



**HAL**  
open science

# Hybridation GPS/Vision monoculaire pour la navigation autonome d'un robot en milieu extérieur

Jean-Marie Codol

► **To cite this version:**

Jean-Marie Codol. Hybridation GPS/Vision monoculaire pour la navigation autonome d'un robot en milieu extérieur. Vision par ordinateur et reconnaissance de formes [cs.CV]. INSA de Toulouse, 2012. Français. NNT : 2012ISAT0060 . tel-01246168

**HAL Id: tel-01246168**

**<https://theses.hal.science/tel-01246168>**

Submitted on 18 Dec 2015

**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.



# THÈSE

En vue de l'obtention du

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Institut National des Sciences Appliquées de Toulouse*

Discipline ou spécialité : *Systèmes Automatiques*

---

Présentée et soutenue par *Jean-Marie CODOL*

Le 15 février 2012

**Titre :**

*Hybridation GPS/Vision monoculaire pour la navigation  
d'un robot en milieu extérieur*

---

**JURY**

<i>M. Jean-Charles</i>	<i>Noyer</i>	<i>Président</i>
<i>M. Philippe</i>	<i>Bonnifait</i>	<i>Rapporteur</i>
<i>M. Emmanuel</i>	<i>Duflos</i>	<i>Rapporteur</i>
<i>M. André</i>	<i>Monin</i>	<i>Directeur de Thèse</i>
<i>M. Michel</i>	<i>Devy</i>	<i>Codirecteur de Thèse</i>

---

**Ecole doctorale :** *Ecole doctorale Systèmes (EDSYS)*

**Unité de recherche :** *LAAS CNRS*

**Directeur(s) de Thèse :** *M. André Monin, M. Michel Devy*



# Hybridation GPS/Vision monoculaire pour la navigation d'un robot en milieu extérieur

Jean-Marie CODOL

Février 2014



## Résumé

On assiste aujourd'hui à l'importation des NTIC<sup>1</sup> dans la robotique. L'union de ces technologies donnera naissance, dans les années à venir, à la robotique de service *grand-public*.

Cet avenir, s'il se réalise, sera le fruit d'un travail de recherche, amont, dans de nombreux domaines : la mécatronique, les télécommunications, l'automatique, le traitement du signal et des images, l'intelligence artificielle . . .

Un des aspects particulièrement intéressant en robotique mobile est alors le problème de la localisation et de la cartographie simultanée. En effet, dans de nombreux cas, un robot mobile, pour accéder à une intelligence, doit nécessairement se localiser dans son environnement. La question est alors : quelle précision pouvons-nous espérer en terme de localisation ? Et à quel coût ?

Dans ce contexte, un des objectifs de tous les laboratoires de recherche en robotique, objectif dont les résultats sont particulièrement attendus dans les milieux industriels, est un positionnement et une cartographie de l'environnement, qui soient à la fois précis, tous-lieux, intègre, bas-coût et temps-réel. Les capteurs de prédilection sont les capteurs peu onéreux tels qu'un GPS standard (de précision métrique), et un ensemble de capteurs embarquables en charge utile (comme les caméras-vidéo). Ce type de capteurs constituera donc notre support privilégié, dans notre travail de recherche.

Dans cette thèse, nous aborderons le problème de la localisation d'un robot mobile, et nous choisirons de traiter notre problème par l'approche probabiliste. La démarche est la suivante, nous définissons nos 'variables d'intérêt' : un ensemble de variables aléatoires. Nous décrivons ensuite leurs lois de distribution, et leur modèles d'évolution, enfin nous déterminons une fonction de coût, de manière à construire un observateur (une classe d'algorithme dont l'objectif est de déterminer le minimum de notre fonction de coût).

Notre contribution consistera en l'utilisation de mesures GPS brutes GPS (les mesures brutes - ou raw-datas - sont les mesures issues des boucles de corrélation de code et de phase, respectivement appelées mesures de pseudo-distances de code et de phase) pour une navigation bas-coût précise en milieu extérieur suburbain. En utilisant la propriété dite 'entière' des ambiguïtés de phase GPS, nous étendrons notre navigation pour réaliser un système GPS-RTK<sup>2</sup> en mode différentiel local précise et bas-coût.

Nos propositions sont validées par des expérimentations réalisées sur notre démonstrateur robotique.

---

1. Nouvelles Technologies de l'Information et de la Télécommunication  
2. Real Time Kinematic

## Abstract

We are witnessing nowadays the importation of ICT<sup>3</sup> in robotics. These technologies will give birth, in upcoming years, to the *general public* service robotics.

This future, if realised, shall be the result of many research conducted in several domains : mechatronics, telecommunications, automatics, signal and image processing, artificial intelligence, etc.

One particularly interesting aspect in mobile robotics is hence the simultaneous localisation and mapping problem. Consequently, to access certain informations, a mobile robot has, in many cases, to map/localise itself inside its environment. The following question is then posed : What precision can we aim for in terms of localisation ? And at what cost ?

In this context, one of the objectives of many laboratories indulged in robotics research, and where results impact directly the industry, is the positioning and mapping of the environment. These latter tasks should be precise, adapted everywhere, integrated, low-cost and real-time. The prediction sensors are inexpensive ones, such as a standard GPS (of metric precision), and a set of embeddable payload sensors (e.g. video cameras). These type of sensors constitute the main support in our work.

In this thesis, we shed light on the localisation problem of a mobile robot, which we choose to handle with a probabilistic approach. The procedure is as follows : we first define our "variables of interest" which are a set of random variables, and then we describe their distribution laws and their evolution models. Afterwards, we determine a cost function in such a manner to build up an observer (an algorithmic class where the objective is to minimize the cost function).

Our contribution consists of using brute GPS measures (brute measures or raw datas are measures issued from code and phase correlation loops, called pseudo-distance measures of code and phase, respectively) for a low-cost navigation, which is precise in an external suburban environment. By implementing the so-called "whole" property of GPS phase ambiguities, we expand the navigation to achieve a GPS-RTK<sup>4</sup> system in a precise and low-cost local differential mode.

Our propositions has been validated through experimentations realized on our robotic demonstrator.

---

3. Information and Communications Technology

4. Real-Time Kinematic

# Table des matières

Résumé . . . . .	iii
Abstract . . . . .	iv
Liste des figures . . . . .	vi
<b>1 Introduction</b>	<b>1</b>
1.1 Enjeux . . . . .	2
1.2 Motivation . . . . .	2
1.3 Choix des capteurs . . . . .	3
1.4 Intérêt du GPS en mode différentiel local . . . . .	4
1.5 Le SLAM Visuel . . . . .	5
1.6 Fusion GPS/Vision . . . . .	6
1.7 Spécifications . . . . .	6
1.8 Architecture proposée . . . . .	6
1.9 Contenu du mémoire . . . . .	7
<b>2 État de l'art</b>	<b>9</b>
2.1 Localisation GPS sub-métrique . . . . .	9
2.1.1 Mesures GPS brutes . . . . .	10
2.1.2 Méthodes de localisation sans infrastructure . . . . .	13
2.1.3 Méthodes de localisation avec infrastructure . . . . .	16
2.1.4 Hybridation avec d'autres capteurs . . . . .	17
2.1.5 GPS en robotique . . . . .	18
2.2 Vision par ordinateur . . . . .	19
2.2.1 Synthèse d'information par capteur optique . . . . .	19
2.2.2 Vision en robotique . . . . .	24
2.3 SLAM : localisation et cartographie simultanées . . . . .	25
2.3.1 Présentation générale . . . . .	25
2.3.2 Pose SLAM . . . . .	25
2.3.3 Landmark-SLAM et Filtrage . . . . .	25
2.3.4 SLAM par lissage incrémental (Pose-SLAM et Landmark-SLAM) . . . . .	27
2.4 Contributions de notre recherche . . . . .	29



<b>3</b>	<b>Estimation par triple-différence des mesures de phase en navigation DGPS-RTK</b>	<b>31</b>
3.1	Modèles dynamiques du mobile . . . . .	31
3.1.1	Modèle odométrique . . . . .	31
3.1.2	Mobile à position, vitesse ou accélération brownienne . . . . .	32
3.2	Modèle des mesures de phase en double différence DGPS . . . . .	33
3.3	Modèle des mesures de phase triple différence DGPS . . . . .	35
3.3.1	Rappel sur les méthodes de relaxation . . . . .	35
3.3.2	Modèle de mesures . . . . .	36
3.3.3	Modélisation de la corrélation du bruit de mesure . . . . .	37
3.4	Filtrage optimal . . . . .	40
3.5	Résultats . . . . .	41
3.6	Conclusions sur le filtre de triple différence . . . . .	41
<b>4</b>	<b>Estimation par lever d’ambiguïté de phase entière</b>	<b>45</b>
4.1	Motivations . . . . .	45
4.2	Méthode LAMBDA . . . . .	45
4.3	Méthode proposée . . . . .	47
4.3.1	Construction arborescente du jeu d’hypothèse . . . . .	48
4.3.2	Elagage séquentiel de l’arbre d’hypothèses . . . . .	55
4.4	Test de consistance sur le caractère gaussien du vecteur de résidu . . . . .	58
4.4.1	Test séquentiel . . . . .	58
4.4.2	Test global . . . . .	59
4.4.3	Adaptation au cas de la navigation avec trajets multiples . . . . .	60
4.5	Résultats . . . . .	62
4.6	Conclusion et perspectives . . . . .	63
<b>5</b>	<b>Approche EKF-SLAM pour la navigation robotique</b>	<b>67</b>
5.1	RT-SLAM . . . . .	67
5.1.1	Objectifs . . . . .	67
5.1.2	Introduction . . . . .	68
5.1.3	Contexte . . . . .	68
5.1.4	SLAM multi-capteur caméra/IMU temps-réel . . . . .	70
5.2	C-SLAM . . . . .	75
5.2.1	Objectifs . . . . .	75
5.2.2	Architecture . . . . .	76
5.2.3	Caractéristiques . . . . .	76
5.2.4	Comparaison avec RT-Slam . . . . .	77
5.2.5	Résultats pour l’hybridation Caméra/IMU . . . . .	77
5.2.6	Conclusion sur C-Slam et les architectures embarquées . . . . .	79

<b>6 Conclusion</b>	<b>81</b>
6.1 Navigation aux triples-différences de phase GPS . . . . .	81
6.2 Navigation aux DGPS-RTK mono-fréquence . . . . .	82
6.3 SLAM visuel . . . . .	82
6.4 Bilan et perspectives . . . . .	83
<b>Appendices</b>	<b>85</b>
<b>A Modèles de bruits Gaussiens</b>	<b>87</b>
A.1 Bruit Gaussien et loi normale . . . . .	87
A.2 Représentation ellipsoïdale . . . . .	87
A.3 Distance de Mahalanobis . . . . .	88
A.4 Probabilités et changement de variable . . . . .	89



# Table des figures

1.1	Principe du GPS différentiel local . . . . .	4
1.2	Architecture temps-réel . . . . .	8
2.1	Suivi du signal de code . . . . .	10
2.2	Suivi du signal de phase . . . . .	11
2.3	Signal réfléchi . . . . .	12
2.4	Effet d'un trajet multiple . . . . .	13
2.5	Allocation des fréquences pour GPS, GLONASS, et GALILEO . . . . .	15
2.6	Une image informative . . . . .	18
2.7	Détecteur de point d'intérêt FAST . . . . .	21
2.8	Contraintes Rt par scans laser 2D . . . . .	26
3.1	Triple-différence : convergence initiale. . . . .	42
3.2	Triple-différence : suivi de trajectoire. . . . .	42
3.3	Triple-différence : Trajectoire typique. . . . .	43
4.1	Décorrélacion LAMBDA. . . . .	47
4.2	Intervalle de confiance optimal. . . . .	51
4.3	Intervalle et contrainte entière. . . . .	52
4.4	Construction récursive des hypothèses. . . . .	64
4.5	Résultats RTK sur 24 heures. . . . .	65
4.6	Terrain d'expérimentation. . . . .	65
5.1	Exemple de graphe des objets RT-Slam . . . . .	71
5.2	Représentations des objets RT-Slam . . . . .	71
5.3	Erreur RT-Slam . . . . .	75
5.4	Architecture C-Slam . . . . .	76
5.5	Image d'une expérience C-Slam . . . . .	77
5.6	Performances de C-Slam sur VIA EPIA . . . . .	78
5.7	Performances de C-Slam sur ARM-9 . . . . .	78
5.8	Performances de C-Slam sur Power-PC . . . . .	79
5.9	Performances comparées en fonction de la taille de la carte SLAM . . . . .	80
5.10	Performances comparées en fonction du nombre de corrections par image . . . . .	80

A.1 Ellipse d'une densité Gaussienne . . . . .	88
------------------------------------------------	----

# Chapitre 1

## Introduction

La navigation autonome d'un véhicule terrestre en milieu extérieur est, encore aujourd'hui, un défi. Si l'on regarde plus en détail ce défi, on distingue un ensemble de verrous technologiques et techniques que l'on tente chaque jour - en tant qu'ingénieurs et chercheurs - de lever. Citons-en deux : d'abord la qualité des capteurs (on parle de rapport qualité/prix), s'en suit immédiatement après la complexité algorithmique des solutions.

Les deux exemples proposés ne sont pas aléatoires. Nous avons tous<sup>1</sup> été confrontés, un jour ou l'autre, aux expériences suivantes :

- Voir ou connaître l'existence d'un capteur aussi utile qu'inaccessible pour des applications grand-public (un Velodyne, un récepteur GPS-RTK bi-fréquence, un IMU de hautes performances, ou une simple horloge atomique ...)
- Obtenir un résultat optimal (ou localement optimal), après une nuit de calcul (sur une machine performante). Avant de nous désoler que cette même procédure ne puisse (pour l'instant) être exécutée en 1 fraction de seconde sur une unité de calcul embarquée.

Cependant, régulièrement, que ce soit le fruit d'un travail dans l'industrie, ou en laboratoire public, certains de ces verrous sont indéniablement levés, ce qui permet d'aller de l'avant. Un résultat récent, l'un des plus parlant sans doute, est le travail de l'équipe de R.A. Newcombe, S.J. Lovegrove et A.J. Davison, dans "*DTAM : Dense Tracking and Mapping in real-time*". Dans ce cas, on démontre qu'il est possible de reconstruire, en 3D dense, et en temps-réel, sur un ordinateur de bureau, une scène observée par une caméra monoculaire, tenue à la main.

L'orientation prise depuis 5 ans, dans certains groupes de travail<sup>2</sup> en navigation robotique nous a influencé vers l'approche statistique pour traiter le problème de localisation robotique. Le fruit de quelques rencontres, de nombreux échanges, et de vives confrontations nous permet d'exposer, dans cette thèse des résultats que j'ose espérer intéressants pour le lecteur. Mais tout d'abord, rappelons quels sont les enjeux de nos travaux.

---

1. "Nous" désigne le lecteur avisé, ayant une expérience en recherche, voire une expérience en robotique.

2. Outre-Manche : *G. Klein, D. Murray, T. Drummond*; outre-Atlantique : *F. Dallaert, M. Kaess, S. Thrun*; et, plus récemment, Européenne : *V.S. Ila*

## 1.1 Enjeux

Les véhicules autonomes terrestres sont présents dans plusieurs domaines d'applications en milieu extérieur. Citons en trois : l'agriculture de précision, le transport de personnes ou de marchandises, et les robots de services (tondeuses, ramasseuses de balles de golf ...).

L'expansion - en nombre et en variété - de ces véhicules dans les prochaines décennies est un phénomène plus que probable. Pour nous en convaincre, observons les travaux récents réalisés dans les équipes du DARPA, de Google, ainsi que d'autres grands acteurs (constructeurs automobiles, laboratoires de recherche ...). Ces acteurs s'intéressent au domaine des ITS (Intelligent Transportation Systems), en raison du potentiel en terme de sûreté, et en terme d'économie d'énergie. Ces sujets sont, de plus, aujourd'hui, au coeur d'enjeux environnementaux. Ces enjeux sont les pierres angulaires de l'avenir de la robotisation, et ils décideront de son expansion, ou pas.

Outre les ITS, les robots autonomes sont commercialisés par des entreprises comme iRobot, aldebaran Robotics ou Boston Dynamics, qui proposent, aujourd'hui, une large gamme de robots de service autonomes, destinés à des applications civiles et militaires.

Ces prototypes de véhicules paraissent évolués, mais plusieurs défis scientifiques demeurent :

- La garantie de sûreté, et de réactivité en cas de panne.
- La zone d'action des véhicules (gestion de l'énergie, navigation mixte indoor/outdoor).
- L'utilisation dans des milieux non-structurés, en particulier sur des terrains inconnus a priori.
- Le coût des composants (capteurs, effecteurs, calculateurs embarqués ...).
- ...

Chacun de ces problèmes est un axe de recherche dans la communauté scientifique, et en particulier la recherche en robotique. On peut enfin dire que l'enjeu du travail des 20 dernières années en robotique reste tout simplement la démocratisation de cette robotique. En effet, bien que nous soyons persuadés de l'importance de cette expansion, elle ne sera envisageable que si les difficultés inhérentes (en particulier la sûreté et le coût des composants) seront mieux traitées.

## 1.2 Motivation

Nous nous intéresserons ici à un cas particulier d'application : la robotique de service accessible à un large public. Domaine dans lequel le prix des capteurs est un critère essentiel dans le processus de conception du système robotique. Ce critère influe sur le design du robot. De plus, en robotique mobile de service, le mobile (par exemple un robot rondier), évolue en milieu partagé avec des humains. De ce fait, il est primordial de caractériser avec précision ce que nous nommerons le "modèle d'erreur". Dans notre cas, l'erreur de navigation est la différence entre la position réelle et la position estimée. Pour des paramètres estimés continus (position, vitesse ...), notre "modèle d'erreur" suit alors une densité continue. Dans le cas de paramètres discrets (ambiguïtés de phase ...), le modèle d'erreur sera une densité discontinue. Dans tous les cas, notre objectif, dans cette thèse, sera de déterminer au mieux le comportement de notre (nécessairement imparfait) système de navigation robotique.

NAV ON TIME est une jeune entreprise possédant une compétence en réseaux de navigation par satellite - ou GNSS, pour Global Navigation Satellite System - (GPS, EGNOS, GALILEO ...).

Cette entreprise souhaite voir se développer l'utilisation des réseaux GNSS dans les applications de type *robotique mobile*, en particulier, dans la robotique de service (robots ronds, robots tondeuses ...). Pour ce faire, cette PME<sup>3</sup> propose l'utilisation de récepteurs GPS sensibles et peu onéreux (des récepteurs mono-fréquence). Utilisés dans un mode de localisation différentiel, nommé L1-RTK<sup>4</sup>. Ce mode de localisation par satellite est la principale solution de navigation développée par cette société.

Conscient des limites de la solution GPS d'alors, notre motivation, dans cette thèse, fut d'étendre la capacité de la navigation robotique d'alors. Ainsi, d'abord limitée aux milieux extérieurs ouverts, nous avons montré qu'il était possible d'étendre la zone de couverture opérationnelle à l'ensemble des milieux extérieurs suburbains. Nos travaux se sont portés sur l'amélioration algorithmique de la navigation au DGPS. Puis nous avons aussi développé, au LAAS-CNRS, un logiciel générique de navigation robotique multi-capteurs. Nous y avons implanté une navigation hybride Caméra monoculaire/IMU, qui sera, pourquoi pas, adapté au cas de la navigation GPS précise.

Justifions le choix des composants.

### 1.3 Choix des capteurs

Nous commençons par constater que la majorité des systèmes de navigation en milieu intérieur actuels se basent sur des capteurs laser 2D (qui permettent aussi la détection d'obstacles). Cependant ce type de capteur est relativement cher. De plus, en milieu semi-urbain - dans lequel les obstacles sont éloignés de plusieurs mètres - les capteurs lasers bas-coût s'avèrent inefficaces. La solution est alors l'utilisation de lasers 3D (de type Velodyne), ils peuvent fonctionner en intérieur comme en extérieur, cependant le budget nécessaire pour acquérir ce type d'appareils est incompatible avec notre étude sur une offre en robotique de service bas-coût.

L'utilisation d'un capteur visuel en mode stéréo apparaît alors comme une solution viable pour naviguer dans une scène en 3D en milieu extérieur, nous allons cependant nous intéresser à la vision monoculaire (et à une version multi-caméra, à champ non-recouvrant), et donc aux méthodes de type '*structure from motion*'. Outre le caractère hautement intéressant<sup>5</sup> de percevoir le monde jusqu'à la ligne d'horizon, la calibration est considérée comme plus simple, et les applications potentielles plus étendues.

En milieu extérieur, le capteur bas-coût par excellence est le récepteur GPS mono-fréquence en mode différentiel local, dont la précision sub-métrique se dégrade rapidement en présence d'arbres, ou de bâtiments (on parle soit de masquage, soit de multi-trajets). D'un autre côté, les performances des récepteurs RTK bi-fréquence - outre trop onéreux pour équiper un robot de service - sont sujettes aux mêmes problèmes en présence d'arbres ou de bâtiments.

Les capteurs alternatifs sont les tags RFID, nécessitant une modification de l'environnement, et dont les performances en localisation sont insuffisantes pour la majorité des applications robotiques, la localisation par *UWB*, dont la mise en place est onéreuse, et enfin le marquage au sol (fil enterré, peintures au sol ...), dont les inconvénients sont l'entretien des marqueurs, la dégradation

---

3. PME : Petite et Moyenne Entreprise.

4. Real Time Kinematic, utilisant les mesures de phase de la fréquence L1 (1575 MHz).

5. scientifiquement.



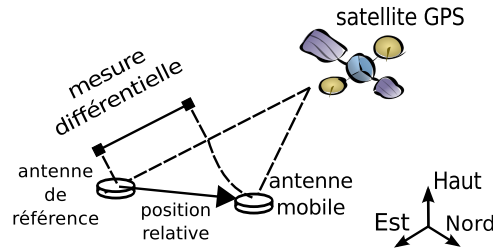


FIGURE 1.1 – Principe du GPS différentiel local : les mesures différentielles dépendent de la position du mobile dans le référentiel local (ENU) centré sur la station de référence. Les biais communs (les biais atmosphériques) sur les mesures sont éliminés par ce mode de positionnement.

visuelle du site, et le manque de flexibilité de la trajectoire du robot en opération.

Dans notre contexte, le couplage GPS/Vision apparaît donc comme une solution idéale pour résoudre notre problème de navigation robotique bas-coût en milieu semi-urbain. Nous proposons en outre un système original appelé "bi-caméra grand angle", et un ensemble d'algorithmes de navigation compatibles, fondés sur les approches récentes du SLAM visuel.

## 1.4 Intérêt du GPS en mode différentiel local

Bien que le réseau GPS soit destiné à la navigation à l'échelle terrestre, son utilisation en mode différentiel local fut introduite dans les années 1990, pour résoudre un problème de précision de positionnement dû au SA - pour Selective Availability - : un brouillage volontaire des signaux par l'armée américaine, retiré le 1<sup>er</sup> Mai 2000. Décrivons ce mode de localisation.

Le mode de localisation dit "différentiel local" nécessite deux récepteurs GPS, l'un fixe (appelé référence) et l'autre mobile. Alors que chacun des récepteurs mesure ses pseudo-distances (distances entre l'antenne au sol et les satellites plus une valeur de biais d'horloge), les mesures dite différentielles, sont les différences satellite par satellite, des mesures de pseudo-distance (voir figure 1.1).

On observe aisément que ces nouvelles observations sont insensibles aux biais communs des récepteurs (en particulier les biais d'horloge des satellites, et les effets atmosphériques pour des distances de quelques kilomètres). En contrepartie, le mode de localisation est rendu relatif à la station de référence : la station de référence devient le centre du repère dans lequel on réalise la navigation.

Pour l'application robotique qui nous intéresse, ce mode de localisation relatif ne pose pas de problème majeur, puisque la carte (de la zone à explorer, à surveiller ...) est définie dans un référentiel local, et que les distances de travail n'excèdent, en général pas, quelques kilomètres. La précision du positionnement relatif classique (non-RTK) est sub-métrique en ciel ouvert.

Les alternatives GNSS de positionnement sub-métriques sont les réseaux RTK, et le PPP.

Les réseaux RTK sont des stations de référence espacées de quelques centaines de kilomètres les unes des autres. Avec l'ensemble des mesures GPS reçues, une interpolation est faite pour générer des corrections (des mesures de pseudo-distance virtuelles), qui sont appliquées aux mesures du

récepteur mobile.

Le PPP (Precise Point Positioning) utilise aussi un réseau des stations, mais réparties plus largement autour du globe, le plus connu de ces réseaux est le réseau IGS. A partir de l'ensemble des mesures de ces réseaux, des modèles fins des horloges satellites, des trajectoires des satellites, et des délais ionosphériques sont estimés. Ces modèles peuvent être utilisés pour accéder à un mode de positionnement centimétrique en mode absolu. Aujourd'hui seuls les récepteurs bi-fréquence accèdent au mode PPP.

Bien que les réseaux RTK, et le PPP soient des méthodes de positionnement sub-métriques, elles nécessitent un lien de communication internet haut débit permanent (Réseaux RTK), ou l'utilisation de récepteurs bi-fréquence. Ces solutions sont évidemment trop onéreuses pour la robotique de service. Rappelons que le coût d'un récepteur bi-fréquence est de l'ordre que 10000 euros, et qu'un ensemble de deux stations fonctionnant en mode RTK bi-fréquence coûte aux alentours de 15000 euros.

## 1.5 Le SLAM Visuel

Souvent perçu de manière anthropomorphique, un capteur optique possède de nombreux avantages.

Comme un télémètre laser, il permet d'accéder à l'observation d'éléments en 3D de notre scène. On observe alors aux éléments en 3D, soit par stéréo-vision, soit en exploitant le mouvement de notre capteur (SFM<sup>6</sup>).

Ce capteur permet d'identifier visuellement les objets présents dans notre scène. Des algorithmes efficaces utilisent la segmentation des images (selon des critères de couleur, de texture ...), qui permet de reconnaître des visages, des obstacles ...

Les applications dans un contexte de localisation robotique restent pour l'instant au stade de recherche, probablement en attendant que les problèmes de complexité<sup>7</sup> et d'inconsistance du SLAM<sup>8</sup> ne soient résolus. Le SLAM est un domaine particulièrement actif en robotique, consistant à 1-cartographier et 2-localiser le mobile dans la carte en cours de création.

L'utilisation de ce capteur reste, dans notre cas, très intéressante dans un contexte de localisation pour deux raisons : d'abord il est souvent monté en tant que charge utile à bord d'un robot mobile (pour la prise de contrôle manuel, la surveillance ...), et ensuite parce qu'il s'agit d'un capteur qui peut réaliser des fermetures de boucles<sup>9</sup> (contrairement aux IMU et aux odomètres qui réalisent de la navigation à l'estime<sup>10</sup>).

Les dix dernières années furent le théâtre de résultats nous permettant d'être confiants dans l'utilisation future du SLAM visuel dans la navigation robotique.

---

6. SFM : Structure From Motion.

7. La complexité affecte le temps de calcul et donc l'aspect "temps-réel" de la navigation.

8. L'inconsistance en SLAM correspond à l'inexistence d'un modèle statistique d'erreur. Des méthodes récentes semblent corriger ce problème.

9. La fermeture de boucle est le retour dans une situation connue (un lieu par exemple), qui permet de stopper la propagation de l'incertitude sur l'état du véhicule.

10. dead-reckoning

## 1.6 Fusion GPS/Vision

Les mesures GPS que nous allons utiliser sont les mesures brutes (nommées pseudo-distances), et les mesures caméra sont les images vidéo. Notre but sera de fusionner ces mesures pour obtenir un positionnement dont la précision sera suffisante pour réaliser une navigation précise (précision sub-métrique).

Il est intéressant de noter que la fusion de capteurs repose sur des fondements forts en théorie des probabilités et, en particulier, en inférence Bayésienne. Nous adopterons l’approche statistique pour décrire nos observations (les données capteur). Ce choix est justifié par les résultats satisfaisants obtenus depuis près de vingt ans dans la communauté de chercheurs en robotique. Les mesures seront ensuite fusionnées pour fournir une *meilleure estimée* de la position du robot.

Aujourd’hui s’opposent deux approches SLAM : l’approche dite par filtrage (proposant de marginaliser les anciennes positions du robot pour réduire le nombre de paramètres, en utilisant le fait que les paramètres de position et d’attitude du robot sont un sous-système markovien), et l’approche dite par lissage (conservant les positions intermédiaires et ré-estimant celles-ci régulièrement). Nous nous intéresserons à la première approche, qui, seule, permet d’atteindre des performances temps-réel en navigation robotique.

## 1.7 Spécifications

Notre type de robot est un robot mobile terrestre. Ce robot évolue en milieu semi-urbain (en présence d’habitations, de végétation ...). Le cas typique d’utilisation est le robot rondier (dans une zone industrielle), ou un robot de tonte de gazon (évoluant dans un parc urbain).

Nous cherchons à créer un système robotique qui puisse naviguer de manière autonome dans son environnement de travail. Cet environnement de travail n’est ni altéré par la présence de marques (plots, lignes au sol ...), ni par la présence de balises invisibles (fil enterré, tags RFID ...).

Pour mener à bien sa tâche, notre système requiert donc un positionnement qui soit temps-réel, et sub-métrique (cela même dans des conditions de visibilité faibles, et en présence d’humains).

Dans ce but, notre problématique est la suivante : proposer un système de navigation qui utilise avantageusement le concept de SLAM visuel, et celui du GPS en mode différentiel local. Puis proposer une solution qui soit viable sur de longues distances (ce qui pose un problème majeur en SLAM visuel), et qui respecte un contrat d’intégrité du positionnement (ce qui pose un problème majeur aussi bien en SLAM que dans l’utilisation du GPS en milieu semi-urbain).

## 1.8 Architecture proposée

Nous verrons que l’approche SAM (Smoothing And Mapping) propose une solution pour réaliser un SLAM sur de grandes superficies, en temps-réel. Cependant le temps de réponse de l’opération de lissage peut excéder la demande fréquentielle en terme de positionnement. Pour cela, notre proposition est une décomposition du processus de géo-localisation en trois modules tournant en parallèle : le front-end (réalisant le traitement d’image et la construction des mesures GPS différentielles), le back-end (réalisant l’opération d’estimation qui peut se révéler complexe et

donc fournissant un résultat avec une périodicité de plusieurs secondes), et enfin la navigation (qui réalise un dead-reckoning depuis la dernière position en sortie du module back-end, en utilisant les images ou simplement l'odométrie). L'architecture proposée est présentée en figure 1.2, nous allons la décrire plus en détail.

On y voit que l'acquisition des mesures (GPS, vision, et odométrie) est réalisée à 30Hz. Le module "front-end" traite les mesures, pour en déduire des mesures utilisables par nos estimateurs d'état. Ces nouvelles mesures (en sortie du front-end) sont des descripteurs (pour les images), des transformations  $R_t$  (transformations rotation/translation) pour l'odométrie, et des mesures GPS-différentielles pour le capteur GPS en mode différentiel. La fréquence de construction des mesures en sortie du bloc "front-end" peut être inférieure à celle des entrées de ce bloc. Cela peut arriver lorsqu'on ne peut pas paramétrer nos capteurs pour acquérir les données à la fréquence du traitement.

Le bloc "back-end" réalise une optimisation paramétrique. Son temps de réponse peut atteindre plusieurs secondes (mais nous considérerons ce temps de réponse comme borné). Notre problème principal ici est que ce temps de réponse est supérieur à notre seuil de sécurité (d'1Hz).

Pour cette raison, un troisième bloc, le bloc "navigation", réalise un dead-reckoning. Entre deux réponses du bloc "back-end", ce bloc propage la dernière position estimée par une méthode simple, sous-optimale, dont le temps de réponse est garanti inférieur à notre seuil de sécurité.

Dans la partie droite de notre figure, on voit en encadré des cercles correspondants aux positions estimées dans les blocs "back-end" et "navigation". On y aperçoit le mode de fonctionnement de l'échanges d'états entre le back-end et la navigation. L'encadré du haut décrit ceci : lorsque le back-end à fini un cycle d'estimation, le résultat de queue (la dernière position estimée :  $P_k$ ) est envoyé à la navigation, qui propage les états jusqu'à la position courante ( $P_t$ ). Dans l'encadré bas, on voit que lorsque le back-end envoie le résultat de queue ( $P_k$ ), la navigation élimine tous les états précédant ( $P_{i < k}$ ), ce qui borne la mémoire nécessaire à notre processus de navigation.

Cette architecture est inspirée de l'approche de SLAM monoculaire PTAM (Parallel Tracking And Mapping) initiée par Klein [1]. La principale différence étant que notre back-end ne se limite pas à une opération de Sparse Bundle Adjustment tel que défini dans [2], mais d'un lissage incrémental utilisant une représentation basée sur la théorie des graphes appliquée à la résolution de systèmes non-linéaires. La seconde évolution est la présence du module appelé *navigation* qui est capable de fournir une estimation de la dernière position en un temps borné.

Le point critique dans ce processus de localisation est donc la garantie d'un temps de réponse borné du module "*back-end*". Des travaux récents tendent à prouver qu'un moyen d'assurer cette caractéristique, appelée "*constant time SLAM*" ou encore "*SLAM in  $\mathcal{O}(1)$* " peut être mis au point.

## 1.9 Contenu du mémoire

Ce mémoire est organisé comme suit :

Le chapitre 2 présentera un état de l'art, en particulier décrivant les approches de SLAM visuel, et la localisation sub-métrique par GPS et GPS différentiel local.

Le chapitre 3 proposera une méthode de navigation basée sur les mesures GPS brutes en mode différentiel local. En utilisant les mesures de phases différenciées en mode dit de *triple-différences*.

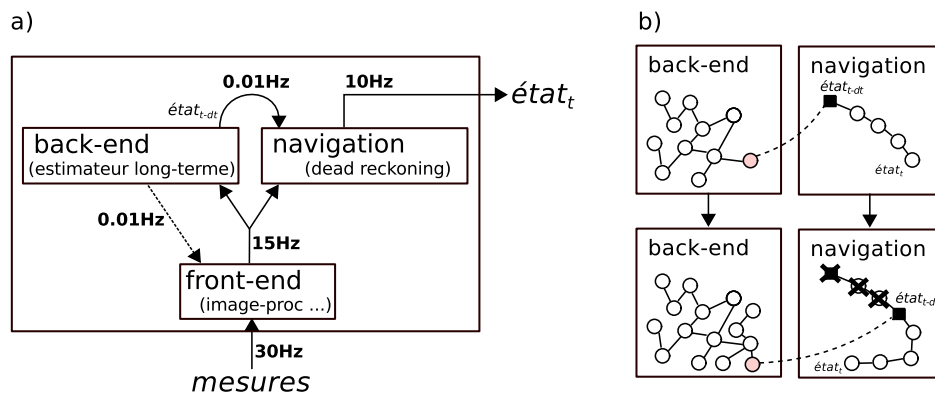


FIGURE 1.2 – Architecture composée de trois modules tournant en parallèle (a), et la liste des paramètres estimés dans le module "back-end" et le module "Navigation" (b).

Le chapitre se terminera par une série de résultats issus de tests en grandeur nature.

Le chapitre 4 présentera une méthode de navigation robotique proche de celle présentée au chapitre 3. En effet, nous nous intéresserons à une navigation au DGPS-RTK mono-fréquence. Une nouvelle approche de levée d'ambiguïté est présentée, et les résultats, comme pour le chapitre précédent, seront issus de tests en milieu extérieur.

Le chapitre 5 présentera deux implantations de SLAM visuel développés au LAAS-CNRS. Ces logiciels sont destinés aussi bien à la recherche de nouvelles méthodes, qu'aux applications embarquées de faible puissances. Un cas d'usage a été notamment développé, dans lequel une hybridation Vision/centrale inertielle tourne à fréquence image (soit, dans notre cas, à 120Hz).

Enfin le chapitre 5 conclura et proposera des directions pour l'extension futur de notre recherche.

## Chapitre 2

# État de l'art

Nous allons passer en revue l'état de l'art sur la navigation robotique, aussi bien utilisant des capteurs GPS, que des capteurs visuels, en considérant aussi les systèmes hybrides.

Dans la première section nous proposerons un état de l'art dans le domaine de la navigation par GPS, et en particulier les méthodes de positionnement sub-métrique (DGPS, DGPS-RTK, et diverses méthodes d'Assisted GPS). Nous concluons avec les travaux les plus récents en robotique, notamment les méthodes d'hybridation du capteur GPS avec d'autres senseurs, dont la vision.

Dans la seconde section, nous aborderons le domaine de la vision par ordinateur. Nous présenterons les modèles et les applications possibles avec ce type de capteur. Nous verrons que les opérations peuvent être hiérarchisées en opérations de détection de primitives (extraction), de reconnaissance (matching) et d'estimation de paramètres stochastiques dans la scène observée (cartographie ...).

Enfin dans la troisième section nous étendrons notre état de l'art au problème particulier du SLAM. Le SLAM étant un domaine très actif en recherche pour traiter la navigation robotique, les développements récents pour la résolution du problème du SLAM permettent des solutions incrémentales, exactes et quasi-temps-réel. Nous passerons en revue les méthodes de filtrage, puis de lissage et enfin les méthodes les plus récentes de lissage incrémental développées notamment à l'Université d'Oxford et de Georgia-Tech.

### 2.1 Localisation GPS sub-métrique

Initialement créé et utilisé dans un contexte militaire, le réseau de satellites de navigation GPS s'est étendu aux applications civiles durant les années 1990. Le réseau est actuellement utilisé aussi bien pour la localisation de véhicules terrestres, que maritimes, aériens, et même spatiaux.

Malgré le fait que le DoD<sup>1</sup> puisse mettre fin, ou brouiller à tout moment la radio-communication du réseau GPS, l'intérêt reste majeur dans les développements d'applications civiles. D'une part parce que des composants électroniques (récepteurs GPS, processeurs et moyens de communication) de bonne qualité sont disponibles à des prix raisonnables (ce qui n'était pas le cas il y a vingt ans), et d'autre part parce que les systèmes embarqués sont sujets à la loi de Moore<sup>2</sup>.

---

1. DoD : Department of Defense (des Etats-unis d'Amérique).

2. La loi de Moore, ou conjecture de Moore, a été énoncée en 1965 et ré-évaluée en 1975 par son auteur. Ce

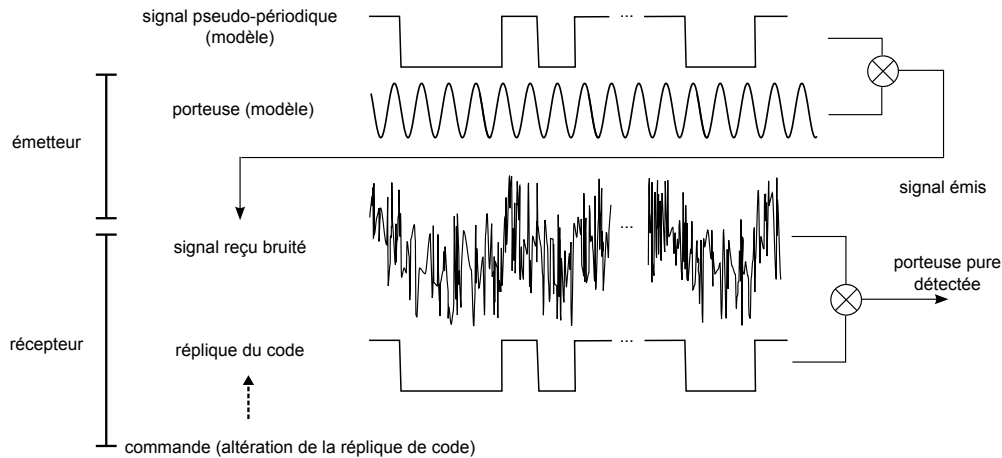


FIGURE 2.1 – La mesure de code est issue de la corrélation d'un signal pseudo-périodique par un corrélateur appelé DLL (Delay Lock Loop), représenté par  $\otimes$ . La commande est un offset temporel appliqué à la réplique. Le pic d'énergie correspond donc à une mesure de biais temporel : la mesure de code.

Alors que l'utilisation classique d'un récepteur GPS permet d'observer la position et la vitesse (nommée mesure *PV*) du mobile qui porte le récepteur, nous choisirons ici de ne pas exploiter les résultats des traitements standards faits dans le récepteur, et de travailler avec les données brutes (appelées raw-data) que sont les mesures GPS de code, de phase et de Doppler, issues des boucles de corrélation internes des récepteurs GPS.

Décrivons les mesures GPS brutes.

### 2.1.1 Mesures GPS brutes

Il existe trois types de mesures GPS brutes :

**La mesure de code** La mesure de code est issue du suivi du code pseudo-aléatoire du signal transmis par les satellites GPS (figure 2.1).

**La mesure de phase** La mesure de phase est issue du suivi de la porteuse pure reconstituée à partir du signal transmis par les satellites GPS (figure 2.2).

**La mesure Doppler** La sortie "Doppler" est égale à la commande de la PLL.

#### *Paramètres "standard" affectant les mesures brutes*

Chaque mesure de code est affecté par :

- La distance entre l'antenne émettrice (satellite) et réceptrice (utilisateur).

---

modèle évolutif est que le nombre de transistors, à prix constant, doublerait tous les deux ans..

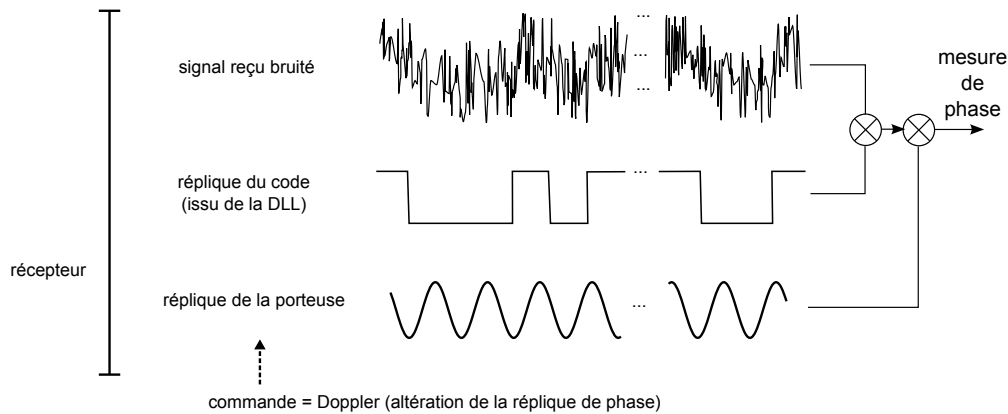


FIGURE 2.2 – La mesure de phase code est issue de la corrélation du signal via un corrélateur appelé PLL (Phase Lock Loop), représenté par  $\otimes$ . La commande est un offset temporel appliqué à la réplique purement périodique. Le pic d'énergie correspond aussi à une mesure de biais temporel : la phase. Cette mesure est ambiguë car le modèle de la réplique est purement périodique.

- Les biais d'horloge des satellites et des récepteurs.
- Les effets atmosphériques : effet de la traversée de la ionosphère, effet de la réfraction sur la troposphère.
- Les réflexions du signal sur les surfaces proches de l'antenne réceptrice (réflexions sur les murs, le sol ...).

Chaque mesure de phase est affectées par les mêmes paramètres (avec des modèles différents), ainsi que d'une variable supplémentaire qui est l'ambiguïté de phase (voir figure 2.2).

La mesure Doppler est égale à la commande du corrélateur de phase. Cette mesure est affectée par :

- La distance entre l'antenne émettrice (satellite) et réceptrice (utilisateur).
- Les dérives d'horloge des satellites et des récepteurs.
- Les dérives des effets atmosphériques.
- Les variations des effets des réflexions du signal (multi-trajets et trajets multiples).

### *Trajet réfléchis*

En positionnement GPS (différentiel ou non), le paramètre le plus difficilement modélisable est le trajet réfléchi[3][4]. Il existe deux phénomènes issus de la réflexion de l'onde GPS sur une surface proche de l'antenne réceptrice : le simple trajet réfléchi (le signal est réfléchi puis arrive sur l'antenne), et le trajet multiple, pour lequel le signal réfléchi atteint l'antenne réceptrice, mais le signal direct aussi, ce qui provoque un phénomène d'interférence non-aléatoire, qu'il est difficile





(a) Trajet réfléchi.

(b) Trajet multiple.

FIGURE 2.3 – Alors qu'un trajet réfléchi (a) provoque un délai dans l'estimation des pseudo-distances, un trajet multiple (b) affecte le signal comme un brouillage non-aléatoire (l'effet sur les mesures de pseudo-distances est donc difficilement modélisable).

de modéliser une fois l'opération du corrélateur passée. La figure 2.1.1 présente ces situations.

L'influence d'un trajet multiple sur la mesure de code est décrite dans [3], il s'agit d'un bruit coloré de moyenne nulle, sur les mesures de pseudo-distance. Bien que cette source d'erreur puisse provoquer des biais dans les mesures de code de plusieurs centaines de mètres, près des bâtiments et des arbres, il existe quatre solutions pour limiter son effet : les deux premières opèrent en amont (au niveau de l'antenne), les deux suivantes identifient et rectifient cette source d'erreur directement sur les mesures :

- L'utilisation d'une antenne équipées du système "shock ring" qui empêche des ondes réfractées d'influer sur la mesure.
- L'utilisation d'une antenne à gain actif (augmentant le gain de la réception dans une certaine direction angulaire).
- L'élimination d'une mesure ou une autre après la détection d'une anomalie de comportement de la mesure (la méthode RAIM<sup>3</sup> est une variante utilisée en aéronautique...). Mais cette méthode nécessite un nombre de mesures élevé.
- La modélisation de l'effet du trajet réfléchi et la construction d'un estimateur adapté[5].

Notons que les méthodes sont soit coûteuses en matériel, soit elle restent impraticables par manque de mesures redondantes<sup>4</sup>, ou encore elles nécessitent une puissance de calcul qui les rend irréalisables sur des systèmes embarqué bas-coût.

Les paramètres standard affectant les mesures étant décrites, nous pouvons décrire le bruit résiduel.

### *Bruit thermique*

Le bruit thermique additionnel, aussi bien sur le code que sur la phase, est considéré comme blanc et Gaussien. Il est de l'ordre de grandeur de la période du signal suivi par le corrélateur,

3. RAIM : Receiver autonomous integrity monitoring.

4. Les trajets réfléchis surviennent quand le porteur du récepteur est proche de bâtiments, donc dans des milieux où le nombre de mesure est réduit, et donc la redondance des mesures est généralement faible.

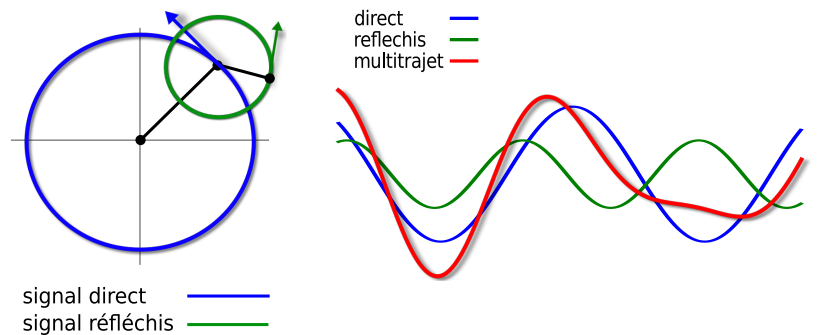


FIGURE 2.4 – Effet d'un trajet multiple sur le signal de phase. A gauche : Un modèle de génération du signal affecté par un trajet multiple (on considère que la réflexion affecte la puissance et la fréquence du signal de phase). A droite : application de ce modèle de trajet multiple sur quelques cycles. Il est clair que la qualité de la mesure de phase est lourdement affectée par un multi-trajet.

ce qui représente quelques mètres pour la mesure de code, et quelques millimètres pour la mesure de phase. Un indicateur, en temps-réel, de la puissance du bruit thermique est le rapport signal sur bruit ( $C/N_0$ ) des corrélateurs de code et de phase.

### *Amélioration des performances*

L'état de l'art propose trois pistes pour améliorer les performances du positionnement par GPS :

- Modéliser le plus précisément possible tous les paramètres des modèles (approche PPP).
- Éliminer les effets des paramètres de nuisance (paramètres que l'on peut mal modéliser) en procédant à des modifications architecturales (différentiations des mesures et changement de repère) (Approche DGPS ou Réseaux RTK).
- Utiliser d'autres capteurs (IMU, Vision, laser ...).

Nous présenterons ces méthodes dans les sections suivantes. Nous commenceront par introduire les méthodes "sans infrastructures" (DGPS, DGPS-RTK, ambiguïtés flottantes et réseaux RTK), puis nous parlerons des méthodes "avec infrastructure" (PPP). Enfin nous aborderons les travaux en fusion de capteurs.

Commençons par les méthodes sans infrastructures (les moins onéreuses aussi bien en mise en place, qu'en maintenance).

### **2.1.2 Méthodes de localisation sub-métriques sans infrastructure : DGPS, RTK, ambiguïtés flottantes**

#### *DGPS : précision métrique*

GPS souffrant historiquement du brouillage SA<sup>5</sup>, la précision du positionnement avant 2000 était de l'ordre de 50m en ciel ouvert. Ce brouillage était un mode commun à tous les récepteurs à

5. SA : Selective Availability, brouillage volontaire retiré en 2000.

un instant donné. Ainsi, le GPS différentiel local (local DGPS) apparaissait comme une solution viable consistant à utiliser un récepteur GPS de référence, et à réaliser un positionnement dit différentiel [6]. La précision du positionnement relatif ne souffre pas (ou peu) du brouillage SA, et peut atteindre une précision de l'ordre du mètre en ciel ouvert, pour une zone de quelques kilomètres autour de la station de référence.

Aujourd'hui, même si la SA n'est plus active, ce mode est toujours utilisé pour obtenir un positionnement qui ne soit pas ou peu affecté, ni par les effets atmosphériques, ni par les biais d'horloge des satellites. En effet, ces paramètres sont les plus difficiles à modéliser (hors multi-trajet). Un autre avantage non négligeable de cette méthode de localisation est l'investissement nécessaire, assez faible vis-à-vis des méthodes sub-métriques alternatives. Dans le cas du DGPS, il suffit d'installer une station de référence fixe, et un moyen de communication entre cette station fixe et le récepteur mobile. La station de référence émet soit les mesures brutes, soit des modèles de correction de mesures (la norme RTCM<sup>6</sup> est un standard dédié à ce type de corrections).

Même si ce mode de localisation est plus précis que le mode de positionnement absolu, la précision métrique peut s'avérer insuffisante pour certaines applications. Nous allons aborder le mode de positionnement RTK<sup>7</sup>, proposant une précision centimétrique.

### *DGPS-RTK : précision centimétrique*

Parmi les travaux les plus aboutis en localisation par GPS, on peut citer l'utilisation des mesures de phase en mode différentiel local pour réaliser un positionnement dit DGPS-RTK (la précision du positionnement relatif est alors centimétrique). Ce mode de positionnement GPS utilise le fait que les mesures de phase sont ambiguës d'un nombre (entier et constant) de longueur d'onde. Lever l'ambiguïté de phase revient à trouver les biais (entiers) pour chaque mesure de phase, par un test statistique.

La plupart des récepteurs GPS réalisant un positionnement DGPS-RTK sont chers car ils utilisent des récepteurs bi-fréquence ( $L_1$  et  $L_2$ ). L'acquisition de la seconde mesure de phase GPS (sur la fréquence militaire  $L_2$ ) étant brevetée. Le prix des licences est le principal frein à l'expansion du mode de localisation DGPS-RTK centimétrique. En effet, le problème de la levée d'ambiguïté de phase elle-même à l'aide de récepteurs bi-fréquence est connue, et de nombreuses publications proposent des solutions dont la complexité algorithmique est largement réalisable par les systèmes embarqués modernes [7, 8, 9, 10, 11]. Ni le prix des composants, ni la complexité algorithmique ne sont des freins à l'expansion du mode de localisation DGPS-RTK  $L_1 + L_2$ .

**Remarque.** Les futurs réseaux de satellites de positionnement, dont GALILEO, ainsi que les nouvelles générations de satellites GPS fournissent plusieurs fréquences civiles, dont l'acquisition des mesures de phase ne sont pas soumises à ces brevets. On peut donc imaginer une expansion prochaine du mode de positionnement centimétrique RTK multi-fréquence [12, 13] (voir figure 2.5).

### *Positionnement DGPS-RTK bas-coût*

En attendant la disponibilité des nouvelles fréquences civiles, certains travaux révèlent que le mode de positionnement DGPS-RTK, à l'aide de récepteurs GPS aujourd'hui bas-coût (mono-

6. RTCM : Radio Technical Commission for Maritime Services

7. RTK : Real Time Kinematic

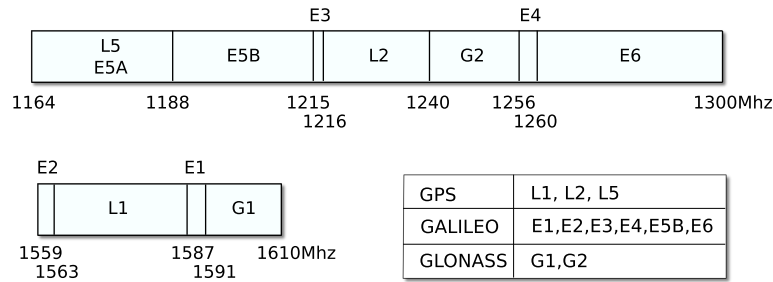


FIGURE 2.5 – L’allocation des fréquences pour GPS, GLONASS, et GALILEO : les principaux réseaux de satellites de navigation.

fréquence  $L_1$ ) [14, 15] est réalisable. Il existe même un logiciel gratuit (et open-source) nommé RTKlib, proposant l’acquisition des mesures de plusieurs modèles de récepteurs GNSS, ainsi que la réalisation d’un ensemble de tests reconnus permettant la levée d’ambiguïté de phase aussi bien en mono-fréquence qu’en bi-fréquence. Le problème de la levée d’ambiguïté entière reposant sur un test d’hypothèse, les travaux démontrent que si l’utilisateur possède les mesures bi-fréquence, alors un test relativement simple peut être construit. Le résultat de ce test est, dans le cas d’un ciel ouvert, avec une forte probabilité, le bon. On montre même que, dans la plupart des cas, les mesures prises en un seul instant permet de lever les ambiguïtés de phase.

Cependant, en mono-fréquence, la puissance du test ”instantané” (c’est à dire en n’utilisant que les mesures d’un instant) est tellement faible, que la probabilité de choisir les bonnes valeurs entières est largement inférieure à 50%. Si l’on accumule les mesures dans le temps, et que l’on construit un test cumulatif, en mono-fréquence, le nombre d’hypothèses étant plus grand, le temps d’initialisation<sup>8</sup> est plus lent, passant de quelques secondes maximum en mode bi-fréquence à quelques minutes, voire quelques dizaines de minutes en mono-fréquence [16], en ciel ouvert.

**Remarque.** Bien que le coût soit moindre, l’idée d’utiliser un mode de positionnement GPS différentiel RTK en mono-fréquence n’est pas répandue. En effet, le temps de convergence de plusieurs minutes, avant d’obtenir une position centimétrique, diminue largement la pertinence dans une opération de recueil de la vérité terrain, et est dommageable pour la plupart des application en robotique. Cependant, certains des robots de service civils fonctionnent en continu, et sont donc peu sensibles au temps d’initialisation qui n’apparaît qu’au démarrage de l’application. Ce mode semble donc proposer une solution viable au problème de localisation centimétrique bas-coût en milieu extérieur. Avec NAV ON TIME, nous avons proposé, dès 2010, la mise en place de ce type de navigation sur une tondeuse à gazon dans le cadre d’un projet Européen nommé ”MOW-BY-SAT” [17, 18].

### *Ambiguïtés flottantes*

Dans le cas de l’utilisation de récepteurs mono-fréquence, pendant l’étape d’initialisation (qui peut durer plusieurs minutes), il est possible d’utiliser la mesure de phase sans lever l’ambiguïté

8. Appelé aussi TFF (Time to First Fix), ou temps de levée d’ambiguïté

entière, en réalisant une relaxation. La relaxation consiste à relâcher la contrainte entière sur l'ambiguïté de phase, et de considérer ce biais comme une variable aléatoire de type réel à densité continue.

Estimer le biais de phase comme une variable réelle dans le processus de positionnement est nommé "mode à ambiguïté flottante" (floating ambiguity mode)[19]. Considérant que la valeur de l'ambiguïté de phase est non seulement continue, mais aussi fixe dans le temps, il est possible d'éviter de l'estimer en différenciant temporellement ces mesures, et en construisant une nouvelle mesure dite de triple différence [20, 21]. Nous avons eu l'occasion, lors de nos travaux de recherche, de développer notre propre approche du mode de localisation par triple-différence de phase [22], notre contribution sera décrite dans le chapitre 3.

**Remarque.** Les mode de navigation dits à "ambiguïtés flottantes" et à "triple différence de phase" sont applicables pour les phases d'initialisations, et pour les phases critiques de la navigation (lorsque le mode RTK n'est plus applicable, par manque de mesures redondantes).

La précision de ces modes DGPS dits "dégradés" est de l'ordre de la dizaine de centimètre, en ciel ouvert. Comparativement, le mode de positionnement RTK, qu'il soit en mono-fréquence, ou en bi-fréquence, est d'une précision de l'ordre d'un centimètre.

### 2.1.3 Méthodes de localisation sub-métriques avec infrastructure : A-GPS, Réseau RTK et PPP

#### *A-GPS*

Certaines infrastructures sont mises en place au niveau national, continental, ou mondial pour permettre d'augmenter les performances du réseau GPS par l'émission, par internet, ou par ondes radio, de corrections locales. Des stations au sol renseignent une modélisation aussi fine que possible des paramètres affectant les mesures GPS. Plus ces paramètres des modèles sont précis, plus l'estimation de la position de l'utilisateur le sera. EGNOS et WAAS sont les SBAS (Satellite Based Augmentation System) les plus importants[23].

L'utilisation de ces modèles pour améliorer le processus de positionnement est nommé A-GPS (Augmentation-GPS). Le gain de performances dépend de la zone géographique cible : les modèles sont plus ou moins précis selon le nombre de station au sol, et de la prédictibilité des effets atmosphériques[24].

Plus récemment, les opérateurs téléphoniques utilisent le réseau 3G pour communiquer les éphémérides GPS pour diminuer le temps d'acquisition du signal. Ce type d'amélioration est nommé A-GPS pour Assisted-GPS et tend à s'étendre avec les nouvelles générations de smartphones.

#### *PPP*

Le PPP est un mode de positionnement absolu de précision centimétrique qui utilise les mesures de phase (et résoud les ambiguïtés entières) [25, 26].

Alors que les réseaux de stations-sol EGNOS et autres WAAS permettent d'assurer un service de précision dans certaines régions (EGNOS en Europe, WAAS aux Etats-Unis), d'autres réseaux

de stations existent. Le plus grand de ces réseaux alternatifs, destiné à l'A-GPS est l'IGS<sup>9</sup>. Il s'agit du fournisseur principal de données GPS dans le domaine du PPP (Precise Point Positioning).

Bien que le PPP propose une solution intéressante pour un positionnement centimétrique en mode absolu, l'utilisation de récepteurs bi-fréquence est nécessaire pour réaliser du PPP. Il est aussi important de noter qu'une connexion à Internet, ou un réseau 3G, reste indispensable.

Enfin, nous pouvons noter que le principe de la localisation par GPS en mode PPP est largement soutenu par un équipe du CNES<sup>10</sup>, basée en France, à Toulouse.

### *Réseaux RTK*

Finalement on observe aujourd'hui le développement de réseaux RTK. Ces réseaux créent des stations de référence RTK virtuelles (par modélisation précise, ou par interpolation des mesures de phase), permettant à l'utilisateur du service de conserver un mode de positionnement RTK, et d'autre part, d'augmenter les distances accessibles par le mode RTK [27, 28, 29, 30].

La solution du réseau RTK est une alternative au PPP, mais nécessite aussi un réseau de communication sans-fil dense (ou un accès Internet). Que se soit dans le cas du PPP ou du réseau RTK, la mise en place et l'entretien du réseau terrestre, et des moyens de communication est une lourde charge. Il est à noter que ces solutions sont aujourd'hui présentées comme fonctionnelles (en France le réseau TERIA en est un exemple), mais on note aussi qu'elles ne fonctionnent uniquement qu'avec des récepteurs GPS bi-fréquence.

On peut imaginer l'introduction future des réseaux GLONASS, et GALILEO. Ce phénomène, bien qu'incertain, permettrait de réaliser une navigation RTK en milieu suburbain. Cette introduction pourra aussi, peut-être, permettre de proposer un service de positionnement RTK en mono-fréquence.

#### 2.1.4 Hybridation avec d'autres capteurs

Les principales limites de la navigation au GPS persistent malgré les solutions disponibles, qu'elles soient avec ou sans infrastructures. Citons la fréquence des mesures (limitée à quelques Hz au maximum) et les durées de pertes/ré-acquisitions du signal en cas de masquage (passage sous un arbre, près d'un bâtiment ...).

Ainsi, la plupart des travaux, en navigation par GPS, à partir des années 2000 se sont portés sur l'hybridation GPS avec une centrale inertielle [31, 32]. La fusion GPS/IMU<sup>11</sup> possède l'avantage d'être 100% passive (aucun signal n'est émis pour la localisation). Il s'agit d'un avantage stratégique dans certains contextes.

**Remarque 2.1.** Il existe trois types de fusions d'un récepteur GPS avec d'autres capteurs. La fusion lâche (utilisant les sorties position et vitesse du récepteur), la fusion serrée (utilisant les mesures brutes de pseudo-distances), et la fusion très serrée (la fusion agit sur les corrélateurs internes du récepteurs GPS). Nous nous placerons dans la seconde catégorie.

---

9. IGS : International GNSS Service

10. CNES : Centre National des Études Spatiales

11. IMU : Inertial Measurement Unit - Centrale Inertielle en Français

Le travail le plus aboutis en hybridation du GPS avec un capteur visuel est celui de Carlson [33]. L'auteur propose d'utiliser un lisseur non-linéaire, basé sur l'inférence dans une représentation par graphe de facteur, pour réaliser un SLAM visuel. Les mesures GPS sont alors considérées comme temporellement corrélées et des paramètres additionnels sont estimés par le lisseur.

Le travail présenté dans [33] est novateur, il est issu de la communauté robotique, et en particulier la section travaillant sur le concept du SLAM.

Mais quel est l'état de l'utilisation du réseau GPS dans la robotique aujourd'hui ?

### 2.1.5 GPS en robotique

En robotique mobile, le réseau GPS en mode DGPS-RTK est couramment utilisé pour fournir une vérité terrain (*ground truth*), lors d'études, et de travaux de recherche et développement. Force est de constater que les récepteurs bi-fréquence représentent, dès lors, l'unique solution disponible sur étagère. Or, parce-que le prix de la licence d'utilisation d'un système DGPS-RTK bi-fréquence ne permet pas de démocratisation de ce moyen de localisation, l'utilisation du GPS hors contexte *ground truth* est impossible.

Une solution est alors l'utilisation de récepteurs bas-coût (mono-fréquence). Cette solution propose une précision d'une dizaine de mètre en mode absolu, et métrique en mode différentiel local. Or, il est difficile de la mettre en place le moyen de communication entre la station de référence, et le robot mobile. Ce moyen de communication devant être fiable, et posséder une bande passante suffisante pour transmettre les mesures brutes.

En conclusion, aussi bien concernant les méthodes d'augmentation de performances avec infrastructure<sup>12</sup> que sans<sup>13</sup>), le réseau GPS reste peu utilisé en robotique mobile. Contrairement à la vision par ordinateur, qui est, depuis longtemps, étudiée et utilisée, comme nous allons le voir dans la section suivante.



FIGURE 2.6 – Une image informative : nous reconnaissons une voie de véhicule probablement en sens unique, avec la présence d'obstacles mobiles. L'architecture et la tour Eiffel en arrière plan permet de dire que le véhicule porteur de la caméra se trouve à Paris (image Google Maps).

12. PPP, A-GPS

13. DGPS, DGPS-RTK et ambiguïtés flottantes

## 2.2 Vision par ordinateur

La vision par ordinateur est un domaine très actif depuis les années 1990. Basé sur une intuition simple : la vision permet d'accomplir de nombreuses tâches. Et pour preuve, privé de ce senseur, nos capacités sont lourdement amoindries.

Basé sur ce constat, nous savons qu'un capteur visuel nous fournit des mesures "riches" (texture, contour ...). La difficulté étant celle-ci : ce capteur nous apporte des mesures riches, mais "brutes". Brutes dans le sens où les informations sont difficiles à extraire de l'image, même en post-traitement. La figure 2.6 nous présente une rue de Paris, dans laquelle les informations extractibles sont riches, mais sont difficile à traiter par un programme informatique, surtout en temps-réel.

### 2.2.1 Synthèse d'information par capteur optique

**Extraction de primitives :**

*Détection de points d'intérêt :*

L'une des premières primitives que l'on peut extraire dans une image est le point. Bien que tout point dans une image corresponde à un rayon optique en 3D, on considère ici qu'un point est remarquable si :

- L'imagerie autour du point correspond à une réflexion de lumière sur une surface (non nécessairement plane) dans l'espace 3D.
- Le point est distinguable parmi tous les points de l'image (il est uniquement identifiable dans l'image).
- Le point possède de bonnes propriétés d'appariement avec un point correspondant au même objet observé dans d'autres images, malgré le changement de point de vue.

Selon ces critères, on peut dire qu'un point est remarquable si il existe un comparateur qui permet de retrouver ce même point, dans d'autres images de la même scène, sous différents points de vue. Or le changement de point de vue modifie notamment l'échelle et la rotation du patch autour du point d'intérêt.

Il existe deux niveaux dans la reconnaissance de points d'intérêt : l'extraction et la caractérisation. Pour obtenir des appariements qui aient le maximum de chance d'être corrélés correctement, il faut extraire un point donc les caractéristiques (la signature), soient invariantes avec le changement de point de vue.

#### **Harris**

L'un des extracteurs de points d'intérêt les plus connus et utilisé est l'extracteur de Harris [34]. Cet extracteur se base sur le fait que, pour qu'un point d'intérêt existe, il est nécessaire que les dérivées spatiales (en 2D) soient localement maximales.

**Remarque.** La notion de dérivée spatiale est définie par la variation du gradient lumineux sur un axe de l'image. Soit  $I$  une image et  $I(u, v)$  la valeur de luminance en  $(u, v)$ . Une des dérivée en  $u$  est  $\partial I(u, v)/\partial u = I(u + 1, v) - I(u - 1, v)$ . Mais il existe d'autres définitions d'une dérivée spatiale.



Le détecteur de point de Harris calcule, pour chaque pixel, la matrice  $M$  définie par

$$M = g(\sigma) * \begin{bmatrix} (\partial I/\partial u)^2 & (\partial I/\partial u)(\partial I/\partial v) \\ (\partial I/\partial u)(\partial I/\partial v) & (\partial I/\partial v)^2 \end{bmatrix} \quad (2.1)$$

ou  $g(\sigma)$  est un masque Gaussien de variance  $\sigma^2$  qui lisse l'image (smoothing factor) grâce à l'opérateur de convolution '\*'. On observe que la matrice  $M$  est le Hessien de l'opération de translation de l'image lissée. Un pixel sera localement identifiable si toute translation en 2D est identifiable, donc si les valeurs propres de  $M$  sont suffisamment grandes.

Harris propose le critère suivant s'appliquant sur  $M$  :

$$C = \det(M) - k \cdot (\text{trace}(M))^2 \quad (2.2)$$

avec  $k = 0.04$ . Les points sont considérés comme remarquables si le critère  $C$  est supérieur à un seuil donné.

On reproche à ce détecteur deux défauts : 1-le facteur  $k$  est arbitraire, et 2-le calcul de la fenêtre Gaussienne ralenti l'opération d'extraction.

### Shi-Tomasi

Un critère plus récent et très utilisé propose une alternative au paramètre  $k$ , il s'agit du critère de Shi-Tomasi [35], étant la plus petite des valeurs propres de  $M$ , définie par

$$C = M_1 + M_3 - \sqrt{(M_1 - M_3)^2 + 4M_2^2} \quad (2.3)$$

avec  $M_1$ ,  $M_2$  et  $M_3$  paramètres de  $M$  tels que

$$M = \begin{bmatrix} M_1 & M_2 \\ M_2 & M_3 \end{bmatrix} \quad (2.4)$$

La fenêtre Gaussienne, quant à elle, peut être remplacée par une fenêtre uniforme, mais au détriment des performances de détection [36].

### FAST

Un des détecteurs de point d'intérêt parmi les plus performants est FAST (Features from Accelerated Segment Test). Cet algorithme compare la valeur d'intensité d'un pixel central à ses voisins répartis circulairement avec un certain rayon. Le critère est alors le nombre de pixels consécutifs dont l'intensité est supérieure (ou inférieure) au pixel central.

### Extraction de segments

La seconde primitive la plus couramment extraite d'une image est la ligne (ou segment). En effet, les segments sont des renseignements sur la structure des objets planaires de la scène. Cette primitive est utile mais malheureusement difficile à extraire en temps-réel. Certains travaux ont cependant rendu possible l'extraction et l'utilisation de segments dans des applications temps-réel de cartographie [37, 38, 39, 40].

**Remarque 2.2.** Admettons le problème d'extraction réglé, le problème suivant est de proposer un modèle de représentation statistique d'un segment dans une carte 3D. Pour ce nouveau problème, une approche comparative des différentes représentations statistiques possibles de segments est présentée dans [41] et [42].

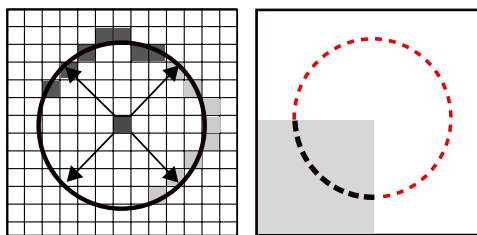


FIGURE 2.7 – Le détecteur FAST compte le nombre maximum d'éléments successifs plus lumineux et moins lumineux que le pixel central sur un cercle. Le score est le nombre maximum d'éléments successifs. Ce détecteur permet de distinguer les coins (figure de droite).

### *Autres primitives*

Les primitives plus complexes qui peuvent être extraites d'images sont des courbes (sphères, ellipses, quadriques . . .) aussi extraites à partir des contours comme les segments, les régions selon des critères d'uniformité de textures, de couleur . . . enfin les facettes planes via un ensemble de points obtenus par une transformation homographique . . . Ces opérations sont souvent réalisables uniquement en post-traitement. En effet, l'extraction robuste de telles observations est considérée comme très difficile.

Citons aussi les méthodes de détection d'objets particuliers : visages, piétons, voitures . . . Ces méthodes d'extraction sont basées sur un apprentissage.

### **Association de primitives :**

Afin d'exploiter plusieurs images, acquises, par exemple, par une caméra déplacée par un robot, il est nécessaire d'associer nos primitives extraites dans ces différentes images, ou d'associer ces primitives à du contenu dans une base de données.

Une des manière d'associer plusieurs primitives entre elles est l'approche "apparence" (appearance based feature matching). Il s'agit d'opposer ou d'associer les primitives entre elles par un critère basé sur l'aspect qu'elles ont dans les images. Cette approche est à opposer aux approches géométriques, où l'on considère uniquement les positions successives de la caméra comme critère pour associer les primitives. L'approche basé sur l'apparence est une base simple et efficace pour une solution d'association, de plus elle présente un potentiel temps-réel reconnu.

Construire une méthode d'association de descripteurs basé sur l'apparence est typiquement un problème de classification. Or il n'existe que deux types de méthodes pour construire un classificateur : les méthodes supervisées et les méthodes non-supervisées. Ces deux approches sont proposées en vision par ordinateur. La plus connue des méthodes supervisée est nommée "bag-of-word" (sac de mots). La plupart des autres méthodes non-supervisées sont identifiées par le terme "feature matching".

### *Association de points d'intérêt*

Concernant les point d'intérêt, il existe plusieurs comparateurs basés sur l'apparence, les plus courants sont définis dans le tableau 2.1.

Nom	Expression
Sum of Absolute Difference :	$SAD = \Sigma  I - J $
Normalized Sum of Absolute Difference :	$NSAD = \Sigma  (I - \bar{I}) - (J - \bar{J}) $
Cross-Correlation :	$CC = \Sigma (I \cdot J)$
Zero-Mean Cross-Correlation :	$ZCC = \Sigma (I - \bar{I})(J - \bar{J})$
Zero-Mean Normalized Cross-Correlation :	$ZNCC = \frac{1}{N} \sum \frac{(I - \bar{I})(J - \bar{J})}{\sigma_I \sigma_J}$

TABLE 2.1 – Méthodes classiques d'association de points,  $I$  et  $J$  sont les intensités des imagerie à comparer (les imagerie peuvent être lissées au préalable),  $\bar{I}$  et  $\bar{J}$  sont les moyennes de  $I$  et  $J$ .  $N$  est le nombre de pixels par imagerie. Enfin  $\sigma_I$  et  $\sigma_J$  représentent les écarts-type de  $I$  et  $J$

D'autres méthodes basées sur l'apparence sont considérées comme plus rapides et/ou plus robustes que celles présentées ci-dessus. Elles sont basées sur le principe du descripteur. Un descripteur est une signature (de dimension plus petite que l'imagerie à comparer). Le critère de comparaison  $C$  est alors exprimé en fonction des descripteurs (et non plus de l'imagerie). La conception d'un descripteur de qualité est décrite de manière claire dans [43]. Le principe décrit est le suivant : un bon descripteur est un descripteur qui soit à la fois invariant aux transformations dans l'image, mais qui permette de distinguer les différentes primitives dans les images.

Partons d'un exemple simple de descripteur : la moyenne :

$$D = \frac{1}{N} \sum_W I \quad (2.5)$$

avec  $W$  une fenêtre qui représente l'imagerie.

Appliquons ensuite une série de transformations à l'image, et observons la qualité de l'invariance de ce descripteur. On observe que la moyenne est robuste à la rotation (la moyenne est robuste à toute permutation), la moyenne est peu robuste à la variation d'illumination, à un changement d'échelle, ou encore à une distorsion.

On peut construire des descripteurs plus complexes comme des histogrammes d'intensité.

$$D[k] = \sum_W (I(u, v) == k) \quad (2.6)$$

dans lequel l'opérateur  $\sum$  agit comme un compteur.

Le critère de comparaison est alors une distance entre les descripteurs (distance Euclidienne, SAD, CC ...). On caractérise un critère d'association de points comme étant le triplet : 1- extracteur, 2- descripteur et 3- fonction de distance, qui conserve le bon appariement de point tout en rejetant les mauvaises paires.

En considérant le cas simple où les points sont extraits sur une surface plane, il est clair que tout changement de point de vue de la caméra provoque une homographie sur l'imagerie autour du

point (plus un changement de luminosité). On cherche donc un descripteur robuste à la transformation affine et au changement de luminosité. Pour être direct, ce descripteur n'existe pas. Dans la littérature, on trouve donc plusieurs approches proposant des solutions d'approximation pour ce problème, les plus courantes sont KLT (méthode itérative), SIFT et SURF (descripteurs efficaces basés sur des principes théoriques forts), et une méthode plus récente : BRIEF (dont la fonction de distance est la distance de Hamming). Les méthodes sont présentées dans le tableau 2.2. Il existe d'autres méthodes mais nous nous limiterons à celles-ci.

Méthode	Description
KLT	Méthode itérative convergente vers un point dans la seconde image.
SIFT	Descripteur peu variant au changement d'échelle (Scale Invariant).
SURF	Basé sur SIFT, plus rapide car l'extraction exploite les images intégrales.
BRIEF	Proche de CENSUS. Descripteur binaire. comparaison par distance de Hamming.

TABLE 2.2 – Principales méthodes d'association d'images. KLT [44, 45], SIFT [46, 47], SURF [48, 49] et BRIEF [50].

Des approches efficaces d'association de points hybrides (critère d'apparence et géométrique) existent.

**Association de points d'intérêt par critère hybride** Agrawal dans [51] propose une méthode d'association de points d'intérêt fonctionnant en temps-réel basé à la fois sur un critère d'apparence et géométrique. Il propose de générer des hypothèses de mouvements de caméra, puis d'associer les points avec une méthode d'apparence, et, pour chaque hypothèse, d'appliquer un algorithme de RANSAC pour éliminer les mauvais appariements. Après cette étape, les associations considérées comme bonnes sont celles de l'hypothèse avec le plus d'appariements corrects (cette hypothèse de mouvement est alors utilisée pour estimer la trajectoire de la caméra).

Dans [52], Lemaire propose une solution d'association de points d'intérêt basée sur la répartition dans l'image de certains groupements de ces points. Après une association individuelle des points d'intérêt réalisée par une méthode d'apparence (ZNCC), une association groupe-groupe est effectuée.

Dans [53], Davison propose une solution pour l'appariement de points utilisant à la fois un critère d'apparence et un critère géométrique. Dans son cas, l'estimation de la position de la caméra, en temps-réel, est utilisée pour projeter les points de la carte dans le plan image. La recherche de correspondance ne se fait alors que dans la zone de projection. Cette méthode, appelée "recherche active" (active search) a permis la première implantation de localisation et cartographie simultanée d'une caméra libre tenue à la main.

Klein dans [1] propose une autre approche, qui consiste à décomposer l'association de points en deux étapes (d'abord basée apparence, puis géométrique). Il s'agit de réaliser une première association non robuste, basée uniquement sur l'apparence, puis, sans détecter les associations aberrantes, utiliser un algorithme robuste aux mauvais appariements (dans son cas un Bundle

Adjustment robuste). Cette seconde étape utilise simplement une fonction de coût robuste (il propose Tukey, Cauchy et Huber comme exemples). Son approche est utilisée dans des algorithmes de SLAM temps-réel appelé PTAM (Parallel Tracking And Mapping) et se révèle suffisamment efficace pour tourner en temps-réel sur un smartphone [54].

### Association de segments

Les segments sont des éléments courants dans notre environnements optiques. Si l'on veut les utiliser depuis plusieurs points de vue, une difficulté est de les associer. Encore une fois on peut citer les travaux réalisés en SLAM visuels qui tentent d'utiliser les lignes dans un contexte de reconstruction (détection de segments en titre 2.2.1).

**Association d'autres primitives** Les primitives plus complexes telles que les plans et les courbes sont difficiles à apparier. Bien que des travaux existent pour la reconstruction de scène 3D utilisant des plans [39, 55], les autres primitives restent rares.

La détection de visages est un cas particulier. En effet, de nombreux travaux ont consisté à détecter et associer des visages dans un flux vidéo. Ces méthodes ont beaucoup progressé ces dernières années.

### Reconnaissance de scène :

La vision par ordinateur possède des techniques pour la reconstruction 3D de l'environnement, mais comprendre le monde "sémantiquement" reste une tâche à réaliser. Les applications inhérentes sont nombreuses (détection de comportements dangereux, suivi d'une personne ...). Cette analyse de scène requiert un algorithme d'inférence, et des données sur lesquelles raisonner (inférer).

Mais où en est la robotique ?

### 2.2.2 Vision en robotique

Les applications en robotiques se heurtent souvent à la difficulté du temps-réel. Les informations doivent être synthétisées avant qu'elles ne soient utilisées pour un traitement ultérieur (navigation ...). L'analyse d'image étant une opération longue pour l'extraction et l'association de primitives (excepté pour les points d'intérêt), et sachant que ces opérations sont essentielles à tout algorithme de plus haut niveau, la robotique consiste aujourd'hui à exploiter un capteur vidéo pour certaines opérations, comme la cartographie.

Ainsi la séquence d'opération la plus utilisée en robotique consiste en quatre tâches :

- L'extraction de primitives.
- L'association éventuelle dans une paire d'images issues d'un banc stéréo .
- Le suivi temporel des primitives.
- L'estimation de la scène (mapping) et le suivi du mobile (localisation).

Avec ce schéma, et les contraintes temps-réel, une ou plusieurs caméras peuvent être utilisées pour détecter des obstacles, identifier des visages, et cartographier une scène. En particulier le SLAM visuel, introduit il y a quelques années, est une application qui tend à s'étendre.

Abordons le sujet du SLAM.

## 2.3 SLAM : localisation et cartographie simultanées

### 2.3.1 Présentation générale

Le SLAM est l'opération de localisation d'un mobile, et de cartographie de l'environnement de ce mobile. L'intérêt est que la cartographie de la région autour du mobile sert de support à la localisation.

Le SLAM permet de se localiser dans des lieux difficiles (dans les bâtiments ...) selon les capteurs disponibles. En effet, les techniques de SLAM s'appliquent aussi bien avec des lasers 2D et 3D qu'avec des caméras (ces deux types de capteurs permettent d'observer l'environnement autour de notre mobile).

### 2.3.2 Pose SLAM

Les méthodes de Pose-SLAM consistent à estimer une trajectoire du robot, en ayant comme observations des mesures de décalage (des contraintes  $Rt$ ) entre les différentes positions. Un capteur laser (2D ou 3D) est capable de mesurer ces contraintes  $Rt$ , permettant de lier non seulement deux positions successives, mais aussi des positions distantes dans le temps (ce que l'on appelle un processus une fermeture de boucle).

**Définition 2.1.** On appelle "contrainte  $Rt$ " une mesure de déplacement de position et d'orientation, en 2D ou en 3D (nommé aussi *changement de repère*). Le nom  $Rt$  est issu de la notation matricielle ou  $R$  est la matrice de rotation (2D ou 3D) et  $t$  est un vecteur de translation en concaténant ces deux éléments en ligne, nous obtenons la matrice  $Rt = [R, t]$ . Cette matrice, appliquée à tout vecteur de l'espace  $v$ , procède au changement de repère approprié.

**Définition 2.2.** On appelle "fermetures de boucle" un processus de détection d'une mesure reliant l'état courant à un état éloigné dans le temps. Par exemple un laser peut réaliser une fermeture de boucle en réalisant un map-matching entre le "scan" de la dernière pose, et un "scan" plus ancien. Si il y a correspondance, on obtient une contrainte  $Rt$  reliant le pose la plus récente à une ancienne pose.

Or le résultat de la trajectoire est trivial lorsque les mesures sont uniquement des contraintes  $Rt$  entre les positions successives (type odométrie). Le résultat ne l'est plus lorsque ces contraintes  $Rt$  existent entre des positions non-successives.

Il est possible de générer des mesures de type  $Rt$  à l'aide d'un capteur odométrique, mais aussi avec un laser 2D ou 3D via une opération de "map-matching" (voir figure 2.8). L'opération de "map-matching" la plus connue est l'*ICP* (Iterative Closest Point), mais il en existe d'autres. La librairie PCL (Point Cloud Library) est une librairie logicielle open-source maintenue par Willow-Garage qui supporte de nombreuses méthodes de map-matching 2D et 3D.

### 2.3.3 Landmark-SLAM et Filtrage

Dans certains cas, plutôt que d'estimer la trajectoire complète, il est préférable de n'estimer que la dernière position du mobile. On parle de filtrage.

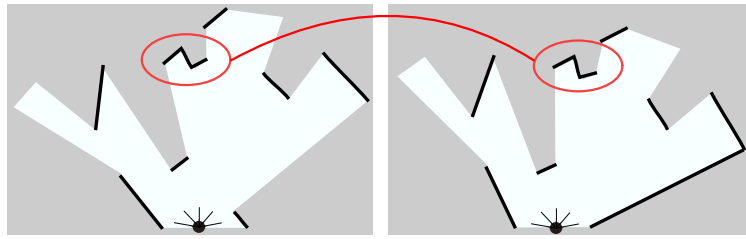


FIGURE 2.8 – Un capteur laser 2D est l'un des capteurs les plus simples à utiliser dans un contexte de SLAM, Les mesures sont des contraintes  $R_t$ , elles sont issues de l'association de deux "scans", et d'une opération d'estimation de la transformation affine (appelée " $R_t$ ") séparant ces deux scans.

**EKF-SLAM** Les premières solutions réalisant du SLAM se basent sur un filtre et une représentation par un vecteur d'état et une matrice de covariance [56, 57]. Le filtre de Kalman étendu (EKF) est la solution qui est retenue pour obtenir une solution du problème linéarisé. Dans ce cas, la phase de prédiction est en complexité linéaire avec la taille du vecteur d'état, et la phase de correction est l'opération la plus gourmande (la complexité étant quadratique avec la taille de la carte).

**EIF-SLAM** La complexité d'un filtre EKF limite cette approche aux environnements clos. Ce problème de complexité peut être grandement réduit par l'utilisation d'un filtre d'information (le résultat est exprimé par un vecteur d'information et une matrice d'information) [58, 59, 60]. Cette seconde solution possède l'avantage de rester peu coûteuse grâce au caractère creux de la matrice d'information<sup>14</sup>, ce qui permet, en utilisant des opérations sur les matrices creuses, d'opérer dans des environnements plus larges.

**Remarque 2.3.** Revenons sur l'approche EIF-SLAM : il est à noter que les blocs non zéros d'une matrice d'information représentent les noeuds dans un réseau Bayésien. Les solveurs de systèmes linéaires creux construisent implicitement la représentation par réseau Bayésien à partir de la matrice d'information dans le cas du filtre d'information [61]. En fait, les travaux les plus récents exploitent les modèles graphiques afin d'opérer directement sur cette structure de réseau Bayésien, sans construire le système sous sa forme matricielle, même creuse.

**EKF avec sous-cartes** Dans le but de limiter la complexité algorithmique, et l'inconsistance du SLAM basé sur un estimateur de type *filtrage*, certains travaux se sont penchés sur la décomposition de l'espace d'état en sous-cartes. L'approche *Divide and Conquer* [62] permet de fusionner les différentes sous-cartes de manière optimale en temps linéaire avec l'algorithme nommé "*map joining*".

**Filtrage particulaire** L'approche FAST-SLAM [63] utilise le fait que l'estimation de la carte est conditionnellement indépendante de la position du mobile. L'algorithme utilise alors une dis-

14. La matrice d'information, en SLAM, est creuse, ce qui n'est pas le cas de la matrice de covariance (son inverse)

tribution de la dernière pose sous forme d'un ensemble de particules (distribution discrète), chaque particule faisant tourner  $N$  filtres de Kalman (avec  $N$  le nombre d'amers dans la carte). La mise à jour du filtre est alors de complexité  $\mathcal{O}(M \log(N))$ , avec  $M$  le nombre de particules.

**Cas particulier de l'initialisation des éléments de la carte** Lorsque la fonction d'observation d'une primitive est inversible et que cette inverse est *suffisamment* linéaire pour conserver une densité Gaussienne, l'initialisation d'un amer visuel peut être instantané (par exemple un scan laser). Lorsque ce n'est pas le cas (caméra monoculaire par exemple), la distribution de l'amer visuel doit se faire avec précaution. En SLAM visuel monoculaire, les méthodes d'initialisation furent initialement basées sur une distribution multimodale Gaussienne [65], puis sur une paramétrisation non-Euclidienne de l'amer observé [66] : la paramétrisation IDP<sup>15</sup> est constituée d'une ancre, de deux angles donnant une direction de l'amer, et de l'inverse de la distance sur cet axe. Plus récemment, la représentations par coordonnées homogènes ancrées fut étudié et généralisée aux lignes [41].

#### *Filtrages : Problème de consistance*<sup>16</sup>

D'un point de vue de l'optimisation de paramètres, nous avons vu que les premières solutions furent basées sur le principe du filtrage stochastique [67, 68, 69, 70]. Cependant il est démontré qu'un filtre stochastique appliqué au problème du SLAM est source d'inconsistance dans le cas de non-linéarités (navigation aux odomètres sur quelques secondes, SLAM visuel 3D ...) [71, 72].

Ce phénomène d'inconsistance est provoqué par le point de linéarisation qui, s'il est suffisamment éloigné de la vérité terrain, provoque un biais de la fonction objectif [71, 72], ce qui provoque une convergence biaisée. On en déduit que l'approche filtrage ne permet pas, pour le problème de SLAM, de garantir l'intégrité du résultat, ce qui pose, en plus du problème de complexité, un problème majeur pour l'utilisation du SLAM par une approche "filtrage" dans des applications industrielles.

### 2.3.4 SLAM par lissage incrémental (Pose-SLAM et Landmark-SLAM)

#### *Modèles graphiques*

Une manière plus naturelle de représenter le problème du SLAM est d'utiliser un graphe. Les modèles graphiques (graphical models) proposent une manière intuitive de représenter et de visualiser les dépendances entre les variables (état du mobile et des éléments de la carte), mais aussi de concevoir des algorithmes d'inférence efficaces basés sur une représentation à la fois simple, et qui tirent parti des travaux existants dans de domaine de l'inférence Bayésienne [73]. Le problème du SLAM possède en effet, intrinsèquement, une structure de graphe, dans laquelle nous cherchons une configuration pour les variables (les noeuds) qui minimise un critère étant donné un ensemble de contraintes (les mesures, représentées par des arrêtes).

L'approche basée sur une représentation par graphe a été développée à travers l'utilisation de réseaux Bayésiens (Bayes nets) [74, 75], de champs de Markov (Markov Random Fields) [76, 77,

15. IDP : Inverse Depth Parametrization

16. Ici "Consistance" est un anglicisme de "Consistency" :cohérence



59], ou encore de graphes de facteur (factor graphs) [59, 78]. Alors que le réseau Bayésien semble proposer la représentation la plus naturelle, le graphe de facteur est lui, lié à la mise en place du processus d'optimisation. C'est donc cette représentation qui servira de base pour la conception de nos algorithmes dans cette thèse.

### *Optimisation non-linéaire et SAM*

Indépendamment de la représentation du système (vecteur d'état et matrice de covariance, vecteur et matrice d'information, graphe de facteur, réseaux Bayésiens, ou réseaux de Markov), la solution du problème de lissage (ne marginalisant aucune variable), appelé SAM (Smoothing And Mapping) reste consistante grâce à une approche d'optimisation non-linéaire. Les solutions proposées pour l'optimisation se basent sur les méthodes directes [59, 79, 80] et itératives [81, 82]. Les solutions basées uniquement sur le lissage ne sont donc pas adaptées aux systèmes robotiques (les systèmes robotiques requérant un estimateur en temps constant). En effet, opérer une optimisation globale à chaque nouvelle mesure est, au mieux, de complexité linéaire avec le nombre de poses (dépendant de la structure creuse du réseau Bayésien).

### *Lissage incrémental : iSAM1 et iSAM2*

Tant que nous n'obtiendrons pas une méthode d'optimisation non-linéaire capable de prendre en compte de nouvelles contraintes (ou mesures) en temps constant, le problème du SLAM (complexité et inconsistance) persistera.

Nous avons vu que la solution consistant à marginaliser certaines variables pour diminuer la taille du vecteur d'état (et ainsi diminuer la complexité de mise à jour de la solution) n'est pas utilisable pour des raisons de complexité (quadratique avec la taille de la carte qui ne fait qu'augmenter) et d'inconsistance du résultat.

Les méthodes de lissage (SAM) proposent à la fois une solution qui se rapproche d'une complexité linéaire avec la taille du vecteur d'état, et ne souffrent pas du problème de consistance. On appelle SAM incrémental (ou iSAM) l'extension des méthodes de lissage, qui prennent en compte les nouvelles contraintes de manière incrémentale. L'objectif est de fournir un résultat de manière incrémentale en temps constant.

Les approches SAM nous montrent que la qualité de la configuration initiale (le premier point de linéarisation ou "solution initiale") est essentiel dans le processus de convergence vers l'optimum global. Le SAM incrémental permet d'obtenir une configuration initiale pour chaque nouvelle itération, qui soit la meilleure estimée (cette configuration est l'optimum d'un sous-problème de lissage).

On comprend que la complexité d'un lisseur incrémental est bornée (supérieurement) par la complexité de la solution de l'approche batch (la seule solution garantissant la consistance de l'estimateur), tout en conservant un résultat optimal.

La première approche pour iSAM [78] se base sur une factorisation  $QR$  incrémentale. Les auteurs proposent, dans leur méthode (que nous nommerons  $iSAM_1$ ) de mettre à jour incrémentalement le facteur  $R$  de la factorisation  $QR$ , et de re-linéariser régulièrement ce système (ce qui revient à appliquer une optimisation "batch"). De plus, à chaque re-linéarisation, un nouveau ré-ordonnement est calculé, garantissant la conservation de la sparsité du facteur  $Q$  et ainsi

augmenter l'efficacité en temps de l'opération de back-substitution. La mise à jour de la trajectoire est alors réalisée en temps constant pour les phases d'exploration et de fermeture de boucle, et équivalent à l'optimisation *batch* à chaque étape de re-linéarisation.

L'extension de cette approche, décrite dans [83, 84], utilise un réseau Bayésien particulier (un *Bayes tree*) pour représenter le système linéaire et opérer les inférences Bayésiennes directement dans ce modèle graphique. La décomposition *QR* incrémentale décrite précédemment est "traduite" en opérations d'édition dans le Bayes tree. La méthode est nommée *iSAM2*, et propose une extension importante : un ré-ordonnancement et une re-linéarisation fluide (fluid relinearisation). La notion de *fluidité* ici consiste à appliquer la propagation des opérations de ré-ordonnancement et de re-linéarisation sur une sous-partie du Bayes tree (seulement la partie touchée par l'opération), ce qui élimine toute nécessité d'opération d'*optimisation batch*. Comparée aux méthodes précédentes, *iSAM2* propose une solution générique (s'appliquant à tout problème de SLAM), dont les performances sont proches du temps constant. Le principal inconvénient d'*iSAM2* est donc la complexité algorithmique du maintien de la structure de Bayes tree, qui crée une surcharge conséquente au processus de résolution (maintenir un Bayes tree nécessite de maintenir toutes les cliques du graphe de facteur).

## 2.4 Contributions de notre recherche

Cet état de l'art nous permet de nous positionner dans notre recherche. Nous avons vu qu'il existait différents moyens d'utiliser chacun des capteurs qui nous intéressent dans notre étude (GPS et caméra). Nous avons aussi vu qu'il existe différents moyens d'hybrider plusieurs capteurs. Et enfin nous avons vu que le domaine du SLAM est une approche cohérente pour traiter notre sujet de recherche.

Or donc, nous allons présenter une contribution dans le domaine de la navigation par GPS en mode différentiel local. En effet, nous allons proposer l'utilisation de mesures de triple-différences de phase GPS, pour réaliser une navigation robotique[22]. Ce qui constitue notre première contribution.

Nous allons aussi proposer une contribution dans le domaine du GPS-RTK mono-fréquence. En effet, nous avons développé une nouvelle méthode de lever d'ambiguïté de phase avec des récepteurs mono-fréquence. Nos résultats ont été publiés en 2011[85].

Enfin, notre dernière contribution consistera dans le développement d'un logiciel de SLAM générique. Ce logiciel est distribué sous licence Open-source. Nous avons implanté plusieurs stratégies de SLAM par filtrage dans ce logiciel. Et, comme cas d'étude d'hybridation multi-capteur, nous avons présenté une hybridation Vision/centrale inertielle tournant à fréquence image. Ces résultats ont également fait l'objet de publications[87, 88].



## Chapitre 3

# Estimation par triple-différence des mesures de phase en navigation DGPS-RTK

Dans ce chapitre, nous nous intéresserons au cas de la localisation par DGPS-RTK, en particulier aux performances du RTK mono-fréquence en mode différentiel local. Nous nous sommes attachés au problème de la localisation dite  $L_1$ -RTK pour sa mise en place relativement simple, et son coût réduit. Nous avons participé ainsi à la réalisation d'une solution de navigation centimétrique montée sur une tondeuse à gazon dans le cadre d'un projet Européen[17, 18]. Par la suite, nous avons étendu nos travaux, et abouti à la conception d'une approche utilisant les mesures de triple-différence de phase. Les travaux présentés dans cette section ont été l'objet d'une présentation lors de la conférence IEEE-SSP<sup>1</sup> durant l'été 2011[22].

### 3.1 Modèles dynamiques du mobile

#### 3.1.1 Modèle odométrique

Nous considérons d'abord un robot mobile à trois roues évoluant en milieu sub-urbain ouvert. Ce robot est équipé d'encodeurs sur ses deux roues arrières motrices. En modélisant notre mobile par un ensemble de variables d'état (typiquement la position et l'orientation), les mesures issues des encodeurs sur le moteur nous fournissent un modèle odométrique qui nous renseigne sur la dynamique de ces variables d'état. Soit  $R_t$  l'état du robot (position et orientation) à l'instant  $t$ , soit  $u_t$  les mesures des encodeurs à l'instant  $t$ , on choisira de modéliser par un processus markovien l'évolution des variables d'état  $R_t$  grâce au modèle suivant :

$$R_t = f(R_{t-1}, u_t, w_t)$$

où  $w_t$  un bruit blanc permettant de prendre en compte les erreurs de mesures odométriques ainsi que les erreurs d'intégration induites par glissement des roues du robot. Ce processus est ici

---

1. IEEE-SSP : International Conference on Statistics and Signal Processing, June 2011, Nice, France

modélisé comme un bruit blanc gaussien. La fonction  $f$  est obtenue à partir de considérations géométriques sur le robot.

### 3.1.2 Mobile à position, vitesse ou accélération brownienne

En absence de mesure odométrique, on peut considérer que la position, la vitesse ou l'accélération suit un mouvement brownien. Soit

$$X_t^U = \begin{bmatrix} x_t^U \\ y_t^U \\ z_t^U \end{bmatrix}$$

la position de l'utilisateur en coordonnées géo-centrées (ECEF). Pour ces trois situations, si  $X_t$  désigne l'état du système, on a :

– **Mouvement brownien**

$$X_t^U = X_{t-1}^U + w_t$$

– **Vitesse brownienne**

$$\begin{aligned} X_t^U &= X_{t-1}^U + V_t^U \cdot \Delta t \\ V_t^U &= V_{t-1}^U + w_t \end{aligned}$$

– **Accélération brownienne**

$$\begin{aligned} X_t^U &= X_{t-1}^U + V_t^U \cdot \Delta t \\ V_t^U &= V_{t-1}^U + \gamma_{t-1}^U \cdot \Delta t \\ \gamma_t^U &= \gamma_{t-1}^U + w_t \end{aligned}$$

Notons qu'on peut étendre ce type de modèle au jerk brownien<sup>2</sup>, sans que celui-ci ne soit limitatif.

**Remarque.** Le problème principal en navigation stochastique est alors - à ce niveau - : quel est le meilleur modèle. La réponse est très simple : le meilleur modèle est le modèle le plus représentatif de la statistique du phénomène que l'on souhaite modéliser. La solution pour choisir un modèle parmi N est, elle, beaucoup plus compliquée. Une solution à laquelle nous avons souvent recourus, lorsque différents choix de modèle rivalisaient, est la *technique de l'histogramme*. Sur un pas de temps représentatif, et à l'aide d'une réalité terrain, on trace les histogrammes des variables d'intérêt, et on observe quels modèles sont les plus représentatifs. Des techniques plus poussées existent, il s'agit du domaine de l'identification de modèles<sup>3</sup>. Il faut aussi être conscient qu'un histogramme seul ne décrit pas à la blancheur des bruits additionnels (seulement leur moyenne et leur distribution). Pour savoir si une variable suit un modèle de bruit filtré (mouvement Brownien à moyenne nulle) ou un bruit blanc, il faut tracer la transformée de fourrier de la fonction d'auto-corrélation de cette variable sur un temps représentatif. Plus le graphe se rapproche d'un Dirak,

2. Le jerk est la dérivée temporelle de l'accélération.

3. L'identification de modèle est un domaine des statistique qui consiste à construire un test sur le modèle.

plus le bruit est blanc, si la fonction est un sinus cardinal, il s'agit d'un bruit filtré, et donc il faut modéliser la variable (que l'on pensait être un bruit blanc) en tant que paramètre d'état<sup>4</sup>.

**Remarque.** La seconde remarque concerne la complexité de l'observateur que l'on va créer par la suite. En effet, plus le modèle est complexe (plus le nombre de variables d'intérêt est grand), plus l'observateur sera lent. On doit donc parfois se limiter à une certaine granularité dans la modélisation du phénomène, et faire la part entre la blancheur des bruits additionnels et la complexité algorithmique de l'observateur<sup>5</sup>.

### 3.2 Modèle des mesures de phase en double différence DGPS

Comme nous l'avons vu dans le chapitre traitant de l'état de l'art, les mesures brutes de phase GPS, dites "*non-différenciées*", sont modélisées comme des variables aléatoires dépendantes de plusieurs paramètres. Elles s'écrivent

$$\varphi_t^{U,i} = \|X_t^U - X_t^i\| + B_t^i + b_t^{S,i} + b_t^U + a_t^{U,i} + v_t^{U,i}$$

où  $\|X_t^U - X_t^i\|$  désigne la distance euclidienne séparant l'antenne du satellite émetteur (de coordonnées  $X_t^i$  dans le repère ECEF) de celle du récepteur,  $B_t^i$  les biais atmosphériques sur les mesures de phases dus à la propagation,  $b_t^{S,i}$  et  $b_t^U$  les biais des horloges des satellites et du récepteur respectivement,  $a_t^{U,i}$  l'ambiguïté de phase et  $v_t^{U,i}$  un bruit additionnel, considéré comme blanc et gaussien, d'écart-type  $\sigma_\varphi$ , modélisant le bruit électronique du récepteur. A noter que les biais induits par la présence de multi-trajets ne sont pas représentés ici.

En mode différentiel, les mesures de simples différences sont alors définies comme les différences de mesures entre celles du mobile ( $\varphi_t^{U,i}$ ) et celles de la station de référence ( $\varphi_t^{R,i}$ ). Les récepteurs étant considérés comme "proches", cette opération permet d'éliminer les biais atmosphériques ainsi que les biais d'horloges satellites. On a alors

$$\Delta\varphi_t^i = \varphi_t^{U,i} - \varphi_t^{R,i} \tag{3.1}$$

$$= \|X_t^U - X_t^i\| - \|X_t^R - X_t^i\| + \Delta a_t^i + \Delta b_t + \Delta v_t^i \tag{3.2}$$

avec  $\Delta a_t^i = a_t^{U,i} - a_t^{R,i}$ ,  $\Delta b_t = b_t^U - b_t^R$ ,  $\Delta v_t^i = v_t^{U,i} - v_t^{R,i}$ .

Notons  $\Delta X_t = X_t^U - X_t^R$  l'écart de position entre utilisateur et référence. On a

$$\|X_t^U - X_t^i\| - \|X_t^R - X_t^i\| = \|X_t^R + \Delta X_t - X_t^i\| - \|X_t^R - X_t^i\|$$

L'écart  $\|\Delta X_t\|$  étant très petit par rapport  $\|X_t^R - X_t^i\|$ , il est légitime de considérer le développement limité de cette différence au voisinage de zéro. On obtient alors

$$\|X_t^U - X_t^i\| - \|X_t^R - X_t^i\| \simeq \frac{(X_t^R - X_t^i)^T}{\|X_t^R - X_t^i\|} \Delta X_t$$

---

4. Une autre solution pour blanchir une variable dynamique est de sous-échantillonner les observations. Pour cela, on peut renvoyer au théorème de Shannon.

5. Il faudra, nous y reviendrons dans les section suivantes, relativiser ce raisonnement. En effet, à travers les algorithmes d'optimisation modernes utilisés dans le contexte du SLAM, on montre qu'il est parfois plus efficace d'observer plus de variables, que de procéder aux marginalisations. Cela montre par l'exemple qu'il n'est pas toujours vrai que le nombre de variables est pénalisant pour la vitesse d'exécution d'un observateur.

où

$$\frac{(X_t^R - X_t^i)^T}{\|X_t^R - X_t^i\|}$$

est le cosinus directeur du rayon entre récepteur et satellite. Si on note  $\mathbf{X}_t$  le vecteur d'état du système dynamique robot, selon le modèle retenu, on a

$$\begin{aligned}\mathbf{X}_t &= \Delta X_t \\ \mathbf{X}_t &= \begin{bmatrix} \Delta X_t \\ V_t^U \end{bmatrix} \\ \mathbf{X}_t &= \begin{bmatrix} \Delta X_t \\ V_t^U \\ \gamma_t^U \end{bmatrix}\end{aligned}$$

La matrice d'observation en mode différentiel local est alors définie par :

$$H_t^{SD,i} = \begin{bmatrix} (X_t^R - X_t^i)^T \\ \|\|X_t^R - X_t^i\|, 0 \end{bmatrix}$$

On a alors finalement

$$\Delta\varphi_t^i = H_t^{SD,i} \mathbf{X}_t + \Delta a_t^i + \Delta b_t + \Delta v_t^i$$

On note que les bruits additifs des mesures brutes étant indépendants entre chaque récepteur implique que les bruits de mesure des simples différences le sont également :  $\Delta v_t^i$  sont donc des bruits blancs gaussiens avec

$$\mathbb{E} \left[ (\Delta v_t^i)^2 \right] = 2(\sigma_\varphi)^2$$

Les mesures de double-différence sont définies comme les différences des simples-différences de chacun des satellites avec un satellite commun appelé *pivot* (indexé  $p$ ). Cette opération permet d'éliminer les biais d'horloge récepteurs  $\Delta b_t$ . On alors, donc  $\forall i \neq p$

$$\begin{aligned}\nabla \Delta\varphi_t^{i,p} &= \Delta\varphi_t^i - \Delta\varphi_t^p \\ &= H_t^{DD,i,p} \Delta X_t + \nabla \Delta a_t^{i,p} + \nabla \Delta v_t^{i,p}\end{aligned}\tag{3.3}$$

où  $\nabla \Delta a_t^{i,p} = \Delta a_t^i - \Delta a_t^p$  sont les ambiguïtés de doubles différences,  $\nabla \Delta v_t^{i,p} = \Delta v_t^i - \Delta v_t^p$ , et  $H_t^{DD,i,p} = H_t^{SD,i} - H_t^{SD,p}$ .

On remarque que les bruits de mesures des double-différences sont intercorrélées. En effet, on a :

$$\begin{aligned}\mathbb{E} \left[ (\nabla \Delta v_t^{i,p})^2 \right] &= \mathbb{E} \left[ (\Delta v_t^i - \Delta v_t^p)^2 \right] \\ &= \mathbb{E} \left[ (\Delta v_t^i)^2 \right] + \mathbb{E} \left[ (\Delta v_t^p)^2 \right] \\ &= 4(\sigma_\varphi)^2\end{aligned}$$

et  $\forall i \neq j$

$$\begin{aligned}\mathbb{E} \left[ \nabla \Delta v_t^{i,p} \nabla \Delta v_t^{j,p} \right] &= \mathbb{E} \left[ (\Delta v_t^i - \Delta v_t^p) (\Delta v_t^j - \Delta v_t^p) \right] \\ &= \mathbb{E} \left[ (\Delta v_t^p)^2 \right] \\ &= 2 (\sigma_\varphi)^2\end{aligned}$$

Rappelons que dans ce modèle  $\nabla \Delta a_t^{i,p}$  est une variable aléatoire de type entier a priori uniformément distribuée sur un horizon infini.

En absence de sauts de phase, le modèle des variables d'ambiguïté  $\nabla \Delta a_t^{i,p}$  est le suivant :

$$\nabla \Delta a_t^{i,p} = \nabla \Delta a_{t-1}^{i,p}$$

### 3.3 Modèle des mesures de phase triple différence DGPS

#### 3.3.1 Rappel sur les méthodes de relaxation

Si on agrège le modèle dynamique représentant les déplacements du robot et le modèle d'observation ainsi construit (doubles-différences), il est nécessaire d'adjoindre à l'état les variables inconnues mais constantes (dans l'hypothèse où il n'y a pas de saut de phase dans les mesures) que sont les ambiguïtés de phase  $\nabla \Delta a_t^{i,p}$ . Soit  $\mathbf{X}_t$  l'état dynamique complet du robot contenant en particulier l'écart de position  $\Delta X_t$  défini plus haut, on obtient alors un système markovien du type

$$\begin{aligned}\mathbf{X}_t &= f(\mathbf{X}_{t-1}, u_t, w_t) \\ \nabla \Delta a_t^{i,p} &= \nabla \Delta a_{t-1}^{i,p}\end{aligned}$$

observé par

$$\nabla \Delta \varphi_t^{i,p} = H_t^{DD,i,p} \Delta X_t + \nabla \Delta a_t^{i,p} + \nabla \Delta v_t^{i,p}$$

Notre représentation d'état est donc hybride. En effet, les composantes de  $X_t$  sont continues alors que les paramètres  $\nabla \Delta a_t^{i,p}$  sont discrets (les valeurs sont nécessairement entières). Nous pourrions considérer un filtre multimodal pour les variables d'ambiguïtés, ce filtre multi-modal estimerait le maximum de vraisemblance statistique sur  $X$  et  $\nabla \Delta a$  conditionnellement aux mesures à disposition. Cependant cette approche est irréalisable en temps-réel embarqué. En effet, le nombre d'hypothèse d'ambiguïté est, par construction, très grand (plusieurs millions à la construction et plusieurs milliers après élagage par une heuristique quelconque). Le fait est qu'un filtre multimodal maintient chaque hypothèse  $k$  (estimant  $\nabla \Delta a_t^{i,p}$ ). Or, après avoir tenté cette approche, nos résultats ne nous ont pas permis de gérer un tel nombre de paramètres, et nous avons été très loin des performances temps réel. Une solution possible passe alors par la relaxation de la contrainte entière.

On appelle "*relaxation sur la contrainte entière de la mesure de phase*" le fait de relaxer (retirer) la contrainte entière sur la variable d'ambiguïté. On considère alors les ambiguïtés de



phases comme des variables aléatoires continues à densités gaussiennes. L'état  $\mathbf{X}_t^A$  contient alors les ambiguïtés  $\nabla\Delta a_t^{i,p}$ . On obtient donc le système Markovien augmenté continu suivant :

$$\begin{aligned}\mathbf{X}_t^A &= f(\mathbf{X}_{t-1}^A, u_t, w_t) \\ y_t &= h(t, \mathbf{X}_t^A) + v_t\end{aligned}$$

Le filtrage d'un tel système est parfaitement réalisable en temps-réel, la solution  $\hat{\mathbf{X}}_t^A$  du filtre est alors non optimale (il s'agit de la solution d'un système relaxé). Cependant elle permet d'obtenir une solution de positionnement en temps-réel. La méthode est nommée filtre à ambiguïtés flottantes.

On a donc, dans le cas d'un filtre à ambiguïtés flottantes, un vecteur d'état de taille variable défini par  $\mathbf{X}_t^A = [\mathbf{X}_t, a_t^{i_1,p}, \dots, a_t^{i_{N-1},p}]^T$ , où  $N$  désigne le nombre de satellites visibles et nos observations sont les mesures de phases de double-différences  $y_t = [\nabla\Delta\varphi_t^{i_1,p}, \nabla\Delta\varphi_t^{i_2,p} \dots]$ . rappelons que l'observabilité est assurée grâce à la variation dans le temps de  $h(t, \mathbf{X}_t^A)$ . Ceci a comme conséquence la nécessité d'attendre suffisamment longtemps, si possible en position statique, avant que la variation de  $h(t, \mathbf{X}_t^A)$  deviennent significative.

Le principal problème d'un filtre à ambiguïtés flottante est la variation de la taille du vecteur de paramètres. En effet, la gestion correcte de l'initialisation d'une valeur d'ambiguïté pose un problème à la fois de gestion de la mémoire, mais aussi en complexité de maintenance et de paramétrisation du filtre stochastique. Le modèle d'évolution des valeurs d'ambiguïtés étant considéré comme constant (sans bruit additionnel), il existe une solution élégante pour retirer les variables d'ambiguïté par différentiation des mesures de phase (cette opération de reparamétrisation<sup>6</sup> est appelée "*ramener les ambiguïtés dans l'espace nul*<sup>7</sup> "). Ainsi, le vecteur d'état ne contiendra plus ces variables ambiguïtés. Pour cela, nous allons différencier les mesures de phases de telle sorte que les nouvelles mesures ne dépendent pas des ambiguïtés de phase et construire un nouveau type de mesure par différentiation, les mesures de triples-différences.

### 3.3.2 Modèle de mesures

On définit la mesure de triple différence de phase comme la différence temporelle (entre deux instants immédiatement successifs) des mesures de phase. On montre que cette mesure ne dépend d'aucune valeur d'ambiguïté de phase<sup>8</sup>.

Rappelons que la mesure double différence a la forme suivante, pour le satellite  $i$

$$\nabla\Delta\varphi_t^{i,p} = H_t^{DD,i,p}\mathbf{X}_t + \nabla\Delta a_t^{i,p} + \nabla\Delta v_t^{i,p} \quad (3.4)$$

avec  $H_{dd,t}^{i,p}$  notre matrice d'observation,  $a^i$  la valeur d'ambiguïté de phase en double-différence,  $p_t$  la position du mobile et  $v_t^i$  notre bruit de mesure.

La mesure de triple différence est alors définie par

$$y_t^{i,p} = \nabla\Delta\varphi_t^{i,p} - \nabla\Delta\varphi_{t-1}^{i,p} = H_t^{DD,i,p}\mathbf{X}_t - H_{t-1}^{DD,i,p}\mathbf{X}_{t-1} + \tilde{v}_t^{i,p}$$

6. On appelle reparamétrisation un changement de variable de l'espace d'état. Par exemple, pour représenter un point en deux dimensions, on peut utiliser des coordonnées cartésiennes, polaire ou encore polaires inversées.

7. Ce type de reparamétrisation est couramment utilisé en théorie de la décision.

8. En effet, la valeur de l'ambiguïté de phase étant constante dans le temps, et la nouvelle mesure étant une différentiation temporelle, cette mesure ne dépend pas de la valeur d'ambiguïté.

Dans notre cas, les équations d'évolution dynamique de l'état s'expriment sous la forme suivante

$$\mathbf{X}_t = F_t \mathbf{X}_{t-1} + G_t w_t$$

où, selon le modèle utilisé,  $F_t$  et  $G_t$  sont les paramètres du modèle dynamique linéaire ou, dans le cas non linéaire, les jacobiens de la fonction dynamique évaluée au voisinage de l'estimateur courant. Dans tous les cas, les trois premières composantes de  $\mathbf{X}_t$  l'état à l'instant  $t$  désignent la position de l'antenne GPS utilisateur dans le référentiel de la station de référence exprimé en coordonnées locales (Est/Nord/Haut, Est/Ouest/Bas ...).  $w_t$  est le bruit de modèle dynamique considéré comme un bruit blanc gaussien.

L'expression complète de la mesure de triple-différence, en absence de saut de phase ( $\nabla \Delta a_t^{i,p} = \nabla \Delta a_{t-1}^{i,p}$ ), est alors

$$\begin{aligned} y_{t-1}^{i,p} &= H_t^{DD,i,p} \mathbf{X}_t + \nabla \Delta a_t^{i,p} + \nabla \Delta v_t^{i,p} - H_{t-1}^{DD,i,p} \mathbf{X}_{t-1} - \nabla \Delta a_{t-1}^{i,p} - \nabla \Delta v_{t-1}^{i,p} \\ &= H_t^{DD,i,p} (F_t \mathbf{X}_{t-1} + G_t w_{t-1}) - H_{t-1}^{DD,i,p} \mathbf{X}_{t-1} + \nabla \Delta v_t^{i,p} - \nabla \Delta v_{t-1}^{i,p} \\ &= \left( H_t^{DD,i,p} F_t - H_{t-1}^{DD,i,p} \right) \mathbf{X}_{t-1} + H_t^{DD,i,p} G_t w_{t-1} + \nabla \Delta v_t^{i,p} - \nabla \Delta v_{t-1}^{i,p} \end{aligned}$$

Ce modèle d'observation peut être réduit à

$$y_t^{i,p} = \tilde{H}_t^{i,p} \mathbf{X}_t + \tilde{v}_t^{i,p}$$

avec

$$\begin{aligned} \tilde{H}_t^{i,p} &= \left( H_{t+1}^{DD,i,p} F_t - H_t^{DD,i,p} \right) \\ \tilde{v}_t^{i,p} &= H_t^{DD,i,p} G_t w_{t-1} + \nabla \Delta v_t^{i,p} - \nabla \Delta v_{t-1}^{i,p} \end{aligned}$$

Il est alors clair qu'en réalisant une mesure de triple différence, nous introduisons une corrélation temporelle dans le terme de bruit  $\tilde{v}_t^{i,p}$  tel que

$$\mathbb{E} \left[ \tilde{v}_t^{i,p} \left( \tilde{v}_t^{i,p} \right)^T \right] = 8\sigma_\varphi^2 \mathbb{E} \left[ \tilde{v}_t^{i,p} \left( \tilde{v}_{t+\tau}^{i,p} \right)^T \right] = 4\sigma_\varphi^2, \forall \tau = \pm 1 \mathbb{E} \left[ \tilde{v}_t^{i,p} \left( \tilde{v}_{t+\tau}^{i,p} \right)^T \right] = 0, \forall |\tau| > 1$$

On note alors que  $\tilde{v}_t^{i,p}$  n'est pas un bruit blanc et qu'il est de plus corrélé au bruit dynamique  $w_t$ .

### 3.3.3 Modélisation de la corrélation du bruit de mesure

Comme décrit dans [20], la prise en compte d'un bruit coloré est faite optimalement par une augmentation de l'espace d'état. On a

$$\tilde{\mathbf{X}}_t = [\mathbf{X}_t, z_t, x_t^{0,1}, x_t^{1,1}, \dots, x_t^{0,N-1}, x_t^{1,N-1}]$$

avec  $N$  le nombre de mesures de triple-différence de phase. Procédons à un décalage dans notre notation (on remplace  $\tilde{v}_t^{i,p}$  par  $\tilde{v}_{t-1}^{i,p}$ ), on a

$$\begin{aligned} \mathbf{X}_t &= F_t \mathbf{X}_{t-1} + G_t z_{t-1} \\ z_t &= w_t \\ x_t^{0,i} &= \nabla \Delta v_t^{i,p}, \forall i = 1 \dots N-1 \\ x_t^{1,i} &= x_{t-1}^{0,i}, \forall i = 1 \dots N-1 \end{aligned}$$

Le modèle d'observation devient alors  $\forall i = 1 \dots N - 1$

$$y_t^{i,p} = \tilde{H}_t^{i,p} \mathbf{X}_t - x_t^{1,i} + \bar{v}_t^{i,p}$$

avec une nouvelle expression pour notre bruit additif  $\bar{v}_t^{i,p}$ , cette fois-ci, blanc, mais toujours corrélé à  $w_t$

$$\bar{v}_t^{i,p} = H_t^{DD,i,p} G_t w_t + \nabla \Delta v_t^{i,p}$$

Le nouveau système dynamique est alors

$$\tilde{\mathbf{X}}_t = F_t \tilde{\mathbf{X}}_{t-1} + \tilde{G}_t \tilde{w}_t$$

avec

$$\tilde{w}_t = [w_t, \nabla \Delta v_t^{1,p}, \dots, \nabla \Delta v_t^{N,p}]^T$$

Les nouvelles matrices sont définies par

$$\tilde{F}_t = \begin{bmatrix} F_t & G_t & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & L & \cdots & 0 \\ 0 & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & L \end{bmatrix}, L = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\tilde{G}_t = \begin{bmatrix} 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & B & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & B \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Et le nouveau modèle de mesure devient

$$y_t = \tilde{H}_t \tilde{\mathbf{X}}_t + \bar{v}_t$$

avec

$$\tilde{H}_t = \begin{bmatrix} L_t^{1,p} & 0 & M & 0 & \cdots & 0 \\ L_t^{2,p} & 0 & 0 & M & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ L_t^{N-1,p} & 0 & 0 & 0 & \cdots & M \end{bmatrix}$$

$$L_t^{i,p} = [H_t^{DD,i,p} F_t - H_{t-1}^{DD,i,p}]$$

$$M = [0 \quad -1]$$

et

$$y_t = \begin{bmatrix} y_t^{1,p} \\ \vdots \\ y_t^{N-1,p} \end{bmatrix}$$

$$\bar{v}_t = \begin{bmatrix} \bar{v}_t^{1,p} \\ \vdots \\ \bar{v}_t^{N-1,p} \end{bmatrix}$$

L'auto-corrélation du bruit dynamique  $\tilde{w}_t$  est

$$\mathbb{E} \left[ \tilde{w}_t (\tilde{w}_t)^T \right] = \begin{bmatrix} Q & 0 & \cdots & \cdots & 0 \\ 0 & 4(\sigma_\varphi)^2 & 2(\sigma_\varphi)^2 & \cdots & 2(\sigma_\varphi)^2 \\ \vdots & 2(\sigma_\varphi)^2 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 2(\sigma_\varphi)^2 \\ 0 & 2(\sigma_\varphi)^2 & \cdots & 2(\sigma_\varphi)^2 & 4(\sigma_\varphi)^2 \end{bmatrix}$$

où  $Q = \mathbb{E} \left[ w_t (w_t)^T \right]$ . Rappelons que la variable  $\sigma_\varphi$  représente l'écart-type du bruit de la mesure de phase non-différenciée.

L'auto-corrélation des bruits de mesure est défini par

$$\begin{aligned} \mathbb{E} \left[ \left( \bar{v}_t^{i,p} \right)^2 \right] &= \mathbb{E} \left[ H_t^{DD,i,p} G_t w_t \left( H_t^{DD,i,p} G_t w_t \right)^T \right] + \mathbb{E} \left[ \left( \nabla \Delta v_t^{i,p} \right)^2 \right] \\ &= H_t^{DD,i,p} G_t Q G_t^T \left( H_t^{DD,i,p} \right)^T + 4(\sigma_\varphi)^2 \end{aligned}$$

La corrélation croisée des bruits de mesures  $\forall i \neq j$  est

$$\begin{aligned} \mathbb{E} \left[ \bar{v}_t^{i,p} \bar{v}_t^{j,p} \right] &= \mathbb{E} \left[ H_t^{DD,i,p} G_t w_t \left( H_t^{DD,j,p} G_t w_t \right)^T \right] + \mathbb{E} \left[ \nabla \Delta v_t^{i,p} \nabla \Delta v_t^{j,p} \right] \\ &= H_t^{DD,i,p} G_t Q G_t^T \left( H_t^{DD,j,p} \right)^T + 2(\sigma_\varphi)^2 \end{aligned}$$

La corrélation croisée entre les bruits dynamique et les bruits de mesure est définie par

$$\mathbb{E} \left[ \tilde{w}_t \left( \bar{v}_t^{i,p} \right)^T \right] = \begin{bmatrix} Q G_t^T \left( H_t^{DD,i,p} \right)^T \\ 2(\sigma_\varphi)^2 \\ \vdots \\ 2(\sigma_\varphi)^2 \\ 4(\sigma_\varphi)^2 \\ 2(\sigma_\varphi)^2 \\ \vdots \\ 2(\sigma_\varphi)^2 \end{bmatrix}$$

où le terme  $4(\sigma_\varphi)^2$  est présent à la  $(i + 1^{\text{ème}})$  ligne.

Notons

$$\tilde{R}_t = \mathbb{E} \left[ \bar{v}_t \bar{v}_t^T \right]$$

$$\tilde{S}_t = \mathbb{E} \left[ \tilde{w}_t \left( \bar{v}_t \right)^T \right]$$

$$\tilde{Q} = \mathbb{E} \left[ \tilde{w}_t (\tilde{w}_t)^T \right]$$

Nous allons décrire les matrices  $\tilde{R}_t$  et  $\tilde{S}_t$ .

$$\tilde{S}_t = \begin{bmatrix} QG_t^T (H_t^{DD,1,p})^T & \dots & \dots & \dots & QG_t^T (H_t^{DD,N-1,p})^T \\ 4(\sigma_\varphi)^2 & 2(\sigma_\varphi)^2 & \dots & 2(\sigma_\varphi)^2 & 2(\sigma_\varphi)^2 \\ 2(\sigma_\varphi)^2 & 4(\sigma_\varphi)^2 & \dots & \dots & 2(\sigma_\varphi)^2 \\ \vdots & \vdots & \ddots & \dots & \vdots \\ 2(\sigma_\varphi)^2 & \vdots & \vdots & 4(\sigma_\varphi)^2 & 2(\sigma_\varphi)^2 \\ 2(\sigma_\varphi)^2 & 2(\sigma_\varphi)^2 & \dots & 2(\sigma_\varphi)^2 & 4(\sigma_\varphi)^2 \end{bmatrix}$$

**Notons**

$$\begin{aligned} \mathbb{E} \left[ \left( \bar{v}_t^{i,p} \right)^2 \right] &= D_t^{i,p} \\ \mathbb{E} \left[ \bar{v}_t^{i,p} \bar{v}_t^{j,p} \right] &= K_t^{i,j,p} \quad \forall i \neq j \end{aligned}$$

On peut en déduire l'expression de  $\tilde{R}_t$  :

$$\tilde{R}_t = \begin{bmatrix} D_t^{1,p} & K_t^{1,2,p} & \dots & K_t^{1,N,p} \\ K_t^{2,1,p} & \ddots & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & K_t^{N-1,N,p} \\ K_t^{N,1,p} & \dots & K_t^{N,N-1,p} & D_t^{N,p} \end{bmatrix}$$

### 3.4 Filtrage optimal

Nous avons donc décrit toutes les variables de notre système sous la forme générique suivante :

$$\begin{aligned} \tilde{\mathbf{X}}_t &= \tilde{F}_t \tilde{\mathbf{X}}_{t-1} + \tilde{G}_t \tilde{w}_t \\ y_t &= \tilde{H}_t \tilde{\mathbf{X}}_t + \bar{v}_t \end{aligned}$$

Nous pouvons appliquer un filtrage optimal, les opérations sont :

– Prédiction.

$$\begin{aligned} \widehat{\tilde{\mathbf{X}}}_{t|t-1} &= \tilde{F}_t \widehat{\tilde{\mathbf{X}}}_{t-1} \\ P_{t|t-1} &= \tilde{F}_t P_{t-1|t-1} \tilde{F}_t^T + \tilde{G}_t \tilde{Q} \tilde{G}_t^T \end{aligned}$$

– Correction.

$$\begin{aligned}
\hat{\mathbf{X}}_{t|t} &= \hat{\mathbf{X}}_{t|t-1} + K_t \left( \tilde{y}_t - \tilde{H}_t \hat{\mathbf{X}}_{t|t-1} \right) \\
\Sigma_t &= \left( \tilde{H}_t P_{t|t-1} \tilde{H}_t^T + \tilde{H}_t \tilde{G}_t \tilde{S}_t + \tilde{S}_t^T \tilde{G}_t^T \tilde{H}_t^T + \tilde{R}_t \right) \\
K_t &= \left( P_{t|t-1} \tilde{H}_t^T + \tilde{G}_t \tilde{S}_t \right) (\Sigma_t)^{-1} \\
P_{t|t} &= P_{t|t-1} - K_t \left( \tilde{H}_t P_{t|t-1} + \tilde{S}_t^T \tilde{G}_t^T \right)
\end{aligned}$$

### 3.5 Résultats

Nous avons intégré ce filtre stochastique dans un logiciel temps-réel embarqué. Ce système de localisation à été porté par un engin agricole pendant sa phase d'opération (un terrain agricole près de Paris, en France). Le modèle utilisé est du type vitesse brownienne.

La figure 3.1 nous présente la convergence initiale de notre filtre, en comparant avec (en rouge) un rayon de protection fixé à  $3\text{-}\sigma$ . Le temps de convergence (relativement long), est issu du fait que nous n'utilisons que des mesures de phase GPS mono-fréquence. La référence est fournie par un système RTK (en post-traitement). Nous remarquons que nous accédons à un positionnement sub-métrique en moins de 500 secondes, et un positionnement centimétrique en moins de 1000 secondes.

Après le départ de l'engin agricole (figure 3.2), notre filtre continue d'estimer convenablement la position du mobile jusqu'à la fin de l'opération agricole qui a duré 2500 secondes, soit environ 30 minutes. Les résultats sont considérés comme satisfaisant dans la mesure où la précision de localisation reste centimétrique tout au long de l'essai. Notre modèle de bruit est volontairement pris plus large que dans le meilleur des cas pour prendre en compte les effets de trajets multiples qui peuvent survenir lors de l'approche d'arbres.

Nos résultats sont compatibles avec les performances requises pour un système agricole d'auto-guidage.

### 3.6 Conclusions sur le filtre de triple différence

Nous avons conçu, puis intégré dans un navigateur agricole autonome un logiciel embarqué de navigation bas-coût centimétrique basé seulement sur des mesures GPS de phase mono-fréquence. Sans opération de levée d'ambiguïté, notre système de navigation obtient des performances très intéressantes, en particulier aux abords des champs, là où une opération de levée d'ambiguïté de phase, aussi-bien en mono-fréquence qu'en bi-fréquence GPS serait impossible.

Pour cela, nous avons étendu la méthode proposée dans [20], et plus récemment dans [21], en proposant un modèle optimal de coloration du bruit.

Cependant, pour augmenter les performances, nous allons nous intéresser à la levée d'ambiguïté de phase.

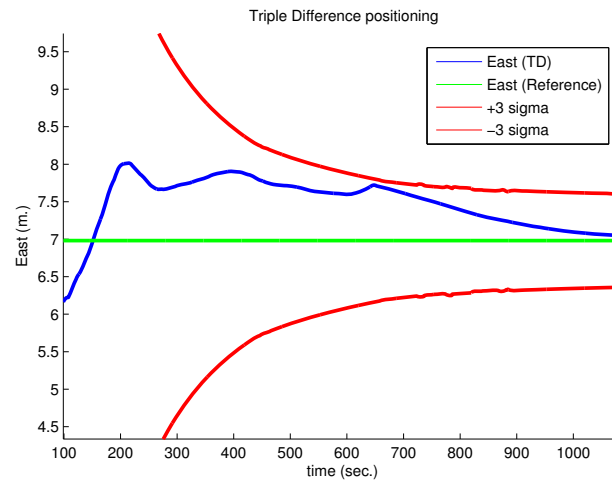


FIGURE 3.1 – La convergence initiale du filtre de triple différence. Le rayon de protection est fixé ici à  $3\sigma$ .

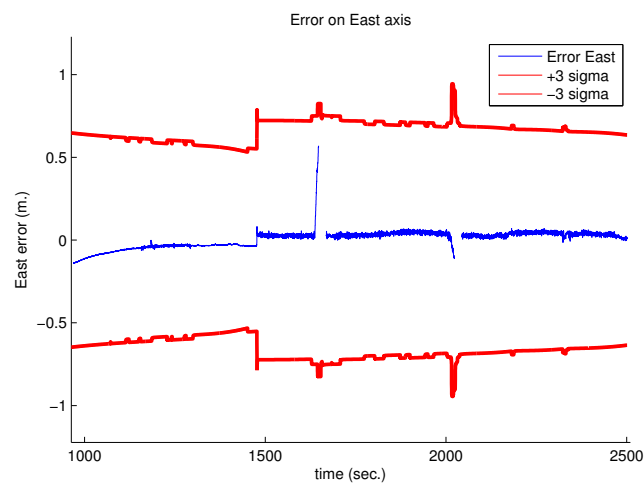


FIGURE 3.2 – Le suivi de trajectoire par notre filtre de triple-différence. Le rayon de protection est fixé à  $3\sigma$ .

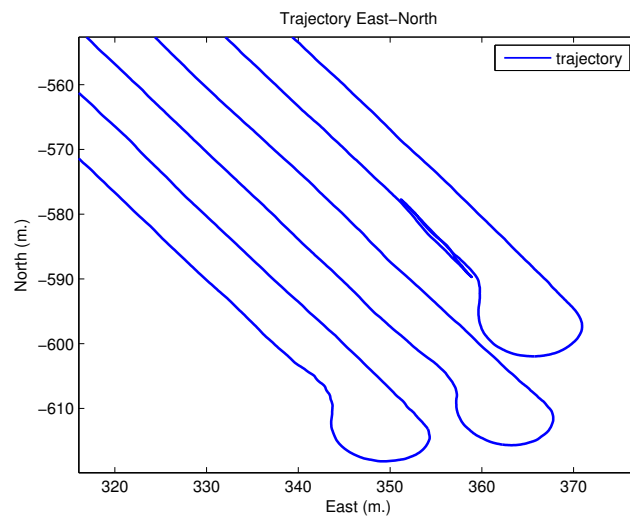


FIGURE 3.3 – Un morceau de trajectoire typique de notre engin agricole.





## Chapitre 4

# Estimation par lever d'ambiguïté de phase entière

Nous avons travaillé sur la modélisation des risques et proposé une solution innovante pour le processus dit de "*lever d'ambiguïté*"<sup>1</sup> de phase[85]. Les travaux présentés dans cette section ont été l'objet d'une présentation lors de la conférence ENC<sup>2</sup>, organisée par le RIN<sup>3</sup>, durant l'automne 2011[85].

### 4.1 Motivations

On ne compte pas les avantages d'un positionnement GPS RTK mono-fréquence. Des centaines de travaux, depuis plus de 20 ans, ont traité de l'opération de lever d'ambiguïté entière, la plupart étudiés initialement dans le cadre de récepteurs bi-fréquence, mais dont l'extension au récepteurs mono-fréquence est explicitement et expérimentalement étudiée.

Le mode de navigation RTK permet une précision de positionnement différentiel inégalée, de l'ordre, au pire, du centimètre, au mieux, du millimètre. Ce mode de positionnement possède théoriquement des applications innombrables notamment dans la robotique mobile.

Parmi les experts des plus reconnus, on peut citer *G. Lachapelle, P. Teunissen, P. De Jonge, C. Tibernius, D. Kim, D. Laurichesse*, et, en France *A. Lannes, C. Macabiau* et *M. Sahnoudi* parmi d'autres.

### 4.2 Méthode LAMBDA

La méthode de référence pour le lever d'ambiguïté entière est la méthode dite "*LAMBDA*"<sup>4</sup>[8]. Cette méthode permet, en procédant tout simplement à la décorrélation de l'espace des ambiguïtés de phase, de minimiser la dimension du problème, mais surtout la zone d'exploration telle que définie en prenant les extrema d'intervalles de solutions entières sur chaque espace d'ambiguïté.

- 
1. Ambiguity resolution
  2. ENC : European Navigation Conference, Novembre 2011, Londres, UK
  3. RIN : Royal Institute of Navigation
  4. LAMBDA : Least-Squares AMBiguity Decorrelation Adjustment.

Par exemple, si nous devons rechercher deux ambiguïtés entières, c'est à dire une recherche sur deux dimensions, on procède ainsi. Dans la première dimension, les extrema de recherche définissent un intervalle  $I_1 = [-10, 10]$  et, dans la seconde dimension, l'intervalle est  $I_2 = [-8, 8]$ . À première vue, il nous faudrait explorer  $N = 17 \times 21 = 357$  hypothèses entières.

Cependant, si on décompose l'exploration d'une hypothèse en deux phases : d'abord le choix de la première valeur (dans  $I_1$ ), puis le choix de la seconde valeur (dans  $I_2$ ), on s'aperçoit alors qu'il n'est pas nécessaire d'explorer tout  $I_2$  après le premier choix dans  $I_1$ . En effet, l'intervalle d'exploration dans la seconde dimension dépend du premier choix, et est borné supérieurement par  $I_2$ . Ce processus de réduction d'incertitude est nommé "exploration conditionnelle", et est basé sur la probabilité conditionnelle. Un arbre de décision est une représentation d'une exploration conditionnelle. La méthode LAMBDA ne souhaite pas réaliser une exploration conditionnelle, et propose une autre voie.

On remarque, en effet, que plus la valeur d'inter-corrélation entre les deux dimensions est faible (plus les choix sont indépendants), plus les hypothèses explorées dans le cas où l'on explore les intervalles extrêmes ( $I_1$  et  $I_2$ ), ne se font pas "*pour rien*". En effet, les tests sont indépendants, et donc les extrema correspondent exactement à la zone à explorer. Cependant, les cas dans lesquels les choix d'hypothèse d'ambiguïtés sont parfaitement indépendants sont extrêmement rares. Pire que cela, la probabilité que cela se produise est, exactement, nulle.

La méthode LAMBDA procède alors à une re-paramétrisation. Dans ce nouvel espace d'état, les inter-corrélations d'ambiguïtés sont faibles, et, ainsi, on minimise le nombre d'hypothèses testées inutilement. Par voie de conséquence, on accélère le processus de lever d'ambiguïté.

Bien que considérée par la communauté du GPS comme une ultime méthode, la méthode LAMBDA est reconnue pour souffrir de deux inconvénients majeurs :

Réduction de dimension : La transformation (appelée Z-transformation) proposée par la méthode LAMBDA occasionne souvent une réduction de dimension<sup>5</sup>. Cette réduction est utile, car elle permet de n'explorer qu'un sous-ensemble de solutions, les solutions complètes sont déduites des solutions partielles. Cependant, il nous arrivait de ne pas contenir, dans l'espace sous-dimensionné, le facteur de la solution entière réelle. Cet inconvénient concerne de la même manière toute la catégorie des méthodes de lever d'ambiguïté entière reposant sur la réduction de dimension. Les effets se font ressentir dans la levée d'ambiguïté en mono-fréquence et lorsque l'observabilité est faible. Il y a donc une élimination aléatoire de certaines solutions dans l'approche LAMBDA.

Explorations indépendantes : Dans la méthode LAMBDA, l'exploration par un test d'hypothèse se fait en ignorant les relations statistiques inter-ambiguïtés (les variances croisées). On peut supposer que les auteurs ont choisis cette solution (sans l'exposer explicitement), dans la mesure où la méthode LAMBDA procède d'abord à la décorrélation des variables d'ambiguïté. Ils ont alors pu légitimement approximer les inter-relations comme étant nulles. Ceci est une double-erreur : d'abord parce-que la décorrélation proposée est une approximation (certains ont d'ailleurs montré que cette approximation pouvait entraîner une augmentation de ces inter-relations), ensuite parce que le but de la méthode est de réduire l'espace de recherche et ignorer les inter-relations augmente cet espace de recherche<sup>6</sup>. Nous allons

5. En effet, si la décomposition n'était pas à valeur entière, il est prouvé que la dimension du problème est exactement '3' lorsque l'on traite des ambiguïtés en double différence.

6. Ignorer les inter-relations dans le cas Gaussien revient, dans le meilleur des cas, à chercher une solution

aborder ce point beaucoup plus en détail dans la suite de cette section et proposer une solution innovante.

En figure 4.1, on peut voir un processus de décorrélation tel que proposé dans la méthode LAMBDA, et identifier notre second inconvénient. Comme on peut le voir, chaque étape de l'opération de décorrélation minimise sur une dimension l'extremum contenant la zone de recherche. Dans la première figure on réduit la zone de recherche (le rectangle englobant), sur l'axe horizontal, et dans la seconde figure on réitère sur l'axe vertical. Les surfaces des ellipses restent identiques (ce qui est normal car, si l'on omet la réduction de dimension, l'incertitude absolue, elle, ne se réduit pas. On remarque au passage que le nombre d'hypothèses entières valides (c'est à dire dans les ellipses) ne varie pas ou peu. Par contre, en ignorant les inter-corrélations (et donc en cherchant dans les zones rectangulaires et non pas dans les ellipses), la surface - et donc le nombre d'hypothèse entières - à tester diminue. En effet, la réduction de surface (rectangulaire) est maximale lorsque les inter-corrélations sont nulles, d'où l'efficacité de la décorrélation et de la méthode LAMBDA.

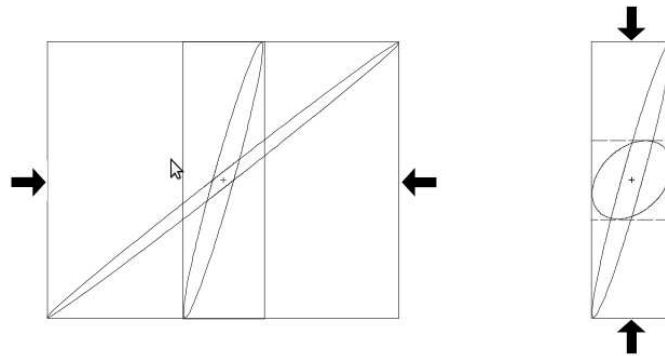


FIGURE 4.1 – Un exemple de décorrélation LAMBDA en deux dimensions. *Image : "the lambda method for fast GPS surveying", Teunissen et al., 1995[86].*

### 4.3 Méthode proposée

Notre méthode sera présentée comme suit :

- Nous commencerons par aborder le processus de création du jeu d'hypothèse en tenant compte des corrélations entre variables d'ambiguïté. L'espace de recherche de ces hypothèses sera restreint grâce à l'estimation de position obtenue par l'exploitation des triple-différences (voir chapitre 3). Nous montrerons dans cette section que la structure de donnée la plus adaptée à cette construction est un arbre : l'arbre d'hypothèse (hérité de l'arbre de décision). Cette structure permet en effet d'assurer avec une certaine probabilité que l'arbre d'hypothèse contient la solution correcte, contrairement à d'autres approches qui fixent a

---

dans la surface d'un carré au lieu du cercle concentrique et dans le pire des cas dans un rectangle au lieu d'une ellipse concentrique. Cet augmentation d'espace de recherche est réalisé à chaque dimension d'ambiguïté (souvent on considère 6 à 10 dimensions d'ambiguïté).

priori le nombre d'hypothèses testées pour n'en retenir que les plus probables. En d'autres termes, le nombre d'hypothèses testées n'est pas fixé a priori mais dépend du niveau de risque qu'on est prêt à accepter.

- Ensuite nous aborderons l'élimination en-ligne d'hypothèse (procédure incrémentale). En effet, au lieu de recommencer la construction complète d'un nouveau jeu d'hypothèse (même de taille inférieure), nous verrons une solution d'élagage d'hypothèse en-ligne sous forme de test statistique (test séquentiel binomial). Enfin nous étendrons ce processus d'élagage en présence de multi-trajet, tout simplement en faisant varier la distribution de probabilité du test
- Enfin nous présenterons nos résultats expérimentaux menés durant l'été 2011.

Considérons que notre mobile se déplace dans un milieu ouvert, dans lequel le récepteur GPS est susceptible d'obtenir des mesures de phase en mode différentiel. On considère le système dynamique Markovien suivant

$$\begin{aligned}\mathbf{X}_t &= f(\mathbf{X}_{t-1}, u_t, w_t) \\ y_t &= h(\mathbf{X}_t) + v_t\end{aligned}$$

où  $y_t$  désigne d'autres observations possibles que les mesures de phase (mesures de code, odomètre,...).

Une mesure de phase, en double différence, est décrite par son modèle statistique et peu s'écrire sous la forme générique suivante (voir équation **3.3**).

$$\nabla\Delta\varphi_t^{i,p} = H_t^{DD,i,p}\mathbf{X}_t + \nabla\Delta a_t^{i,p} + \nabla\Delta v_t^{i,p} \quad (4.1)$$

Pour simplifier la présentation, nous écrirons cette mesure sous la forme synthétique suivante :

$$\varphi_t = H_t\mathbf{X}_t + a + v_t$$

où  $\varphi_t$  désigne le vecteur de mesures de double différence de dimension  $n$  égale au nombre de satellites visibles moins l'unité,  $a$  le vecteur d'entiers représentant les ambiguïtés et  $v_t$  le bruit blanc de mesure. A noter que même pour un nombre de mesures supérieur à trois, le rang de la matrice  $H_t$  reste égal à trois. Ceci a pour conséquence d'induire de fortes corrélations entre ambiguïtés.

### 4.3.1 Construction arborescente du jeu d'hypothèse

Au moment de la création du jeu d'hypothèse, on suppose avoir obtenu une statistique de l'état  $\mathbf{X}_t$  de notre mobile. Dans notre cas, cette statistique est issue du filtre décrit dans le chapitre qui donne une estimation gaussienne de la position à partir des triples-différences. Cette estimation est gaussienne si le système est linéaire et une approximation gaussienne dans le cas général. On dispose donc de la donnée suivante :

$$p(\mathbf{X}_t) = \Gamma(\mathbf{X}_t - \hat{\mathbf{X}}_t, P_t)$$

où  $\Gamma$  désigne la densité de probabilité gaussienne centrée

$$\Gamma(x, P) = \frac{1}{\sqrt{(2\pi)^n |P|}} \exp\left(-\frac{1}{2}x^T P^{-1}x\right)$$

Nous allons montrer comment construire notre jeu d'hypothèse d'ambiguïtés à partir de cette statistique.

On remarque que le modèle d'observation est inversible (ce qui permet de lier l'état du mobile et la mesure de phase, à la valeur ambiguïté correspondante) :

$$a^{DD} = \varphi_t^{DD} - H_t \mathbf{X}_t - v_t$$

La variable d'ambiguïté de phase influe sur la mesure de phase de manière additive.

Donc, à partir d'une mesure de phase  $\varphi_t^{DD}$ , et de notre modèle d'incertitude sur  $\mathbf{X}_t$ , nous pouvons en déduire un modèle d'incertitude sur  $a^{DD}$ . Nous avons une approximation locale d'une distribution gaussienne pour la variable d'ambiguïté

$$p(a^{DD} | \varphi_t^{DD}) = \Gamma(a^{DD} - \varphi_t^{DD} + H_t \hat{\mathbf{X}}_t, H_t P_t H_t^T + R)$$

où

$$R = \mathbb{E}[v_t v_t^T]$$

À partir des modèles d'incertitude sur les variables d'ambiguïtés ( $p(a^{DD} | \varphi_t^{DD})$ ), nous allons construire nos intervalles de recherche. De manière récursive, nous allons nous limiter au jeu d'ambiguïté entier strictement nécessaire pour respecter un certain niveau de risque de ne pas inclure la bonne hypothèse dans notre jeu final. Commençons par définir quelques termes.

**Définition 4.1.** On appelle "hypothèse nulle", notée  $H_0$ , une hypothèse postulant une égalité entre deux données d'un modèle. Elle est toujours testée contre une hypothèse alternative qui postule au contraire la différence entre les données.

**Définition 4.2.** On appelle "risque" la probabilité que notre test nous retourne la mauvaise réponse sur  $H_0$ .

Nous allons construire récursivement notre jeu d'hypothèse. Sur la première mesure de phase, nous avons vu que nous pouvions déduire du modèle d'incertitude sur notre mobile et sur la mesure, un modèle d'incertitude continu (gaussien) sur notre variable de phase. Notons  $p(a^{DD} | \varphi_t^{DD})$  notre densité de probabilité de la variable d'ambiguïté de type continu, et  $\mathbb{P}(a^{DD} = k | \varphi_t^{DD})$ ,  $\forall k \in \mathbb{Z}$  notre probabilité de cette même variable mais cette fois-ci restreinte au domaine des entiers. Nous allons intégrer la "contrainte entière". Rappelons que la définition d'une probabilité conditionnelle s'écrit

$$\mathbb{P}(a^{DD} = k | \phi_t^{DD} = \varphi_t^{DD}) = \lim_{d\varphi \rightarrow 0} \frac{\mathbb{P}(a^{DD} = k, \phi_t^{DD} \in [\varphi_t^{DD}, \varphi_t^{DD} + d\varphi])}{\mathbb{P}(\phi_t^{DD} \in [\varphi_t^{DD}, \varphi_t^{DD} + d\varphi])}$$

Comme  $\varphi_t^{DD} = H_t \mathbf{X}_t + a^{DD} + v_t$ ,  $\phi_t^{DD} \in [\varphi_t^{DD}, \varphi_t^{DD} + d\varphi]$  si et seulement si

$$\varphi_t^{DD} < H_t \mathbf{X}_t + k + v_t < \varphi_t^{DD} + d\varphi$$

Comme  $x_t$  et  $v_t$  sont gaussiens, la densité de probabilité de  $z_t = H_t \mathbf{X}_t + v_t$  peut s'écrire

$$p_{Z_t}(z_t) = \Gamma(z_t - H_t \hat{\mathbf{X}}_t, H_t P_t H_t^T + R)$$

On a alors

$$\mathbb{P}(a^{DD} = k, \phi_t^{DD} \in [\varphi_t^{DD}, \varphi_t^{DD} + d\varphi]) = p_{Z_t}(\varphi_t^{DD} - k) d\varphi$$

Par ailleurs,  $\mathbb{P}(\phi_t^{DD} \in [\varphi_t^{DD}, \varphi_t^{DD} + d\varphi])$  peut être vu comme la marginale sur  $a^{DD}$  de la densité définie par

$$\mathbb{P}(a^{DD} = k, \phi_t^{DD} \in [\varphi_t^{DD}, \varphi_t^{DD} + d\varphi])$$

D'où

$$\begin{aligned} \mathbb{P}(\phi_t^{DD} \in [\varphi_t^{DD}, \varphi_t^{DD} + d\varphi]) &= \sum_{k \in \mathbb{Z}} \mathbb{P}(a^{DD} = k, \phi_t^{DD} \in [\varphi_t^{DD}, \varphi_t^{DD} + d\varphi]) \\ &= \sum_{k \in \mathbb{Z}} p_{Z_t}(\varphi_t^{DD} - k) d\varphi \end{aligned}$$

On a donc finalement

$$\mathbb{P}(a^{DD} = k | \varphi_t^{DD}) = \frac{p_{Z_t}(\varphi_t^{DD} - k)}{\sum_{k \in \mathbb{Z}} p_{Z_t}(\varphi_t^{DD} - k)}$$

Dans le cas gaussien, on voit que  $\mathbb{P}(a^{DD} = k | \varphi_t^{DD})$  s'étend sur l'intervalle  $[-\infty; +\infty]$ . Nous allons construire un algorithme qui permet de réduire cet intervalle au risque de rejeter la bonne hypothèse d'ambiguïté dans notre ensemble final.

Rappelons que  $H_0$  désigne l'hypothèse selon laquelle notre jeu d'ambiguïté final contiendra le bon vecteur d'entiers. Nous allons choisir le premier sous-ensemble  $E$ , qui affectera notre risque  $\alpha$ . La méthode optimale pour sélectionner un ensemble de taille minimal, en contrôlant le risque que la solution ne soit pas dans l'intervalle, consiste à sélectionner cet ensemble en prenant les zones où la probabilité est la plus forte. Pour simplifier la présentation, nous écrirons  $\mathbb{P}(a = k)$  la probabilité conditionnelle de l'ambiguïté entière  $\mathbb{P}(a^{DD} = k | \varphi_t^{DD})$  et  $p(a)$  celle de l'ambiguïté continue  $p(a^{DD} | \varphi_t^{DD})$ . Soit

$$E = \{a | p(a) \in \max p(a)\}$$

tel que

$$\int_E p(a) da = 1 - \alpha$$

L'opérateur  $\max$  désigne ici l'enveloppe supérieure de  $p(a)$ .

La figure 4.2 présente le cas d'une densité continue. On observe graphiquement un intervalle minimal respectant un certain risque  $\alpha$ , il s'agit de l'union des sous-intervalles pour lesquels la probabilité est maximale.

Dans le cas continu gaussien, la densité de probabilité étant symétrique, l'ensemble  $E$  se réduit à un intervalle  $I$  centré sur la valeur la plus probable  $\bar{a}$  (la moyenne de la densité gaussienne) :

$$I = [\bar{a} - r; \bar{a} + r]$$

où  $r$  est tel que

$$\operatorname{erf}\left(\frac{r}{\sigma}\right) - \operatorname{erf}\left(-\frac{r}{\sigma}\right) = 1 - \alpha \quad (4.2)$$

et où  $\sigma$  est la variance de la loi normale définissant  $p(a)$ .  $\operatorname{erf}$  désigne la fonction erreur.

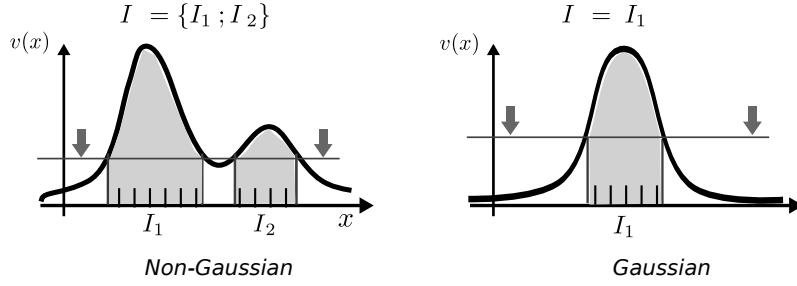


FIGURE 4.2 – L'ensemble minimal de plus grande probabilité est l'ensemble des sous-intervalles pour lesquelles les probabilités sont maximales.

Si on échantillonne la densité gaussienne, l'ensemble devient alors un ensemble d'entiers tel que

$$L = \{k | \mathbb{P}(a = k) \in \max \mathbb{P}(a = k)\}$$

défini tel que

$$\sum_{k \in L} \mathbb{P}(a = k) > 1 - \alpha$$

pour une certaine valeur de  $\alpha$  évaluant le risque. En effet, on ne peut pas garantir l'égalité stricte,  $\alpha$  étant un réel quelconque fixé a priori.

Il est clair que la densité de probabilité étant continue,  $I \cap \mathbb{Z} \subset L$ , il suffit, pour garantir le risque  $\alpha$ , de considérer l'ensemble des entiers contenus dans  $I$ .

La propriété précédente nous permettra, en choisissant l'ensemble  $I$ , de contrôler notre risque. Nous définissons donc, dans l'espace de notre première ambiguïté, le plus petit intervalle  $I_1$ , le risque  $\alpha_1$  étant un paramètre d'entrée. Rappelons que l'ambiguïté relative à la première mesure s'écrit

$$a_1 = \varphi_t^1 - H_t^1 \mathbf{X}_t - v_t^1$$

et que que  $\mathbf{X}_t$  et  $v_t^1$  étant gaussiens, on a, pour  $a_1 \in \mathbb{R}$ , la densité de probabilité suivante

$$p(a_1 | \varphi_t^1) = \Gamma(\varphi_t^1 - a_1 - H_t^1 \hat{\mathbf{X}}_t, H_t^1 P_t (H_t^1)^T + R_1)$$

et pour  $a_1 \in \mathbb{Z}$ , la probabilité suivante

$$\mathbb{P}(a_1 = k_1 | \varphi_t^1) = \frac{\Gamma(\varphi_t^1 - k_1 - H_t^1 \hat{\mathbf{X}}_t, H_t^1 P_t (H_t^1)^T + R_1)}{\sum_{k_1 \in \mathbb{Z}} \Gamma(\varphi_t^1 - k_1 - H_t^1 \hat{\mathbf{X}}_t, H_t^1 P_t (H_t^1)^T + R_1)}$$



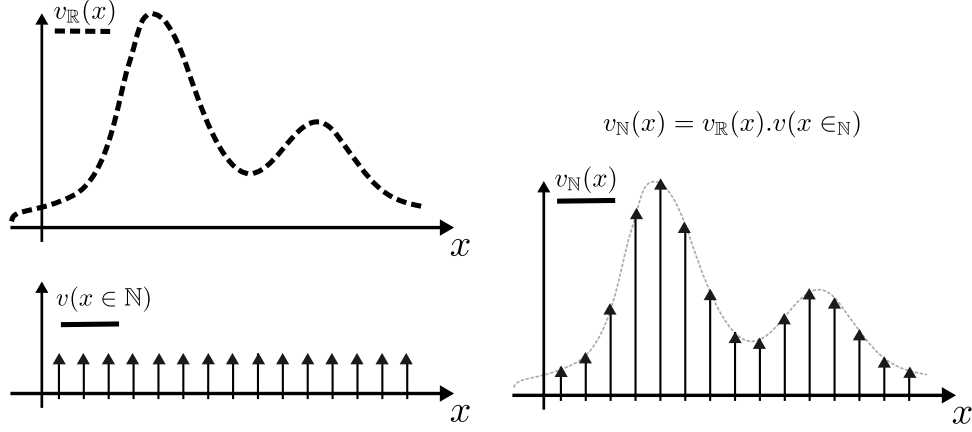


FIGURE 4.3 – L'ajout de la contrainte entière sur une densité de probabilité revient à échantillonner cette densité puis à la normaliser.

avec  $R_1 = 4(\sigma_\varphi)^2$ . L'intervalle  $I_1$  est donc centré sur  $\varphi_t^1 - H_t^1 \hat{\mathbf{X}}_t$ , sa largeur est définie par la relation 4.2 avec

$$\sigma_1 = \sqrt{H_t^1 P_t (H_t^1)^T + R_1}$$

On obtient alors, à partir des entiers  $L_1$  contenus dans  $I_1$ , notre espace de recherche d'entiers pour la première variable d'ambiguïté  $a_1$ .

Notre arbre d'hypothèse possède donc maintenant  $N_1 = \#L_1$  feuilles, les  $N_1$  valeurs d'ambiguïtés entières contenues dans  $I_1$ , l'ensemble pour lequel  $\alpha_1$  est le risque de ne pas contenir le bon entier dans notre arbre. Pour chaque hypothèse entière  $k_1 \in L_1$ , feuille de l'arbre, nous allons étendre notre choix en exploitant une seconde mesure.

Notons

$$\begin{aligned} z_t^1 &= H_t^1 \mathbf{X}_t + v_t^1 \\ z_t^2 &= H_t^2 \mathbf{X}_t + v_t^2 \end{aligned}$$

Soit

$$z_t = \begin{bmatrix} z_t^1 \\ z_t^2 \end{bmatrix}, H_t = \begin{bmatrix} H_t^1 \\ H_t^2 \end{bmatrix}, v_t = \begin{bmatrix} v_t^1 \\ v_t^2 \end{bmatrix}, a = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}, \varphi_t = \begin{bmatrix} \varphi_t^1 \\ \varphi_t^2 \end{bmatrix}$$

$\mathbf{X}_t$  et  $v_t^i$  étant gaussiens, la densité de probabilité conjointe de  $a_1 = \varphi_t^1 - z_t^1$  et  $a_2 = \varphi_t^2 - z_t^2$  est alors

$$p(a_1, a_2 | \varphi_t^1, \varphi_t^2) = \Gamma(\varphi_t - a - H_t \hat{\mathbf{X}}_t, H_t P_t H_t^T + R_2)$$

avec

$$R_2 = \begin{bmatrix} 4(\sigma_\varphi)^2 & 2(\sigma_\varphi)^2 \\ 2(\sigma_\varphi)^2 & 4(\sigma_\varphi)^2 \end{bmatrix}$$

Notons  $M_2 = H_t P_t H_t^T + R_2$ . La densité de probabilité de  $a_2$  conditionnellement à  $a_1$  est gaussienne de moyenne et variance définies en utilisant le complément de Schur par

$$\begin{aligned} \bar{a}_2(k_1) &= \mathbb{E}[a_2 | a_1 = k_1] \\ &= \mathbb{E}[a_2] + [M_2]_2^1 \left( [M_2]_2^2 \right)^{-1} (k_1 - \mathbb{E}[a_2]) \\ &= H_t^2 \hat{\mathbf{X}}_t + [M_2]_2^1 \left( [M_2]_2^2 \right)^{-1} (k_1 - H_t^1 \hat{\mathbf{X}}_t) \\ (\sigma_2)^2 &= \mathbb{E}[(a_2 - \bar{a}_2)^2 | a_1 = k_1] \\ &= [M_2]_2^2 - [M_2]_2^1 \left( [M_2]_2^2 \right)^{-1} [M_2]_2^1 \end{aligned}$$

où  $[M_2]_i^j$  désigne le terme de  $M_2$  de ligne  $i$  et de colonne  $j$ . A noter que la variance de  $a_2$  conditionnelle à  $a_1 = k_1$  est toujours inférieure ou égale à sa variance a priori. Ceci a pour effet de réduire l'espace de recherche de  $a_2$  après que  $a_1$  ait été fixé. On peut alors écrire :

$$\mathbb{P}(a_2 = k_2 | a_1 = k_1, \varphi_t^2) = \frac{\Gamma(\varphi_t^2 - \bar{a}_2(k_1), (\sigma_2)^2)}{\sum_{k_2 \in \mathbb{Z}} \Gamma(\varphi_t^2 - \bar{a}_2(k_1), (\sigma_2)^2)}$$

Ainsi, pour chaque hypothèse  $k_1 \in L_1$ , on construit l'intervalle  $I_2$  centré sur  $\bar{a}_2(k_1)$  de largeur définie par 4.2 avec l'écart-type  $\sigma_2$  et le risque  $\alpha_2$ . On considère alors les entiers inclus dans cet intervalle  $k_2 \in L_2$ . La figure 4.4 présente la stratégie de construction arborescente.

A l'étape  $n$ , on obtient

$$p(a_1, \dots, a_n | \varphi_t^1, \dots, \varphi_t^n) = \Gamma(\varphi_t - a - H_t \hat{\mathbf{X}}_t, H_t P_t H_t^T + R_n)$$

avec

$$H_t = \begin{bmatrix} H_t^1 \\ \vdots \\ H_t^n \end{bmatrix}, a = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}, \varphi_t = \begin{bmatrix} \varphi_t^1 \\ \vdots \\ \varphi_t^n \end{bmatrix}$$

$$R_n = \begin{bmatrix} 4(\sigma_\varphi)^2 & 2(\sigma_\varphi)^2 & \cdots & 2(\sigma_\varphi)^2 \\ 2(\sigma_\varphi)^2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 2(\sigma_\varphi)^2 \\ 2(\sigma_\varphi)^2 & \cdots & 2(\sigma_\varphi)^2 & 4(\sigma_\varphi)^2 \end{bmatrix}$$

Notons  $M_n = H_t P_t H_t^T + R_n$ . La densité de probabilité de  $a_n$  conditionnellement à  $a_1, \dots, a_{n-1}$

est gaussienne de moyenne et variance définies par

$$\begin{aligned}
 \bar{a}_n(k_1, \dots, k_{n-1}) &= \mathbb{E}[a_n | a_1 = k_1, \dots, a_{n-1} = k_{n-1}] \\
 &= \mathbb{E}[a_n] + [M_n]_n^{1:n-1} ([M_n]_n^n)^{-1} \begin{bmatrix} k_1 - \mathbb{E}[a_1] \\ \vdots \\ k_{n-1} - \mathbb{E}[a_{n-1}] \end{bmatrix} \\
 &= H_t^n \hat{\mathbf{X}}_t + [M_n]_n^{1:n-1} ([M_n]_n^n)^{-1} \begin{pmatrix} k_1 - H_t^1 \hat{\mathbf{X}}_t \\ \vdots \\ k_{n-1} - H_t^{n-1} \hat{\mathbf{X}}_t \end{pmatrix} \\
 (\sigma_n)^2 &= \mathbb{E}[(a_n - \bar{a}_n)^2 | a_1 = k_1, \dots, a_{n-1} = k_{n-1}] \\
 &= [M_n]_n^n - [M_n]_n^{1:n-1} ([M_n]_n^n)^{-1} [M_n]_n^{1:n-1}
 \end{aligned}$$

On obtient alors

$$\mathbb{P}(a_n = k_n | a_1 = k_1, \dots, a_{n-1} = k_{n-1}, \varphi_t^n) = \frac{\Gamma(\varphi_t^n - \bar{a}_n(k_1, \dots, k_{n-1}), (\sigma_n)^2)}{\sum_{k_n \in \mathbb{Z}} \Gamma(\varphi_t^n - \bar{a}_n(k_1, \dots, k_{n-1}), (\sigma_n)^2)}$$

On construit alors l'intervalle  $I_n$  centré sur  $\bar{a}_n(k_1, \dots, k_{n-1})$  de largeur définie par 4.2 avec l'écart-type  $\sigma_n$  et le risque  $\alpha_n$  et on construit l'ensemble  $L_n$  des entiers contenus dans cet intervalle. La probabilité conjointe des ambiguïtés ainsi explorées s'obtient par le produit suivant :

$$\begin{aligned}
 \mathbb{P}(a_1 = k_1, \dots, a_n = k_n | \varphi_t^1, \dots, \varphi_t^n) &= \mathbb{P}(a_n = k_n | a_1 = k_1, \dots, a_{n-1} = k_{n-1}, \varphi_t^n) \\
 &\quad \times \mathbb{P}(a_1 = k_1, \dots, a_{n-1} = k_{n-1} | \varphi_t^1, \dots, \varphi_t^{n-1}) \\
 &= \mathbb{P}(a_n = k_n | a_1 = k_1, \dots, a_{n-1} = k_{n-1}, \varphi_t^n) \\
 &\quad \times \mathbb{P}(a_{n-1} = k_{n-1} | a_1 = k_1, \dots, a_{n-2} = k_{n-2}, \varphi_t^{n-1}) \\
 &\quad \dots \\
 &\quad \times \mathbb{P}(a_2 = k_2 | a_1 = k_1, \varphi_t^2) \times \mathbb{P}(a_1 = k_1 | \varphi_t^1)
 \end{aligned}$$

A chaque niveau  $i$ , les intervalles d'exploration  $L_i$  sont définis par

$$\sum_{k_i \in L_i} \mathbb{P}(a_i = k_i | a_1 = k_1, \dots, a_{i-1} = k_{i-1}, \varphi_t^i) > 1 - \alpha_i$$

A chaque niveau  $L_n$ , le risque de ne pas contenir la bonne combinaison d'ambiguïtés entières est défini par les risques  $\alpha_i, i \leq n$  accordés aux niveaux supérieurs de telle sorte que pour  $L_n$ , nous obtenions le risque :

$$\mathbb{P}(H_0) = \sum_{k_1 \in I_1} \dots \sum_{k_n \in L_n} \mathbb{P}(a_1 = k_1, \dots, a_n = k_n | \varphi_t^1, \dots, \varphi_t^n) > (1 - \alpha_1) \dots (1 - \alpha_n)$$

La méthode de construction proposée fait apparaître deux avantages :

- Une méthode qui ne nécessite pas de procéder à une décorrélation des ambiguïtés telle que proposé dans la méthode LAMBDA. La corrélation entre ambiguïtés est directement prise en compte par la définition récursive des intervalles de recherche sans nécessiter l'approximation "changement de variable entier" de la méthode LAMBDA
- En définissant à chaque niveau un risque, on contrôle le risque global que le vrai jeu d'ambiguïté ne soit pas une des feuilles.

### 4.3.2 Elagage séquentiel de l'arbre d'hypothèses

A ce stade, nous disposons d'un arbre représentant l'ensemble des hypothèses à tester afin de garantir notre risque. Chaque feuille est une hypothèse qui est relié à la racine par un chemin ascendant unique dont chaque niveau définit l'ambiguïté entière du satellite concerné.

On peut alors procéder à l'élagage "en-ligne" de l'arbre d'hypothèse. Pour ce faire, on peut envisager différentes approches :

- **Filtrage optimal.** Dans ce cas, on cherche à construire la probabilité conjointe de l'état d'intérêt  $X_t$  (la position) et du vecteur d'ambiguïté  $a$  ( $p(X_t, a = k | \varphi_{0:t})$ ).  $k$  désigne le vecteur d'entiers des ambiguïtés de phase dont le domaine d'appartenance est défini comme ci-dessus, l'ensemble des feuilles de l'arbre. On peut alors montrer que, si cette densité est gaussienne :

$$p(X_t, a = k | \varphi_{0:t}) = \rho_t^k \Gamma(X_t - \hat{X}_{t|t}^k, P_{t|t})$$

où  $\rho_t^k$  désigne la probabilité a posteriori de  $a = k$  ( $\rho_t^k = \mathbb{P}(a = k | \varphi_{0:t})$ ), alors, ces paramètres sont calculables récursivement selon

- Prédiction

$$\begin{aligned} \hat{X}_{t|t-1}^k &= F \hat{X}_{t-1|t-1}^k \\ P_{t|t-1} &= F P_{t-1|t-1} F^T + Q \end{aligned}$$

- Correction

$$\begin{aligned} \hat{X}_{t|t}^k &= \hat{X}_{t|t-1}^k + K_t (\varphi_t - H_t \hat{X}_{t|t-1}^k - k) \\ \Sigma_t &= H_t P_{t|t-1} H_t^T + R \\ K_t &= P_{t|t-1} H_t^T (\Sigma_t)^{-1} \\ P_{t|t} &= P_{t|t-1} - K_t H_t P_{t|t-1} \end{aligned}$$

- Pondération

$$\rho_t^k = \frac{\rho_{t-1}^k \Gamma(\varphi_t - H_t \hat{X}_{t|t-1}^k - k, \Sigma_t)}{\sum_k \rho_{t-1}^k \Gamma(\varphi_t - H_t \hat{X}_{t|t-1}^k - k, \Sigma_t)}$$

A noter que dans ce cas, les matrices de covariances ainsi que les gains des filtres de Kalman en parallèle ne dépendent pas de la valeur d'ambiguïté estimée. Avec l'accumulation des mesures, les poids  $\rho_t^k$  évoluent de sorte que le poids de la bonne hypothèse tend vers l'unité, alors que les autres poids tendent tous vers zéro. L'élagage est alors le résultat de l'élimination des feuilles de probabilités inférieures à un certain seuil  $\tau$ , définissant notre risque a posteriori.

- **Approximation du filtrage optimal.** Dans ce cas, afin d'éviter de calculer pour chaque feuille  $k$  de l'estimation de position  $\hat{X}_{t|t}^k$ , on remplace celle-ci par l'estimation délivrée en parallèle par le filtre aux triples différences  $\hat{X}_{t|t}^{TD}$ . On s'intéresse alors à la vraisemblance de chaque jeu d'ambiguïté définie par

$$\rho_t^k = \frac{\rho_{t-1}^k \Gamma(\varphi_t - H_t \hat{X}_{t|t}^{TD} - k, \Sigma_t)}{\sum_k \rho_{t-1}^k \Gamma(\varphi_t - H_t \hat{X}_{t|t}^{TD} - k, \Sigma_t)}$$

Comme dans le cas précédent, l'élagage va consister à éliminer les feuilles de moindres poids. A noter que l'estimateur à triple différence ayant été obtenu par le traitement des mesures de phase, celui-ci n'est pas indépendant des mesures  $\varphi_t$ .

- **Estimateur à maximum de vraisemblance.** Dans cette approche, on ne tient pas compte de l'autocorrélation temporelle de la position. Par conséquent, celle-ci est considérée à chaque instant comme une variable aléatoire indépendante dans le temps. Rappelons que le rang de  $H_t$  est au plus égal à trois, et égale à trois si le nombre de satellites visibles est supérieur à quatre. Dans ce cas, on peut inverser à chaque instant la relation entre état et observation, c'est-à-dire

$$\varphi_t = H_t X_t + k + v_t$$

en

$$\begin{aligned} X_t(k) &= (H_t^T H_t)^{-1} H_t^T (\varphi_t - k - v_t) \\ &= (H_t^T H_t)^{-1} H_t^T (\varphi_t - k) - (H_t^T H_t)^{-1} H_t^T v_t \end{aligned}$$

Par conséquent,  $X_t(k)$  est une variable gaussienne de moyenne

$$\hat{X}_t^{MV}(k) = (H_t^T H_t)^{-1} H_t^T (\varphi_t - k)$$

et de variance

$$\Sigma_t = (H_t^T H_t)^{-1} H_t^T R H_t (H_t^T H_t)^{-1}$$

indépendante de  $k$ . La vraisemblance de chaque hypothèse s'écrit alors récursivement

$$\rho_t^k = \frac{\rho_{t-1}^k \Gamma(\varphi_t - k - H_t \hat{X}_t^{MV}(k), \Sigma_t)}{\sum_k \rho_{t-1}^k \Gamma(\varphi_t - k - H_t \hat{X}_t^{MV}(k), \Sigma_t)}$$

Comme dans le cas précédent, l'élagage va consister à éliminer les feuilles de moindres poids.

- **Test de consistance sur le caractère gaussien du vecteur de résidus.** On considère le vecteur de résidu construit à partir de l'estimation de position  $\hat{X}_t^{MV}$  issue de l'estimateur à maximum de vraisemblance défini ci-dessus

$$\begin{aligned} \tilde{\varphi}_t^1(k_1) &= \varphi_t^1 - H_t^1 \hat{X}_t^{MV}(k) - k_1 \\ &\dots \\ \tilde{\varphi}_t^n(k_n) &= \varphi_t^n - H_t^n \hat{X}_t^{MV}(k) - k_n \end{aligned}$$

Pour chaque valeur de  $k$ , ces résidus s'écrivent

$$\begin{aligned}\tilde{\varphi}_t(k) &= \varphi_t - k - H_t \hat{X}_t^{MV}(k) \\ &= \varphi_t - k - H_t (H_t^T H_t)^{-1} H_t^T (\varphi_t - k) \\ &= \left( I - H_t (H_t^T H_t)^{-1} H_t^T \right) (\varphi_t - k)\end{aligned}$$

Notons

$$S_t = \left( I - H_t (H_t^T H_t)^{-1} H_t^T \right)$$

Remarquons alors que

$$\begin{aligned}S_t H_t &= \left( I - H_t (H_t^T H_t)^{-1} H_t^T \right) H_t \\ &= 0\end{aligned}$$

On constate que, pour la bonne hypothèse  $a = k^*$ ,  $\varphi_t - k^* = H_t X_t + v_t$ , et par conséquent

$$\begin{aligned}\tilde{\varphi}_t(k^*) &= S_t (H_t X_t + v_t) \\ &= S_t v_t\end{aligned}$$

suit une loi normale de moyenne nulle et de variance, en l'occurrence, égale au moment d'ordre deux, définie par :

$$\begin{aligned}\Sigma_t &= \mathbb{E} \left[ (\tilde{\varphi}_t(k^*)) (\tilde{\varphi}_t(k^*))^T \right] \\ &= S_t R (S_t)^T\end{aligned}$$

Pour toute valeur  $k \neq k^*$ , si on note  $\Delta k = k - k^*$ , on a

$$\begin{aligned}\tilde{\varphi}_t(k) &= S_t (H_t X_t + \Delta k + v_t) \\ &= S_t (\Delta k + v_t)\end{aligned}$$

$\tilde{\varphi}_t(k)$  est à nouveau une variable aléatoire gaussienne de moyenne

$$\mathbb{E} [\tilde{\varphi}_t(k)] = S_t \Delta k$$

et de variance  $\Sigma_t$ . L'expression de son moment d'ordre deux est différent de sa variance, il s'écrit

$$\mathbb{E} \left[ (\tilde{\varphi}_t(k)) (\tilde{\varphi}_t(k))^T \right] = \Sigma_t + (\Delta k)^T S_t \Delta k$$

la matrice  $S_t$  étant nilpotente d'ordre un ( $S_t S_t = S_t$ ). On construit alors le résidu quadratique normalisé défini par

$$\tilde{\varphi}_{t,\Sigma}^2(k) = (\tilde{\varphi}_t(k))^T \Sigma^{-1} \tilde{\varphi}_t(k)$$

Ce résidu quadratique devant suivre une certaine loi du  $\chi^2$  pour  $k = k^*$ , la technique consiste à tester la consistance de cette variable scalaire avec cette loi.

La méthode que nous développerons ici repose sur ce test de consistance du résidu quadratique normalisé et est détaillée dans la section suivante.

## 4.4 Test de consistance sur le caractère gaussien du vecteur de résidu

On constate que, pour la bonne hypothèse d'ambiguïté  $k^*$ , le résidu quadratique normalisé  $\tilde{\varphi}_{t,\Sigma}^2(k^*)$  suit une loi du  $\chi^2$  centré à  $n - 3$  degrés de liberté (le rang de  $S_t$ ) alors que pour toute autre valeur de  $k \neq k^*$ , ce moment d'ordre deux suit une loi du  $\chi^2$  décentrée à, également,  $n - 3$  degrés de liberté.

Nous allons utiliser cette propriété. Nous nous proposerons en effet ici de tester, pour chaque feuille de l'arbre, l'hypothèse du caractère gaussien du vecteur de résidus. Pour ce faire, nous allons utiliser un test de consistance sur le résidu quadratique normalisé  $\tilde{\varphi}_{t,\Sigma}^2(k)$ .

**Proposition. Construction d'une loi du  $\chi^2$  à partir d'une loi gaussienne centrée.** Soit  $v$  un vecteur de variables aléatoires de distribution gaussienne, de moyenne nulle, et de moment d'ordre deux  $\Sigma$ . La variable scalaire  $v_{\Sigma}^2$ <sup>7</sup>, ayant comme expression

$$\begin{aligned} v_{\Sigma}^2 &= v^T \Sigma^{-1} v \\ &= v^T \mathbb{E} [v^T v]^{-1} v \end{aligned}$$

suit une loi du  $\chi^2$  centré à  $n$  degrés de liberté, où  $n$  désigne le rang de  $\Sigma$  ( $n$  est donc aussi le nombre de degrés de liberté de  $v$ ).

**Proposition. Construction d'une loi du  $\chi^2$  à partir d'une loi gaussienne quelconque.** Soit  $v$  un vecteur de variables aléatoires de distribution gaussienne, de moyenne  $\bar{v}$  et de moment d'ordre deux  $\Sigma$ . La variable scalaire  $v_{\Sigma}^2$ , ayant comme expression

$$\begin{aligned} v_{\Sigma}^2 &= (v - \bar{v})^T \Sigma^{-1} (v - \bar{v}) \\ &= (v - \bar{v})^T \mathbb{E} [(v - \bar{v})^T (v - \bar{v})]^{-1} (v - \bar{v}) \end{aligned}$$

suit également une loi du  $\chi^2$  centré à  $n$  degrés de liberté, où  $n$  désigne le rang de  $\Sigma$  ( $n$  est donc aussi le nombre de degrés de liberté de  $v$ ).

**Remarque. Cas de l'application d'un moment d'ordre deux erroné ou d'une moyenne erronée.** Il est intéressant de noter que, pour  $v$  un vecteur de variables aléatoires gaussiennes, de moyenne  $\bar{v}$  et de moment d'ordre deux différent de sa variance  $\Sigma$ , la variable  $v_{\Sigma}^2 = (v - \bar{v})^T \Sigma^{-1} (v - \bar{v})$  suit une loi du  $\chi^2$  décentré, à  $n$  degrés de liberté, où  $n$  désigne le rang de  $\Sigma$  ( $n$  est donc aussi le nombre de degrés de liberté de  $v$ ). À noter que l'expression du décentrage est connue si et seulement si on connaît le véritable moment d'ordre deux de  $v$ . On note aussi que si l'expression de  $\bar{v}$  est erronée,  $v_{\Sigma}^2$  suit également une loi du  $\chi^2$  décentré, à  $n$  degrés de liberté. L'expression du décentrage dépend alors du véritable moment d'ordre deux de  $v$ , et de la véritable moyenne  $\bar{v}$ .

### 4.4.1 Test séquentiel

Il est naturel dans un premier temps d'appliquer séquentiellement dans le temps un test d'intervalle  $T_t(k)$  à chaque hypothèse d'ambiguïté  $k$ . Nous choisissons de mettre en place un test

---

7. Nous appellerons cette variable *residu quadratique normalisé* dans le cas où  $v$  est un vecteur de résidu

spécifique, dont le résultat est binaire. C'est à dire que, pour toute hypothèse d'ambiguïté  $k$  testée, le test  $T_t(k)$  renvoie "faux" si l'hypothèse est jugée fautive avec un certain risque maîtrisé, que l'on définira, et renvoie "vrai" sinon.

La procédure est la suivante. Toutes les hypothèses  $k$ , pour lesquelles le test  $T_t(k)$  échoue sont éliminées définitivement du jeu d'hypothèse. Naturellement, celles dont le test réussit sont conservées dans le jeu d'hypothèse et restent disponibles pour le test à l'instant suivant. Notre procédure d'élimination basé sur un test  $T_t(k)$  possède certaines caractéristiques statistiques que nous allons présenter ci-dessous. Rappelons que l'objectif majeur de cette démarche est de maîtriser le risque d'éliminer la bonne hypothèse. Ainsi, le nombre d'hypothèses conservées pour être testées à l'étape suivante n'est pas fixé a priori, comme ce serait le cas dans la méthode "filtrage optimal", mais est déterminé au fur et à mesure de l'accumulation des données.

A chaque instant  $t$ , nous disposons, en entrée de notre algorithme, d'un ensemble  $\mathcal{H}_t$  d'hypothèses d'ambiguïtés à tester. Parmi ces hypothèses, la solution vraie  $k^*$  y est présente avec une probabilité que nous souhaitons maîtriser à partir du concept de risque d'absence ou probabilité de non-détection. Le vecteur de résidu  $\tilde{\varphi}_t(k)$  construit à partir des mesures courantes permet de définir le résidu quadratique pondéré associé  $\tilde{\varphi}_{t,\Sigma}^2(k)$  censé suivre une loi du  $\chi^2$  centré à  $n - 3$  degrés de liberté pour  $k = k^*$ , et une loi du  $\chi^2$  décentré pour tout  $k \neq k^*$ . A noter que pour  $k \neq k^*$ , la distribution de  $\tilde{\varphi}_{t,\Sigma}^2(k)$  est partiellement inconnue, car elle dépend du biais d'ambiguïté  $\Delta k = k - k^*$  (qui est un paramètre inconnu).

**Définition 4.3. PND.** On appelle **PND**( $T$ ) (**P**robabilité de **N**on-**D**étection) la probabilité que le test  $T_t(k)$  échoue en étant appliqué à la bonne hypothèse d'ambiguïté  $k^*$ .

**Définition 4.4. PFP.** On appelle **PFP**( $T$ ) (**P**robabilité de **F**aux **P**ositif) la probabilité que le test  $T_t(k)$  réussisse, en étant appliqué à une hypothèse d'ambiguïté  $k$  fautive ( $k \neq k^*$ ).

Nous allons donc construire un test par intervalle  $T_t(k)$ , sur chaque résidu quadratique normalisé  $\tilde{\varphi}_{t,\Sigma}^2(k)$ , qui permettra de classifier  $k$  en maîtrisant le risque **PND**, que l'on souhaite minimal, mais naturellement pas le risque **PFP**.

Le test par intervalle  $T_t(k)$  optimal consiste à décider en l'élimination **définitive** de l'hypothèse  $k$  si la valeur de résidu quadratique normalisé  $\tilde{\varphi}_{t,\Sigma}^2(k)$  n'est pas dans un certain intervalle  $I$ . Cet intervalle  $I$  est défini comme l'intervalle tel que la valeur de vraisemblance est la plus grande sur la loi du  $\chi^2$  centré à  $n$  degrés de liberté et garantissant  $\int_I \chi^2(x) dx = 1 - \text{PND}$ .

Les premiers essais que nous avons réalisés en appliquant ce test à un ensemble de cas d'étude se sont avérés insatisfaisants, la bonne ambiguïté étant trop souvent éliminée trop tôt. C'est pourquoi nous avons adapté ce test et construit un test que nous qualifierons de "global".

#### 4.4.2 Test global

**Définition 4.5. Test global.** On appelle test global  $T(k)$  un test dont la décision (ici l'élimination ou pas de l'hypothèse  $k$ ) est réalisée seulement après plusieurs sous-tests répartis dans le temps. Le test global diffère de notre stratégie précédente. En effet, dans notre approche précédente, la décision d'éliminer une hypothèse  $k$  est conditionnée à chaque instant  $t$  par l'application du test  $T_t(k)$ . On désirera, à partir d'ici, que cette décision d'élimination soit le fruit d'une compilation de plusieurs tests antérieurs ( $T_{t_1}, \dots, T_{t_N}$ ).



**Remarque. Test séquentiel et tests successifs.** On peut toujours construire un test d'élimination séquentiel  $T_{t_1 \dots t_N}^S(k)$ , constitué de  $N$  sous-tests  $T_{t_1} \dots T_{t_N}$  qui soit plus performant que l'application successive des  $N$  tests d'élimination  $T_{t_1} \dots T_{t_N}$  successivement.

L'objectif est d'améliorer les performances de notre test séquentiel précédent en prenant en compte plusieurs mesures de résidu quadratique normalisés  $\tilde{\varphi}_{t,\Sigma}^2(k)$  à différents instants. Ainsi, dans notre cas, au lieu d'éliminer une hypothèse  $k$  la première fois où  $\tilde{\varphi}_{t,\Sigma}^2(k)$  n'est pas dans l'intervalle de confiance  $I$ , nous définirons en lieu et place un test global. Ce test global est défini comme suit : à chaque instant  $t$  entre  $t_1$  et  $t_N$ , pour chaque hypothèse  $k$ , on réalise un test par intervalle dont la valeur du risque  $PND$  est maîtrisée. Pour chaque hypothèse  $k$ , on compte alors le nombre de réussites, et le nombre d'échecs, parmi ces  $N$  tests. Le résultat du test global consiste alors à éliminer l'hypothèse  $k$  si et seulement si le nombre d'échec est supérieur à un certain seuil  $F < N$ .

Nous allons nous intéresser aux propriétés statistiques de ce test global. Mais tout d'abord nous allons supposer que les  $N$  tests successifs sont indépendants. Si tel n'était pas le cas, on pourra augmenter l'intervalle de temps entre les tests pour rendre les  $N$  tests successifs quasiment indépendants.

**Proposition. Tests successifs (rendus) indépendants.** Dans le cas d'une navigation au DGPS-RTK, les mesures de phases peuvent être considérées comme corrélées (sur quelques secondes). Cette corrélation est inhérente à la technique de poursuite de phase utilisée par le récepteur. Cependant, on montre qu'en espaçant les tests au delà de l'intervalle de corrélation des mesures de phases, on rend chacun des tests  $T_{t_1} \dots T_{t_N}$  quasiment indépendants.

Notons  $PND^{global}$  le risque global de non-détection du test global, et  $PND^{interv}$  le risque individuel de non-détection de chaque test par intervalle, réalisé  $n$  fois aux instants  $t_1 \dots t_n$  sur une hypothèse d'ambiguïté  $k$ . Soit  $F$  le seuil de décision qui permet d'éliminer une hypothèse. Rappelons qu'on élimine définitivement une hypothèse si le nombre d'échec de chaque test est supérieur à  $F$ . Le test séquentiel proposé étant un test binomial, l'expression du risque d'échec du test global est la suivante

$$PND^{global}(N, F) = \sum_{F < j < N} \mathbb{C}_N^j (PND^{interv})^j (1 - PND^{interv})^{N-j}$$

avec  $\mathbb{C}_n^k$  le nombre de combinaisons de  $k$  objets discernables dans un ensemble de  $n$  objets discernables. Cette expression permet alors de dimensionner le test (définitions de  $N$ ,  $F$  et  $I$ ) à partir des exigences a priori de fiabilité de l'estimation (exigence sur  $PND^{global}$ ).

Or, après avoir appliqué ce test à un ensemble de cas d'étude, et bien que les résultats soient meilleurs qu'avec l'utilisation d'un test séquentiel, il s'avère que les résultats furent à nouveau insuffisants. Soupçonnant la présence de multi-trajets cause des défaillances de cet estimateur, nous avons étendu ce test, et construit un test robuste aux trajets multiples.

#### 4.4.3 Adaptation au cas de la navigation avec trajets multiples

Dans le cas de la navigation en milieux suburbain, les mesures de phases sont souvent entachées de bruits temporellement corrélés, qui ont pour origine la présence de trajets multiples ou réfléchis.

Ceci impacte notre test global de deux manières. D'abord nous avons vu qu'il était indispensable, dans notre modélisation, que les tests par intervalles restent indépendants. Ensuite nous avons vu l'importance de l'hypothèse que le vecteur de résidu suive une loi multi-normale de moyenne nulle.

Si nous pouvons contourner, comme nous l'avons vu dans la section précédente, le premier problème (celui de l'indépendance des tests par intervalle) en augmentant l'intervalle de temps entre nos tests par intervalle, le second problème (vecteur de résidu ne suivant pas une loi multi-normale de moyenne nulle) nécessite que nous nous y attachions un peu plus longtemps.

Nous savons que le vecteur de résidu, dans le cas de trajets multiples et réfléchis en milieux sub-urbain, doit être sous-échantillonné à quelques secondes pour assurer l'indépendance temporelle, n'est pas de moyenne nulle, et possède un moment d'ordre deux différent de sa covariance.

Il faudra donc reprendre le mode de fonctionnement de notre test par intervalle. En effet, pour garantir un risque  $PND$  pour notre test par intervalle en présence de trajets multiples et réfléchis, il va nous falloir déterminer une autre loi de distribution de la variable de résidu quadratique normalisé  $\tilde{\varphi}_{t,\Sigma}^2(k^*)$ . Or pour ce faire, et étant donné que les paramètres affectant cette distribution ne sont pas observables (puissance du trajet réfléchis et délai de celui-ci sur chaque mesure), nous avons procédé à la modélisation par une méthode empirique, c'est à dire en analysant un certain nombre de valeurs de résidus quadratiques normalisés, dans différentes situations, et à différents moments.

Nous avons pu observer deux choses. D'abord que, comme prévu, la distribution de  $\tilde{\varphi}_{t,\Sigma}^2(k^*)$  suit bien, toujours, une loi du  $\chi^2$  décentré à  $n - 3$  degrés de liberté. Et ensuite que les valeurs de ce décentrage, qui varient dans le temps, sont, dans tous les cas étudiés, bornées. Une explication plausible viendrai du fait que la valeur de la moyenne sur chaque valeur de résidu est elle-même bornée par la valeur de délai entre la mesure de phase du trajet direct et du trajet réfléchis. En effet, au delà de ce seuil (défini classiquement par le quart d'une longueur d'onde de la porteuse du signal de phase), la mesure est perdue. Cette propriété nous arrange, car elle permet de définir le pire cas de notre décentrage sur la distribution de  $\tilde{\varphi}_{t,\Sigma}^2(k^*)$ .

Ainsi, pour garantir un risque  $PND$  pour notre test par intervalle, il suffit de garantir ce risque dans le pire cas. Notons  $\tau_{MAX}(k)$  le décentrage théorique maximum (dans le pire cas). Notre intervalle de confiance  $I$  sera défini comme l'intervalle allant de zéro à la valeur  $g(k)$ , de telle sorte que

$$\int_0^{g(k)} \chi^2(x, \tau_{MAX}(k)) dx = 1 - PND.$$

avec  $\chi^2(x, \tau)$  la loi de distribution du  $\chi^2$  décentré de décentrage  $\tau$  évaluée en  $x$ . Ainsi, si nous définissons  $I$  de cette manière, nous garantissons que le risque  $PND$  réel est bien inférieur à la valeur théorique de ce risque, et ce, dans tous les cas de décentrage  $\tau(k) \in 0 \dots \tau_{MAX}(k)$ . Ceci est prouvé par le lemme suivant.

**Lemme. Critère de détection sur une loi du  $\chi^2$  à décentrage borné.** Soit  $\tau$  une valeur de décentrage d'une loi du  $\chi^2$  décentré. Soit  $\tau_{MAX}$  une valeur maximale pour  $\tau$ . On montre que pour toute valeur de risque  $R$  (une probabilité de non détection), que l'on se fixe en contrainte, il existe un seuil  $g(R)$  tel que

$$\forall g > g(R), \int_0^g \chi^2(x, \tau) dx > 1 - R$$

On montre aussi que

$$g(R) < G(R)$$

avec  $G(R)$  aisément définissable par

$$\int_0^{G(R)} \chi^2(x, \tau_{MAX}) dx = 1 - R$$

*Démonstration.*  $\forall G(R) > g(R)$ , et quelque-soit la valeur du décentrage  $\tau$  fixée, on a  $1 - R < \int_0^{g(R)} \chi^2(x, \tau) dx < \int_0^{G(R)} \chi^2(x, \tau) dx$ . D'autre part, on montre que pour deux valeurs de décentrage  $\tau_1$  et  $\tau_2 > \tau_1$ , on a  $\int_0^{g(R)} \chi^2(x, \tau_2) dx < \int_0^{G(R)} \chi^2(x, \tau_1) dx$ . On en conclut donc que  $1 - R < \int_0^{G(R)} \chi^2(x, \tau) dx < \int_0^{G(R)} \chi^2(x, \tau_{MAX}) dx$ .  $\square$

On a donc montré que, dans tous les cas de figure, le seuil  $G(R)$  déterminé par l'intégrale ci-dessus pour un décentrage borné par  $\tau_{MAX}$ , garantit un risque  $R$  de non-détection.

Nous avons appliqué cette nouvelle distribution empirique au test d'intervalle. Nous avons donc altéré la définition du test d'intervalle sans modifier la définition du test global. De cette manière seulement, les résultats de navigation au DGPS-RTK ont été satisfaisantes en milieu suburbain.

En effet, l'élimination des hypothèses fausses a été cohérente avec les observations que nous avons réalisées hors-ligne (ce que nous avons appelé l'apprentissage empirique). Le risque est maîtrisé. Nous avons pu donc proposer une solution de navigation au DGPS-RTK, en milieu suburbain, dont les risques liés à la levée d'ambiguïté, sont maîtrisés. Nous avons placé le risque de non-détection du test séquentiel à une valeur de  $1e - 9$ . Les résultats expérimentaux que nous avons obtenus ont été satisfaisants et conformes à ce risque donné en paramètre.

## 4.5 Résultats

L'algorithme présenté à été développé sur un prototype robotique d'environ 50cm x 50cm. Le terrain d'expérimentation est une zone gazonnée appartenant à une zone industrielle (figure 4.6). En figure 4.5, on peut voir le résultat d'une trajectoire réalisée durant l'été 2011. Le robot, rendu autonome par un pilotage au GPS-RTK, a réalisé plus de 24 heures de mission de navigation, ce qui représente environ 6 retours automatiques à sa station de rechargement. Le retour à la station est, en effet, une phase critique car la station est située au pied d'un bâtiment et les pôles de rechargement mesurent seulement quelques centimètres de largeur, ce qui met la précision et l'intégrité du positionnement, de la navigation au GPS-RTK, à l'épreuve. Les derniers essais ont été très concluants, avec un taux de succès de retour à la station de rechargement de 100%. A noter que la vérité terrain n'était pas disponible pour cette expérimentation.

## 4.6 Conclusion et perspectives

Les résultats en terme de navigation par GPS et GPS-RTK ont été très concluants. Les résultats sont très intéressants et nous avons pu proposer deux innovations en terme de navigation au GPS en mode différentiel local.

Cependant, l'utilisation d'un algorithme de filtrage (de type EKF) pose encore certains problèmes en terme de temps-réel dus à la latence des communications. Dans l'architecture utilisée pour nos expériences, les données issues du récepteur mobile sont transférées à la station de référence, qui calcule une position et une vitesse, qu'elle renvoie au robot mobile. Ces communications prennent du temps et le robot manque de réactivité. Pour cela, nous avons mis en place un système d'interpolation. Cependant, un capteur de type centrale inertielle serait le bienvenu pour gérer ces latences.

Nous avons ensuite souhaité étendre ces résultats, en intégrant un capteur optique à notre prototype. Nous nous sommes basés sur les travaux récents issus de la communauté du SLAM. Nous présenterons ces travaux dans le chapitre suivant.

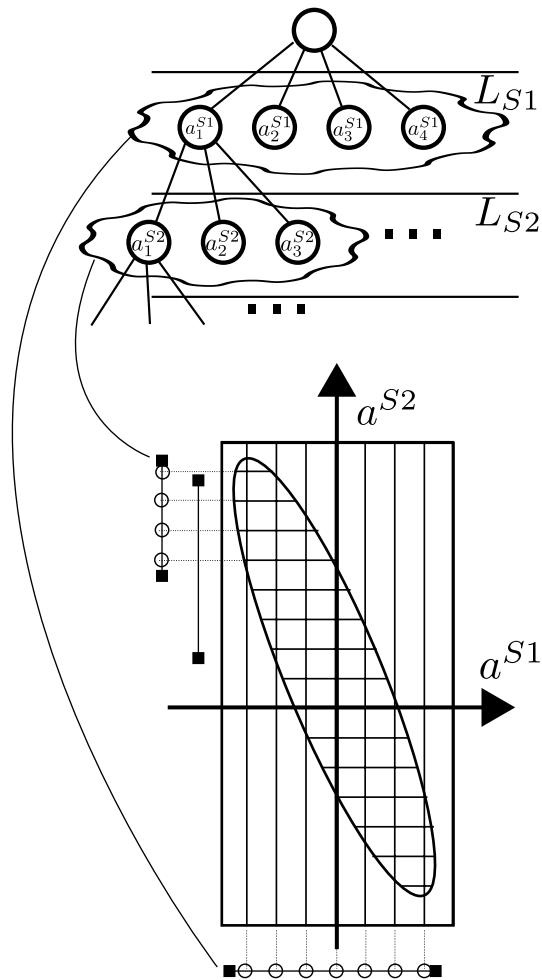


FIGURE 4.4 – La construction des hypothèses commence par la création d'intervalles (lignes à extrémités carrées). Ensuite on extrait les entiers contenus dans les intervalles pour peupler l'arbre.

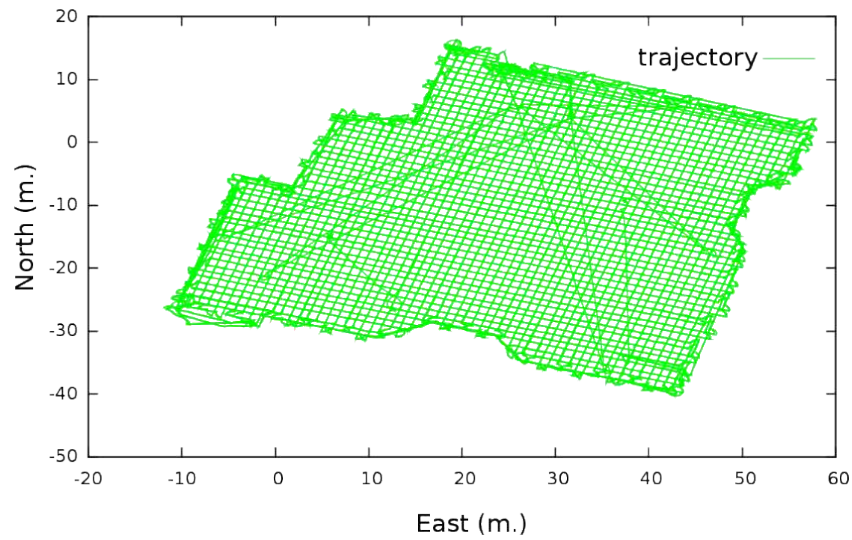


FIGURE 4.5 – La trajectoire de notre robot sur environ 24 heures. Il y a eu 6 retours automatiques à la station de rechargement entièrement par guidage GPS-RTK. Le rechargement se faisant avec une marge autorisée de quelques centimètres, le taux de réussite est de 100%.



FIGURE 4.6 – Le terrain d'expérimentation est un terrain sub-urbain. La zone de transit passe le long de certains bâtiments. Le robot mesure environ 50cm x 50cm.



## Chapitre 5

# Approche EKF-SLAM pour la navigation robotique

Dans ce chapitre, nous aborderons une série de travaux réalisés entre 2008 et 2012 dans le groupe de robotique du LAAS-CNRS. Ces travaux portent sur la localisation et cartographie simultanées, réalisé sur un montage de type caméra monoculaire/centrale Inertielle. L'objectif est alors que le logiciel de SLAM s'exécute à la fréquence image (soit à 30Hz pour une caméra USB, et 120Hz pour les modèles de caméra firewire disponibles chez le fabricant PointGrey). L'approche retenue est alors le filtrage stochastique non particulière (approche dire EKF-SLAM). Ces travaux ont permis de réaliser et distribuer deux logiciels complémentaires. Ces deux logiciels sont disponibles sous licences libres : le premier est nommé RT-Slam [87] (pour Real-Time SLAM), et le second C-Slam [88] (pour C-language SLAM).

### 5.1 RT-SLAM

#### 5.1.1 Objectifs

Dans cette section, nous aborderons l'implantation en C++ du logiciel de SLAM visuel RT-Slam. Ce logiciel est remarquable par la présence de deux objectifs : la généricité dans le développement, et la rapidité d'exécution. Le premier objectif (généricité) est résolu par l'utilisation d'un langage orienté objet (le C++) et en utilisant des fonctionnalités avancées introduites par ce langage de programmation (comme l'héritage de classes et l'utilisation des templates). Le second objectif (rapidité d'exécution) a nécessité l'introduction de plusieurs stratégies, dont les implantations ont été jusqu'alors, réalisées isolément dans diverses implémentations (algèbre des matrices symétriques, RANSAC ...).

Or la généricité permet la programmation facilitée (pour la recherche scientifique par exemple). Ainsi il est possible d'adjoindre de manière aisée de nouveaux modèles de capteurs, de nouveaux types d'amers, et enfin de nouvelles méthodologies de résolution du problème du SLAM (tel qu'active search, la reparamétrisation d'amers, one-point RANSAC, robot-centric SLAM ...).

Nous avons choisi de considérer, comme cas pratique, l'utilisation d'une caméra monoculaire et d'une centrale inertielle de type MTi, montées sur un socle commun, pour démontrer les per-



performances de notre application open-source. Nous avons alors réalisé un SLAM Visuel + IMU tournant à 120Hz (donc à fréquence image).

### 5.1.2 Introduction

Les progrès récents en traitement d'image, l'augmentation des puissances de calculs, et les avancées des procédés de SLAM ont rendu populaire, ces dernières années, le SLAM visuel en tant que procédé de localisation en robotique. On sait que l'approche filtrage mono-hypothèse (EKF) est nécessaire pour résoudre le problème du SLAM visuel en temps-réel. Pour cela nous avons débuté par l'implantation de cette méthode. En parallèle, l'arrivée de nouvelles méthodes d'optimisation globales (utilisables pour générer, a posteriori, des trajectoires complètes plus précises) est une solution que nous voyons comme complémentaire, et à laquelle le groupe de robotique du LAAS-CNRS attache une grande importance. À noter que l'étape de lissage est en cours de réalisation, dans le logiciel RT-Slam, au jour d'écriture de ce manuscrit.

### 5.1.3 Contexte

Le SLAM visuel a subi, depuis dix ans, de nombreuses évolutions. En commençant par (la liste est loin d'être complète) l'utilisation de filtres stochastique (EKF, filtrage particulaire, rao-blackwellisation), l'active search, les approches Divide & Conquer, le Robot-centric Slam, et enfin les méthodes d'initialisation, de paramétrisation et de reparamétrisation d'amers. Concevoir un logiciel de SLAM générique permet non seulement de tester et de comparer les performances des méthodes existantes, mais aussi de mixer différentes de ces méthodes entre elles, et aussi d'en développer de nouvelles.

### Architecture orientée objet

La première tâche dans un développement logiciel utilisant comme support un langage de programmation orienté-objet (ici le langage C++), est de construire un tableau des classes les plus abstraites, et de leur attribuer une sémantique (une description). Ces classes seront ensuite déclinées (dérivées, instanciées ...) dans des objets concrets, manipulés lors de l'exécution du programme (runtime). Voici la liste de ces classes abstraits.

#### Map

La classe "Map" ne contient pas seulement l'estimation de la carte de l'environnement (le M de SLAM). En effet, ici, l'objet "Map" est capable de contenir et de manipuler toute représentation d'un ensemble de variables gaussiennes (identifiée par un vecteur d'état  $X$  et une matrice de covariance  $P$ ). Les opérations de base sont l'ajout/suppression d'éléments dans  $X$  et  $P$ . Les autres opérations sont la reparamétrisation linéaire d'éléments du vecteur d'état (utilisé lors de la prédiction d'un filtre de Kalman, l'initialisation de nouveaux éléments dans un contexte de SLAM, ou encore la phase de reparamétrisation d'amers visuels). Les accès aux données (vecteur d'état et matrice de covariance) sont réalisés par indexage indirect, ce qui permet de gérer la lectures/écriture/insertion d'état comme un accès aux caractères dans un fichier sur tout périphérique de stockage en informatique. Les bibliothèques et structures sous-jacentes sont celles de Boost

et de `ublas`. Ce sont des bibliothèques de manipulation d'algèbre linéaire open-sources. Ces bibliothèques sont choisies parce qu'elles sont aujourd'hui reconnues, et en développement, par une communauté active de contributeurs.

## Robot

La classe "Robot" est un objet représenté par un ensemble de variables d'intérêt de type gaussien. Cet objet possède donc une référence dans la classe `Map`, que nous venons de voir. Mais la classe *Robot* doit aussi définir son modèle d'évolution (donc un modèle dynamique tel qu'un modèle à vitesse brownienne, accélération brownienne ... ou encore un modèle d'évolution dépendant des mesures d'un capteur proprioceptif telle qu'une centrale inertielle). Cette classe contient aussi des références vers un ensemble de capteurs qui lui sont propres (de la classe "Sensor" que l'on détaillera). Nous avons choisi ici de distinguer d'une part le type "Robot" (une plate-forme et son modèle d'évolution) et d'autre part le type "Sensor" (un capteur placé sur une plate-forme et capable d'acquérir et de transmettre des mesures), pour une raison précise. En effet, lorsque le robot, muni de capteurs, se déplace, l'ensemble des capteurs de ce même robot se déplacent de manière liée. Donc la distinction entre un "Robot" et un "Sensor" est qu'un "Sensor" est lié de manière fixe au repère d'un robot. La conséquence principale est qu'on ne peut pas définir, dans RT-Slam, une caméra tenue à la main comme un simple "Sensor" (sans objet "Robot" associé). En effet, il faudra définir un robot muni d'un modèle dynamique (accélération brownienne par exemple) contenant une caméra.

## Sensor

Comme nous l'avons vu, un "Sensor" représente un périphérique de type *capteur* (tel une caméra, un laser 2D ...) lié, nécessairement, au repère d'un "Robot". La position du "Sensor" dans son "Robot" n'est pas nécessairement fixée au démarrage de l'algorithme. En effet, il est possible d'estimer, dans le processus de filtrage, la position du capteur dans le repère du robot. Dans tous les cas, un "Sensor" possède un vecteur d'état (potentiellement nul) dans la "Map" : on peut citer les paramètres intrinsèques d'une caméra, les biais d'une centrale inertielle, la position de ce capteur dans le repère du robot ...

## Landmark

Les amers (classe "landmark") possèdent donc un état dans la "Map". Ils peuvent être de différents types (points, ligne, plan ...) et, pour un même type, de différentes paramétrisations (Euclidien ou point homogène pour un point '). Une paramétrisation d'un type peut varier dans le temps (le processus s'appelle une reparamétrisation). Enfin, un "landmark" possède un descripteur, qui permet d'associer deux observations comme provenant du même amer. Pour les points dans les images, on peut citer des descripteurs comme des patches, des histogrammes, des descripteurs vectoriels (SURF, SIFT, BRIEF ...) et d'autres. Dans un sens, le descripteur d'un "landmark" est la partie de l'état qui n'est pas représentée dans "Map".

## Observation

La classe "Observation" est né du concept de rencontre entre un "Sensor", et un "landmark". Dans RT-slam, il y a autant d'objet "Observation" que de combinaisons entre un "Sensor" et un "landmark". Ainsi, l'objet "Observation" ne correspond pas à une observation réalisée, mais à une observation possible. Ainsi, c'est cet objet qui est le cœur de la généricité du logiciel RT-Slam. En effet, un type d'observation dépend de trois paramètres, le modèle de capteur, le modèle d'amer, et le modèle d'observation. C'est donc cette classe qui sert au développement de nouveaux modèles de capteurs et d'amers dans RT-Slam. Ces modèles peuvent donc être évalués et comparés.

## Managers

Les classes "DataManager" et "MapManager" sont des gestionnaires de données de SLAM.

La classe "MapManager" définit la stratégie de gestion de la "Map". Ceci inclut la suppression d'amers si ils ne sont plus observés depuis un moment, ou si la cartographie atteint une trop grande taille. Le "MapManager" peut aussi réaliser l'archivage de l'environnement (multi-map) ou d'autres stratégies comme l'odométrie visuelle.

La classe "DataManager" récupère les données capteur et les datent. Elle opère sur les données capteurs pour détecter les observations des amers dans ces données dites brutes (raw-data), et le renseigne dans l'objet "Observation" correspondant. Ainsi l'objet "DataManager" peut appliquer des stratégies de détection actives d'amer dans les images (appelé active search) mais on peut aussi améliorer ce processus en fonction d'une stratégie de type multimap, ou d'information mutuelle, ou de détection aléatoire ou autre. Il est possible de comparer les résultats facilement.

## Conclusion sur l'architecture objet

La conception objet RT-Slam (résumés en figures 5.1 et 5.2) permet de définir une généricité indispensable au développement et à l'évaluation de nouvelles méthodes de SLAM. En premier lieu, et grâce à cet outil, nous avons mis en place une fusion de deux capteurs : une caméra monoculaire tournant à 120Hz, et une centrale inertielle de type MTi. Ce SLAM tourne à fréquence image avec un montage tenu à la main. Nous allons détailler nos résultats.

### 5.1.4 SLAM multi-capteur caméra/IMU temps-réel

Nous allons débiter par décrire la stratégie au niveau capteur (traitement d'image), puis nous aborderons le modèle du robot et des scapteurs, avant de décrire notre paramétrisation d'amers, et enfin nous présenterons nos résultats.

## Traitement d'image

L'observation d'un point en 3 dimensions (landmark), dans une image vidéo, correspond à la projection d'un point 3D, dans un plan en 2 dimensions. Ainsi, tous les points de l'image correspondent à un point unique dans l'espace tridimensionnel, mais tous les points 3D ne sont pas projetés dans toutes les images.

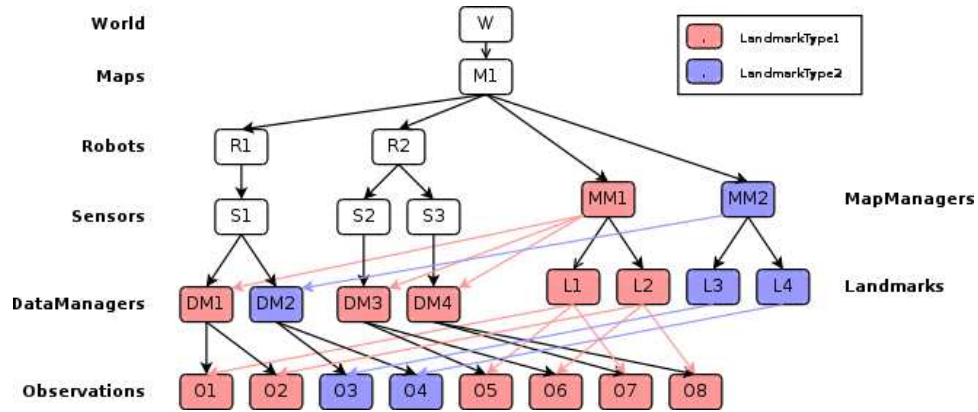


FIGURE 5.1 – Cartographie des objets abstraits dans RT-Slam. Ici on voit 2 robots, le premier contient un capteur ( $S1$ ), et le second deux capteurs ( $S2$  et  $S3$ ). La carte est constituée de quatre amers ( $L1 \dots L4$ ). On voit aussi, clairement, les positions des Managers et des Observations.

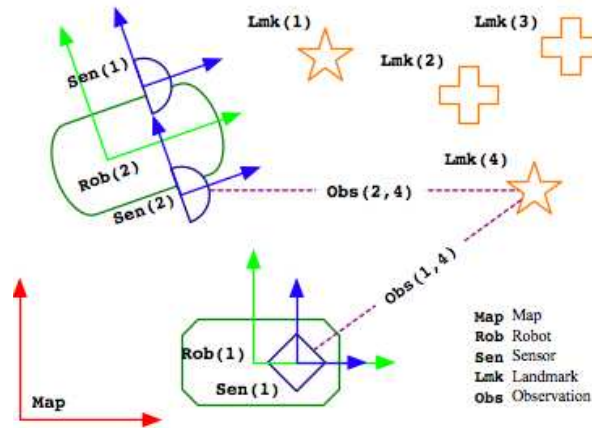


FIGURE 5.2 – Autre représentation de la cartographie des objets abstraits dans RT-Slam présenté en figure 5.1. On voit 2 robots, le premier contient un capteur ( $S1$ ), et le second deux capteurs ( $S2$  et  $S3$ ). La carte est constituée de quatre amers ( $L1 \dots L4$ ). On voit aussi, clairement, les positions des Managers et des Observations.

Pour réaliser du SLAM visuel de type *Landmark-SLAM*, il faut détecter le même point 3D, dans plusieurs images. Pour faciliter cette détection, il est nécessaire de choisir des points d'intérêts dans les images possédant certaines propriétés dites d'associabilité.

### Extraction d'un point

Après avoir comparé expérimentalement différentes méthodes, les meilleures performances furent atteintes avec une extraction de points dans l'image basée sur le détecteur de Harris incluant quelques améliorations. Ainsi pour garantir une exécution à fréquence image, la fenêtre gaussienne de l'extracteur de Harris est remplacée par une fenêtre uniforme.

L'utilisation d'une technique de *tessellation* (c'est-à-dire la segmentation de l'image en sous-zones) permet de lancer la détection d'un nouveau point d'intérêt uniquement dans une sous-zone sans points. L'avantage immédiat est l'évitement de l'étape de seuillage, dans la phase d'extraction du nouveau point d'intérêt de notre l'image.

### Appariement : *Active search*, changement de repère, et *One point Ransac*.

Notons qu'un point en 3D possède une coordonnée estimée dans "Map", et par une imagerie (souvent appelée 'patch') qui contient une image de la zone autour du point au moment de l'extraction du point d'intérêt. Le patch est souvent de taille réduite et carré (allant de 7 à 33 pixels de côté).

Notre objectif est de mettre en correspondance un amer, présent dans "Map", et un point de l'image courante. Nous avons pu mettre en avant, expérimentalement, l'efficacité d'un algorithme basé sur ZNCC incluant trois améliorations : la recherche active (*active search*), un changement de repère de l'imagerie, et la mise en place de l'algorithme *one point Ransac*.

**Active search :** La recherche active consiste à projeter dans l'espace image le point estimé dans "Map", puis de déterminer une zone de recherche de correspondance dans un périmètre autour de la projection théorique. Ce périmètre correspond à l'incertitude associée à la projection (qui prend en compte l'incertitude sur la position du robot, de la position de l'amer en trois dimensions, et du bruit du capteur).

**Changement de repère de l'imagerie :** La seconde amélioration consiste à chercher une correspondance, non pas entre un patch récupéré lors de l'extraction du point et l'image courante, mais bien d'une déformation du patch récupéré lors de l'extraction et l'image courante. La déformation est une transformation affine correspondante à la différence de point de vue entre l'estimation de la position de la caméra lors de l'extraction du patch, et l'estimation de la position courante de notre caméra. On considère alors que le patch est l'observation du contour du point en trois dimensions, et que ce contour est un plan perpendiculaire à la direction de la projection lors de l'extraction.

**One point Ransac :** Dans le but de rendre robuste l'algorithme de localisation et cartographie simultané, nous avons mis en place l'algorithme de *One point Ransac*. Cet algorithme consiste à 1-*Mémoriser* l'état de notre estimateur avant d'appliquer le résultat d'un appariement, et 2-*Annuler* l'opération de mise à jour si l'on détecte une inconsistance. Cette inconsistance est détectée si plusieurs appariements ne trouvent pas leurs correspondances.

### Modèle du robot

Nous avons choisi, dans le cas de l'hybridation caméra/IMU, d'utiliser un modèle dynamique de robot avec commande. La valeur de commande, dans notre cas, est la mesure fournie par la centrale inertielle.

### Mesures de la centrale inertielle

La centrale inertielle fournit des mesures d'accélération linéaires et vitesses angulaires, qui servent de commande. Le modèle d'observation est :

$$u_t = [am_t, wm_t]^T \quad (5.1)$$

$$am_t = R_t^T \cdot (a_t + g_t) + ab_t + \epsilon_t^a \quad (5.2)$$

$$wm_t = R_t^T \cdot (w_t) + wb_t + \epsilon_t^w \quad (5.3)$$

avec  $u_t$  la commande.  $am_t$  et  $wm_t$  sont respectivement l'accélération (en trois dimensions) et la vitesse angulaire (en trois dimensions) mesurées par la centrale inertielle.  $g$  est le vecteur de gravitation,  $a_t$  l'accélération de la centrale dans le repère global,  $w_t$  la vitesse angulaire de la centrale dans le repère global,  $R_t$  la matrice de rotation définissant l'orientation de la centrale inertielle dans le repère global.  $ab_t$  et  $wb_t$  les biais du capteur,  $\epsilon_t^a$  et  $\epsilon_t^w$  sont les bruits thermiques additionnels, considérés comme blancs et gaussiens.

### Modèle d'évolution du robot

Notons  $\mathbf{R}$  l'état du robot. Cet état contiens la position  $p$ , l'orientation  $q$ , la vitesse linéaire  $v$ , mais aussi les biais de la centrale inertielle ( $ab_t$  et  $wb_t$ , que nous estimerons) et le vecteur de gravitation  $g$ . On a :

$$\mathbf{R}_t = [p_t, q_t, v_t, ab_t, wb_t, g_t]^T \quad (5.4)$$

Le modèle dynamique (non-linéaire) s'écrit alors :

$$\mathbf{R}_{t+1} = f(\mathbf{R}_t, u_t) + \epsilon_t \quad (5.5)$$

avec  $f$  la fonction d'évolution d'état,  $u_t$  la commande, et  $\epsilon_t$  un bruit additionnel considéré comme blanc et gaussien. On a :

$$p_{t+1} = p_t + v_t \cdot \Delta_t \quad (5.6)$$

$$v_{t+1} = v_t + R(q_t) \cdot ((am_t - ab_t) - g_t) \cdot \Delta_t + \epsilon_t^v \quad (5.7)$$

$$q_{t+1} = q_t \otimes q((wm_t - wb_t) \cdot \Delta_t) + \epsilon_t^q \quad (5.8)$$

$$ab_{t+1} = ab_t + \epsilon_t^a \quad (5.9)$$

$$wb_{t+1} = wb_t + \epsilon_t^w \quad (5.10)$$

$$g_{t+1} = g_t \quad (5.11)$$

avec  $\Delta_t$  le temps écoulé entre les indexes temporels  $t$  et  $t + 1$ ,  $R(q)$  la matrice de rotation correspondant au quaternion  $q$ ,  $\otimes$  l'opérateur de composition de quaternion, et  $\epsilon_t^v$ ,  $\epsilon_t^q$ ,  $\epsilon_t^a$ , et  $\epsilon_t^w$  les bruits de modèles considérés comme blancs et gaussiens.

### Modèle de mesure de la caméra

Notre caméra est placée sur une plate-forme rigide. Cette plate-forme permet de rendre rigide la transformation entre la centrale inertielle et cette caméra. Ainsi notre objet abstrait de type "Robot" est modélisé comme l'état de cette plate-forme.

Notons  $\mathbf{F}_C^W$  la position et l'orientation de la caméra dans le repère global. Notons  $\mathbf{F}_R^W$  et  $\mathbf{F}_C^R$  respectivement la position et l'orientation du robot dans le repère global, et de la caméra dans le repère du robot. On a naturellement :

$$\mathbf{F}_C^W = \mathbf{F}_R^W \oplus \mathbf{F}_C^R \quad (5.12)$$

$$\mathbf{F}_C^R = \mathbf{F}_C^W \ominus \mathbf{F}_R^W \quad (5.13)$$

avec  $\oplus$  l'opérateur de composition de repère et  $\ominus$  l'opérateur de décomposition de repère.

Soit  $F_L^W$  et  $F_L^C$  la position d'un amer dans le repère global et le repère de la caméra. De la même manière, on a :

$$\mathbf{F}_L^W = \mathbf{F}_R^W \oplus \mathbf{F}_C^R \oplus \mathbf{F}_L^C \quad (5.14)$$

$$\mathbf{F}_L^C = \mathbf{F}_L^W \ominus \mathbf{F}_R^W \quad (5.15)$$

On considère que notre caméra mesure une projection de points en trois dimensions dans une image, en deux dimensions. On a :

$$u = \text{proj}(\mathbf{F}_L^W, \mathbf{F}_C^W) + \epsilon^u \quad (5.16)$$

$$= \text{proj}(\mathbf{F}_R^W \oplus \mathbf{F}_C^R) + \epsilon^u \quad (5.17)$$

avec  $\epsilon^u$  un bruit de mesure, considéré comme blanc et gaussien.

Finalement la projection est paramétrisée par les paramètres intrinsèques  $k$  de notre caméra. On a :

$$u = \text{proj}(\mathbf{F}_L^W, \mathbf{F}_C^W, k) + \epsilon^u \quad (5.18)$$

$$= \text{proj}(\mathbf{F}_R^W \oplus \mathbf{F}_C^R, k) + \epsilon^u \quad (5.19)$$

### Paramétrisation des points

Nous avons donc d'abord utilisé la paramétrisation IDP (Inverse Depth Parametrization, de 6 paramètres), avant de proposer une paramétrisation nommée AHP (Anchored Homogeneous Point, de 7 paramètres). Les fonctions de passage d'une coordonnée euclidienne  $l_{euc}$ , avec une ancre euclidienne  $a_{euc}$ , d'un point de paramétrisation IDP ( $l_{idp}$ ) ou AHP ( $l_{ahp}$ ) sont :

$$l_{idp} = [a_{euc}, \theta, \psi, \rho] \quad (5.20)$$

$$l_{euc} = a_{euc} + \rho^{-1} R(\theta, \psi) \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (5.21)$$

$$l_{ahp} = [a_{euc}, u_{euc}, \rho] \quad (5.22)$$

$$l_{euc} = a_{euc} + \rho^{-1} u_{euc} \quad (5.23)$$

avec  $\theta$  et  $\psi$  les angles de tangage et lacet définissant la direction de l'amer en fonction de l'ancre,  $u_{euc}$  le vecteur directionnel de l'amer en fonction de l'ancre.

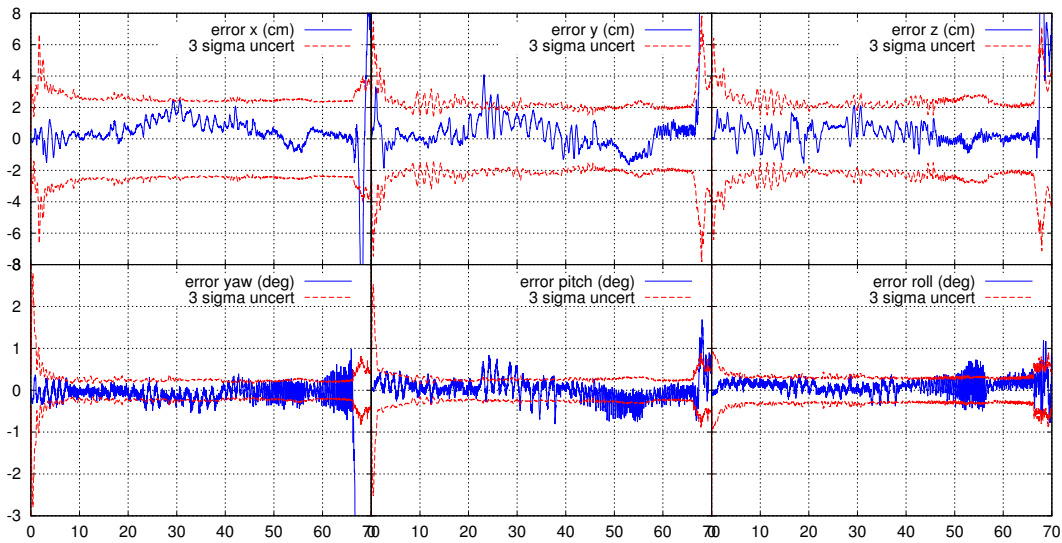


FIGURE 5.3 – Erreurs en positions et orientations à forte dynamique. La vérité terrain est assurée par un système de capture de mouvements. Les vitesses angulaires excèdent  $400 \text{ deg.sec}^{-1}$  et les accélérations angulaires excèdent  $3000 \text{ deg.sec}^{-2}$ . L'axe horizontal est défini en secondes.

## Résultats

Un ensemble de marqueurs de motion-capture sont placés sur la plate-forme rigide. Ce qui permet de déterminer une vérité terrain. Des expérimentations furent réalisées en 2011, et les résultats furent satisfaisant puisque notre implémentation de SLAM tourne à fréquence image, qu'il ne diverge pas, et que l'erreur réelle correspond à l'estimation théorique du modèle d'erreur.

La figure 5.3 présente l'erreur dans un contexte de dynamique forte. En effet, les vitesses angulaires excèdent  $400 \text{ deg.sec}^{-1}$  et les accélérations angulaires excèdent  $3 \text{ deg.sec}^{-2}$ . On voit que l'incertitude théorique est cohérente avec l'erreur réelle. L'utilisation d'une centrale inertielle permet d'observer le facteur d'échelle, RANSAC permet de rendre robuste notre SLAM, et les optimisations en traitement d'images permettent de garantir une exécution à fréquence image.

## 5.2 C-SLAM

### 5.2.1 Objectifs

Le logiciel C-SLAM (pour C-language SLAM) est une implémentation en ANSI-C d'un logiciel EKF-SLAM dédié aux applications critiques (typiquement pour l'aéronautique). Il a pour but de fournir une base pour le développement d'applications de SLAM embarquées (donc il n'y a pas de dépendances externes ni pour les opérations d'algèbre, ni pour le traitement d'image). Les sources sont disponibles à l'adresse <http://c-slam-laas.googlecode.com/svn/trunk/>.

Les résultats furent embarqués sur trois types de cartes qui sont présentées dans le tableau 5.1.



Carte	Modèle	Fréquence	Mémoire
VIA EPIA	EK8000EG	800MHz	1Go RAM
ARM9	926-EJ-S	400Mhz	64MB RAM
Xilinx	Virtex 5	400MHz	256kB RAM

TABLE 5.1 – Liste des architectures sur lesquelles le logiciel C-SLAM a été embarqué.

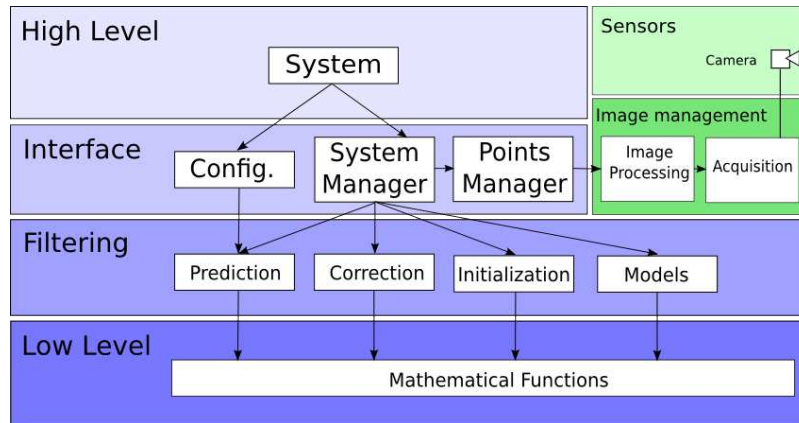


FIGURE 5.4 – L'architecture du logiciel C-Slam. Il a été choisis ici de séparer clairement les traitements d'acquisition et de traitement d'image, de la partie algébrique, ceci afin de permettre l'utilisation de cartes hybrides CPU/GPU ou d'autres cartes embarquées sur des plates-formes robotiques.

### 5.2.2 Architecture

La clé de l'architecture du logiciel C-Slam est la séparation en deux classes d'opérations. D'une part les opérations d'acquisition et de pré-traitement d'images (qui peuvent être externalisée sur une unité de calcul annexe), et l'algèbre d'autre part. Les opérations matricielles sont elles-mêmes hiérarchisées, de telle sorte qu'elle puissent aussi, en partie, tourner sur une unité dédiée. La figure 5.4 présente cette architecture.

### 5.2.3 Caractéristiques

C-Slam étant dédié aux applications critiques, une attention particulière a été donnée à la gestion de la mémoire. Ainsi il n'y a pas d'utilisation du *tas* (donc aucun appel à la fonction système *malloc()*). Il est possible de paramétrer (à l'aide d'une macro) le type des valeurs scalaires réelles à virgule flottante (double ou float), et enfin, partout où cela est possible, l'utilisation des structures de type *union* est privilégiée.

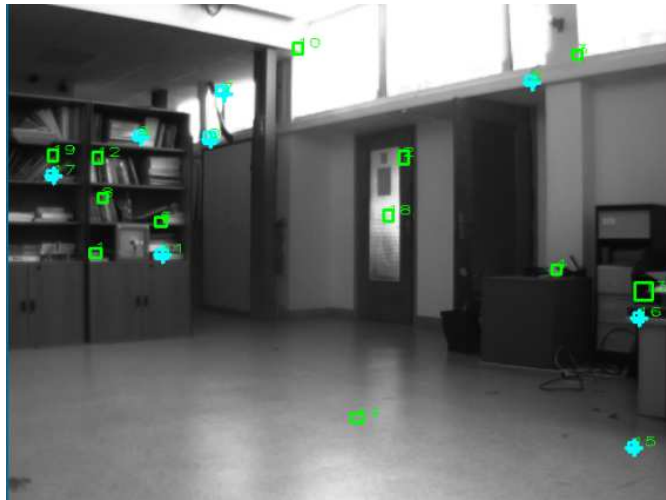


FIGURE 5.5 – Image prise d’une séquence utilisée pour mettre en place l’évaluation de C-Slam.

#### 5.2.4 Comparaison avec RT-Slam

Dans cette implantation de SLAM, l’accès à la généricité est moindre que dans RT-Slam. En effet, nous avons privilégié la clarté du code source (en C), et donc nous n’avons pas mis en place le système de ‘Manager’ tel que défini dans RT-Slam. Cependant, le fait de limiter l’abstraction, a permis, dans un sens, de clarifier les algorithmes sous-jacents.

Ainsi l’algorithme de One-Point-Ransac est explicitement défini dans C-Slam (alors qu’il fait appel à plusieurs managers dans RT-Slam), en contrepartie, l’implantation d’une éventuelle nouvelle méthode, remplaçant ce One-Point-ransac serait difficile à mettre en place dans C-Slam.

Les choix des algorithmes dans C-Slam et les modèles d’observation et de dynamique du robot sont les mêmes dans RT-Slam et C-Slam, l’objectif étant, dans les deux cas, de concevoir une expérimentation en hybridation Caméra/IMU.

#### 5.2.5 Résultats pour l’hybridation Caméra/IMU

Les résultats principaux portent sur les fréquences de mise à jour de vecteur d’état. Pour garantir un SLAM qui ne diverge pas, il faut garantir une fréquence d’au moins 30Hz, voire au delà pour des dynamiques fortes.

Pour cela, nous avons réalisé des acquisitions (figure 5.5), et, à partir de ces séquences, procédé à l’évaluation de deux manières.

D’abord nous avons fait varier le nombre de corrections par image (chaque image possède une résolution de 640x480 pixels), et le nombre d’amers dans la carte, et nous avons observé l’effet sur la fréquence image. Les résultats sont visibles sur les figures 5.6, 5.7 et 5.8. On y voit que le nombre de corrections influe sur la fréquence image. On voit aussi que les meilleurs résultats proviennent de l’implantation sur la carte Xilinx Virtex 5.

Ensuite nous avons comparé les différentes cartes entre elles. La fréquence de mise à jour

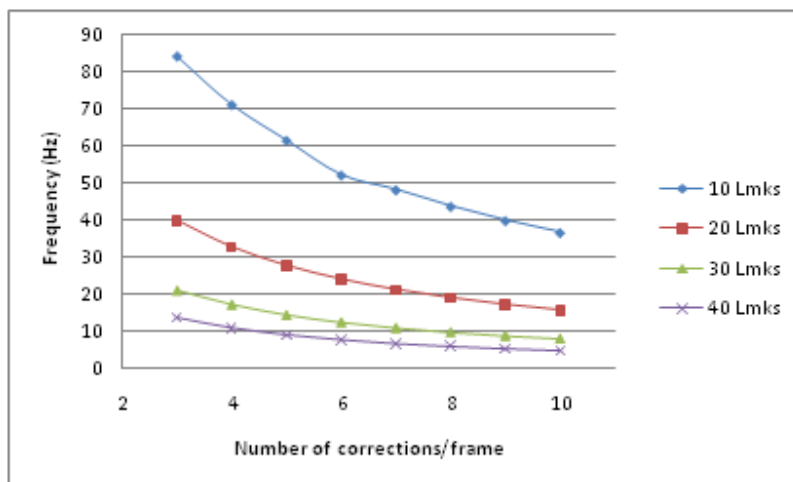


FIGURE 5.6 – Performances de C-Slam sur une carte VIA EPIA. On voit que le nombre de corrections influe sur la fréquence de mise à jour de l'état. Ainsi pour atteindre 20 images par secondes, il faudrait se limiter à 4 corrections par image, et 20 amers dans la carte.

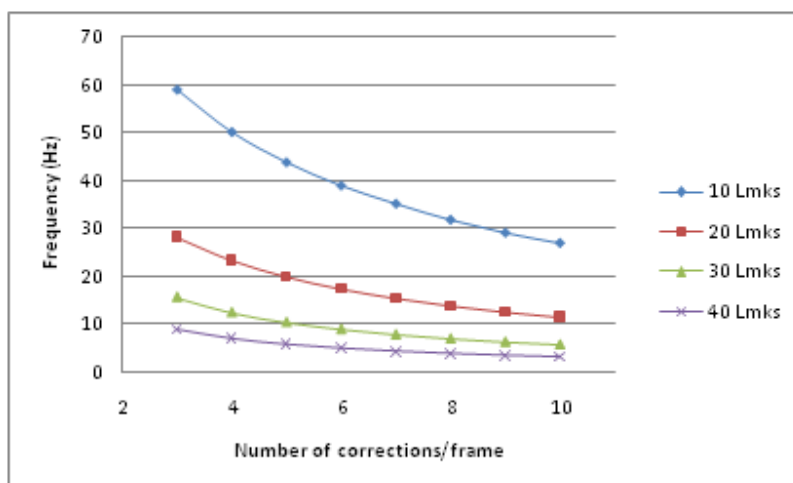


FIGURE 5.7 – Performances de C-Slam sur une carte ARM-9. On voit que pour atteindre 20 images par secondes, il faudrait se limiter à 5 corrections par image, et 20 amers dans la carte.

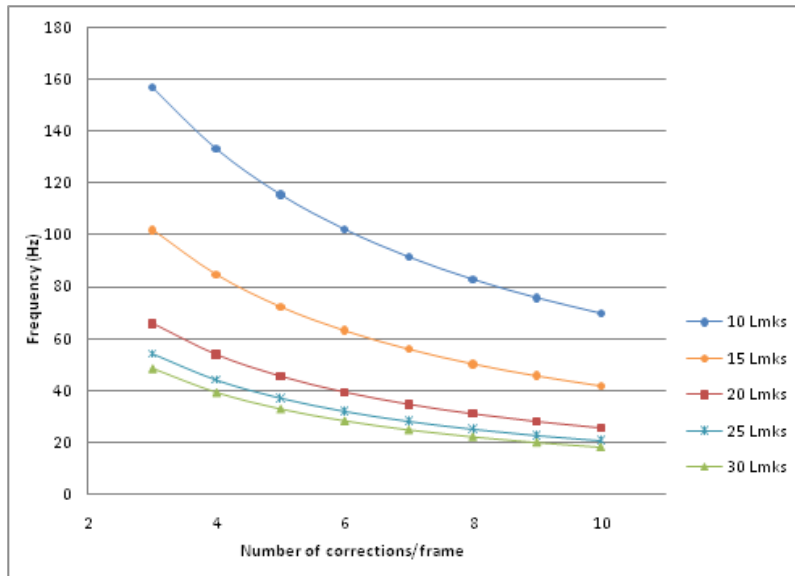


FIGURE 5.8 – Performances de C-Slam sur une carte Power-PC embarqué sur un Xilinx Virtex 5. On voit que pour atteindre 20 images par secondes, on peut monter quasiment à 10 corrections par image, sur une carte de 30 amers.

est mesurée pour chaque architecture avec, en figure 5.9, trois amers corrigés par image, pour différentes tailles de carte, et, en figure 5.10, avec vingt amers dans la carte, en faisant varier le nombre de corrections par image. Encore une fois, les performances de C-Slam sur Power-PC supplantent celles des autres architectures.

### 5.2.6 Conclusion sur C-Slam et les architectures embarquées

Nous avons donc conçu un logiciel destiné à être embarqué sur différentes architectures. Nous avons réalisé des tests en milieu ouvert en hybridation Vision/IMU. Les algorithmes sont sensiblement les mêmes que pour le logiciel RT-Slam. Cependant, l'architecture et la gestion des objets abstraits est différente.

Ce travail nous a permis d'évaluer différents scénarios de réalisation de SLAM en temps-réel, sur systèmes embarqués à faible capacités de calculs. Le travail est aujourd'hui poursuivi par de nouvelles équipes. Il nous a aussi permis de voir qu'il est recommandé d'embarquer un logiciel de type EKF-SLAM sur une carte de type Xilinx, comparativement à une carte contenant une unité de calcul standard, même de fréquence supérieure.

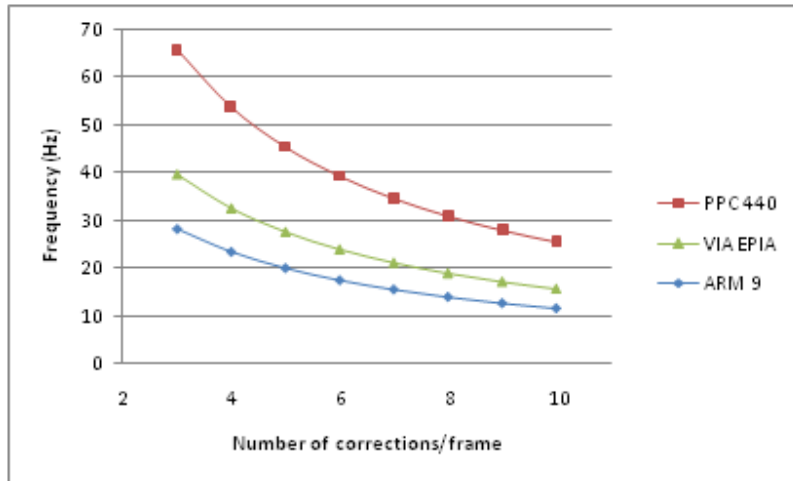


FIGURE 5.9 – Performances de C-Slam, comparées sur différentes architectures, en fonction de la taille de la carte de SLAM. Ici le nombre de correction par image est fixé à trois.

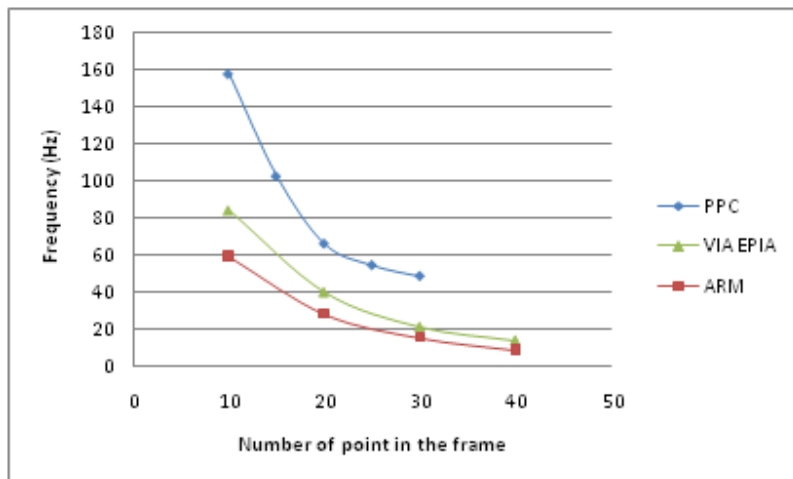


FIGURE 5.10 – Performances de C-Slam, comparées sur différentes architectures, en fonction du nombre de corrections par image. La taille de la carte SLAM est fixée à vingt.

# Chapitre 6

## Conclusion

Dans cette thèse, ont été présentées trois méthodes de navigation robotiques.

**Navigation aux triple différences de phases GPS** D’abord une navigation robotique au décimètre, utilisable en milieu sub-urbain. Cette méthode est une extension de la triple différence originale, initialement présentée dans [20].

**Navigation au DGPS-RTK** Ensuite une navigation au DGPS-L1-RTK. Dans ce contexte, on montre qu’un récepteur du commerce mono-fréquence est capable de fournir, en différentiel local, un positionnement centimétrique. Cette précision est rendue possible grâce à une approche statistique rigoureuse. Des essais ont été réalisés sur robot rondier, et sur des machines agricoles.

**SLAM Visuel** Enfin nous avons réalisés une navigation visuelle de type EKF-SLAM, dont l’implantation permet un exécution du SLAM à fréquence image. Pour éviter les phénomènes d’inconsistance et de divergences, certaines procédures (One-point RANSAC, tessellation ...) ont été mises en place. Nos deux implantations EKF-SLAM sont livrées sous licence Open-source, et elles sont destinées à deux types d’applications : la première est destinée à la recherche en robotique, et la seconde à l’utilisation dans des architecture embarquées et à l’aéronautique.

### 6.1 Navigation aux triples-différences de phase GPS

Dans le chapitre 3, nous présentons une approche de navigation au GPS différentiel mono-fréquence sans levée d’ambiguïté de phase. En utilisant un modèle de mesure de type *triple-différence* de phase.

Nous avons décrit les modèles d’évolution de notre plate-forme robotisée, et nous avons observé qu’il s’agissait d’un modèle markovien. Ensuite nous avons décrit le modèle de mesure de phase GPS en mode différentiel local, puis différentiel temporel (ce que l’on nomme triple-différence). Enfin nous avons pu créer un observateur optimal.

Cet observateur optimal apparaît sous la forme d’un filtre de Kalman, filtre qui prend exactement en compte les corrélations de bruits de mesures de phase GPS. En effet, bien que les modèles de bruits de mesure de phases GPS non différenciés soient considérés comme indépendants, ce

n'est pas de le cas pour les mesures différentielles. Les bruits de mesures de triple-différences de phases étant interdépendantes d'une mesure à l'autre, et d'un instant à l'autre, nous avons pris soin de modéliser ces corrélations et de les inclure de manière rigoureuse dans le processus de filtrage optimal.

Le résultat est une navigation au DGPS mono-fréquence précis au décimètre. Les tests réalisés nous permettent d'être confiants sur le potentiel, dans l'industrie, de cette approche.

Ces travaux ont été publiés en 2011 [22].

## 6.2 Navigation aux DGPS-RTK mono-fréquence

Dans le chapitre 4, nous présentons une autre approche pour la navigation au DGPS : le mode RTK<sup>1</sup>, dans lequel les ambiguïtés de phases (connues pour être un facteur de longueur d'onde), sont estimées. Le gain en terme de navigation est conséquent, puisqu'il permet d'atteindre une précision centimétrique en mode différentiel local.

Nous avons décrit la méthode de lever d'ambiguïté la plus utilisée aujourd'hui (la méthode LAMBDA). Nous avons mis en avant ses caractéristiques essentielles, et, en particulier, ses points faibles. Puis nous avons présenté notre approche. Cette approche repose sur une construction arborescente du jeu d'hypothèse, et sur l'élimination des hypothèses qui sont jugées fausses, grâce à un test séquentiel.

A chaque étape de la méthode présentée, nous avons décrit l'approche optimale, et les alternatives non-optimales qui permettent d'atteindre des performances temps-réel.

Le résultat est une navigation au DGPS-RTK mono-fréquence précis au centimètre. Les tests réalisés nous permettent d'être confiants sur son potentiel dans l'industrie.

Ces travaux ont été publiés en 2011 [85].

## 6.3 SLAM visuel

Dans le chapitre 5, nous présentons deux implantations de SLAM visuel. Ces deux implantations possèdent des avantages de par : la modularité du premier (destiné à la recherche en robotique), et le minimalisme du second (destiné aux application embarquées de faible puissance).

Ces deux implantations servent de moyens de recherche et d'expérimentation, au LAAS-CNRS, dans le groupe de robotique. Elles sont également utilisées dans plusieurs projets au niveau mondial.

En plus de décrire les avantages architecturaux (modularité, bibliothèques génériques ...) de ces implantations, nous avons pris soin de développer un cas d'usage simple et dans l'état de l'art. Ainsi, nous avons développé un SLAM Visuel/inertiel à fréquence image (120Hz), tolérant aux mauvais appariements (dans une certaine mesure), et résistant aux fortes dynamiques linéaire et angulaires (résistant aux accélérations de plus de 3g).

Ces travaux ont été publiés en 2012 [88, 87].

---

1. RTK : Real Time Kinematic

## 6.4 Bilan et perspectives

Le bilan concernant l'exploitation des capteurs GPS, et de la vision monoculaire, destinés à la navigation robotique est très positif. En effet, ce travail permet de mettre en avant deux faits complémentaires :

- L'analyse fine des modèles de mesure GPS, et des bruits afférents, est un élément clef de la réussite de la navigation à la fois précise, et bas-coût.
- L'implantation d'une navigation de type SLAM, utilisant une caméra-vidéo, est possible en temps-réel.

Nous travaillons aujourd'hui à deux améliorations :

**Introduction de mesures GPS brutes dans le système de SLAM** En partant des implantations de SLAM visuel, il est possible d'ajouter de nouveaux capteurs. En particulier il serait intéressant d'introduire les données GPS brutes.

**Méthodes de lissage dans le système de SLAM** Le LAAS-CNRS travaille actuellement sur l'introduction d'une méthode de lissage dans le logiciel de SLAM présenté dans le chapitre 5. Les performances seraient accrues, et il serait théoriquement possible de fermer les boucles de plus grande taille, en améliorant la consistance du résultat.





# Appendices



# Annexe A

## Modèles de bruits Gaussiens

### A.1 Bruit Gaussien et loi normale

Une variable aléatoire  $x$  est distribuée selon une loi Normale de moyenne  $\bar{x}$  et de variance  $\sigma^2$  si sa densité est continûment distribuée sur l'intervalle  $[-\infty; +\infty]$  et que sa fonction de distribution suit le modèle

$$\begin{aligned} p(x) &= \Gamma(x - \bar{x}; \sigma^2) \\ &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x - \bar{x}}{\sigma}\right)^2\right) \end{aligned}$$

Dans le cas d'un vecteur de variables aléatoires  $X$ , la distribution est appelée multi-Normale est est définie par

$$\begin{aligned} p(X) &= \Gamma(X - \bar{X}; P) \\ &= \frac{1}{\sqrt{(2\pi)^n |P|}} \exp\left(-\frac{1}{2} (X - \bar{X})^T P^{-1} (X - \bar{X})\right) \end{aligned}$$

avec  $|P|$  le déterminant de la matrice de variance-covariance  $P$ .

Par définition, la moyenne  $\bar{X}$  (sous forme vectorielle) et la variance-covariance  $P$  (sous forme matricielle) sont

$$\begin{aligned} \bar{X} &= \mathbb{E}[X] \\ P &= \mathbb{E}[(X - \bar{X})(X - \bar{X})^T] \end{aligned}$$

### A.2 Représentation ellipsoïdale

Une distribution Normale  $\Gamma(X - \bar{X}; P)$  possède la particularité d'être monotone autour de son maximum de densité  $\bar{X}$ . On peut définir une "*région de confiance*" comme un intervalle multi-dimensionnel  $I$ , dans lequel la probabilité d'apparition d'un échantillon  $u$  est exactement égale à

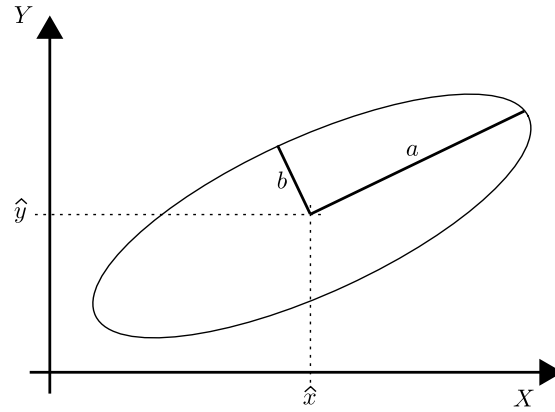


FIGURE A.1 – L'ellipse d'une densité Gaussienne en deux dimensions. Les valeurs  $a$  et  $b$  sont les valeurs propres de l'ellipse. Le centre de l'ellipse est situé à la moyenne statistique.

une valeur  $J$ . On montre alors que la plus petite région de confiance  $I$  correspondant à une valeur de probabilité  $J = P(I)$  est, dans le cas d'une distribution multi-Normale, une ellipse.

Il est alors possible de représenter graphiquement une distribution multi-Normale par une ellipse de centre  $\bar{X}$  et ayant pour contour un tracé correspondant à une probabilité de contenance  $J = 99\%$ , ou encore  $J = 99.99\%$ .

Il est aussi possible de représenter graphiquement une distribution multi-Normale par une ellipse de centre  $\bar{X}$  et ayant pour contour l'ensemble des points  $u$  distants d'un certain écart-type  $\sigma_{ref}$ , on a

$$\sqrt{(u - \bar{X})^T P^{-1} (u - \bar{X})} = \sigma_{ref}$$

avec  $u$  l'ensemble des points (formant une ellipse) à une "distance"  $\sigma_{ref}$ .

Une représentation d'une densité Gaussienne en deux dimensions est présentée en figure A.1.

### A.3 Distance de Mahalanobis

Une distance particulière entre un échantillon  $u$  et une distribution Gaussienne  $\Gamma(X - \bar{X}; P)$  est la distance de Mahalanobis, définie comme

$$M = \sqrt{(u - \bar{X})^T P^{-1} (u - \bar{X})}$$

On a vu précédemment que cette distance est liée à la représentation ellipsoïdale de la distribution multi-Normale.

## A.4 Probabilités et changement de variable

Soit  $X$  un vecteur de variables aléatoires suivant un modèle d'incertitude. Soit  $y = f(X)$  une fonction s'appliquant à  $X$ , la probabilité d'un intervalle  $I$  dans le domaine de  $y$  vaut

$$\begin{aligned} p_Y(y \in I) &= \int_{x \in f^{-1}(I)} p_X(x) dx \\ &= \int_y p_X(f^{-1}(y)) \left| \frac{\partial f^{-1}(y)}{\partial y} \right| dy \end{aligned}$$

On montre alors que si  $X$  suit une densité multi-Gaussienne, et que  $f()$  est une fonction linéaire, on obtiens

$$p(y) = \Gamma(F.X - \bar{Y}; FPF^T)$$

avec  $F$  la matrice définissant la fonction  $f()$ .

Dans le cas d'une fonction non-linéaire appliquée à une densité multi-Gaussienne, on approxime généralement la densité de  $p(y)$  par

$$p(y) \approx \Gamma(f(X) - \bar{Y}; FPF^T)$$

avec  $F$  la matrice Jacobienne de la fonction  $f()$  évaluée en  $\bar{X}$ .



# Bibliographie

- [1] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2007, pp. 1–10.
- [2] M. Lourakis and A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm," *Institute of Computer Science-FORTH, Heraklion, Crete, Greece, Tech. Rep.*, vol. 340, 2004.
- [3] E. Kaplan and C. Hegarty, *Understanding GPS : principles and applications, second edition*. Artech House Publishers, 2006.
- [4] B. Parkinson, J. Spilker, P. Axelrad, and P. Enge, "Global Positioning System : Theory and Applications Volume II," *Progress in astronautics and aeronautics*, vol. 163, 1996.
- [5] M. Sahmoudi, R. Landry, and F. Gagnon, "A constrained maximum-likelihood approach for efficient multipath mitigation in gnss receivers," in *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*. IEEE, 2009, pp. 569–572.
- [6] E. Kaplan and C. Hegarty, *Understanding GPS : principles and applications, second edition*. Artech House Publishers, 2006, ch. 8 DGPS, pp. 379 –458.
- [7] C. Tiberius and P. De Jonge, "Fast positioning using the LAMBDA method," in *Proceedings of the 4th International Symposium on Differential Satellite Navigation Systems DSNS*, vol. 95. Citeseer, 1995, pp. 24–28.
- [8] P. De Jonge and C. Tiberius, "The LAMBDA method for integer ambiguity estimation : implementation aspects," *Publications of the Delft Geodetic Computing Centre*, vol. 12, 1996.
- [9] M. MACABIAU, "Etude de l'applicabilite des techniques de lever d'ambiguite de la mesure de phase gps aux approches de precision," Ph.D. dissertation, INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE, 1997.
- [10] D. Kim and R. Langley, "GPS ambiguity resolution and validation : methodologies, trends and issues," in *Proceedings of the 7th GNSS Workshop-International Symposium on GPS/GNSS, Seoul, Korea*, vol. 30, no. 2.12, 2000.
- [11] S. Durand, "Amélioration de la Précision de la Localisation Différentielle Temps Réel par Mesure de Phase des Systèmes GNSS," *Observatoire de Paris*, 2003.
- [12] C. Tiberius, T. Pany, B. Eissfeller, K. de Jong, P. Joosten, and S. Verhagen, "Integral gps-galileo ambiguity resolution," in *Proc. GNSS*, 2002.



- [13] B. Eissfeller, C. Tiberius, T. Pany, R. Biberger, T. Schuelers, and G. Heinrichs, "Instantaneous ambiguity resolution for gps/galileo rtk positioning," in *9 th Saint Petersburg International Conference on Integrated Navigation Systems, St. Petersburg*, 2002, pp. 72–86.
- [14] J. Pinchin, C. Hide, D. Park, and X. Chen, "Precise kinematic positioning using single frequency gps receivers and an integer ambiguity constraint," in *Position, Location and Navigation Symposium, 2008 IEEE/ION*, may 2008, pp. 600–605.
- [15] C. Park, I. Kim, J. G. Lee, and G.-I. Lee, "Efficient technique to fix gps carrier phase integer ambiguity on-the-fly," *Radar, Sonar and Navigation, IEE Proceedings -*, vol. 144, no. 3, pp. 148–155, jun 1997.
- [16] S. Han and C. Rizos, "The impact of two additional civilian gps frequencies on ambiguity resolution strategies," in *55th National Meeting US Institute of Navigation, " Navigational Technology for the 21st Century"*, Cambridge, Massachusetts. Citeseer, 1999, pp. 28–30.
- [17] J. Codol and M. Poncelet, "GNSS Technologies in robotic applications. MOW-BY-SAT, an autonomous robotic lawnmower using GPS, EGNOS, and GALILEO ," *Toulouse Space Show, Toulouse, France*, 2010.
- [18] D. Melita, G. Muscato, and M. Poncelet, "Performances evaluation of global navigation satellite systems and satellite/ground based augmentation systems in mobile robotics," *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1–8, june 2010.
- [19] T. Ford and J. Neumann, "NovAtel s RT20-A Real Time Floating Ambiguity Positioning System," in *Proceedings of ION GPS 94, Salt Lake City, Utah, Sept*, 1994, pp. 20–23.
- [20] B. Remondi and G. Brown, "Triple differencing with Kalman filtering : making it work," *GPS Solutions*, vol. 3, no. 3, pp. 58–64, 2000.
- [21] M. Petovello, K. O'Keefe, G. Lachapelle, and M. Cannon, "Consideration of time-correlated errors in a Kalman filter applicable to GNSS," *Journal of Geodesy*, vol. 83, no. 1, pp. 51–56, 2009.
- [22] J. Codol and A. Monin, "Improved triple difference GPS carrier phase for RTK-GPS positioning," *International Symposium on Signal Processing SSP-2011, Nice France*, 2011.
- [23] M. Pflieger and J. S andpac andek, "Real-time egnos monitoring," in *ELMAR, 2010 PROCEEDINGS*, sept. 2010, pp. 425–428.
- [24] J. Oliveira and I. Tiberius, "Egnos performance for landing," *European Journal of Navigation*, vol. 6, no. 2, pp. 1–7, 2008.
- [25] D. Laurichesse, F. Mercier, J. Berthias, P. Broca, L. Cerri, and F. CNES, "Integer ambiguity resolution on undifferenced GPS phase measurements and its application to PPP and satellite precise orbit determination," *Navigation*, vol. 56, no. 2, pp. 135–149, 2009.
- [26] S. Bisnath and Y. Gao, "Current state of precise point positioning and future prospects and limitations," *Observing our Changing Earth*, pp. 615–623, 2008.
- [27] G. Wübbena, A. Bagge, and M. Schmitz, "Network-based techniques for rtk applications," *GPS JIN*, pp. 14–16, 2001.

- [28] P. Cruddace, I. Wilson, M. Greaves, H. Euler, R. Keenan, and G. Wübbena, "The long road to establishing a national network rtk solution," in *FIG XXII International Congress, Session no. TS5*, vol. 6, 2002, pp. 19–26.
- [29] J. Meguro, T. Hashizume, J. Takiguchi, and R. Kurosaki, "Development of an autonomous mobile surveillance system using a network-based rtk-gps," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, april 2005, pp. 3096 – 3101.
- [30] M. Saito, Y. Ohmura, A. Onoda, K. Nishikawa, Y. Shibahara, and H. Higuchi, "Network-based rtk-gps for nation-wide high accuracy positioning and navigation in japan," in *International Symposium International Space University*, 2003.
- [31] S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "A high integrity imu/gps navigation loop for autonomous land vehicle applications," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 3, pp. 572 –578, jun 1999.
- [32] X. Yun, E. Bachmann, R. McGhee, R. Whalen, R. Roberts, R. Knapp, A. Healey, and M. Zyda, "Testing and evaluation of an integrated gps/ins system for small auv navigation," *Oceanic Engineering, IEEE Journal of*, vol. 24, no. 3, pp. 396 –404, jul 1999.
- [33] J. Carlson, "Mapping large, urban environment with gps-aided slam," Ph.D. dissertation, Georgia Institute of Technology, 2010.
- [34] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.
- [35] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.
- [36] J. Ortega, "Towards visual localization, mapping and moving objects tracking by a mobile robot : a geometric and probabilistic approach," Ph.D. dissertation, 2007.
- [37] C. Berger and S. Lacroix, "Dseg : Détection directe de segments dans une image," in *Reconnaissance des Formes et Intelligence Artificielle*, 2010.
- [38] P. Smith, I. Reid, and A. Davison, "Real-time monocular slam with straight lines," in *British Machine Vision Conference*, vol. 1, 2006, pp. 17–26.
- [39] T. Lemaire, "Localisation et cartographie simultanées avec vision monoculaire," Ph.D. dissertation, 2006.
- [40] G. Klein and D. Murray, "Improving the agility of keyframe-based slam," *Computer Vision—ECCV 2008*, pp. 802–815, 2008.
- [41] J. Solà, T. Vidal-Calleja, J. Civera, and J. Martinez-Monti, "Impact of landmark parametrization on monocular ekf-slam with points and lines," 2010.
- [42] J. Sola, T. Vidal-Calleja, and M. Devy, "Undelayed initialization of line segments in monocular slam," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 1553–1558.
- [43] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," *Computer Vision ?ECCV 2002*, pp. 128–142, 2002.

- [44] B. Lucas, T. Kanade, *et al.*, “An iterative image registration technique with an application to stereo vision,” in *International joint conference on artificial intelligence*, vol. 3. Citeseer, 1981, pp. 674–679.
- [45] J. Bouguet *et al.*, “Pyramidal implementation of the lucas kanade feature tracker description of the algorithm,” *Intel Corporation, Microprocessor Research Labs, OpenCV Documents*, vol. 3, 1999.
- [46] D. Lowe, “Object recognition from local scale-invariant features,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.
- [47] ———, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [48] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf : Speeded up robust features,” *Computer Vision–ECCV 2006*, pp. 404–417, 2006.
- [49] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [50] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief : Binary robust independent elementary features,” *Computer Vision–ECCV 2010*, pp. 778–792, 2010.
- [51] M. Agrawal and K. Konolige, “Real-time localization in outdoor environments using stereo vision and inexpensive gps,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 1063–1068.
- [52] T. Lemaire, C. Berger, I. Jung, and S. Lacroix, “Vision-based slam : Stereo and monocular approaches,” *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–364, 2007.
- [53] A. Davison, “Active search for real-time vision,” in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 66–73.
- [54] G. Klein and D. Murray, “Parallel tracking and mapping on a camera phone,” in *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*. Ieee, 2009, pp. 83–86.
- [55] A. Argiles, J. Civera, and L. Montesano, “Dense multi-planar scene estimation from a sparse set of images,” *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.
- [56] M. S. R. Smith and P. Cheeseman, “A stochastic map for uncertain spatial relationships,” in *4th Int. Symp. Robot. Res.*, 1988.
- [57] M. Dissanayake, P. Newman, H. Durrant-Whyte, S. Clark, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 229–241, 2001.
- [58] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *The International Journal of Robotics Research*, vol. 23, no. 7-8, p. 693, 2004.
- [59] F. Dellaert and M. Kaess, “Square Root SAM : Simultaneous localization and mapping via square root information smoothing,” *Intl. J. of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec 2006.

- [60] V. Ila, J. M. Porta, and J. Andrade-Cetto, "Information-based compact Pose SLAM," *IEEE Trans. Robotics*, vol. 26, no. 1, 2010, In press. [Online]. Available : <http://dx.doi.org/10.1109/TRO.2009.2034435>
- [61] T. Davis, *Direct methods for sparse linear systems*. Society for Industrial Mathematics, 2006, vol. 2.
- [62] L. Paz, P. Jensfelt, J. Tardós, and J. Neira, "EKF slam updates in  $O(n)$  with divide and conquer slam," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 1657–1663.
- [63] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam 2.0 : An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *International Joint Conference on Artificial Intelligence*, vol. 18. LAWRENCE ERLBAUM ASSOCIATES LTD, 2003, pp. 1151–1156.
- [64] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," 2003.
- [65] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 2499–2504.
- [66] J. Montiel, J. Civera, and A. Davison, "Unified inverse depth parametrization for monocular SLAM," *analysis*, vol. 9, p. 1.
- [67] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Intl. J. of Robotics Research*, vol. 5, no. 4, p. 56–68, 1987.
- [68] J. Leonard, H. Durrant-Whyte, and I. Cox, "Dynamic map building for an autonomous mobile robot," *Intl. J. of Robotics Research*, vol. 11, no. 4, pp. 286–289, 1992.
- [69] J. Tardós, J. Neira, P. Newman, and J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *Intl. J. of Robotics Research*, vol. 21, no. 4, pp. 311–330, 2002.
- [70] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, April 2005, pp. 2417–2424.
- [71] S. Julier and J. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 4, 2001, pp. 4238–4243.
- [72] T. Bailey, J. Nieto, J. Guivant, M. Stevens, , and E. Nebot, "Consistency of the EKF-slam algorithm," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006, p. 3562–3568.
- [73] D. Heckerman, "A tutorial on learning with bayesian networks," *Innovations in Bayesian Networks*, pp. 33–82, 2008.
- [74] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM : A factored solution to the simultaneous localization and mapping problem," in *Proc. 19<sup>th</sup> AAAI National Conference on AI*, Edmonton, Alberta, Canada, 2002.
- [75] M. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Intl. Joint Conf. on AI (IJCAI)*, 2003.

- [76] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT press, Cambridge, MA, 2005.
- [77] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *Intl. J. of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.
- [78] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM : Incremental smoothing and mapping,” *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1365–1378, Dec 2008.
- [79] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent, “Sparse pose adjustment for 2d mapping,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
- [80] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C. Thorpe, “Subgraph-preconditioned conjugate gradient for large scale slam,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
- [81] E. Olson, J. Leonard, and S. Teller, “Fast iterative alignment of pose graphs with poor initial estimates,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2006.
- [82] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, “A tree parameterization for efficiently computing maximum likelihood maps using gradient descent,” in *Robotics : Science and Systems (RSS)*, Jun 2007.
- [83] M. Kaess, V. Ila, R. Roberts, and F. Dellaert, “The Bayes tree : An algorithmic foundation for probabilistic robot mapping,” in *Intl. Workshop on the Algorithmic Foundations of Robotics*, Dec 2010.
- [84] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2 : Incremental smoothing and mapping with fluid relinearization and incremental variable reordering,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.
- [85] J. Codol and M. Poncelet, “Safe L1-only RTK navigation for outdoor service robotic,” *European Navigation Conference ENC-2011, London UK*, 2011.
- [86] P. Teunissen, P. De Jonge, and C. Tiberius, “The lambda method for fast gps surveying,” in *International Symposium ?GPS Technology Applications ? Bucharest, Romania*. Citeseer, 1995.
- [87] C. Roussillon, A. Gonzalez, J. Solà, J. Codol, N. Mansard, S. Lacroix, and M. Devy, “Rt-slam : a generic and real-time visual slam implementation,” *Computer Vision Systems*, pp. 31–40, 2011.
- [88] A. Gonzalez, J. Codol, and M. Devy, “A c-embedded algorithm for real-time monocular slam,” in *Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 665–668.

# Hybridation GPS/Vision monoculaire pour la navigation d'un robot en milieu extérieur

---

## GPS and monocular camera fusion in outdoor robotic navigation problem

---

### ABSTRACT

We are witnessing nowadays the importation of ICT in robotics. These technologies will give birth, in upcoming years, to the \textit{general public} service robotics. This future, if realised, shall be the result of many research conducted in several domains: mechatronics, telecommunications, automatics, signal and image processing, artificial intelligence, etc.

In this thesis, we shed light on the localisation problem of a mobile robot, which we choose to handle with a probabilistic approach. The procedure is as follows: we first define our ?variables of interest? which are a set of random variables, and then we describe their distribution laws and their evolution models. Afterwards, we determine a cost function in such a manner to build up an observer (an algorithmic class where the objective is to minimize the cost function).

Our contribution consists of using brute GPS measures (brute measures or raw datas are measures issued from code and phase correlation loops, called pseudo-distance measures of code and phase, respectively) for a low-cost navigation, which is precise in an external suburban environment. By implementing the so-called "whole" property of GPS phase ambiguities, we expand the navigation to achieve a GPS-RTK\ system in a precise and low-cost local differential mode. Our propositions has been validated through experimentations realized on our robotic demonstrator.

*(french abstract behind this page)*

---

### KEYWORDS

Robotic, Navigation, SLAM, Localisation, Optimisation, GPS, RTK, Bundle Adjustment, Smoothing and mapping, Graphical model.

**AUTEUR :** Jean-Marie CODOL

**TITRE :** Hybridation GPS/Vision monoculaire pour la navigation d'un robot  
en milieu extérieur

**DIRECTEURS DE THESE :** M. André Monin, M. Michel DEVY

**LIEU ET DATE DE SOUTENANCE :** LAAS-CNRS, 7 Avenue du Colonel Roche, 31400 Toulouse  
Soutenance le 15 Février 2012

---

## RESUME EN FRANCAIS

On assiste aujourd'hui à l'importation des NTIC dans la robotique. L'union de ces technologies donnera naissance, dans les années à venir, à la robotique de service grand-public.

Cet avenir, s'il se réalise, sera le fruit d'un travail de recherche, amont, dans de nombreux domaines : la mécatronique, les télécommunications, l'automatique, le traitement du signal et des images, l'intelligence artificielle...

Dans cette thèse, nous aborderons le problème de la localisation d'un robot mobile, et nous choisirons de traiter notre problème par l'approche probabiliste. La démarche est la suivante, nous définissons nos 'variables d'intérêt' : un ensemble de variables aléatoires. Nous décrivons ensuite leurs lois de distribution, et leur modèles d'évolution, enfin nous déterminons une fonction de coût, de manière à construire un observateur (une classe d'algorithme dont l'objectif est de déterminer le minimum de notre fonction de coût).

Notre contribution consistera en l'utilisation de mesures GPS brutes GPS (les mesures brutes - ou raw-datas - sont les mesures issues des boucles de corrélation de code et de phase, respectivement appelées mesures de pseudo-distances de code et de phase) pour une navigation bas-coût précise en milieu extérieur suburbain. En utilisant la propriété dite 'entière' des ambiguïtés de phase GPS, nous étendrons notre navigation pour réaliser un système GPS-RTK en mode différentiel local précise et bas-coût. Nos propositions sont validées par des expérimentations réalisées sur notre démonstrateur robotique.

*(résumé en anglais au recto de cette page)*

---

## MOTS-CLES

Robotique, Navigation, SLAM, Localisation, Optimisation, GPS, RTK, Bundle Adjustment, Smoothing and mapping, Graphe.

---

**DISCIPLINE ADMINISTRATIVE :** Systèmes Automatiques

---

**INTITULE ET ADRESSE DE L'U.F.R. OU DU LABORATOIRE :**

LAAS-CNRS,  
7 Avenue du Colonel Roche, 31400 Toulouse