur et transmise au drone par liaison Wi-Fi, Bluetooth ou Ethernet.
- de plusieurs capteurs parmi lesquelles
 - la centrale de mesure inertielle ou encore appelée (IMU) « *Inertial Measurement Unit* » (voir figure A.8a) permet la mesure la position angulaire (encore appelée attitude : roulis, tangage et lacet), de la vitesse linéaire (selon les trois axes orthogonaux) et de la vitesse angulaire (vitesse de rotation autour des trois axes orthogonaux).
 - l'accéléromètre (voir figure A.8b) permet de mesurer les accélérations linéaires non-gravitationnelles du drone selon ses trois axes orthogonaux.
 - le magnétomètre (voir figure A.8c) ou compas magnétique permet de mesure la direction (ou intensité) du champ magnétique terrestre du drone (très sensible aux perturbations extérieures du drone).
 - le gyromètre permettant de mesurer la vitesse instantanée de rotation du drone sur les trois axes du repère mobile (vitesses de roulis, de tangage et de lacet). Le gyroscope quant-à lui mesure la position angulaire suivant sa configuration de montage sur le drone. En fonction de cette configuration, un gyro (voir figure A.8d) peut fonctionner comme un gyroscope ou un gyromètre.

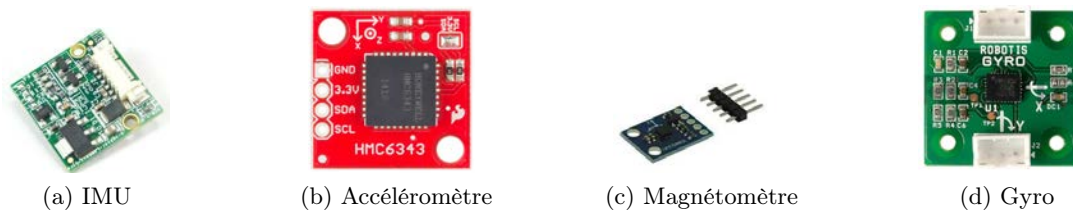


FIGURE A.8. – Exemples de quelques capteurs présents sur un drone.

- le récepteur GPS (pour « *Global Positionning System* ») qui mesure la position du drone. Le système GPS (situé dans la classe des systèmes de positionnement et de navigation par satellites), il est constitué d'un réseau de satellites en orbite autour de la terre à une altitude fixe, qui transmettent des signaux à tous les équipements munis d'un récepteur GPS. Ces signaux permettent d'obtenir la position exacte du récepteur (en indiquant sa longitude, sa latitude et son altitude) quel que soit l'endroit où il se trouve. Le drone est donc muni d'un circuit électronique qui sert de récepteur GPS.
- le télémètre qui mesure la distance du drone à une objet (ou obstacle) grâce à un signal qui peut être laser, ultrasonique ou des ondes radios. Il sert à la mesure de la distance du drone au sol afin de permettre son décollage.
- l'altimètre barométrique qui sert à déterminer l'altitude du drone, très sensible aux changements extérieurs, il mesure le niveau du drone par rapport au niveau de référence (sol) et est très souvent utiliser sur les drones dans le cadre d'un

A. Types de drones à voilure tournante

vol en intérieur

- la caméra permet d'obtenir la position et la vitesse de la caméra par rapport aux objets observés dans l'environnement. Elle est très souvent utilisée pour pallier à l'indisponibilité du GPS dans des environnements contraints et permet d'augmenter la précision du positionnement lors de la navigation. L'utilisation des caméras s'avère extrêmement utile pour réaliser des tâches de localisation, de navigation et d'exploration.

Tous les capteurs cités ci-dessus sont utilisés pour la navigation et la localisation qui constituent l'une des plus importantes tâches du drone DAMIEN 2011. En effet, la stabilité et l'autonomie de l'appareil en dépendent totalement car ils permettent de fournir les informations sur la position, l'attitude et la vitesse permettant de mieux manœuvrer celui-ci.

B. Architecture de l'environnement expérimental

Dans cette annexe, nous présentons brièvement l'environnement expérimental dans lequel les acquisitions des données réelles utilisées pour valider les méthodes proposées dans ce manuscrit ont été menées.

B.1. architecture générale

L'architecture générale de l'environnement expérimentale est constituée de la partie matérielle (Système de capture Vicon et les drones) et de la plateforme logicielle (Telekyb, Matlab) pour l'acquisition.

Les drones sont équipés de marqueurs réfléchissants afin de rendre possible la capture de leurs mouvement par le Système de motion Capture Vicon (décrit à la section B.2). Une fois la position drone mesurée par le Vicon, cette position est transmise à la plateforme logicielle d'expérimentation Telekyb-V-Rep.

Telekyb-V-Rep est utilisée pour contrôler les drones. Elle sert de cerveau et intègre :

- le logiciel de simulation V-REP¹ qui communique avec VISP (*Visual Servoing Platform* développée par l'équipe Rainbow de l'INRIA) utilisé pour le traitement d'images
- la plateforme ROS (Robotic Operating System) qui rend possible les échanges avec le drones
- Matlab/Simulink qui est utilisé pour la programmation des vols et communique avec Telekyb au travers des messages ROS
- un système de communication XBee qui permet l'échange avec les drones

Dans nos expériences, toutes les données nécessaires sont recueillies au travers des messages ROS émis par Telekyb-V-Rep, des messages ROS émis par les cartes embarquées présentent sur les drones et des lectures des données transmises par le système de Capture Vicon.

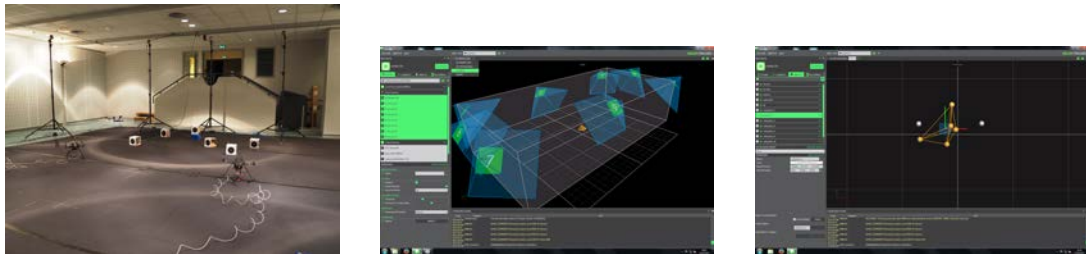
1. <http://www.coppeliarobotics.com/>

B.2. Système de capture de mouvement Vicon

L'arène de vol utilisée dans cette thèse est une salle de 10m de long, 6m de largeur et haute de 3m (figure B.1a). Elle est équipée d'un système de Motion Capture Vicon². La capture du mouvement se fait au travers de 8 caméras infrarouge montées sur des pieds de part et d'autre de la salle. Ces caméras sont connectées à un ordinateur central, servant de station de base. Le logiciel « *Vicon Tracker* » est utilisé sur l'ordinateur afin de pouvoir recueillir les informations des différentes caméras.

La figure B.1b présente une vue du logiciel, montrant les caméras et la zone couverte par celles-ci.

Un objet pour être suivi, doit être équipé de réflecteurs (marqueurs) et son modèle doit être construit dans le logiciel. L'objet représentant le drone Mk-Quadro (utilisé dans cette thèse) est présenté à la figure B.1c. Une fois l'objet construit, sa position dans la salle est suivie et mise à disposition sur l'ordinateur central par le système de capture à une fréquence de 100Hz.



(a) Vue extérieure de la salle de motion capture, avec en plus dans la salle une exemple de *setup* pour une acquisition. (b) Vue de la scène d'acquisition à partir du logiciel de tracking « *Vicon Tracker* » du système de capture de mouvement Vicon. (c) Exemple de création d'un objet dont la mouvement sera suivi depuis le système de capture.

FIGURE B.1. – Constituants de la salle de motion capture.

B.3. Drones expérimentaux

Deux drones de type quadrirotor ont été utilisés menés nos expériences. Chacun d'eux est connecté à l'ordinateur central de la plateforme et leur pilotage peut se faire par un joystick ou le clavier connecté à l'ordinateur au ou par tout autre programme personnalisé.

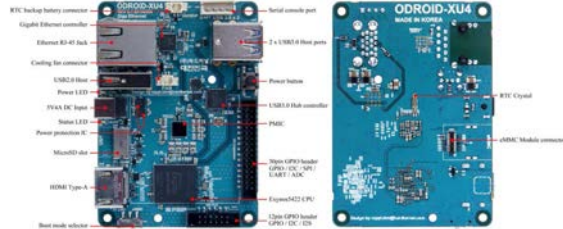
2. <https://www.vicon.com/motion-capture>

B.3.1. Drone de classe MK-Quadro

Le premier drone est le quadrirotor de classe Mk-Quadro de MikroKopter³.



(a) Drone MK-Quadro de MikroKopter.



(b) Carte ODROID-XU4

FIGURE B.2.

La Mk-Quadro habituelle a été étendue avec une carte ODROID-XU4 sur laquelle est installée ROS sous Ubuntu (14.04 ou 16.04).

Grâce aux capteurs présents sur la carte, il est possible pour nous d'avoir les mesures proprioceptives des vitesses linéaires et angulaires du drone à chaque instant. Le drone est également équipé d'une caméra embarquée de classe Flea FL3-U3-32S2C-CS de FLIR (ancien Point-Grey⁴). Elle permet l'acquisition d'images de l'environnement de navigation du drone à chaque instant. Grâce au module de tracking de points $VpDot2$ ⁵ de , il est possible de mesurer les positions 2D d'objets de la scène visibles dans l'image acquise.

B.3.2. Drone de classe AR.Drone

Le second drone est un quadrirotor du fabricant Parrot, de classe AR.DRONE 2.0 Elite Edition (figure B.3).

Il est équipé d'une **batterie** présentée à la figure B.4a, qui permet d'alimenter les moteurs et la **carte électronique** qui est le cerveau du drone (voir figure B.4b). À la carte, sont connectés un accéléromètre, un baromètre, un gyroscope, un magnétomètre et en dessous du drone, un capteur ultrasonique (voir figure B.4c).

3. <http://wiki.mikrokopter.de/en/MK-Quadro>

4. <https://eu.ptgrey.com/>

5. <https://visp.inria.fr/blob/>

B. Architecture de l'environnement expérimental



FIGURE B.3. – PARROT AR.DRONE 2.0 Elite Edition.

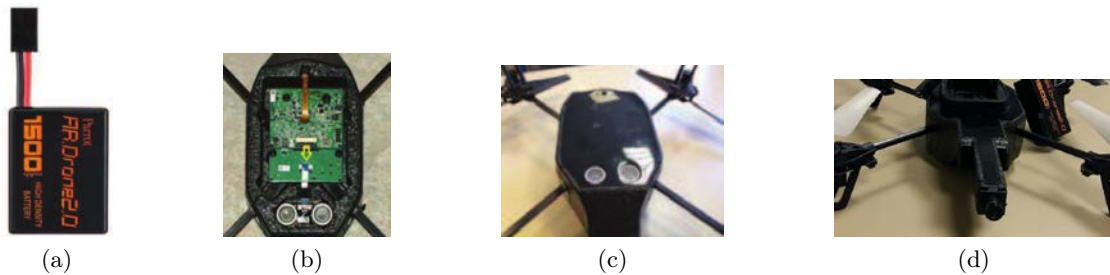


FIGURE B.4. – Composants de l'AR.Drone 2.0

Deux caméras sont présentes sur le drone, une en dessous, d'une résolution de 640×480 pixels et une caméra HD de résolution 1280×720 pixels située à l'avant du drone (voir figure B.4d). Les données des capteurs sont transmises à un ordinateur connecté par Wi-Fi au drone.

Grâce à chacun de ces capteurs, il est possible pour nous d'obtenir des informations sur les vitesses de translation et rotation du drone, sur son altitude, des valeurs incertaines du tangage et du roulis et des positions 2D dans l'image des objets observés dans l'environnement de navigation du drone.

C. Estimation robuste en présence de mesures aberrantes dans l'image

Cette annexe présente quelques résultats sur la localisation garantie à partir de mesures dans l'image. Nous avons observé au chapitre 4 que les mesures dans l'image peuvent être manquantes ou les bornes de leurs erreurs de mesures violées. Cette situation conduit à un ensemble vide ($\bar{\mathbb{S}}_{\text{img}} = \emptyset$) ou alors un ensemble qui ne contient pas la vraie pose. Dans ce cas, l'inversion ensembliste q -relaxée peut être utilisée pour avoir une solution qui prend en compte un nombre q de mesures aberrantes.

C.1. Solution q -relaxée

Nous supposons que le drone R est capable de mesurer les positions de m points 3D et leurs correspondants incertains 2D dans l'image. Parmi ces m points, il y en a q pour lesquelles les bornes d'erreurs sont violées, considérés comme mesures aberrantes. Une solution est de relaxer les contraintes du CSP \mathcal{H}_{img} .

Soit $\bar{\mathbb{S}}_{\text{img}}(i)$ l'ensemble solution des poses compatibles avec les mesures d'un point 3D ${}^w\mathbf{X}_i$. La solution robuste $\bar{\mathbb{S}}_{\text{img}}^{\{q\}}$ au problème complet est l'ensemble des poses compatibles avec $m - q$ mesures donnée par :

$$\bar{\mathbb{S}}_{\text{img}}^{\{q\}} = \bigcap_{i=1\dots m}^{\{q\}} \bar{\mathbb{S}}_{\text{img}}(i) \quad (\text{C.1})$$

L'algorithme RSIVIA (algorithme 4 présenté au chapitre 3) a été utilisé pour calculé l'ensemble solution $\bar{\mathbb{S}}_{\text{img}}^{\{q\}}$, les résultats sont présentés à la section suivante.

C.2. Résultats de la q -relaxée dans l'image

La cas où un drone capable d'observer un nombre $m = 5$ amers a été testé. Nous avons calculé l'ensemble solution $\bar{\mathbb{S}}_{\text{img}}^{k\{q\}}$ pour des valeurs croissantes de q et le résultat de la superposition des sous-pavages obtenus est explicité à la figure C.1 aux instants $t = 0\text{s}$ et $t = 0.7\text{s}$.

C. Estimation robuste en présence de mesures aberrantes dans l'image

$\overline{\mathbb{S}}_{\text{img}}^{\{0\}}$ est représenté par le sous-pavage en vert, il est équivalent au cas où tous les amers sont observés et les positions mesurées ne sont pas erronées. En rouge, nous avons $\overline{\mathbb{S}}_{\text{img}}^{\{1\}}$ qui suppose que la position d'un des 5 amers est une mesure aberrante. En cyan, nous avons le sous-pavage constitué de l'ensemble des poses consistantes avec 3 mesures sur les 5 ($\overline{\mathbb{S}}_{\text{img}}^{\{2\}}$). $\overline{\mathbb{S}}_{\text{img}}^{\{3\}}$ (en bleu) représente l'ensemble des poses compatibles 2 mesures.

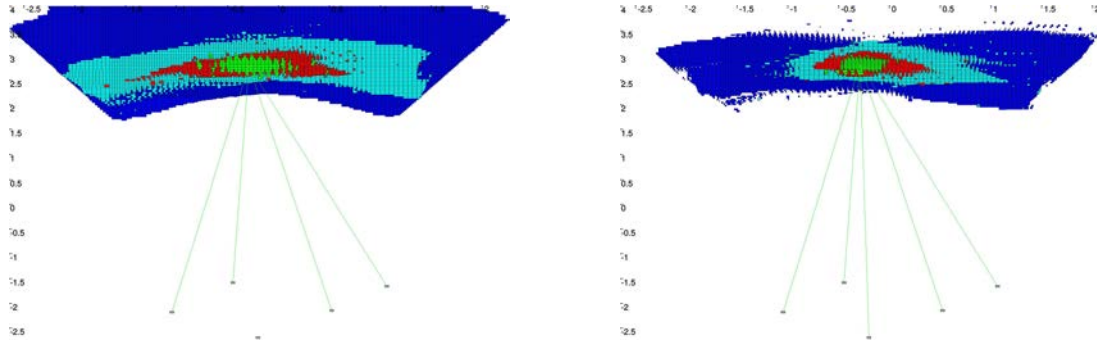


FIGURE C.1. – Plan (x, y) du sous-pavage résultat de RSIVIA sur les mesures dans l'image.

C. Estimation robuste en présence de mesures aberrantes dans l'image

Titre: Liste des publications

Kenmogne, I., Drevelle, V. et Marchand, E.(2017). "Vision based Pose domain characterization of an Unmanned Aerial Vehicle using Interval Analysis" In: 10th Workshop on Interval Methods.

Kenmogne, I., Drevelle, V. et Marchand, E.(2017). "Image-based UAV localization using Interval Methods" In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, p. 5285–5291.

Kenmogne, I., Drevelle, V. et Marchand, E.(2018). "Quantifier elimination for robot positioning with landmarks of uncertain position" In: 8th Int. Symp. on Scientific Computing, Computer Arithmetic and Verified Numerical Computations, SCAN2018.

Kenmogne, I., Drevelle, V. et Marchand, E.(2018). "Using constraint propagation for cooperative UAV localization from vision and ranging" In: Constraint Programming and Decision Making Workshop, CoProd 2018.

Kenmogne, I., Drevelle, V. et Marchand, E.(2018). "Cooperative Localization of Drones by using Interval Methods" In: SWIM 2018-11th Summer Workshop on Interval Methods, p. 1–4.

Kenmogne, I., Drevelle, V. et Marchand, E.(2018). "Interval-based cooperative UAVs pose domain characterization from images and ranges" In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),2018, p. 6349–6356.

Bibliographie

- 2.4Ghz Esky Tandem Rotor 4CH Chinook Helicopter Camo RC Remote Control Radio (2018). Online. Accessed : 2018-10-22. URL : <https://www.xheli.com/eskyheli-2328-chinook-camo.html>.
- ALTMANN, SL (1986). *Rotations, quaternions, and double groups : Oxford University Press*.
- AMELLER, Marc-andre, Bill TRIGGS et Long QUAN (2000). “Camera Pose Revisited - New Linear Algorithms”. In :
- AUSTIN, Reg (2011). *Unmanned aircraft systems : UAVS design, development and deployment*. T. 54. John Wiley & Sons.
- BAHR, Alexander, John J. LEONARD et Maurice F. FALLON (2009). “Cooperative Localization for Autonomous Underwater Vehicles”. In : *The Int. Journal of Robotics Research* 28.6, p. 714–728. DOI : 10.1177/0278364908100561. eprint : <http://ijr.sagepub.com/content/28/6/714.full.pdf+html>.
- BENEDETTI, Marco, Arnaud LALLOUET et Jérémie VAUTARD (2006). “Reasoning on quantified constraints”. In : *Rappresentazione della conoscenza e ragionamento automatico*.
- BENHAMOU, Frédéric, David A MCALLESTER et Pascal VAN HENTENRYCK (1994). “CLP (Intervals) Revisited.” In : *ILPS*. T. 94. Citeseer, p. 1–21.
- BENHAMOU, Frédéric et al. (1999). “Revising Hull and Box Consistency”. In : *Int. Conf. on Logic Programming*. MIT press, p. 230–244.
- BENHAMOU, Frédéric et Frédéric GOUALARD (2000). “Universally quantified interval constraints”. In : *International Conference on Principles and Practice of Constraint Programming*. Springer, p. 67–82.
- BENHAMOU, Frédéric et Laurent GRANVILLIERS (2006). “Continuous and interval constraints”. In : *Foundations of Artificial Intelligence*. T. 2. Elsevier, p. 571–603.
- BESSIERE, Christian (2006). *Handbook of constraint programming*.
- BESSIERE, Christian et Romuald DEBRUYNE (2008). “Theoretical analysis of singleton arc consistency and its extensions”. In : *Artificial Intelligence* 172.1, p. 29–41. ISSN : 0004-3702.
- BETHENCOURT, Aymeric et Luc JAULIN (2013). “Cooperative localization of underwater robots with unsynchronized clocks”. In : *Paladyn, Journal of Behavioral Robotics* 4.4, p. 233–244.
- BEVACQUA, Giuseppe et al. (2015). “Mixed-Initiative Planning and Execution for Multiple Drones in Search and Rescue Missions.” In : *ICAPS*, p. 315–323.

Bibliographie

- BORDEAUX, Lucas (2005). “Boolean and interval propagation for quantified constraints”. In : *Proceedings 1st international workshop on quantification in constraint programming (at CP 2005)*.
- BORDEAUX, Lucas et Eric MONFROY (2002). “Beyond NP : Arc-consistency for quantified constraints”. In : *International Conference on Principles and Practice of Constraint Programming*. Springer, p. 371–386.
- BRADSKI, G. et A. KAEHLER (2008). *Learning OpenCV : Computer vision with the OpenCV library*. O’Reilly Media, Inc.
- BROWN, D.C. (1971). “Close-Range Camera Calibration”. In : *Photogrammetric Engineering* 4.2, p. 127–140.
- BUCHER, Christian G. (1988). “Adaptive sampling - an iterative fast Monte Carlo procedure”. In : *Structural Safety* 5.2, p. 119 –126. ISSN : 0167-4730. DOI : [http://dx.doi.org/10.1016/0167-4730\(88\)90020-3](http://dx.doi.org/10.1016/0167-4730(88)90020-3).
- CAMILLO JOSE TAYLOR, John Spletzer (2007). “A Bounded Uncertainty Approach to Cooperative Localization Using Relative Bearing Constraints”. In : *IEEE/RSJ Int. Conf.on Intelligent Robots and Systems (IROS)*, p. 2500–2506.
- CASTILLO, Pedro, Rogelio LOZANO et Alejandro E DZUL (2006). *Modelling and control of mini-flying machines*. Physica-Verlag.
- CHABERT, Gilles et Luc JAULIN (2009). “Contractor programming”. In : *Artificial Intelligence* 173, p. 1079–1100.
- CLEARY, John (1987). “Logical arithmetic”. In : *Future computing systems* 2.2, p. 125–149.
- COLLE, E. et S. GALERNE (2013). “Mobile robot localization by multiangulation using set inversion”. Anglais. In : *Robotics and Autonomous Systems* 61.1, p. 39–48.
- COLLINS, George E (1975). “Quantifier elimination for real closed fields by cylindrical algebraic decomposition”. In : *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*. Springer, p. 134–183.
- DAMIEN, Eynard (2011). “Capteur de stéréovision hybride pour la navigation des drones”. Thèse de doct. Université de Picardie Jules Verne, Page 19.
- DAVIS, Robert B (2014). *Applying cooperative localization to swarm uavs using an extended kalman filter*. Rapp. tech. NAVAL POSTGRADUATE SCHOOL MONTEREY CA DEPT OF COMPUTER SCIENCE.
- DEMENTHON, D. et L. DAVIS (1995). “Model-Based Object Pose in 25 Lines of Codes”. In : *Int. J. of Computer Vision* 15.1-2, p. 123–141.
- DREVELLE, V. et P. BONNIFAIT (2013). “Localization Confidence Domains via Set Inversion on Short-Term Trajectory”. In : *IEEE Trans. on Robotics* 29.5, p. 1244–1256. ISSN : 1552-3098. DOI : [10.1109/TR0.2013.2262776](https://doi.org/10.1109/TR0.2013.2262776).
- DREVELLE, Vincent (2011). “Étude de méthodes ensemblistes robustes pour une localisation multisensorielle intégrée. Application à la navigation des véhicules en milieu urbain.” Thèse de doct. Université de Technologie de Compiègne.
- DREVELLE, Vincent et Philippe BONNIFAIT (2011). “A set-membership approach for high integrity height-aided satellite positioning”. In : *GPS solutions* 15.4, p. 357–368.

Bibliographie

- DREVELLE, Vincent et Philippe BONNIFAIT (2016). “Interval-based fast fault detection and identification applied to radio-navigation multipath”. In : *Int. Journal of Adaptive Control and Signal Processing* 30.2, p. 154–172.
- Drone de surveillance, pour recherche et sauvetage, hélicoptère, bi-rotor FH-420 (2018). Online. Accessed : 2018-10-22. URL : <http://www.aeroexpo.online/fr/prod/flint-hill-solutions/product-181381-26949.html>.
- FAUGERAS, O. (1993). *Three-dimensional computer vision : a geometric viewpoint*. Cambridge, Massachusetts : MIT Press.
- FAUGERAS, O.D et G. TOSCANI (1987). “Camera calibration for 3D computer vision.” In : *Proc Int. Workshop on Machine Vision and Machine Intelligence*. Tokyo, p. 240–247.
- FISCHLER, N. et R.C. BOLLES (1981). “Random Sample Consensus : A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography”. In : *Communication of the ACM* 24.6, p. 381–395.
- FOX, D. et al. (2006). “Distributed Multirobot Exploration and Mapping”. In : *Proceedings of the IEEE* 94.7, p. 1325–1339. ISSN : 0018-9219. DOI : 10.1109/JPROC.2006.876927.
- GAO, Xiao-Shan et al. (2003). “Complete solution classification for the perspective-three-point problem”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.8, p. 930–943. ISSN : 0162-8828. DOI : 10.1109/TPAMI.2003.1217599.
- GARCIA, Dorian (2001). “Mesure de formes et de champs de déplacements tridimensionnels par stéréo-corrélation d’images”. Thèse de doct. Institut National Polytechnique de Toulouse-INPT.
- GIOIOSO, Guido et al. (2014). “The flying hand : A formation of UAVs for cooperative aerial tele-manipulation”. In : *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, p. 4335–4341.
- GOLDSZTEJN, Alexandre et al. (2005). “Modal intervals revisited : a mean-value extension to generalized intervals”. In : *Proceedings of QCP-2005 (Quantification in Constraint Programming), Barcelona, Spain*. T. 2005.
- GOLDSZTEJN, Alexandre et Luc JAULIN (2006). “Inner and outer approximations of existentially quantified equality constraints”. In : *International Conference on Principles and Practice of Constraint Programming*. Springer, p. 198–212.
- GOLDSZTEJN, Alexandre, Claude MICHEL et Michel RUEHER (2009). “Efficient handling of universally quantified inequalities”. In : *Constraints* 14.1, p. 117–135.
- GUYONNEAU, Rémy (2013). “Méthodes ensemblistes pour la localisation en robotique mobile”. Thèse de doct. Université d’Angers.
- HAJDER, Levente (2017). “W-PnP Method : Optimal Solution for the Weak-Perspective n-Point Problem and Its Application to Structure from Motion”. In :
- HANSEN, E (1992). “Global optimization using interval analysis”. In : *Marcel Dekkar, New York*.
- HANSEN, Eldon (1978). “Interval forms of Newtons method”. In : *Computing* 20.2, p. 153–163.
- HARTLEY, R. et A. ZISSERMAN (2001). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Bibliographie

- HARTLEY, Richard et Andrew ZISSERMAN (2004). *Multiple View Geometry in Computer Vision*. 2^e éd. Cambridge University Press. DOI : 10.1017/CB09780511811685.
- HONG, Hoon (1992). “Simple solution formula construction in cylindrical algebraic decomposition based quantifier elimination”. In : *Papers from the international symposium on Symbolic and algebraic computation*. ACM, p. 177–188.
- HORN, Berthold KP (1987). “Closed-form solution of absolute orientation using unit quaternions”. In : *JOSA A* 4.4, p. 629–642.
- HUANG, Guoquan P et al. (2011). “Observability-based consistent EKF estimators for multi-robot cooperative localization”. In : *Autonomous Robots* 30.1, p. 99–122.
- JAULIN, Luc (2001). “Reliable Minimax Parameter Estimation”. In : *Reliable Computing* 7.3, p. 231–246. ISSN : 1573-1340. DOI : 10.1023/A:1011451021517. URL : <https://doi.org/10.1023/A:1011451021517>.
- (2009). “Robust set-membership state estimation ; application to underwater robotics”. In : *Automatica* 45.1, p. 202–206.
- JAULIN, Luc et Eric WALTER (1993). “Set inversion via interval analysis for nonlinear bounded-error estimation”. In : *Automatica* 29.4, p. 1053 –1064. ISSN : 0005-1098.
- JAULIN, Luc, Eric WALTER et Olivier DIDRIT (1996). “Guaranteed robust nonlinear parameter bounding”. In : *Proc. CESA*. T. 96, p. 1156–1161.
- JAULIN, Luc et al. (1997). “Light scattering data analysis via set inversion”. In : *Journal of Physics A : Mathematical and General* 30.22, p. 7733.
- JAULIN, Luc et Alain GODON (1999). “Motion planning using interval analysis”. In : *MISC*. T. 99, p. 335–346.
- JAULIN, Luc et al. (2001). *Applied interval analysis : with examples in parameter and state estimation, robust control and robotics*. Springer Verlag.
- JAULIN, Luc et Gilles CHABERT (2008). “QUIMPER : un langage de programmation pour le calcul ensembliste ; Application à l’automatique”. In : *CIFA*.
- (2010). “Resolution of nonlinear interval problems using symbolic interval arithmetic”. In : *Engineering Applications of Artificial Intelligence* 23.6, p. 1035–1040.
- JIANG, Jingjing et Alessandro ASTOLFI (2015). “Shared-Control for a UAV Operating in the 3D Space”. In : *Control Conference (ECC), 2015 European*. IEEE, p. 1633–1638.
- KENMOGNE, Ide-Flore, Vincent DREVELLE et Eric MARCHAND (2017). “Image-based UAV localization using Interval Methods”. In : *IROS 2017 - IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. Vancouver, Canada, p. 5285–5291.
- (2018a). “Interval-based cooperative UAVs pose domain characterization from images and ranges”. In : *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS’18*.
- (2018b). “Quantifier elimination for robot positioning with landmarks of uncertain position”. In : *8th Int. Symp. on Scientific Computing, Computer Arithmetic, and Verified Numerical Computations, SCAN2018*.
- (2018c). “Using constraint propagation for cooperative UAV localization from vision and ranging”. In : *Constraint Programming and Decision Making Workshop, CoProd 2018*.
- KIEFFER, M. (1999). “Estimation ensembliste par analyse par intervalles. Application à la localisation d’un véhicule”. 1999PA112034. Thèse de doct., 176 P. URL : <http://www.theses.fr/1999PA112034>.

Bibliographie

- KIM, H Jin, David H SHIM et Shankar SASTRY (2002). “Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles”. In : *American Control Conference, 2002. Proceedings of the 2002*. T. 5. IEEE, p. 3576–3581.
- KNEIP, Laurent, Hongdong LI et Yongduck SEO (2014). “Upnp : An optimal $O(n)$ solution to the absolute pose problem with universal applicability”. In : *European Conference on Computer Vision*. Springer, p. 127–142.
- KOEHL, Arnaud (2012). “Modélisation, observation et commande d’un drone miniature à birotor coaxial”. Thèse de doct. Université Henri Poincaré-Nancy I.
- LASSOUED, K., P. BONNIFAIT et I. FANTONI (2017). “Cooperative Localization with Reliable Confidence Domains Between Vehicles Sharing GNSS Pseudorange Errors with No Base Station”. In : *IEEE Intelligent Transportation Systems Magazine* 9.1, p. 22–34. ISSN : 1939-1390. DOI : 10.1109/MITS.2016.2630586.
- LE BARS, Fabrice (2011). “Analyse par intervalles pour la localisation et la cartographie simultanées ; Application à la robotique sous-marine”. Thèse de doct. Université de Bretagne occidentale-Brest.
- LEPETIT, V. et P. FUA (2005). “Monocular Model-Based 3D Tracking of Rigid Objects : A Survey”. In : *Foundations and Trends in Computer Graphics and Vision* 1.1, p. 1–89.
- LEPETIT, V., F. MORENO-NOGUER et P. FUA (2009). “EPnP : An accurate $O(n)$ solution to the PnP problem”. In : *Int. Journal of Computer Vision* 81.2, p. 155–166.
- LEVENBERG, Kenneth (1944). “A method for the solution of certain non-linear problems in least squares”. In : *Quarterly of applied mathematics* 2.2, p. 164–168.
- LHOMME, Olivier (1993). “Consistency techniques for numeric CSPs”. In : *IJCAI*. T. 93, p. 232–238.
- MANZANERA, Antoine (2005). “Traitement d’images et vision artificielle”. In : *Unité d’Électronique et d’Informatique*.
- MARCHAND, E., F. SPINDLER et F. CHAUMETTE (2005). “ViSP for visual servoing : a generic software platform with a wide class of robot control skills”. In : *IEEE Robotics and Automation Magazine* 12.4, p. 40–52.
- MARCHAND, E., H. UCHIYAMA et F. SPINDLER (2016). “Pose estimation for augmented reality : a hands-on survey”. In : *IEEE Trans. on Visualization and Computer Graphics* 22.12, p. 2633–2651.
- MARSHALL, Douglas M et al. (2016). *Introduction to unmanned aircraft systems*. Crc Press.
- MAZA, Ivan et Anibal OLLERO (2007). “Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms”. In : *Distributed Autonomous Robotic Systems 6*. Tokyo : Springer Japan, p. 221–230. ISBN : 978-4-431-35873-2.
- MELCHERS, R.E. (1989). “Importance sampling in structural systems”. In : *Structural Safety* 6.1, p. 3 –10. ISSN : 0167-4730. DOI : [http://dx.doi.org/10.1016/0167-4730\(89\)90003-9](http://dx.doi.org/10.1016/0167-4730(89)90003-9). URL : <http://www.sciencedirect.com/science/article/pii/0167473089900039>.
- MICHAEL, Nathan, Jonathan FINK et Vijay KUMAR (2011). “Cooperative manipulation and transportation with aerial robots”. In : *Autonomous Robots* 30.1, p. 73–86.
- MOORE, Ramon E (1979). *Methods And Applications Of Interval Analysis*. T. 2. Siam.

Bibliographie

- MOORE, R.E. (1966). *Interval analysis*. Prentice Hall.
- MOORE, R.E., R.B. KEARFOTT et M.J. CLOUD (2009). *Introduction to Interval Analysis*. SIAM e-books. Society for Industrial et Applied Mathematics (SIAM). ISBN : 9780898717716. URL : <https://books.google.fr/books?id=tT7ykKbqfEwC>.
- MORAS, Julien et al. (2012). “Drivable space characterization using automotive lidar and georeferenced map information”. In : *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, p. 778–783.
- MOURAD, Farah et al. (2012). “Robust bounded-error tracking in wireless sensor networks”. In : *SYSID 2012*, p. 1–6.
- NEUMAIER, Arnold (1990). *Interval methods for systems of equations*. T. 37. Cambridge university press.
- NEVEU, Bertrand, Gilles TROMBETTONI et Ignacio ARAYA (2015). “Adaptive constructive interval disjunction : algorithms and experiments”. In : *Constraints* 20.4, p. 452–467.
- NONAMI, Kenzo et al. (2010). *Autonomous Flying Robots-Unmanned Aerial Vehicles and Micro Aerial Vehicles*.
OctoCopter, Drone de série, homologué par la DGAC. Online. Accessed : 2018-10-22. URL : <https://www.flyingeye.fr/product/octocopter/>.
- Parrot AR.Drone 2.0 'Edition Elite, The high-definition quadricopter drone*. Online. Accessed : 2018-10-22. URL : <https://www.parrot.com/global/drones/parrot-ardrone-20-elite-edition>.
- QU, Yaohong et Youmin ZHANG (2011). “Cooperative localization against GPS signal loss in multiple UAVs flight”. In : *Journal of Systems Engineering and Electronics* 22.1, p. 103–112.
- QUAN, Long et Zhongdan LAN (1999). In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21. ISSN : 0162-8828. DOI : 10.1109/34.784291.
- RIDA, Mokhtari Mohammed (2015). “Observation et Commande de Drones Miniatures à voilures tournantes”. Thèse de doct. Thèse de Doctorat de l’Université de Aboubekr Belkaid Tlemcen (spécialité automatique), Page 11.
- ROBUFFO GIORDANO, P (2012). “Multi-UAV Bilateral Shared Control and Decentralization”. In : *Summer School Airobots : Aerial Service Robotics, Eidgenössische Technische Hochschule*.
- ROUMELIOTIS, Stergios, George A BEKEY et al. (2002). “Distributed multirobot localization”. In : *Robotics and Automation, IEEE Trans. on* 18.5, p. 781–795.
- SAHINOGLU, Z. et S. GEZICI (2006). “Ranging in the IEEE 802.15.4a Standard”. In : *2006 IEEE Annual Wireless and Microwave Technology Conference*, p. 1–5. DOI : 10.1109/WAMICON.2006.351897.
- SCHIANO, Fabrizio et Paolo Robuffo GIORDANO (2017). “Bearing rigidity maintenance for formations of quadrotor UAVs”. In : *ICRA 2017 - IEEE Int. Conf. on Robotics and Automation*. Singapore, Singapore, p. 1467–1474. DOI : 10.1109/ICRA.2017.7989175.
- Skyborne Technologies - New Tri-rotor UAVs, Project CERBERUS*. Online. Accessed : 2018-10-22. URL : <https://www.qldstartupshowcase.com/uav--unmanned-vehicles-etc/skyborne-technologies-new-tri-rotor-uavs>.

Bibliographie

- SPLETZER, John et Camillo J. TAYLOR (2003). “A Bounded Uncertainty Approach to Multi-Robot Localization”. In : *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, p. 1258–1264.
- STEIN, G.P. (1997). “Lens Distortion Calibration Using Point Correspondences”. In : *IEEE Int. Conf on computer Vision and Pattern Recognition, CVPR97*, p. 602–608.
- STIPANOVIĆ, Dušan M et al. (2004). “Decentralized overlapping control of a formation of unmanned aerial vehicles”. In : *Automatica* 40.8, p. 1285–1296.
- SUTHERLAND, Ivan E (1974). “Three-dimensional data input by tablet”. In : *Proceedings of the IEEE* 62.4, p. 453–461.
- TARSKI, Alfred (1998). “A decision method for elementary algebra and geometry”. In : *Quantifier elimination and cylindrical algebraic decomposition*. Springer, p. 24–84.
- TELLE, B., M. J. ALDON et N. RAMDANI (2003). “Guaranteed 3D visual sensing based on interval analysis”. In : *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. T. 2, 1566–1571 vol.2. DOI : 10.1109/IROS.2003.1248867.
- TRAXXAS DR-1 Double-Rotor coaxial RTF (2018). Online. Accessed : 2018-10-22. URL : <http://www.modelisme-guadeloupe.com/helico-rc/377-traxxas-dr-1-double-rotor-coaxial-rtf.html>.
- TROMBETTONI, Gilles et Gilles CHABERT (2007). “Constructive interval disjunction”. In : *International Conference on Principles and Practice of Constraint Programming*. Springer, p. 635–650.
- TSAI, R.Y. (1986). “An efficient and accurate camera calibration technique for 3D machine vision”. In : *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR’86*. Miami, Floride, p. 364–374.
- Typhoon H, L’hexaoptère primé avec caméra. Online. Accessed : 2018-10-22. URL : https://www.yuneec.com/fr_FR/drones-camera/typhoon-h/aperçu.html.
- VIÑAS, Pau Herrero i et al. (2006). *Quantified real constraint solving using modal intervals with applications to control*. Universitat de Girona.
- VOGES, Raphael et Bernardo WAGNER (2018). “RGB-Laser Odometry Under Interval Uncertainty for Guaranteed Localization”. In : *SWIM 2018*, p. 61.
- WALTZ, David L (1972). “Generating semantic descriptions from drawings of scenes with shadows”. In :
- WAN, Eric A et Rudolph VAN DER MERWE (2000). “The unscented Kalman filter for nonlinear estimation”. In : *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. Ieee, p. 153–158.
- WENG, J., P. COHEN et M. HERNIOU (1992). “Camera Calibration with Distorsion Models and Accuracy Evaluation”. In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 14.10, p. 965–980.
- YORKE-SMITH, Neil et Carmen GERVET (2005). “Closures of uncertain constraint satisfaction problems”. In : *Proceedings 1st International Workshop on Quantification in Constraint Programming (at CP 2005)*.
- ZELLER, Cyril et al. (1996). “Calibration projective, affine et Euclidienne en vision par ordinateur et application a la perception tridimensionnelle”. Thèse de doct.

Bibliographie

ZHANG, Z. (2000). "A Flexible New Technique for Camera Calibration". In : *IEEE Trans. on Pattern Analysis and Machine Intelligence* 22.11, p. 1330–1334.

Bibliographie

Mots clés : Méthodes ensemblistes, Intervalles, Localisation collaborative, Vision, Drones

résumé : Dans cette thèse, on cherche à caractériser un domaine de pose pour la localisation de drones, en utilisant des méthodes ensemblistes par intervalles.

Cette caractérisation a pour but de quantifier l'incertitude de localisation. En absence ou défaillance du GPS, en milieux contraints et intérieurs, une alternative est d'utiliser la caméra. Les mesures dans l'image et celles provenant des capteurs du drone ainsi que les paramètres des modèles sont très souvent tachés d'erreurs. Les méthodes classiques d'estimation ne fournissent pas une estimation ponctuelle de la pose (position et orientation) du drone, en modélisant les erreurs par leur distribution de probabilité. Toutefois, il est parfois difficile, voire impossible, de décrire précisément les lois de probabilité de ces perturbations.

Dans le cadre ensembliste, on peut représenter les erreurs par des intervalles. L'analyse par intervalles permet de propager l'incertitude de manière garantie les incertitudes, même en présence de données aberrantes. Ces travaux proposent une méthode de localisation ensembliste par intervalles basée sur l'observation d'amers géoréférencés, l'élimination de quantificateur pour la prise en compte des incertitudes sur les positions des amers ; et une méthode pour résoudre le problème de localisation coopérative ensembliste. Chacune d'elles a été testée en simulation et sur des données réelles acquises dans un environnement intérieur, permettant leurs comparaisons aux méthodes classiques existantes.

Titre : Set-membership localization of drones using interval methods

Keywords : Set-membership methods, Intervals, Cooperative localization, Vision, Drones

Interval set-membership methods. This characterization enables to quantify the positioning uncertainty. In the absence or failure of GPS, in constrained and indoor environments, an alternative is to use the camera. The image measurements and those coming from the drone sensors as well as the models parameters are very often tainted by errors. Classical estimation methods provide a joint estimate of the drone pose (position and orientation), assuming a probabilistic model of errors. However, it is sometimes difficult or impossible to describe precisely the probability distributions of these disturbances.

Analysis can then be used to propagate uncertainties, even in the presence of outliers. This work proposes an interval set-membership localization method based on image measurements; a quantifier elimination method taking into account the uncertainties on landmarks positions; and a method for solving set-membership cooperative localization problem. Each of these methods has been implemented and tested in simulation and on data acquired in an indoor environment, allowing their comparisons with existing classical methods possible.