



HAL
open science

Agrégation d'information pour la localisation d'un robot mobile sur une carte imparfaite

Laurent Delobel

► **To cite this version:**

Laurent Delobel. Agrégation d'information pour la localisation d'un robot mobile sur une carte imparfaite. Robotique [cs.RO]. Université Clermont Auvergne [2017-2020], 2018. Français. NNT : 2018CLFAC012 . tel-01982085

HAL Id: tel-01982085

<https://theses.hal.science/tel-01982085>

Submitted on 15 Jan 2019

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

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

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

THÈSE

en vue de l'obtention du grade de

Docteur d'Université

Spécialité : VISION POUR LA ROBOTIQUE

présentée par

Laurent DELOBEL

**Agrégation d'Information Pour la Localisation
d'un Robot Mobile sur une Carte Imparfaite**

soutenue le 04 Mai 2018.

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

Jury:

Fawzi NASHASHIBI	Directeur de Recherches	INRIA Paris	Rapporteur
Philippe BONNIFAIT	Professeur	UTC Compiègne	Rapporteur
Samia BOUCHAFA	Professeur	Université Evry	Examineur
Roland CHAPUIS	Professeur	Université Clermont Auvergne	Directeur de Thèse
Thierry CHATEAU	Professeur	Université Clermont Auvergne	Directeur de Thèse
Romuald AUFRÈRE	Maitre de Conférences	Université Clermont Auvergne	Examineur
Christophe DEBAIN	Chargé de Recherches	IRSTEA Clermont-F ^d	Invité

Résumé

La plupart des grandes villes modernes mondiales souffrent des conséquences de la pollution et des bouchons. Une solution à ce problème serait de réglementer l'accès aux centres-villes pour les voitures personnelles en faveur d'un système de transports publics constitués de navettes autonomes propulsées par une énergie n'engendrant pas de pollution gazeuse. Celles-ci pourraient desservir les usagers à la demande, en étant déroutées en fonction des appels de ceux-ci. Ces véhicules pourraient également être utilisés afin de desservir de grands sites industriels, ou bien des sites sensibles dont l'accès, restreint, doit être contrôlé.

Afin de parvenir à réaliser cet objectif, un véhicule devra être capable de se localiser dans sa zone de travail. Une bonne partie des méthodes de localisation reprises par la communauté scientifique se basent sur des méthodes de type « Simultaneous Localization and Mapping » (SLAM). Ces méthodes sont capables de construire dynamiquement une carte de l'environnement ainsi que de localiser un véhicule dans une telle carte. Bien que celles-ci aient démontré leur robustesse, dans la plupart des implémentations, le partage d'une carte commune entre plusieurs robots peut s'avérer problématique. En outre, ces méthodes n'utilisent fréquemment aucune information existant au préalable et construisent la carte de leur environnement à partir de zéro.

Nous souhaitons lever ces limitations, et proposons d'utiliser des cartes de type sémantique, qui existent au-préalable, par exemple comme OPENSTREETMAP, comme carte de base afin de se localiser. Ce type de carte contient la position de panneaux de signalisation, de feux tricolores, de murs de bâtiments etc... De telles cartes viennent presque à-coup-sûr avec des imprécisions de position, des erreurs au niveau des éléments qu'elles contiennent, par exemple des éléments réels peuvent manquer dans les données de la carte, ou bien des éléments stockés dans celles-ci peuvent ne plus exister.

Afin de gérer de telles erreurs dans les données de la carte, et de permettre à un véhicule autonome de s'y localiser, nous proposons un nouveau paradigme. Tout d'abord, afin de gérer le problème de sur-convergence classique dans les techniques de fusion de données (filtre de Kalman), ainsi que le problème de mise à l'échelle, nous proposons de gérer l'intégralité de la carte par un *filtre à Intersection de Covariance Partitionnée*. Nous proposons également d'effacer des éléments inexistant des données de la carte en estimant leur probabilité d'existence, calculée en se basant sur les détections de ceux-ci par les capteurs du véhicule, et supprimant ceux doté d'une probabilité trop faible. Enfin, nous proposons de *scanner* périodiquement la totalité des données capteur pour y chercher de *nouveaux amers potentiels* que la carte n'intègre pas encore dans ses données, et de les y ajouter.

Des expérimentations montrent la faisabilité d'un tel concept de *carte dynamique de haut niveau* qui serait mise à jour au-vol.

Mots clés : Sur-Convergence, filtre de Kalman, filtre à Intersection de Covariance, Construction/Gestion d'une Carte Dynamique, Réseaux Bayésiens, Gestion de l'Intégrité.

Abstract

Most large modern cities in the world nowadays suffer from pollution and traffic jams. A possible solution to this problem could be to regulate personal car access into center downtown, and possibly replace public transportations by pollution-free autonomous vehicles, that could dynamically change their planned trajectory to transport people in a fully on-demand scenario. These vehicles could be used also to transport employees in a large industrial facility or in a regulated access critical infrastructure area.

In order to perform such a task, a vehicle should be able to localize itself in its area of operation. Most current popular localization methods in such an environment are based on so-called « Simultaneous Localization and Mapping » (SLAM) methods. They are able to dynamically construct a map of the environment, and to locate such a vehicle inside this map. Although these methods demonstrated their robustness, most of the implementations lack to use a map that would allow sharing over vehicles (map size, structure, etc. . .). On top of that, these methods frequently do not take into account *already existing information* such as an existing city map and rather *construct it from scratch*.

In order to go beyond these limitations, we propose to use in the end semantic high-level maps, such as OPENSTREETMAP as *a-priori* map, and to allow the vehicle to localize based on such a map. They can contain the location of roads, traffic signs and traffic lights, buildings etc. . . Such kind of maps almost always come with some degree of *imprecision* (mostly in position), they also can be *wrong, lacking* existing but undescribed elements (landmarks), or *containing* in their data elements that do not exist anymore.

In order to manage such imperfections in the collected data, and to allow a vehicle to localize based on such data, we propose a new strategy. Firstly, to manage the classical problem of data incest in data fusion in the presence of strong correlations, together with the map scalability problem, we propose to manage the whole map using a *Split Covariance Intersection filter*. We also propose to remove possibly *absent landmarks* still present in map data by estimating their probability of being there based on vehicle sensor detections, and to remove those with a low score. Finally, we propose to periodically *scan* sensor data to detect *possible new landmarks* that the map does not include yet, and proceed to their integration into map data.

Experiments show the feasibility of such a concept of *dynamic high level map* that could be updated on-the-fly.

Keywords : Overconvergence, Kalman filter, Covariance Intersection filter, Dynamic Map Construction/Management, Bayes Networks, Integrity Management.

Remerciements

Ce travail n'aurait pu voir le jour sans ROMUALD AUFRÈRE, qui m'a proposé une thèse, et qui s'est battu pour obtenir un financement. J'associe bien évidemment à ces remerciements ROLAND CHAPUIS qui m'a accueilli dans son équipe de l'INSTITUT PASCAL, et qui par ses remarques, parfois critiques, mais aussi constructives, m'a beaucoup appris sur le sens du mot *recherche*. Je remercie également THIERRY CHATEAU et CHRISTOPHE DEBAIN qui ont co-encadré ces travaux.

M^r MICHEL DHOME, directeur à la fois de l'INSTITUT PASCAL et du LABEX IMOBS3 au démarrage de ma thèse, et qui m'a accueilli au sein d'un laboratoire aux multiples thématiques, a toujours été disponible pour échanger de manière informelle en se mettant à la hauteur du doctorant que je suis. J'aimerais l'en remercier. Je remercie également les chercheurs et enseignants-chercheurs de la thématique ISPR, et parmi eux FRANÇOIS BERRY, YOUSSEF MEZOUHAR, CÉLINE TEULIÈRE et MAXIME LHULLIER pour les discussions intéressantes et qui m'ont ouvert à des domaines scientifiques que je connais moins.

Je veux également témoigner ma reconnaissance à JEAN-THIERRY LAPRESTÉ, mathématicien émérite de l'INSTITUT PASCAL, qui m'a, à certaines occasions, grâce à un mail de deux lignes, aidé à trouver des solutions alors que je bloquais. Mr BRUNO BACHELET, maître de conférences à l'ISIMA en C++ et méta-programmation, a été d'un conseil inappréciable dans la mise au point de certaines particularités de mes développements, notamment celui de la bibliothèque [LibK2](#) de filtres génériques. Dans la même thématique, je remercie également JOËL FALCOU, ancien ZZ comme moi, et ancien doctorant de l'INSTITUT PASCAL, maître de conférence au LRI, Paris XI Orsay, et *référence* dans le domaine de la méta-programmation C++, pour nos divers échanges.

Une mention spéciale pour les ingénieurs support de la thématique ISPR, qu'il soient titulaires ou contractuels, qui m'ont aidé à configurer les véhicules, les ont conduits, et ont veillé aux maintenances et mises à niveau. Je remercie LAURENT MALATERRE, FRANÇOIS MARMOITON, SERGE ALIZON et STÉPHANE WITZMANN pour leur support et leur disponibilité. CLAUDE AYNAUD, mon prédécesseur, m'a pendant plusieurs mois décrit de A à Z l'algorithme qu'il avait développé, afin que je puisse le ré-écrire intégralement durant sa rédaction. Ses explications le font mériter le qualificatif de *bénédictin* pour sa patience envers moi. Sans celles-ci, mon travail de re-codage n'aurait pu être aussi réussi. Je remercie également CORALIE BERNAY-ANGELETTI, ma collègue doctorante au début de mes travaux et qui a soutenu depuis, pour les discussions à bâtons rompus que nous avons eues, pour sa perpétuelle bonne humeur et son calme olympien. Elle a également présenté à la conférence RoBio 2015 mon tout premier article de conférence, je l'en remercie infiniment.

Mes collègues doctorants avec qui j'ai pu échanger, notamment FRANÇOIS DE LA BOURDONNAYE, NGYEN THAN TIN, ANGE NIZARD pour n'en citer que quelques uns, m'ont permis par les diverses discussions que j'ai eues avec eux, d'humaniser mon tra-

vail de recherche, bien solitaire par ailleurs. THOMAS FÉRAUD, qui finissait son contrat de post-doctorant quand j'ai démarré ma thèse, mérite également toute ma reconnaissance pour m'avoir introduit aux techniques particulières de la robotique mobile. Son expérience dans ce domaine m'a fait avancer à pas de géants au début de mes travaux et je l'en remercie.

Pour clore les remerciements relatifs au travail technique de recherche que j'ai effectué depuis 2014, je remercie le programme *Equipement d'Excellence* ROBOTEX (ANR-10-EQPX-44), le programme « Investissements d'Avenir », géré par l'*Agence Nationale de la Recherche*, le fond européen FEDER à travers la région AUVERGNE (désormais la région AURA) pour avoir financé ces travaux, dans le cadre du *Laboratoire d'Excellence* IMOBS3 (ANR-10-LABX-16-01).

Un proverbe dit « *Qui veut voyager loin ménage sa monture.* ». Une thèse étant un travail de longue haleine et non un sprint, je remercie les membres de ma famille pour leur soutien moral. Je remercie tout spécialement mes parents, ainsi que ma sœur Christelle qui, bien souvent malgré les distances, m'ont soutenu par leurs conseils avisés. Je remercie également les gens qui m'ont aidé mais que j'aurai omis de mentionner. J'ai une grande gratitude envers le corps professoral de l'ISIMA, sans lequel je n'aurais pu développer mes compétences en informatique dans divers domaines qui m'ont été utiles pour mes travaux. Celles-ci m'ont permis d'avancer bien plus vite que je n'aurais pu le faire sinon. À travers l'ISIMA, c'est également toute l'UNIVERSITÉ BLAISE PASCAL (désormais devenue UNIVERSITÉ CLERMONT-AUVERGNE) que je souhaite remercier pour m'avoir permis de retourner « sur les bancs de l'école ».

Table des matières

Introduction	19
1.1 Différents domaines de la robotique	20
1.1.1 Robotique domestique	20
1.1.2 Robotique industrielle	21
1.1.3 Robotique médicale	23
1.1.4 Robotique de transport	25
1.2 Contexte de la robotique de transport de personnes	26
1.2.1 Technologies d'aide à la conduite	26
1.2.2 Les véhicules autonomes	28
1.3 Contexte des travaux	29
2 Localisation et cartographie pour la robotique mobile	33
2.1 Introduction	33
2.2 Principaux problèmes liés à la localisation d'un robot mobile	35
2.2.1 Erreurs intégrales de positionnement	35
2.2.2 Formalisation des données dans une carte	36
2.2.3 Problèmes d'association carte–observations	37
2.2.4 Carte et évolution de l'environnement avec le temps	37
2.3 Types de capteurs utilisés en robotique mobile	38
2.3.1 Données et capteurs proprioceptifs	38
2.3.2 Capteurs extéroceptifs	40
2.3.3 Exceptions	45
2.4 Méthodes de Cartographie en Robotique	48
2.4.1 Techniques Basées sur un Système d'Information Géographique	48
2.4.2 Techniques basées sur une acquisition orthophotographie–LIDAR vertical	51
2.4.3 Cartographies des voies de circulation basée GPS	53
2.4.4 Cartographies Basées SONAR	54
2.4.5 Cartographies Basées LIDAR	55
2.4.6 Cartes prenant en compte l'évolution dans le temps et les configurations	56
2.4.7 Cartes séparant les éléments fixes des éléments mobiles	57
2.4.8 Cartes obtenues par SLAM	59
2.4.9 Tableaux Récapitulatif	63
2.5 Techniques de localisation basées sur une carte	66
2.5.1 Description des grands principes	66
2.5.2 Alignement de données sous forme de nuages de points	67
2.5.3 Détection du marquage au sol	68

2.5.4	Localisation par association de vue caméra	69
2.5.5	Alignement sur les bords de la route	70
2.5.6	Synthèse	70
2.6	Techniques de localisation et de gestion d'une carte dynamique	71
2.6.1	Localisation et fusion de données	72
2.6.2	Localisation collaborative par filtre de Kalman	74
2.6.3	Approches par fusion de cartes locales	75
2.6.4	Approches par Intersection de Covariance	75
2.6.5	Approches de type Maximum-A-Posteriori	76
2.6.6	Approches de type Channel Filter	77
2.6.7	Approches Multi-Hypothèses	77
2.6.8	Synthèse sur la cartographie et la gestion d'une carte dynamique	78
3	Approche Proposée	81
3.1	Choix de la technique de localisation sur carte	81
3.2	Données contenues dans la carte et méthode de mise à jour	83
3.3	Méthode choisie et développée	84
3.4	Propriétés des travaux antérieurs	86
4	Techniques Probabilistes et Théorie de l'Information	87
4.1	Introduction au formalisme Bayésien	88
4.1.1	Variable Aléatoire Normale	88
4.1.2	Distance de Mahalanobis	88
4.1.3	Variable Aléatoire Binaire	90
4.1.4	Règle de Bayes	90
4.2	Introduction à la Théorie de l'Information	91
4.3	Entropie de Shannon	92
4.3.1	Définition	92
4.3.2	Exemple	93
4.4	Entropie Différentielle	93
4.4.1	Cas général	94
4.4.2	Analogie	94
4.4.3	Cas d'une distribution Normale	94
4.5	Divergence de Kullback-Leibler	95
4.5.1	Cas discret	95
4.5.2	Cas continu	95
4.6	Information Mutuelle	96
4.6.1	Entropie conditionnelle	96
4.6.2	Information mutuelle de deux variables	96
4.6.3	Impact de la connaissance d'une variable sur une autre	97
4.6.4	Inégalité relative au traitement de l'information	97
4.6.5	Exemples	98
4.6.6	Généralisations	98
4.6.7	Applications	99
4.7	Estimation Probabiliste Continue	100
4.7.1	Modèle d'état	100
4.7.2	Filtre de Kalman	101
4.7.3	Filtres de Kalman Corrélés	101
4.7.4	Pourquoi une fusion corrélée ne fonctionne-elle pas?	104

4.7.5	Filtre à Intersection de Covariance	106
4.7.6	Coût calculatoire d'un filtre à Intersection de Covariance	109
4.7.7	Filtre à Intersection de Covariance Partitionnée	110
4.8	Estimation Probabiliste Discrète	111
4.8.1	Formalisme des Réseaux Bayésiens Discrets	111
4.8.2	Table de Probabilités Conditionnelles	112
4.8.3	Chaînage des Probabilités Conditionnelles	113
4.8.4	Marginalisation	113
4.8.5	Inférence	114
4.8.6	Réseaux Bayésiens Chaînés	115
4.9	Approximation dite de « Champ Moyen »	116
4.9.1	Exemple visuel	117
4.9.2	Familles Exponentielles	117
4.9.3	Approximation de « Champ Moyen »	118
4.9.4	Discussion	119
5	Localisation dans une Carte Existante	121
5.1	Introduction	121
5.1.1	Historique des Travaux	121
5.2	Concept de <i>Triplet Perceptif</i>	122
5.2.1	Concepts de « proprioceptivité » et d'« extéroceptivité »	122
5.2.2	Présentation du Concept de Triplet	124
5.2.3	Choix du Triplet	125
5.2.4	Représentation de l'état du Véhicule	126
5.2.5	Structure de l'Algorithme de Localisation	127
5.2.6	Contenu de la Carte	128
5.3	Estimation de position d'un véhicule dans une carte précise	129
5.3.1	Visibilité des Murs	129
5.3.2	Modèle d'Observation des Murs Fixes	130
5.3.3	Détection des Murs	132
5.4	Gestion de l'Intégrité	133
5.4.1	Concept représenté par la probabilité d'intégrité	133
5.4.2	Réseau Bayésien	134
5.4.3	Coefficient de corrélation p_{CI_k}	136
5.5	Critère de choix du Triplet	137
5.6	Véhicules de test	138
6	Localisation dans une carte imprécise	141
6.1	Introduction	141
6.2	Utilisation du filtre à intersection de covariance	142
6.2.1	Rappel	142
6.2.2	Modélisation	144
6.2.3	Observation et mise à jour de l'état des amers	146
6.2.4	Validation par simulation de la méthode choisie	149
6.3	Critère de sélection des amers proposé	156
6.3.1	Pourquoi utiliser la notion d'entropie ?	156
6.3.2	Remarque sur le calcul de l'entropie sur une carte	156
6.3.3	Nouveau critère	157
6.3.4	Simulation qualitative du nouveau critère	160

6.4	Améliorations spécifiques	161
6.4.1	Améliorations du réseau bayésien	161
6.4.2	Modification du détecteur de murs	162
6.4.3	Modification du critère théorique proposé	164
6.4.4	Estimation des temps de calcul <i>Kalman</i> vs <i>SCI</i>	167
6.5	Expérimentation	168
6.5.1	Différentes expérimentations	168
6.5.2	Test avec capteurs LIDAR, GPS et magnétomètre	171
6.5.3	Test avec capteurs LIDAR, GPS-RTK et magnétomètre dans une carte imprécise	178
6.5.4	Comparaison au filtre de Kalman	179
6.5.5	Test du « robot kidnappé »	181
7	Suppression d'Amers de la Carte	185
7.1	Introduction	185
7.2	Problématique des Réseaux Couplés	186
7.2.1	Estimation exacte et explosion combinatoire	186
7.2.2	Problème théorique et entropie	188
7.2.3	Factorisation et sur-convergence	189
7.2.4	Réseaux bayésiens couplés et approximations	190
7.3	Test des solutions proposées	191
7.3.1	Description des simulations réalisées	191
7.3.2	Réseau modifié n°1 avec marginalisation	192
7.3.3	Réseau modifié n°1 avec approximation « mean-field »	194
7.3.4	Réseau hiérarchique n°2	195
7.4	Expérimentation sur véhicule	199
7.4.1	Description	199
7.4.2	Résultats	199
8	Ajout d'Amers dans la Carte	205
8.1	Introduction	205
8.2	Idée de base de l'ajout de murs dans la carte	206
8.3	Principe du processus BOTTOM-UP	207
8.3.1	Transformée de Hough	207
8.3.2	Processus d'alignement entre scans LIDAR	210
8.3.3	Association avec la carte existante	214
8.3.4	Processus de « bloc-diagonalisation »	217
8.4	Résultats de simulation	217
8.5	Résultats expérimentaux	221
	Conclusion et Perspectives	225
A	Articles Publiés dans le cadre de ces travaux	231
B	Tables de probabilité conditionnelles	233
B.1	Tables de probabilités originelles	233
B.2	Tables de probabilités modifiées	234

C	Estimation continue et théorie de l'information	235
C.1	Représentation d'Etat et Dérivation du Filtre de Kalman	235
C.1.1	Développement des équations	235
C.1.2	Optimalité du filtre de Kalman	236
C.2	Conservativité du filtre à Intersection de Covariance	236
C.2.1	Estimation gaussienne conservative	236
C.2.2	Principe du filtre à Intersection de Covariance	237
C.2.3	Choix de ω	239
C.3	Filtre à Intersection de Covariance Partitionnée	239

Notations utilisées

x	Scalaire (dans \mathbb{R})
\mathbf{X}	Vecteur (dans un \mathbb{R} -espace-vectoriel)
Σ	Matrice symétrique définie positive (représentant une matrice de covariance)
$\frac{d\mathbf{Z}}{d\mathbf{T}}$	Dérivée de la fonction vectorielle \mathbf{Z} par rapport au vecteur \mathbf{T}
\mathbf{M}^\top	Transposée de la matrice \mathbf{M}
$ \mathbf{M} $	Déterminant de la matrice carrée \mathbf{M}
$p(A = a)$	Probabilité discrète de faire un tirage de la variable A ayant la valeur a
$p(A = a B = b)$	Probabilité discrète de faire un tirage de la variable A ayant la valeur a , sachant que dans le tirage en question, la variable B a la valeur b .
\underline{p}	Densité de probabilité continue
$\mathcal{N}(x; x_0, \sigma_x^2)$	Loi normale sur la variable x , de moyenne x_0 et de variance σ_x^2
\cup	Union d'ensembles
\cap	Intersection d'ensembles
\setminus	Soustraction ensembliste
$O(x)$	Notation <i>grand-O</i>
$\arg \min_x f(x)$	Valeur de la variable x pour laquelle $f(x)$ est minimale
$\arg \max_x f(x)$	Valeur de la variable x pour laquelle $f(x)$ est maximale
$\Sigma_{\mathbf{A}} \succeq \Sigma_{\mathbf{B}}$	Indique que la matrice $\Sigma_{\mathbf{B}}$ est « plus fine » que $\Sigma_{\mathbf{A}}$ au sens de la positivité, c'est à dire que $\Sigma_{\mathbf{A}} - \Sigma_{\mathbf{B}}$ est semi-définie positive
$\Sigma_{\mathbf{A}} \succ \Sigma_{\mathbf{B}}$	Indique que la matrice $\Sigma_{\mathbf{B}}$ est « strictement plus fine » que $\Sigma_{\mathbf{A}}$ au sens de la positivité, c'est à dire que $\Sigma_{\mathbf{A}} - \Sigma_{\mathbf{B}}$ est définie positive
$\sum_x f(x)$	Somme sur les indices x des valeurs $f(x)$
$\prod_x f(x)$	Produits sur les indices x des valeurs $f(x)$

À ma famille, qui fût toujours à mes côtés,
À tous ces anonymes qui m'ont un jour tendu la main,
et que je n'oublie pas.

Introduction

Depuis longtemps déjà, l'être humain rêve de pouvoir se décharger de tâches répétitives ou sans intérêt majeur pour lui afin de pouvoir se focaliser sur ce qu'il estime important, des activités créatives, d'organisation ou bien à composante sociale importante. Qui n'a jamais rêvé de pouvoir faire en sorte que le sol de son domicile soit nettoyé sans son intervention pendant qu'il s'occupe de l'un de ses enfants.

Le terme robot, inventé en 1920 dans la littérature d'anticipation, désigne actuellement une machine composée d'un système électronique de contrôle, d'actionneurs et de capteurs renseignant sur l'environnement, destiné à réaliser des tâches dites « simples » pour le commun des mortels.

Alors que le terme « ordinateur » n'a été associé au concept de « computer » qu'une décennie après l'invention de l'**ENIAC** en 1946, le mot robot a précédé de trente ans la première création cybernétique. Celle-ci consistait en un « bras robotisé programmable » nommé UNIMATE, qui fut utilisé sur les chaînes de montage du géant de l'automobile GENERAL MOTORS. C'est dire que l'idée de pouvoir déléguer des tâches secondaires à une entité non-humaine préexistait dans l'inconscient collectif. Cette force motrice a progressivement implanté les robots dans tous les domaines de l'industrie de fabrication, allant de l'industrie automobile à la fabrication d'emballages en passant par la réalisation de pièces usinées...

Développons ici les deux définitions du mot robot, qui nous ont paru avoir un intérêt, définitions issues du site <http://www.larousse.fr> :

ROBOT :

- *Dans les œuvres de science-fiction, machine à l'aspect humain, capable de se mouvoir, d'exécuter des opérations, de parler.*
- *Appareil automatique capable de manipuler des objets ou d'exécuter des opérations selon un programme fixe, modifiable ou adaptable.*

La première de ces définitions correspond au contexte de l'œuvre littéraire dans laquelle ce terme fut pour la première fois inventé. La seconde, quant à elle, décrit ce que l'humanité a pu construire de plus approchant du concept décrit dans la précédente définition. Il y a donc, dans la robotique, deux aspects :

- l'aspect social ou imaginaire, qui correspond à l'aspiration des hommes, la volonté motrice,
- l'aspect technologique, qui consiste à trouver des moyens de réaliser la première, qui en est l'exécution.

Il n'est pas étonnant que ce domaine exacerbe les passions, les oppositions les plus farouches comme les vocations les plus précoces, étant donné son impact potentiel sur la société et sur la technologie qui la rend réalisable et crédible.

Nous allons tenter ici de faire un tour d’horizon du vaste domaine de la robotique, qui est souvent mal connu du grand public, tant par les techniques utilisées que les objectifs déjà atteints.

1.1 Différents domaines de la robotique

La robotique est un domaine actuellement très divers, tant du point de vue des techniques utilisées pour les concevoir, les construire que de celui de leurs objectifs. On peut citer, de manière non-exhaustive, les robotiques dédiées aux applications :

- domestiques,
- industrielles,
- médicales,
- de transport de personnes,
- de transport de biens,
- de type spatial,
- de type militaire.

On peut également séparer la robotique en termes de technologies mises en oeuvre pour le concevoir ou l’opérer :

- la robotique manipulatrice,
- la robotique mobile,
- la micro-robotique.

Nous allons décrire quelques exemples de réalisations dans les différents domaines d’application, avant de nous intéresser aux différentes techniques mises en oeuvre. Nous passons sous silence les applications de type spatial et militaire, bien que ce soient des applications de type robotique (sondes spatiales, atterrisseurs, rovers, drones aériens, robots terrestres tels le BIG-DOG) car trop spécifiques.

1.1.1 Robotique domestique

Après la robotique industrielle, première application à grande échelle de la robotique, la première application disponible pour les particuliers a été la robotique domestique.

Un exemple d’application fréquemment utilisé pour introduire la robotique est le robot ROOMBA[®], commercialisé par la société iROBOT en 2002 pour la première fois, et utilisé pour nettoyer sans intervention humaine le sol d’un logement. On peut voir l’une des dernières générations de celui-ci sur la figure 1.1. Ce type de robot a été un succès commercial immédiat en atteignant le million d’exemplaires vendus en 2004.

Un autre exemple de robot domestique, dans le domaine ludique est le robot NAO, présenté sur la figure 1.2, commercialisé par la société SOFTBANK ROBOTICS, anciennement ALDEBARAN ROBOTICS et nom sous lequel elle est plus connue. NAO est un robot humanoïde commercialisé pour la première fois en 2008. Sa principale application est le divertissement mais il est également utilisé comme plateforme de développement pour les universités ou pour des challenges en robotique comme la ROBOCUP (NAO ACADEMICS EDITION). De nouveaux robots comme PEPPER ou ROMEO sont également au catalogue de cette société et indiquent la vitalité de cette branche de la robotique.

La robotique domestique se démocratise. Ses coûts de production et de vente sont en baisse avec une augmentation de la production concomitante. Les particuliers y sont de



FIGURE 1.1 – Robot ROOMBA[®] 880, commercialisé par la société iROBOT.

plus en plus sensibles, on peut légitimement parler d'engouement. Cette branche de la robotique est appelée à évoluer de manière importante, par exemple avec le développement de la robotique de service.



FIGURE 1.2 – Robot NAO, commercialisé par la société SOFTBANK ROBOTICS.

1.1.2 Robotique industrielle

La robotique industrielle a été le premier secteur d'utilisation des robots. Ceux-ci y ont longtemps été cantonnés, car en raison de leurs coût de développement, leurs applications à d'autres domaines étaient non-viables économiquement. Le premier robot était le bras robotisé UNIMATE, commercialisé par la société UNIMATION, représenté sur la figure 1.3, conçu dans les années 1950 et utilisé sur les chaînes de montage à partir des années 1960 aux ETATS-UNIS.

La régie RENAULT a été en France la première société du secteur automobile à utiliser de tels robots à la fin des années 1970, et fut bientôt imitée par d'autres constructeurs

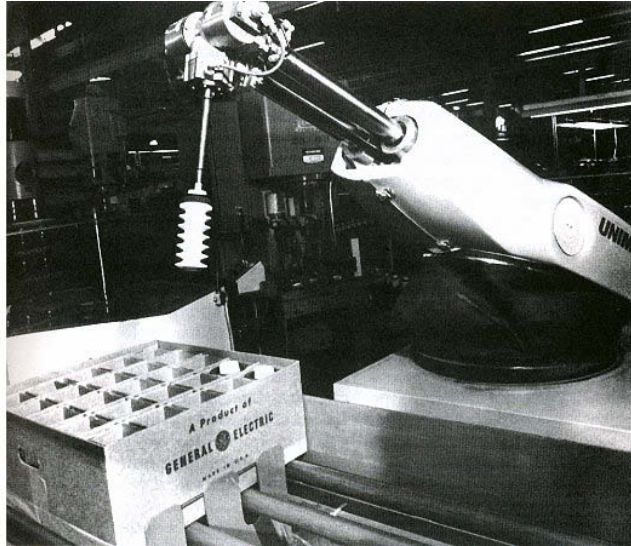


FIGURE 1.3 – UNIMATE, le premier robot manipulateur au monde, inventé par Georges DEVOL et commercialisé par la société UNIMATION.

de cette industrie. La figure 1.4 montre une chaîne de montage de véhicules automobiles du constructeur **PSA-PEUGEOT-CITROËN**, dans laquelle les opérations de manutention d'objets lourds (portières, sièges par exemple), de soudure précise et reproductibles sont exécutées par des robots industriels. Ces robots présentent des avantages certains en termes de rapidité d'exécution, de précision, de force et de capacité d'opération dans des conditions dangereuses pour des êtres humains (en présence de produits chimiques comme des solvants aux postes de peinture, des vapeurs toxiques aux postes de soudure à l'arc, dans des postes de zingage, ou autre).

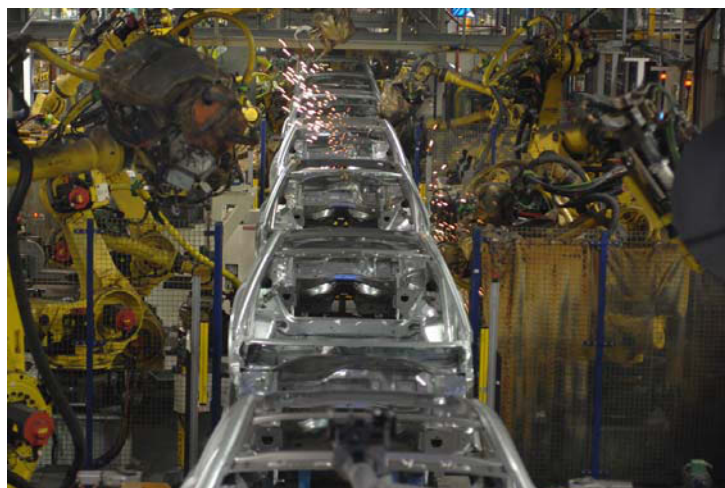


FIGURE 1.4 – Exemple de chaîne de montage du groupe **PSA-PEUGEOT-CITROËN**.

Cette branche est le cheval de bataille historique de la robotique. Elle est actuellement en évolution stable, la technologie dans le monde industriel étant considérée comme « mature ». Les robots de cette branche sont typiquement chers, lourds, complexes, encombrants, puissants, rapides, mais à poste fixe et peuvent fonctionner dans des environ-

nements interdits à l'homme (température, polluants chimiques, etc...). Ce domaine est l'exact contraire de la robotique domestique.

1.1.3 Robotique médicale

L'usage des robots dans le domaine médical est relativement récent. Le tout premier robot chirurgical, destiné à l'orthopédie, a été testé dans les années 1980 à Vancouver. Ce robot, nommé ARTHROBOT, avait pour fonction de positionner les articulations des patients dans des configurations déterminées à l'avance, selon les ordres vocaux du chirurgien. Les robots sont actuellement utilisés spécifiquement pour des opérations dans lesquelles une intervention classique aurait plus de risques de complications ou poserait plus de difficultés.

On peut citer par exemple le robot DA-VINCI de la société INTUITIVE SURGICAL, qui est présenté sur la figure 1.5. Il est constitué d'un poste de contrôle opéré par un chirurgien, à gauche au premier plan sur la figure, et d'un bloc opératoire équipé de multiples bras robotisés portant des instruments compatibles avec des opérations de type endoscopique, à droite au fond sur la figure.

Il permet, grâce aux instruments multiples portés par les bras, de réaliser des opérations moins invasives, plus précises, et sous des angles impossibles à réaliser par un être humain. En raison de la rareté de ce type de robots, ces interventions sont en règle générale réservées à des cas d'emploi spécifiques. Les interventions dans lesquelles ceux-ci n'ont pas de plus-value importante sont réalisées de manière classique. La rareté de ces robots médicaux peut être une conséquence de leur coût comme de l'acquisition de la pratique nécessaire pour les utiliser.

On songe à télé-opérer ce type d'intervention, c'est à dire opérer depuis un terminal de contrôle à distance importante ($\geq 100 \text{ km}$) du bloc opératoire. Actuellement, ce type de poste est situé dans une pièce attenante, pour limiter le risque de défaut de connexion de type réseau qui aurait des conséquences sérieuses dans le cas d'une télé-opération. D'autres robots sont également utilisés pour de la neuro-chirurgie ou de la chirurgie cardiaque par exemple.

D'autres systèmes robotiques, utilisés non pas en chirurgie, mais en rééducation fonctionnelle, permettent de s'attaquer à des problèmes tels les conséquences d'accidents vasculaires cérébraux, ou bien à la perte de motricité. Un exemple est le LOKOMAT commercialisé par la société HOCOMA, que l'on peut voir sur la figure 1.6.

Ce domaine de la robotique est, dans le domaine de la médecine, assez marginal. Néanmoins, des opérations sont possibles là où un être humain les réaliserait de manière moins précise, plus invasive ou avec un risque de complication plus élevé. Ces robots peuvent être considérés comme la « rolls-royce » dans un certain nombre de techniques médicales. Ils requièrent par contre, afin de pouvoir les utiliser, des compétences allant au-delà de la simple médecine, ce qui en fait des outils à l'apprentissage potentiellement long. La robotique médicale parvient néanmoins à convaincre progressivement, bien que ce phénomène soit lent à se produire.



FIGURE 1.5 – Robot médical DA-VINCI commercialisé par la société INTUITIVE SURGICAL.



FIGURE 1.6 – Exosystème LOKOMAT de la société HOCOMA.

1.1.4 Robotique de transport

Le développement de robots comme ceux décrits dans les précédentes sections, ainsi que des avancées théoriques ont permis de commencer à développer des robots mobiles autonomes. Plusieurs exemples d'applications de tels robots existent, comme par exemple le transport de personnes sur site industriel, ou le transport de matériel dans une usine.

Un exemple de robot de transport de personnes sur sites industriels, c'est à dire une navette autonome, est présenté sur la figure 1.7. Il s'agit de la navette autonome ARMA de la société NAVYA, basée à Lyon. Elle est capable de se localiser dans un environnement dont elle a déjà une carte et de parcourir sans conducteur des trajets « courts ». Cette navette a été testée sur le centre EDF de production électro-nucléaire de Civaux dans le département de la Vienne, pour le transport d'employés, et a été mise en service en avril 2016.



FIGURE 1.7 – Navette automatique de transport de personnes ARMA, de la société NAVYA. Photographie prise à proximité du centre EDF de production d'électricité de Civaux.

Un autre exemple d'un tel véhicule est présenté sur la figure 1.8, représentant un modèle Ez10. Ce véhicule est commercialisé par la société EASY-MILE, et le châssis propulseur est conçu et produit par l'industriel automobile français LIGIER.

Des robots autonomes de manutention commencent également à exister. Leur degré d'autonomie actuel ne leur permet pas de s'insérer dans une foule. Ils sont donc limités à des applications en site contrôlé, comme des sites industriels.

Tout comme la robotique domestique, cette application de la robotique est en plein essor et, bien que les applications actuelles soient cantonnées à des environnements limités, une évolution rapide va très probablement se produire dans les années à venir. Le groupe automobile DAIMLER a par exemple annoncé récemment vouloir passer à un test en environnement routier dans un flot de circulation humain, d'un prototype de voiture entièrement autonome d'ici 2022. Il s'agit sans doute de l'un des tests les plus difficiles dans le monde de la robotique.

Cette robotique de transport est, après la robotique domestique, la seconde révolution attendue de la robotique, avec par exemple l'automatisation de la logistique par robots sur les sites industriels, le concept de distribution de produits au consommateur envisagé par



FIGURE 1.8 – Véhicule autonome EZ10, de la société EASY-MILE.

drones, ou bien la voiture intelligente. Le concept de camions intelligents est également envisagé.

C'est à cette robotique liée aux transports que nous allons nous attacher dans le reste de cette introduction, qui est le domaine d'application directe des travaux présentés dans ce mémoire.

1.2 Contexte de la robotique de transport de personnes

Le domaine de la robotique appliqué au transport de personnes recouvre deux approches distinctes :

- l'aide à la conduite et les fonctions de sécurité, qui s'adresse à des conducteurs humains dans des véhicules non-autonomes (Advanced Driver-Assistance Systems, ADAS),
- la conduite autonome dans des véhicules conçus dans ce but.

Si le premier élément a actuellement une réalité tangible, le second ne compte pour l'instant que quelques applications, par exemple la GOOGLE-CAR, ou les développements récents de l'industriel TESLA. Ces derniers, bien que montrant la faisabilité technique, sont actuellement positionnés encore trop loin de l'industrialisation à grande échelle pour la permettre, tant du point de vue du coût que de la sécurité. Néanmoins, les « road-maps » des grands constructeurs visent à la commercialisation d'un premier modèle totalement autonome dans un flot de circulation d'ici à 5 ans.

Nous allons présenter ces deux approches à travers quelques exemples, puis à la lumière de ces explications, introduire le contexte dans lequel nous avons réalisé les travaux décrits dans ce mémoire.

1.2.1 Technologies d'aide à la conduite

Parmi les principaux types de dispositifs d'aide à la conduite existant depuis un certain temps, on peut citer de manière non-exhaustive :

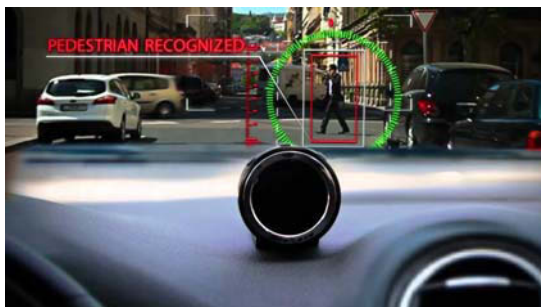
- l'ABS (ANTI-SKID BRAKING SYSTEM),
- l'ESP (ELECTRONIC STABILITY PROGRAM),
- l'aide à la navigation,
- le contrôle automatique de vitesse,
- la mesure de pression des pneus en ligne,
- la détection d'endormissement du conducteur.

Ce sont principalement des technologies ayant atteint un niveau de maturité suffisant pour qu'on les considère comme relevant purement de l'ingénierie.

De nouvelles fonctions plus avancées, se rapprochant de plus en plus de la conduite semi-automatique sont par exemple :

- le dépassement de véhicule en conduite semi-automatique,
- la conduite semi-autonome en suivant une voie de circulation,
- la détection automatique des panneaux indicateurs, de signalisation, et des feux de signalisation,
- la détection active de collision imminente,
- la détection de piétons,
- le rangement et la sortie autonome d'un véhicule dans un emplacement de stationnement (« parking valet »),
- la conduite en convoi.

Quelques exemples de telles méthodes, comme la détection de piéton et de panneau de limitation de vitesse sont présentés sur la figure 1.9. Un autre exemple, le suivi de route, est illustré sur la figure 1.10.



(a) Détection de piéton.



(b) Détection de limitation de vitesse.

FIGURE 1.9 – Exemples de détection visuelle d'éléments de sécurité, réalisés par la société MOBILEYE.

Ces éléments sont actuellement plus dans des thématiques de recherche que d'ingénierie, étant donné qu'aucune méthode validée en contexte réel n'a été qualifiée à notre connaissance. Différentes méthodes sont envisagées afin de permettre la réalisation de ces fonctions.



FIGURE 1.10 – Exemple de suivi de route. Source publique, DAIMLER.

1.2.2 Les véhicules autonomes

Les navettes autonomes présentées à la section 1.1.4 sont actuellement limitées à des applications en environnement contrôlé (site industriel), et à des vitesses faibles, ce qui fait que le risque, qui existe toujours, reste limité. Les perspectives à moyen et long terme des constructeurs automobiles montrent l'intention de certains d'entre eux de passer à une phase de commercialisation de masse de ce type de véhicules d'ici à une dizaine d'années.

Les technologies permettant ce passage à l'échelle industrielle soit sont encore in-existantes pour certaines, soit ne sont pas encore matures. Ceci rend cet objectif assez « stimulant » mais risqué. Un effort conséquent est donc consenti par nombre d'entre eux afin de rester en course le jour où ces applications seront commercialisables.

Plusieurs obstacles s'opposent à cet objectif :

- la capacité d'un robot à avoir « conscience » de son environnement est limitée malgré l'amélioration des techniques d'analyse de scène,
- la capacité d'un robot à se localiser en présence d'incertitude sur son environnement avec une qualité suffisante pour éviter les collisions et avec robustesse en utilisant les techniques actuelles n'est pas prouvée,
- la planification d'une trajectoire en temps réel et acceptable par un être humain peut se heurter à des décisions dans lesquelles un humain agit « par instinct », mais que l'on peut être incapable de modéliser pour la prise de décision par une machine (décision de dépassement par exemple),
- l'autonomie de décision d'un robot dans des situations où un risque vital existe est un problème ouvert, tant du point de vue technique qu'éthique.

L'état de l'art actuel consiste en une liste de techniques d'ingénierie, testées séparément et qui fonctionnent convenablement dans des cas précis. Leur concept est basé sur une programmation impérative des opérations. Le devenir de ces méthodes en cas de sortie des conditions d'utilisation prévues est indéterminé.

D'autres techniques plus limitées font appel aux techniques d'apprentissage, dans lesquelles l'objectif n'est pas d'exécuter des opérations fixées à l'avance, mais d'évaluer la validité d'une hypothèse tout en agrégeant de l'information. Ces techniques, plus flexibles, sont également extrêmement ardues à prouver d'un point de vue sécuritaire, notamment

en raison de l'impossibilité pour un être humain d'appréhender toutes les combinaisons possibles relatives au test du système et à sa validation.



FIGURE 1.11 – Exemple de classification sémantique des objets sur une scène routière. Source publique, DAIMLER.

Un exemple d'une telle méthode est présenté sur la figure 1.11. Une analyse sémantique du contenu d'une image y est réalisée grâce à un apprentissage. Chaque forme (être humain, voiture, sol, route, transport en commun, trottoirs, panneaux de signalisation) est catégorisée, localisée et répertoriée.

1.3 Contexte des travaux

Dans le domaine des véhicules intelligents, on distingue les thèmes suivants :

- l'estimation de l'état du véhicule (localisation),
- la prise de décision et la construction de la trajectoire du véhicule autonome,
- le calcul de commande et le suivi de la trajectoire précédemment définie.

Nos travaux se positionnent dans la catégorie relative à l'estimation de la localisation d'un véhicule autonome. Ceci peut se faire de différentes manières, mais l'une des plus semblable à ce que font les êtres humains consiste à se repérer par rapport à un environnement connu. Une carte peut permettre de se remémorer cet environnement.

Nos travaux ont donc pour objectif de pouvoir utiliser des cartes ayant un degré connu d'imprécision, contenant des informations potentiellement erronées, éléments manquants ou faux, dans un objectif de localisation robuste dans des environnements évolutifs et partiellement inconnus, tels les centres urbains.

A terme, un tel système aura pour objectif :

- qu'une flotte de véhicules autonomes de type navette puisse transporter des personnes dans un centre ville, entre deux points définis à l'avance par l'utilisateur, sur une carte existante, commerciale ou bien libre,
- qu'un système de coordination en gère l'organisation de manière à pouvoir dérouter l'un des véhicules si besoin,

- que chaque véhicule puisse communiquer avec les autres, par exemple en cas de panne pour redistribuer le trafic en question,
- de conduire en toute sécurité pour tous (usagers de la chaussée, sur les trottoirs et dans la navette).

Un tel système est complexe à plusieurs niveaux :

- au niveau localisation de chaque véhicule (basée sur une carte),
- au niveau localisation de la flotte de véhicules,
- au niveau commande locale de chacun des véhicules (trajectoire donnée comme objectif, évitement d'obstacles),
- au niveau organisation globale (quel véhicule va chercher quel usager à quel endroit pour l'amener où).

Nos travaux s'insèrent dans l'objectif « **localisation de chaque véhicule** », c'est à dire de pouvoir faire en sorte qu'un véhicule puisse se localiser malgré une carte potentiellement imparfaite puisque l'environnement change au cours du temps, que les cartes les plus précises comportent toujours des imprécisions et qu'elles ne peuvent être parfaites.

Nos objectifs de développement sont :

1. de pouvoir faire se localiser un véhicule dans une carte imprécise,
2. de pouvoir corriger la situation d'éléments fixes de l'environnement (appelés amers) lorsqu'ils sont incorrectement positionnés dans la carte en question,
3. d'être capable de détecter lorsqu'un amer précédemment cartographié est impossible à retrouver dans l'environnement (peut-être a-t-il disparu ?),
4. de pouvoir détecter dans l'environnement des amers qui ne correspondent encore à aucun élément cartographié, et à pouvoir les ajouter à la carte en question.

L'environnement dans un centre ville change en permanence, par exemple des travaux peuvent modifier un carrefour, un sens de circulation, détruire un immeuble ou en construire un autre. Ce sont autant d'informations qui pourraient être utilisées par un véhicule autonome pour se localiser ou prévoir une trajectoire de contournement. Une navette autonome doit pouvoir être capable de supporter de tels changements de l'environnement de manière robuste. Les quatre opérations mentionnées ci-dessus ont pour objectif de pouvoir rendre dynamique la carte de l'environnement et à permettre à un véhicule son utilisation comme sa mise à jour **en ligne**. Ces quatre objectifs sont ceux des travaux rapportés dans le présent mémoire.

Afin de pouvoir aboutir à cet objectif, on peut songer utiliser des cartes numériques de l'environnement. En effet, depuis une dizaine d'années, l'emploi de cartes numériques se généralise dans les domaines des transports (cartes utilisées par les aides à la navigation type « GPS », cartes numériques des zones urbaines par l'IGN et d'autres. . .). De nouveaux acteurs mettent à disposition du grand public de telles cartes obtenues par les mesures de milliers de contributeurs volontaires anonymes (on peut citer les contributeurs du projet OPENSTREETMAP, bien que d'autres projets du même type existent). Ce type de cartes permettraient de pouvoir accéder à des informations permettant à un véhicule de se localiser plus finement.

C'est dans ce cadre qu'il a été envisagé d'utiliser des cartes numériques de diverses sources, afin de n'être captif d'aucune, afin d'intégrer ces données dans un système de localisation de flotte de véhicules autonomes se déplaçant en zone urbaine. Ces cartes pourraient contenir, outre le tracé des voies de circulation, d'autres éléments aisément observables (enseignes lumineuses, murs de maçonnerie, panneaux de signalisation. . .),

qui, par leur observation, permettraient à un véhicule de s’assurer de sa position, ou bien de la préciser.

Dans cette optique, nous apportons plusieurs contributions, à l’état de l’art actuel. Afin de les énoncer, nous allons détailler la stratégie de localisation envisagée. Tout d’abord, de par la nature des cartes envisagées afin de permettre la localisation d’un véhicule, nous proposons l’utilisation de cartes contenant divers types d’éléments (amers), quels qu’ils soient. Ces éléments peuvent être, dans le concept envisagé, de différents types (poteaux, murs, arrêtes verticales de murs). Dans le cadre d’une carte de source variable, on ne peut garantir la qualité des données, qu’il faudra bien corriger à un moment ou un autre. Cette correction de cartes peut être faite par exemple en permettant à un véhicule de corriger les amers erronés d’après ses propres observations.

La stratégie classique de gestion d’une telle carte consiste à utiliser une description dans laquelle ces différents éléments de la carte sont interdépendants. Ceci pose des problèmes de mise à l’échelle, ou bien de gestion des interdépendances, qui peuvent être complexes.

Notre **première contribution** est de proposer de gérer chacun des éléments cartographiés de la carte indépendamment des autres, par l’utilisation d’une méthode d’estimation appelée **filtre à Intersection de Covariance Partitionnée**. Ce type de filtre permet de gérer les corrélations entre éléments tout en ayant la simplicité de rendre l’état de chacun *indépendant* de celui des autres.

Une autre manière de corriger une carte extérieure de l’environnement peut consister à supprimer un amer que l’on ne détecte plus (par exemple un mur que l’on vient de détruire). La gestion des corrélations de l’existence des amers pose également problème comme c’était le cas dans le paragraphe précédent. Nous proposons une optique permettant de gérer **indépendamment** les uns des autres l’existence de chacun de ces différents amers de la carte.

Notre **seconde contribution** consiste à le faire grâce à l’utilisation d’un **réseau hiérarchique**, qui limite l’ampleur du phénomène de corrélation entre l’amer considéré, et d’autres éléments de la carte (véhicule ou autres amers).

Un troisième objectif est de pouvoir ajouter à la carte de l’environnement utilisée, et potentiellement erronée, d’autres amers qui existent, mais n’y figurent pas encore. Afin de permettre de réaliser cet objectif, les données issues des capteurs doivent être alignées **au mieux** sur la carte existante.

Notre **troisième contribution** consiste à réaliser cet alignement des données issues des capteurs sur la carte en utilisant le formalisme de notre première contribution, le **filtre à Intersection de Covariance Partitionnée**, ce qui est une approche non-standard de localisation.

Ce sont ces trois contributions que nous allons détailler dans le présent rapport et que nous allons expliciter. Après une rapide bibliographie de la thématique de « cartographie » dans le sens le plus général possible, nous présenterons l’approche du problème qui nous a paru la meilleure. Nous décrirons au chapitre 4 les théories que nous avons utilisées dans le cadre de ces travaux. Le chapitre 5 est dédié à la description de la méthode générique de localisation utilisant une carte et ayant servi de support à nos travaux. Les chapitres 6, 7 et 8 décrivent nos contributions applicatives aux problématiques mentionnées ci-dessus.

Chapitre 2

Localisation et cartographie pour la robotique mobile

La navigation d'une flotte de véhicules dans un environnement ouvert pose plusieurs problèmes, dont celui de la localisation de celle-ci. C'est à l'analyse du problème de localisation que nous allons nous attacher dans ce chapitre. On peut avoir collationné de l'information relativement à l'environnement immuable par des observations, quelles qu'en soit la nature ou l'origine, dans une carte. On peut alors se servir de points de repère référencés par celle-ci afin de connaître sa position dans un cadre donné, et donc de pouvoir s'y référer dans le temps, ou l'utiliser comme base commune de localisation dans le cadre de travail en équipe (flotte de véhicules par exemple).

Que ce soit en intérieur ou en extérieur, un environnement évolue potentiellement au cours du temps. Lorsque l'environnement change, le contenu de la carte doit alors refléter ce changement de manière tant quantitative que qualitative, afin de rester à jour. Ensuite, la gestion d'une flotte de véhicule, comme d'une quantité évoluant d'amers de nature différente, implique des techniques de fusion de données permettant d'utiliser un nombre variable d'éléments de la carte. Enfin, la source d'information dans une carte peut évoluer en fonction des circonstances et de la localisation du véhicule (milieu urbain ou non, inséré dans une flotte de véhicules à proximité ou seul dans la zone, du type et du nombre de capteurs disponibles sur le véhicule en question).

Dans ce chapitre, nous nous intéresserons à plusieurs thèmes. Tout d'abord, nous nous intéresserons aux techniques de cartographie existantes. Ensuite, les techniques de localisation utilisant une carte de l'environnement seront investiguées. Enfin, une analyse des techniques existantes de fusion de données pour la gestion d'ensemble d'une carte d'éléments observables sera faite.

2.1 Introduction

On pourrait penser que depuis l'invention et le développement du « Global Positioning System » plus connu sous le nom de *NAVSTAR-GPS*, le problème de la localisation d'un véhicule ou d'un élément fixe à la surface de la **Terre** est un problème résolu. Bien que dans certains cas, cette mesure puisse aider à la localisation, elle ne permet pas à elle seule la localisation d'un véhicule afin de lui permettre de naviguer de manière autonome dans la circulation.

Premièrement, la précision de la localisation que donne le signal *public* du GPS est actuellement estimée à environ 5 *m*. Dans le cadre des applications de conduite auto-

nome d'un véhicule, cette précision est supérieure à l'écart existant entre deux voies de circulation, sans parler de la distance latérale entre deux véhicules circulant sur le même axe routier. De plus, le signal émis par le système GPS peut être altéré ou stoppé par des obstacles (effet de « canyon urbain », absence du signal à l'intérieur de bâtiments, parkings souterrains...), et n'est donc pas toujours disponible.

Dans le cas où une analyse permettrait au véhicule de suivre sa trajectoire de manière autonome à partir de l'observation du « contexte routier » (comme le font les humains), une telle localisation serait suffisante. Bien que des travaux industriels existent dans ce domaine (DAIMLER par exemple, voir la figure 1.11), ces techniques ne sont pas considérées comme matures d'un point de vue sécuritaire (sûreté des analyses effectuées, et de l'interaction avec les manœuvres qui en découlent).

La solution choisie actuellement est donc de faire suivre au véhicule une trajectoire **précise au décimètre** dans un référentiel lié à la Terre, afin de se positionner précisément par rapport à la route sans dépendre du contexte de cette dernière. L'incertitude de positionnement donnée par le système GPS standard, de 5 m, est alors insuffisante pour assurer un tel degré de précision.

Secondement, dans le cadre de la conduite d'une flotte de véhicules autonomes, il est important que deux véhicules puissent comparer leur positionnement sur une base absolue et plus précise que les 5 m évoqués précédemment, afin d'éviter collisions et indications erronées entre véhicules, et permettre une évolution coordonnée.

De ce fait, parmi les solutions possibles, une localisation basée sur des éléments fixes de l'environnement peut permettre d'améliorer cette situation. Ce type de technique de localisation est en tout point similaire aux techniques utilisées depuis l'antiquité par l'humanité sur terre et sur mer. Ce type de méthode utilise des éléments préalablement localisés (même de manière approximative) et historiquement notés sur un support physique plan. La donnée essentielle de ce type de cartes est qu'elle ne contient que des éléments « fixes ». On peut donc repasser à un emplacement préalablement noté (même des années auparavant) simplement en se référant aux éléments immuables de l'environnement notés sur cette carte.

Cette simple constatation a des répercussions directes sur la manière dont on peut chercher à faire évoluer une flotte de véhicules autonomes. On peut se demander comment obtenir une telle carte de l'environnement ; doit-on la faire réaliser de manière externe et indépendante ou la faire générer par des véhicules autonomes au cours de leurs missions. On peut également se demander quels types d'objets cette carte peut ou doit contenir afin d'être utilisable au maximum. Enfin, le niveau de précision atteignable, *absolue* (un élément par rapport à la planète **Terre** par exemple) ou *relative* (deux éléments cartographiés l'un envers l'autre) doit également faire l'objet d'une analyse.

Dans ce chapitre, nous allons tout d'abord expliquer pourquoi une source de données fixe et absolue est souhaitable pour qu'un robot mobile ou un véhicule puisse se localiser. Nous nous focaliserons ensuite sur les capteurs et techniques permettant la localisation d'un tel robot ou véhicule dans une carte. Nous nous intéresserons ensuite aux différentes manières de cartographier l'environnement, de manière statique et hors-ligne comme de manière dynamique et en temps-réel. Nous nous intéresserons enfin aux problèmes techniques qui peuvent limiter cette capacité à naviguer dans une carte, ou à réaliser une telle carte.

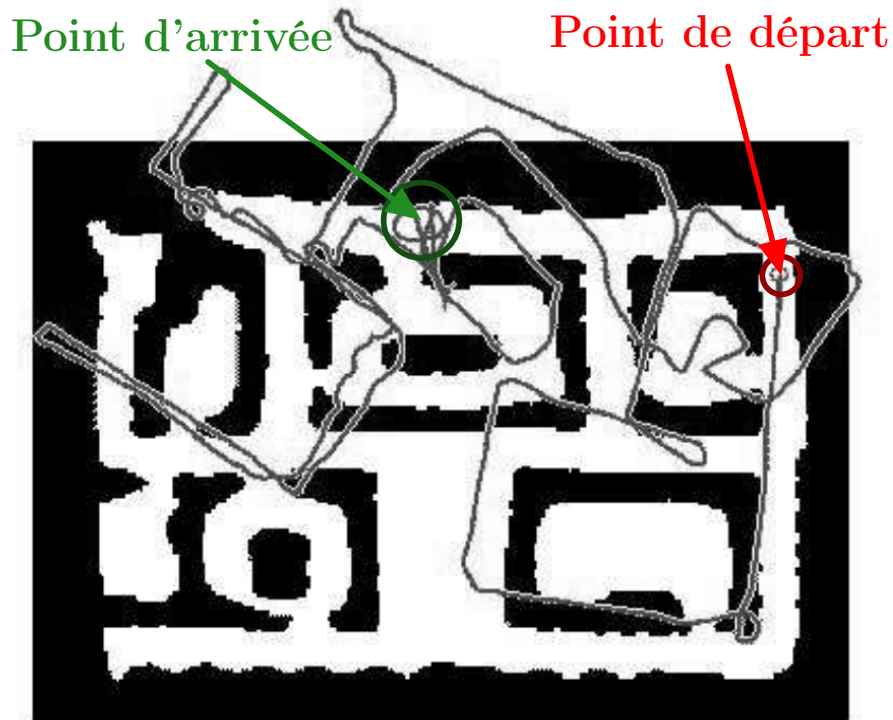


FIGURE 2.1 – Exemple de dérive de la localisation d’un robot par pure intégration d’estimations de l’évolution de son état (reproduit d’après [Thrun et al., 2002](#), figure 1a).

2.2 Principaux problèmes liés à la localisation d’un robot mobile

Dans cette section, nous prenons comme référence ([Thrun et al., 2002](#), section 3), qui reste pertinente malgré son ancienneté.

2.2.1 Erreurs intégrales de positionnement

Supposons qu’un robot autonome connaisse parfaitement sa position. Afin de se déplacer, celui-ci doit choisir une action à réaliser, puis en fonction de celle-ci, calculer des commandes afin de le faire se déplacer, puis enfin estimer sa nouvelle position dans l’espace. Si les deux premiers éléments sont en dehors du périmètre des travaux rapportés ici, le troisième reçoit tout notre intérêt.

On peut en tant qu’être humain se repérer dans l’espace à partir de ses propres mouvements et en connaissant sa position initiale. C’est par exemple ce que nous faisons lorsque nous nous déplaçons dans le noir, pour chercher un interrupteur par exemple. Mais chacun sait que plus nous nous déplaçons ainsi, sans autre source de données nous permettant de nous localiser, moins notre position est certaine.

La problématique est la même pour un robot mobile. La commande calculée par un robot peut par exemple contenir une vitesse de déplacement et une vitesse de rotation. Une caractéristique commune à toutes ces informations, quel que soit le format sous lequel elles sont calculées, est qu’elles indiquent (et renseignent) sur le *changement d’état* du robot, plutôt que sur l’état atteint à la fin de la manœuvre. Le mouvement réellement effectué

par le robot mobile diffère toujours en raison de la différence entre la commande calculée et la commande appliquée réellement. Il s'en suit un écart entre évolution prédite et mesurée. Cet écart de la commande est toujours constitué d'une composante de moyenne non-nulle (appelée *biais*, ou, dans le cas où elle est lentement variable, *dérive*) et d'une composante aléatoire de moyenne nulle (appelée *bruit*). Cet écart est additionné (lors d'un processus d'*intégration numérique*) pour obtenir le nouvel état du robot et cause la dérive de l'estimation de l'état du robot par rapport à l'état réel.

Le biais augmente de manière systématique l'évolution dans un certain sens de cette estimation de l'état du robot. Par exemple, sur la figure 2.1, la position estimée du véhicule, partant du bord en haut à droite de la carte, dévie systématiquement vers la droite lors de son déplacement. A la fin du déplacement, on observe que ni l'orientation du véhicule, ni sa position ne sont correctes.

De plus, le bruit aléatoire, qui s'additionne au biais, augmente l'imprécision de la localisation du robot, par le biais d'un processus dit *de marche aléatoire*. On peut voir que l'ellipse matérialisant ce bruit de moyenne nulle et obtenue à la fin du déplacement (milieu-haut de la figure 2.1) est plus large que celle de départ (bord haut-droit de la même figure).

Ces deux sources d'erreur s'ajoutent l'une à l'autre. C'est pourquoi, dans le cadre de la commande d'un véhicule autonome, il est indispensable de disposer d'une source de localisation de type *absolu* de manière à pouvoir borner cette erreur par celle de la mesure associée à cette source de données. On comprend bien que, dans le noir, on dispose d'une information absolue lorsque l'on parvient à toucher un mur. C'est bien l'intérêt d'une carte que de pouvoir, par identification des éléments qui la constituent (on sait quel mur on vient de heurter) avec la position d'éléments identifiables par les capteurs du robot ou du véhicule (par le sens du toucher, on sait quelle est notre position par rapport au mur identifié). Ceci nous amène au sujet du contenu de la carte de l'environnement.

2.2.2 Formalisation des données dans une carte

Pour un robot mobile d'intérieur, on peut imaginer une carte comme un dessin approximatif des cloisons dans une maison sur un schéma 2D. Ce type de représentation est appelée sémantique, car seuls des éléments de haut niveau ayant un fort degré de symétrie (les cloisons) y sont représentés. Elle ne requiert que quelques nombres dans une carte, ces nombres pouvant représenter les coordonnées des angles de chaque pièce.

Une carte plus élaborée peut utiliser un plus grand nombre de ces segments, modélisant des volumes plus faibles, ou prenant en compte l'épaisseur des murs. Une centaine de nombres représentant chaque face de tout volume pérenne serait typiquement nécessaire dans ce cas.

On peut imaginer une carte 2D qui indiquerait si un emplacement au sol est, à une hauteur déterminée, occupé, vide, ou si l'on ne peut déterminer son état. Ce type de carte s'appelle une carte d'occupation bidimensionnelle. Une telle carte, reproduite d'après (Manfredi et al., 2016) est visible sur la figure 2.2. Le nombre de valeurs numériques pour stocker une telle carte dépasse la dizaine de milliers et peut aller à la centaines de milliers, en fonction de la résolution au sol de la carte.

Dans le cas où le robot en question doit être capable de chercher un objet dans un placard, une cartographie 3D de haute résolution serait nécessaire, comme par exemple dans (Manfredi et al., 2016). Une telle carte en rendu 3D–vue de dessus est reproduite, toujours

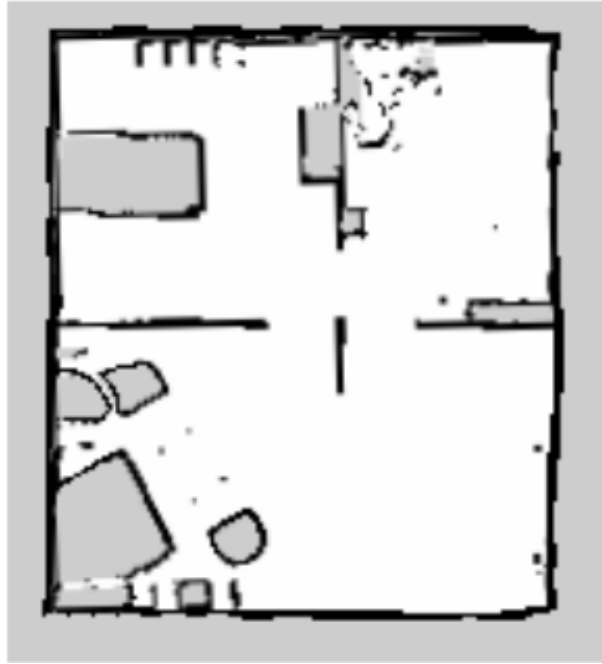


FIGURE 2.2 – Exemple de carte d’occupation 2D sur un appartement reconstitué en laboratoire, vu de dessus, d’après (Manfredi et al., 2016, figure 2b).

d’après (Manfredi et al., 2016), dans la figure 2.3. Le nombre de valeurs numériques pour une telle carte dépasse facilement le million.

D’après (Thrun et al., 2002, section 3), si l’on veut analyser chaque technique de représentation d’une carte afin de les comparer, il est nécessaire de prendre en compte ce nombre de valeurs comme des composantes d’un vecteur dont le nombre de dimensions indique la complexité de la représentation sous-jacente.

2.2.3 Problèmes d’association carte–observations

À la fin de la section 2.2.1, il était question d’associer la détection à tâtons d’un mur dans le noir avec un mur de la carte. Imaginons que la pièce dans laquelle on se situe soit grande, avec une géométrie complexe, et qu’au départ, on se situe au centre de celle-ci. Lorsque l’on va se déplacer dans le noir, l’estimation de notre position réelle sera d’autant plus imprécise que l’on se sera déplacé loin. Une erreur d’association du mur détecté avec le mur que l’on pense avoir trouvé est de ce fait possible.

2.2.4 Carte et évolution de l’environnement avec le temps

L’environnement change comme chacun le sait. Dans un environnement urbain, de nouveaux bâtiments se construisent, d’autres sont rasés, de nouveaux axes routiers sont percés. Dans la nature, les arbres croissent, meurent, puis sont abattus. Certes, cette évolution est lente, mais elle se produit, et il est donc indispensable de remettre à jour la carte en fonction des nouvelles données relatives à la nouvelle configuration des zones cartographiées.

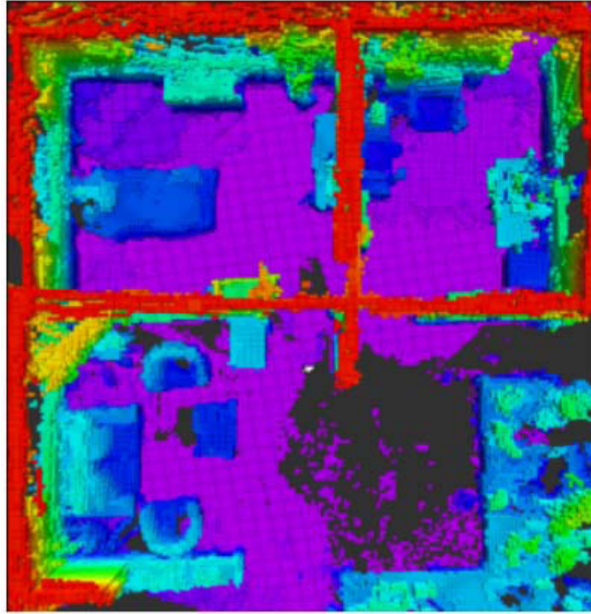


FIGURE 2.3 – Exemple de carte 3D sur le même appartement que sur la figure 2.2, vu de dessus, d’après (Manfredi et al., 2016, figure 3a).

2.3 Types de capteurs utilisés en robotique mobile

Nous allons ici donner une liste non-exhaustive des différents types de mesures que l’on peut avoir dans le domaine de la robotique mobile. Nous avons parlé dans la section 2.2.1 de types de données ou de mesures renseignant sur l’évolution du robot. Ces données sont appelées *proprioceptives*. Nous avons également mentionné les capteurs et données informant le robot sur son environnement extérieur, et appelés *extéroceptifs*.

2.3.1 Données et capteurs proprioceptifs

- La plupart des données à caractère proprioceptif peuvent être listées ainsi :
- Position ou commande d’actionneurs liés au mouvement de l’*ego*-véhicule,
 - Capteurs de type *inertiels*.

Position et commande d’actionneurs

La plupart des véhicules de tourisme à 4 roues sont habituellement basés sur un train directeur et moteur de deux roues, positionné à l’avant, et un train non directeur à l’arrière. Ces deux données définissent la géométrie dite d’*Ackermann*. Un exemple typique d’une telle structure est présentée sur la figure 2.4 pour un véhicule à quatre roues.

La géométrie d’*Ackermann* consiste en un train avant orientable, et un train arrière fixe. L’orientation des roues avant est réalisée de telle manière que les axes de rotation de chacune, ainsi que l’axe de rotation arrière soient concourants en un unique point, le centre de rotation instantané du véhicule. Ceci se produit quelle que soit la valeur du braquage des roues avant.

Les deux paramètres permettant de calculer l’évolution de l’état du véhicule sont l’angle de braquage équivalent δ d’une part, et la vitesse instantanée du centre du train

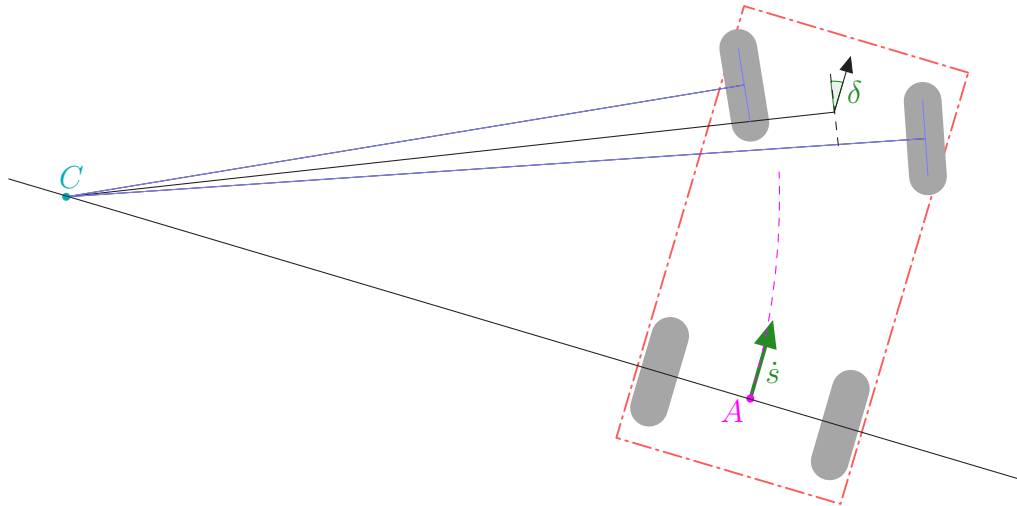


FIGURE 2.4 – Géométrie d’Ackermann pour un véhicule à quatre roues. Le train avant se commande en angle de braquage, noté δ . Le train arrière peut mesurer la vitesse d’avancée du véhicule, notée \dot{s} . Le centre de l’essieu arrière, noté A , évolue en rotation instantanée autour du centre noté C .

arrière, notée \dot{s} . Une mesure de ces valeurs (ou à défaut, une commande de ces deux grandeurs) permet donc d’estimer l’évolution par unité de temps de l’état du véhicule.

Capteurs de type inertiel

L’exemple type d’un tel capteur est le gyromètre, qui mesure une vitesse angulaire par rapport à un référentiel galiléen. Ce capteur informe sur la variation par unité de temps de l’orientation du véhicule. D’autres capteurs se basent sur la mesure des accélérations du véhicule, afin d’estimer les changements de vitesse et/ou de trajectoire de ce dernier. Ces deux types de capteurs sont en général utilisés dans les cas où une mesure directe de la commande au niveau du véhicule n’est pas possible, n’est pas suffisamment précise ou bien ne prend pas en compte tous les degrés de liberté d’évolution de l’état du véhicule. Dans ce cas, ils remplacent le vecteur de commande mesuré au niveau des essieux du véhicule, par exemple.

Ils sont également utiles dans le cas de glissements de ce dernier par rapport au sol. On peut penser à un véhicule dérapant sur de la terre meuble. Les données fournies par les trains du véhicule ne représentent correctement la réalité que dans le cas où un contact solide s’établit avec le sol; on imagine aisément que lorsque les roues motrices patinent, la vitesse de rotation de l’arbre moteur peut être élevée tandis que le véhicule reste immobile par rapport au sol. Les capteurs inertiels sont capables d’identifier de tels scénarios, lorsque couplés aux grandeurs de commande du véhicule, et sont typiquement utilisés sur les système ESP des véhicules automobiles.

2.3.2 Capteurs extéroceptifs

Ces capteurs informent de manière directe le véhicule sur son environnement, et donc de manière indirecte sur son état ou sa localisation. On peut citer parmi ceux-ci :

- le capteur RADAR,
- le capteur LIDAR 2D mono nappe,
- le capteur LIDAR 3D multi nappes,
- la caméra monoculaire,
- la caméra binoculaire à champs recouvrants...

Capteur RADAR

Un capteur RADAR fonctionne par émission et réception d’ondes radiofréquence. Deux types principaux de RADAR existent dans le domaine de la robotique :

- les RADARs automobiles, en règle générale fixes, qui mesurent un écart de position et de vitesse par rapport au capteur (Kellner et al., 2014; Viquerat et al., 2008),
- les RADARs panoramiques, dont l’antenne est en rotation autour d’un axe vertical, qui permettent de reconstruire une image plane de l’environnement (El Natour, 2016; Vivet et al., 2012).

Les RADARs automobiles permettent à un véhicule d’identifier des éléments en déplacement relatif par rapport à ce dernier. Dans le cas d’éléments fixes, il s’agit de mesurer la vitesse du véhicule (Kellner et al., 2014). Si l’on détecte un obstacle sur la ligne de trajet, on peut détecter un danger de collision (Viquerat et al., 2008) et prévoir une manœuvre d’évitement.

Un RADAR panoramique permet de cartographier l’environnement à proximité du capteur. Un exemple de reconstruction de données d’un RADAR panoramique est présenté sur la figure 2.5 et est tirée des travaux décrits dans (El Natour, 2016). On observe la structure des bâtiments d’une zone urbaine, ainsi que les axes routiers, comme des zones rectilignes vides d’échos RADAR.

La portée de tels capteurs est dans le domaine de la robotique de l’ordre de 100 à 200 m.

Capteur LIDAR 2D mono-nappe

Un capteur LIDAR utilise une lumière LASER infrarouge, mais fonctionne sur le principe du RADAR impulsionnel originel, dont le principe est la mesure de temps de vol, principe également utilisé dans les RADARs de contrôle aérien. Une onde électromagnétique est émise à un instant t_E à partir du capteur. Cette onde électromagnétique se propage de manière rectiligne jusqu’à heurter un hypothétique objet, situé dans son champ. Une partie de cette onde électromagnétique est réfléchiée et/ou rétro diffusée, et revient sur le capteur du LIDAR, où l’on mesure sa date de détection t_D . La distance d_O entre l’objet réfléchissant et le capteur LIDAR est donnée par l’équation 2.1.

$$d_O = \frac{c \cdot (t_D - t_E)}{2} \quad (2.1)$$

c célérité de la lumière

La plupart des capteurs LIDAR, outre cette distance, mesurent également la réflectivité de l’objet. Cette information peut être utilisée afin d’éliminer un éventuel bruit de

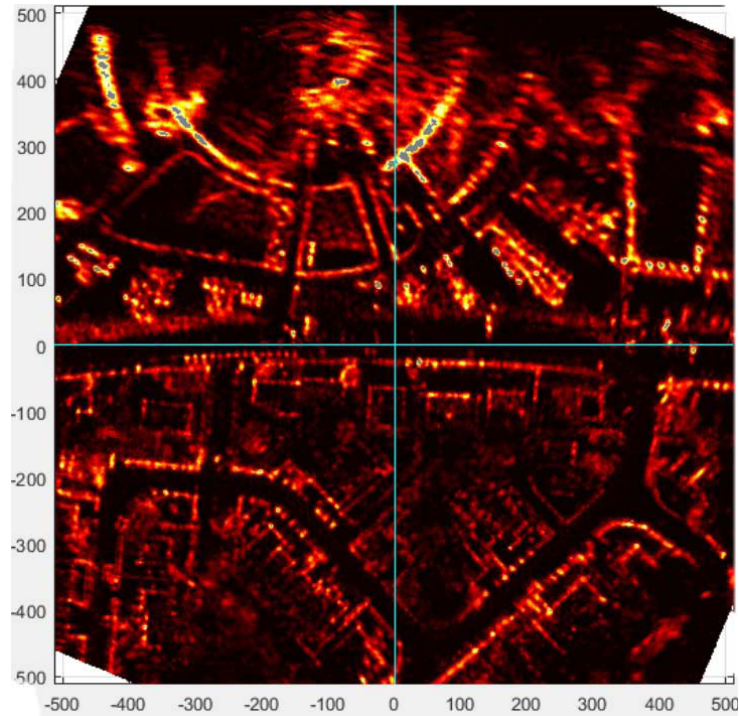


FIGURE 2.5 – Exemple de données mesurées par un RADAR et reconstruites en 2D, d’après (El Natour, 2016, figure 1.1).

détection ou afin d’identifier la nature de l’objet. Par exemple, on pourrait détecter s’il s’agit d’une paroi transparente, une fenêtre, ou d’une paroi plus réfléchissante, un mur.

Le faisceau émis effectue un mouvement plan autour de son axe de révolution, ce qui permet d’obtenir une série de points de mesures dans un espace bidimensionnel de même nature. Un exemple d’une telle série de données, prise dans un plan horizontal, est présentée sur la figure 2.6. On observe les divers murs, identifiables à leur structure en ligne droite.

Les capteurs LIDAR plans peuvent également être montés sur un plan orthogonal au sol (détection de bords de trottoirs), ou encore sur un plan orienté à 45° afin d’identifier des volumes non-plans sur le chemin du véhicule (nids-de-poule ou cassis, par exemple). La plupart des modèles disponibles de manière commerciale ont une portée allant de quelques mètres (LIDAR utilisés en environnement intérieur), à quelques centaines de mètres (LIDAR prévus pour un emploi en environnement extérieur). La faible portée des capteurs LIDAR est en relation directe avec la puissance optique émise par le LASER. Pour des raisons de sécurité évidentes (dommages potentiels à la cornée et à la rétine des passants), la puissance incidente du faisceau fait l’objet de sévères limitations.

Capteur LIDAR 3D

Le principe d’un capteur LIDAR 3D est sensiblement identique à celui du LIDAR 2D. La différence notable consiste en l’utilisation de plusieurs pinceaux LASER, orientés en élévations régulièrement réparties sur une échelle symétrique par rapport au plan normal avec l’axe de rotation du capteur, comme dessiné sur la figure 2.7.

Un exemple de reconstitution en 3D de points obtenus par une telle méthode est visible sur la figure 2.8. Le capteur, de la marque *Velodyne* HDL 64 et utilisant 64 nappes

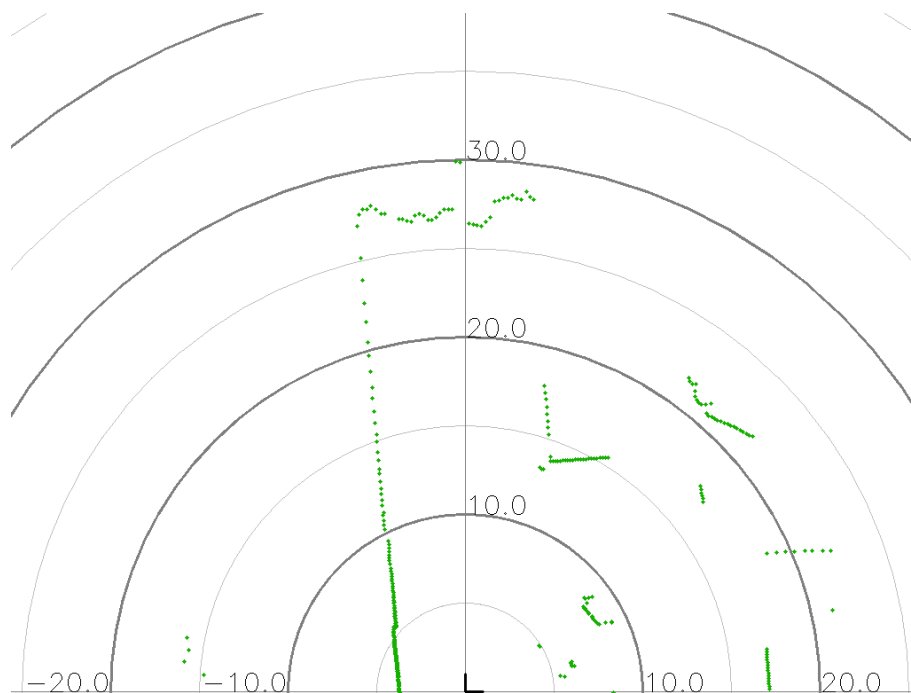


FIGURE 2.6 – Un exemple de données obtenues par un capteur LIDAR 2D plan. Les cercles concentriques indiquent la distance (en mètres) des échos au capteur LIDAR. On observe bien la structure rectiligne des murs.

LASER, est positionné sur le toit d'un véhicule de loisir, utilisé pour l'acquisition des données. On reconnaît la structure de révolution des données générées par le capteur en rotation. Chaque nappe distincte correspond à une élévation distincte du faisceau LASER par rapport à l'horizontale.

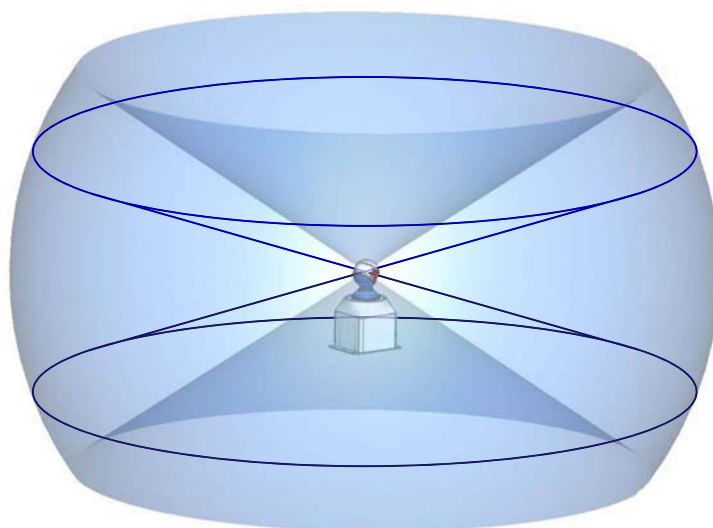


FIGURE 2.7 – Nappes LASER générées par un LIDAR 3D. Image originale téléchargée à partir du site <http://www.ocularrobotics.com>.



FIGURE 2.8 – Un exemple de données issues à partir d’un LIDAR 3D utilisant 64 nappes LASER. Données reproduites d’après (Levinson and Thrun, 2010, figure 3b).

Les capteurs LIDAR, 2D comme 3D, sont robustes en règle générale, mais néanmoins sensibles aux conditions météorologiques (brouillard, pluie, pluie verglaçante, givre ou neige) qui génèrent de faux échos ou atténuent le faisceau LASER.

Capteur caméra monoculaire

Une caméra est un capteur bidimensionnel. Son fonctionnement est basé sur l’optique géométrique, et peut être modélisé par exemple avec le modèle *sténopé* (*pinhole* en anglais), ou bien le *modèle unifié*. Dans le cas du modèle *sténopé*, on obtient une image d’un environnement 3D en réalisant une projection centrale sur le plan focal de la caméra. Ce modèle est présenté sur la figure 2.9.

Utilisant une représentation bidimensionnelle, une partie de l’information spatiale sur les éléments observés est perdue. La position d’un objet projeté sur le plan focal de la caméra est équivalente à une donnée de position angulaire dans les deux directions normales à l’axe de vue. La donnée profondeur est totalement supprimée, à la différence du capteur LIDAR que nous venons de voir. L’utilisation d’une caméra embarquée comme moyen de localiser un véhicule autonome est de ce fait soumise à quelques limitations (fonction d’observation fortement non-linéaire, possible ambiguïté d’association et sur la profondeur pour des éléments voisins au niveau image).

Néanmoins, les caméras sont des capteurs de prix accessibles, que l’on peut monter quasiment sans contrainte sur un véhicule autonome. Ce faisant, ce sont des capteurs largement utilisés dans la communauté robotique. Plusieurs types de méthodes de localisation, telles le « Simultaneous Localization And Mapping » visuel (SLAM visuel), basées soit sur un filtre de Kalman (voir plus loin la section 4.7.2) soit sur un *ajustement de faisceau* (se référer à la section 2.4.8, et à (Strasdat et al., 2010) pour un exemple de réalisation) sont spécifiques de ce type de capteur.

Pour ce type de mesure, les conditions d’illumination sont très importantes. Sur les caméras de type *Charge-Coupled Device* (CCD), l’effet dit de « blooming » peut se produire, voir la figure 2.10. Les caméras de type *Complementary Metal–Oxide–Semi-conductor*

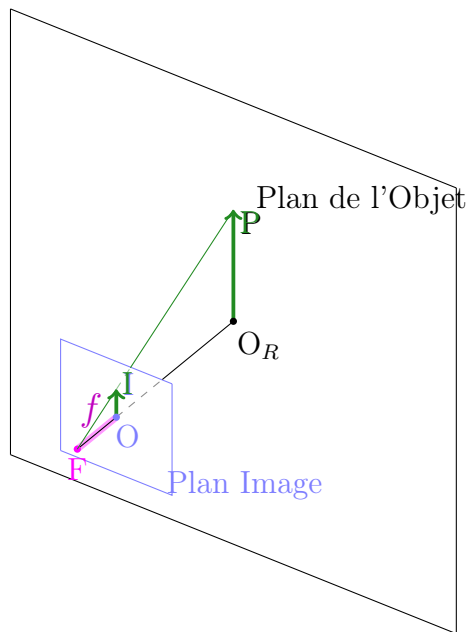


FIGURE 2.9 – Modèle *sténopé* de caméra 2D. Le point P représente l'objet, le point I son image. F est le foyer, et O le centre optique de la caméra. La variable f représente la distance focale de la caméra.

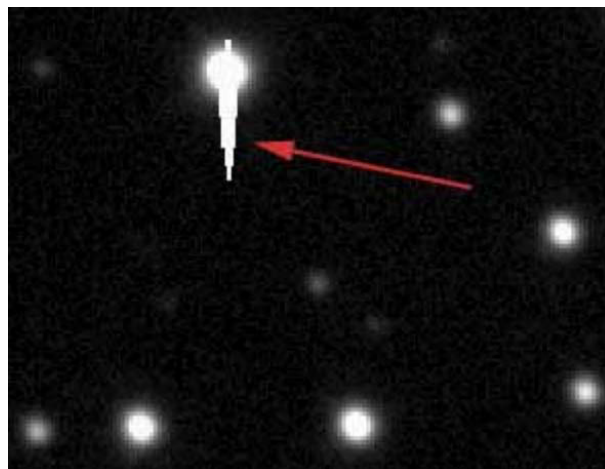


FIGURE 2.10 – Exemple d'effet de « blooming » sur un capteur CCD. Téléchargé à partir du site <http://www.adimec.com>.

(CMOS) s'affranchissent de ce type de problème. Par contre, la dynamique de mesure doit être correctement ajustée en fonction de l'illumination, celle-ci variant (potentiellement rapidement) en fonction de l'orientation du véhicule et de la luminosité ambiante. En extérieur, les caméras sont sensibles aux conditions d'éclairage (soleil de face, entrée ou sortie de tunnel, éclairage naturel ou artificiel), à la pluie et au brouillard comme à la condensation.

Caméra binoculaire

Une caméra binoculaire est un système basé sur l'emploi de deux caméras, d'axes optiques quasiment parallèles, et décalées d'une distance suffisamment importante pour permettre la triangulation, mais suffisamment faible pour que les champs de vue de chacune soient recouvrant. Un exemple de travail utilisant un tel capteur, conjointement à un capteur de type RADAR est présenté dans (El Natour, 2016). Ce type de capteur est également sensible aux mêmes conditions atmosphériques et d'illumination.

2.3.3 Exceptions

Dans ce qui précède, nous avons supposé que les capteurs extéroceptifs retournaient des d'informations sur l'état du véhicule provenant de leur environnement. Nous avons également supposé que les capteurs proprioceptifs renvoyaient des informations provenant du véhicule et relatives à l'évolution de l'état de ce dernier.

Certains capteurs sont à mi-chemin de ces deux définitions. Par exemple, un magnétomètre retourne une information sur l'état du véhicule (et non son évolution), on pourrait donc le classer parmi les capteurs extéroceptifs. Néanmoins, deux points de vue relatifs à la nature de la source de données existent :

1. le magnétomètre donne une information « venant du véhicule » (son orientation), d'où un potentiel caractère **proprioceptif**, ou alors,
2. le magnétomètre mesure un élément de l'environnement (le champ magnétique) qui, s'agissant du champ magnétique terrestre, peut fournir une information d'orientation (capteur vu comme **extéroceptif**).

De la même manière, un capteur GPS donne des informations sur un état du véhicule (les coordonnées cartésiennes bidimensionnelles d'un point du véhicule). Mais le classement du GPS peut être fait ainsi :

1. le couple antenne/capteur GPS peut être vu comme retournant une donnée relative à l'état interne du véhicule (sa position horizontale approximative), donc vu comme **proprioceptif**, ou bien,
2. ce couple antenne/capteur GPS peut être vu comme un récepteur d'un signal radiofréquence, donnant des indications de temps et des informations permettant de les convertir en coordonnées sphériques à la surface de la **Terre**, d'où un capteur **extéroceptif**.

Nous prenons pour ces deux capteurs le choix de les considérer comme **extéroceptifs**. Ce choix est fait pour deux raisons. Prenons l'exemple du magnétomètre. Tout d'abord, celui-ci peut mesurer un champ magnétique, résultat de la superposition du champ terrestre, mais potentiellement également celui venant d'une autre source, par exemple un moteur électrique ou un matériau ferromagnétique de l'environnement ayant un fort champ rémanent (fer à béton ou autre). Il peut donc dans certains cas ne pas mesurer la grandeur voulue. Secondement, si un champ magnétique extérieur, dont la source est fixe par rapport au sol, génère une telle perturbation, celle-ci est liée à la localisation géographique du véhicule, donc à l'environnement extérieur. On peut mener le même raisonnement avec le capteur GPS en remplaçant un champ magnétique se superposant, et une zone fixe de l'espace où le champ magnétique terrestre n'est pas mesurable, respectivement, par des perturbations radiofréquence, et un bâtiment occultant le signal GPS.

Nous allons détailler brièvement les caractéristiques de ces deux capteurs dans ce qui suit.

Capteur magnétomètre

Les capteurs magnétomètres peuvent être de plusieurs types, mais la plupart de ceux utilisés en robotique mobile sont des capteurs intégrés fonctionnant sur la base de l'effet Hall. Ils sont constitués d'une puce de silicium, comme un circuit intégré, combinant des circuits électroniques de mesure (amplificateurs, convertisseurs analogiques numériques) intégrés à un processeur de signal numérique (également intégré), couplé à une structure de type Hall, qui génère le signal provenant du champ magnétique (la tension de Hall). Un exemple de réalisation d'une structure Hall sur une puce silicium est présenté dans (Racz et al., 2004).

La plupart des capteurs magnétomètres à effet Hall sont 3-axes, c'est à dire qu'ils retournent les 3 composantes du champ magnétique selon une base de vecteurs proche de l'orthonormale.

Capteur GPS

Un capteur GPS est un appareil de géolocalisation sur la surface de la **Terre**, utilisant des signaux radiofréquence sur deux bandes dites L1-L2 (dont l'une, L2, émet un signal crypté) émis par une constellation de satellites, la constellation NAVSTAR, opérée par le *DoD*, le « Department of Defence » américain. Celle-ci est constituée d'une vingtaine de satellites se déplaçant sur plusieurs orbites d'inclinaison différente appartenant à une coque sphérique positionnée à une altitude d'environ 20 000 *km*. D'autres systèmes de ce type sont déjà en service dans d'autres pays, notamment GLONASS (FÉDÉRATION DE RUSSIE), BAÏDU (RÉPUBLIQUE POPULAIRE DE CHINE), ou vont l'être dans un avenir relativement proche, comme GALILEO (UNION EUROPÉENNE).

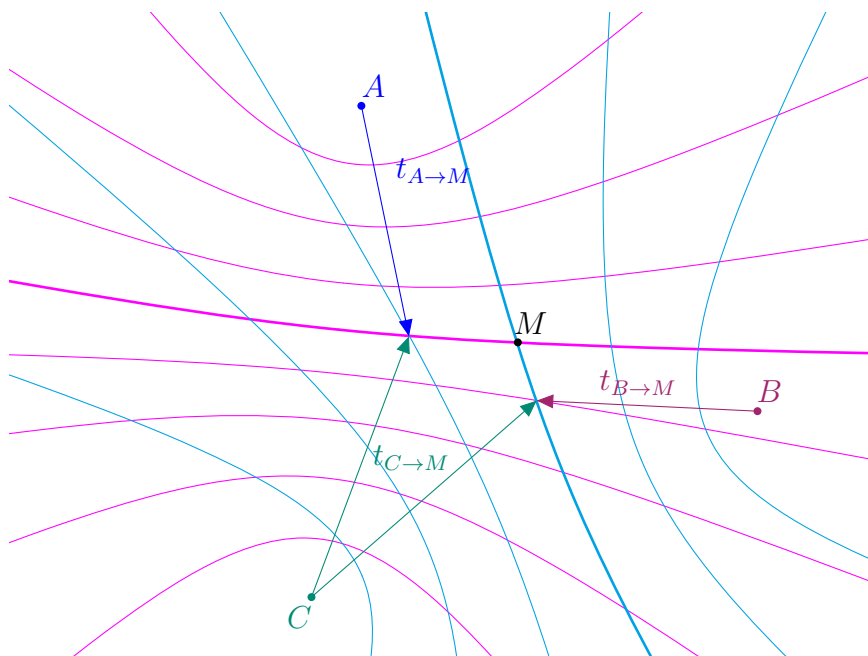


FIGURE 2.11 – Exemple de multilatération, ici une bilatération basée sur 3 antennes, **A**, **B** et **C**, émettant un signal périodique. Chacune des branches d'hyperbole en **magenta** représente l'ensemble des points de l'espace plan pour lesquels $t_{C \rightarrow M} - t_{A \rightarrow M}$ est constant. Celles pour lesquelles $t_{C \rightarrow M} - t_{B \rightarrow M}$ est constant sont représentées en **cyan**.

Le principe de base de ce type de localisation est la multilatération. Un tel système utilise plusieurs émetteurs à une position donnée, précisément connue. Un signal est émis de manière périodique par chaque émetteur, tous étant calés sur la même période, mais avec un décalage fixe, également précisément connu. Prenons l'exemple de la figure 2.11. Trois points, A, B et C, représentent trois émetteurs. La position du récepteur est symbolisée par le point M.

Ce dernier reçoit les trois signaux à des instants différents, mais il ignore à quel instant chacun des trois émetteurs envoie son signal. Dans le référentiel de la **Terre**, la date d'arrivée d'un signal au point M, $t_{r\ M}$, émis de A à l'instant $t_{e\ A}$ par exemple, est donnée par l'équation 2.2.

$$t_{r\ M} = t_{e\ A} + \frac{d_{AM}}{c} \quad (2.2)$$

$$t_{A \rightarrow M} = t_{r\ M} - t_{e\ A} \quad (2.3)$$

$$= \frac{d_{AM}}{c} \quad (2.4)$$

Dans le cas général, $t_{e\ A/B/C}$ est inconnu de M. M a donc accès à deux valeurs de différences de temps fixe, $t_{C \rightarrow M} - t_{A \rightarrow M}$ et $t_{C \rightarrow M} - t_{B \rightarrow M}$. Chacune de ces deux valeurs définit une branche d'hyperbole. Celles-ci s'intersectent en un unique point, M. Ce principe de base est utilisé dans certains systèmes de localisation, tels le *LORAN-C* (voir [Fehlner, 1972](#)).

Le système de localisation GPS basé sur la constellation NAVSTAR est fondé sur le même principe, à deux exceptions majeures près. D'abord, la résolution est faite en 3D au lieu de 2D, ce qui implique une trilatération, basée sur au minimum 4 signaux provenant d'émetteurs différents. D'autre part, les satellites sont en mouvement autour de la **Terre**, et leur position au cours du temps, comme leur dérive (dérive de l'horloge de précision, de l'orbite) fait l'objet d'ajustements et de compensations permanentes, temps réel comme de manière périodique (mise à jour télé-opérée des données émises par les satellites toutes les 24H).

Ces données de compensation sont émises sur un canal de données dans chaque bande de fréquence de la porteuse. Le signal émis en bande L2, crypté, couplé au signal L1, permet d'obtenir des performances de localisation meilleures que le signal en bande L1 seul, qui, lui, est dédié à un usage civil.

La donnée fournie par un tel capteur consiste, de manière minimale, en un vecteur de coordonnées sphériques, $[\phi, \theta]^T$ (ϕ représentant la longitude et θ la latitude), sur le système de coordonnées de référence *WGS84*¹, et éventuellement une troisième coordonnée, Z , l'altitude dans ce même système. La plupart des capteurs commerciaux fournissent beaucoup d'autres données, et parmi elles, le mode d'opération du capteur GPS. Celui-ci peut être en mode *GPS-Standard* (la précision la plus lâche, environ 5 m), *GPS-Differential* (utilisant également l'évolution dans le temps du signal reçu pour affiner la localisation). Dans le cas où une station de base de type *Real-Time Kinematic (RTK-GPS)* est à proximité (de l'ordre de quelques kilomètres) et émet un signal contenant les données permettant la

1. Originellement, ce système de référence WGS84 était basé sur une légère translation de l'ellipsoïde GRS80. Cependant, des corrections ultérieures ont fait diverger les deux systèmes de référence. Les données GPS obtenues à partir du signal GPS par les récepteurs comportent les paramètres les plus à jour du système de référence WGS84. Ce dernier est exclusivement mis à jour par le DoD à partir des données fournies par les stations de base du système GPS.

compensation d'une partie des erreurs de localisation, le mode de localisation *RTK-GPS* permet d'obtenir une précision centimétrique.

Il est à noter que le prix d'un récepteur GPS standard (5 m) est de l'ordre de l'euro jusqu'à la centaine d'euros donc très abordable. Tandis que celui d'un récepteur compatible *RTK-GPS* (5 cm) est de l'ordre de la dizaine de milliers d'euros, prohibitif pour la plupart des conducteurs de véhicules de tourisme. Ceci le réserve donc pour des usages de type gouvernemental, sécuritaire ou de recherche. Un autre facteur limite son emploi, la nécessité d'installer une station générant les compensations, de portée kilométrique.

Néanmoins, des solutions permettant de pallier ce problème existent. On peut par exemple citer les systèmes américains WAAS (Pullen et al., 2002), européens EGNOS (Salvatori et al., 2014), qui sont des systèmes de compensation des erreurs du système GPS. Dans ces systèmes, des stations de base (réparties sur le continent nord-américain pour WAAS, européen pour EGNOS) mesurent les erreurs du signal GPS, estiment les corrections y remédiant, et les transmettent via satellites géosynchrones sur leur zone d'opération. On estime que la précision obtenue est, dans les deux cas, de l'ordre du mètre, au lieu de 5 mètres pour le signal GPS-L1 standard. Le signal WAAS, couplé au signal GPS-L1, est qualifié par l'administration américaine (FAA) pour l'atterrissage aux instruments. D'autres systèmes pour les zones asiatiques (MSAS, GAGAN) ont le même but.

2.4 Méthodes de Cartographie en Robotique

Nous allons maintenant parler des méthodes de cartographie. Dans notre objectif qu'une flotte de véhicules puisse se localiser en utilisant une carte commune, la création de celle-ci et sa mise à jour sont des fonctionnalités importantes. Nous allons tenter de donner au lecteur une idée des méthodes qui existent dans ce cadre, avant de passer à l'analyse des techniques de localisation basées sur l'exploitation de cartes.

Il existe plusieurs manières de différencier les méthodes de cartographie dans le domaine de la robotique. Tout d'abord, le caractère *en ligne-hors ligne* sépare les techniques qui mettent à jour la carte au fur et à mesure de celles qui sont exécutées une fois pour toutes, et pour lesquelles ensuite la carte reste immuable. Ensuite, les méthodes de cartographie *basée sur des données véhicule* se distinguent d'avec celles *basées sur des données externes* (données aériennes, satellite, cadastre...). Enfin, on peut séparer les approches qui permettent de réaliser une carte *dynamique* de celles ne permettant de réaliser que des cartes *statiques*. Nous allons présenter ci-après quelques-unes de ces techniques, qui sont classiques dans les domaines de la robotique ou de la cartographie.

2.4.1 Techniques Basées sur un Système d'Information Géographique

Sources de données géoréférencées

Comme nous l'avons dit précédemment, une personne, ou un véhicule autonome, peuvent se localiser en se référant à une carte de l'environnement, par exemple, un plan de ville. Les cartes de l'environnement utilisées par un être humain sont fréquemment au format papier. Mais une fois mises au format informatique, ces dernières pourraient permettre à un véhicule autonome de se localiser. Ce type de carte au format informatique, appelé Système d'Information Géographique (SIG, *Geographic Information System-GIS*)

en anglais) est une base de données contenant des éléments géoréférencés, généralement de type sémantique.

Il existe un nombre conséquent de GIS, notamment à usage pour la statistique et la gestion au niveau étatique (environnement, démographie, statistiques, installations critiques etc.) ou européen (voir par exemple les listes recensées aux États-Unis par [Cornell University](#); [University of Pennsylvania](#)). Pour les applications de transport, nous sommes intéressés par ceux ayant trait aux cartes à haute résolution (à l'échelle du mètre). Ceux-ci sont de deux types, *propriétaires* (par exemple [IGN](#), pour la France), ou *collaboratifs-ouverts* ([Mairie de Paris](#), pour l'agglomération parisienne) ou bien ([OpenStreetMap](#) [OpenStreetMap contributors](#), 2017, pour un site collaboratif mondial).

Nous allons dans ce qui va suivre nous focaliser sur l'exemple du GIS collaboratif *OpenStreetMap*. Le GIS *OpenStreetMap* est un système de gestion de données géoréférencées collaboratif soutenu et financé principalement par l'*University College London*, l'*Imperial College London* et *Bytemark*, au *Royaume Uni*, bien que d'autres soutiens moins importants contribuent également.

Un exemple de vue d'une carte obtenue à partir de ce GIS est visible sur la figure 2.12. Les données sont publiées par des utilisateurs ou contributeurs volontaires qui réalisent les mesures de position par leurs propres moyens. Une problématique que cela pose est relative à la qualité des données. On peut se poser la question si les mesures ne sont pas de qualité variant avec l'identité des contributeurs ([Arsanjani et al.](#), 2013). Des estimations récentes attestent néanmoins de la bonne tenue de la base de données et de la qualité des mesures obtenues ([Ather](#), 2009; [Fan et al.](#), 2016; [Girres and Touya](#), 2010; [Helbich et al.](#), 2012; [Kessler](#), 2013; [Mooney et al.](#), 2010)².

Pour fixer les idées, nous allons mentionner quelques chiffres. On estime dans ([Helbich et al.](#), 2012) que l'erreur de positionnement des éléments cartographiés dans *OpenStreetMap* est de l'ordre de 5 m. La taille des données gérées par ce système de cartographie est indiquée dans la table 2.1 et est de l'ordre de 2 To (2 TB). Sur un serveur miroir³ local ([nepomuk.openstreetmap-LyonIX](#)), les latences moyennes d'accès disque sont de l'ordre de 100 ms à 150 ms, les temps d'accès réseau au serveur sont de l'ordre de 25 ms à 30 ms et les temps de réponse à une requête apparaissent bornés par valeur supérieure à 1 s. Celles-ci correspondent au téléchargement d'une portion *locale* ou *abrégée précalculée* ([/stores/tiles](#)) de la carte complète ([/stores/database](#)).

Sur un véhicule disposant d'une connexion sans-fil au réseau, une utilisation d'*OpenStreetMap*, ou éventuellement de tout autre GIS, serait possible afin de permettre la localisation d'un véhicule, voire même la mise à jour de la carte par un « véhicule contributeur » équipé en conséquence.

Localisation et cartographie utilisant des GIS

Ce type de méthodes se fonde sur une carte pré-existante provenant d'un GIS. Cette carte peut être utilisée conjointement à d'autres capteurs, comme une caméra ou potentiellement un capteur LIDAR. Dans ([Brenner and Elias](#), 2003), une comparaison entre une base de données de type cadastre, et des acquisitions de type LIDAR aérien est réalisée. Celle-ci permet de générer une carte dans laquelle hauteur (provenant des données

2. Les mesures d'intégrité sont généralement réalisées par calcul du taux de recouvrement des surfaces définies, plutôt que par mesure d'écart de position.

3. Serveur dupliquant les données d'un serveur de base. Plusieurs sont généralement répartis sur la Terre de manière sensiblement uniforme. Ils sont utilisés car, étant plus proches de l'utilisateur final, leurs latence d'accès et débit sont meilleurs.

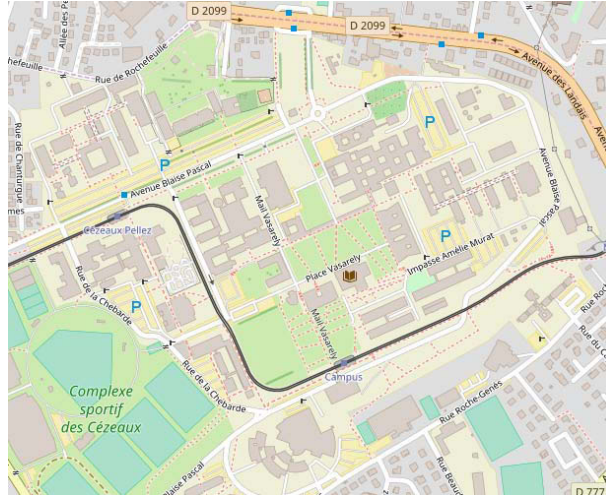


FIGURE 2.12 – Exemple de vue sur la zone du campus des Cèzeaux (Université Clermont-Auvergne) des données du GIS publique OpenStreetMap (OpenStreetMap contributors, 2017).

Systèmes de fichiers	Description	Taille disque (GB/Go)	Usage (%)	Utilisé (GB/Go)
/stores/database	Base de données principale	939	71.74	673
/stores/tiles/default-high	Tuile de vue précalculée à haute résolution	931	85.91	800
/stores/tiles/default-low	Tuile de vue précalculée à basse résolution	550	83.31	458

TABLE 2.1 – Taille des bases de données principales du GIS OpenStreetMap à la date du 02 Septembre 2017, d’après (OpenStreetMap contributors, a) et (OpenStreetMap contributors, b).

LIDAR) et extension en surface des bâtiments (provenant d’une fusion entre la base cadastre et les données LIDAR) sont disponibles. Une telle carte est visible sur la figure 2.13, d’après (Brenner and Elias, 2003, figure 2). Un calcul de la visibilité de bâtiments par un observateur depuis un certain point de vue est réalisé afin de déterminer si celui-ci peut utiliser le bâtiment en question comme amer pour se repérer. Des travaux plus récents du même auteur confirment la pertinence de cette approche (Brenner, 2010), en permettant la localisation d’un véhicule à partir d’images caméra prises à bord de ce dernier et identifiant les arbres de l’environnement. Ceux-ci ont été préalablement cartographiés grâce aux données acquises par un véhicule équipé de capteurs LIDAR, lors d’une opération de génération hors ligne. Cette technique est assez similaire à celle décrite dans (Kais et al., 2004).

Une autre utilisation d’un GIS est présentée également dans (Majdik et al., 2014), que la section précédente mentionne également. Une base de données de type cadastre est utilisée conjointement à des micro-drones équipés de caméras, de manière à pouvoir permettre la reconstruction des façades de bâtiments d’une ville. Ce type de données seraient à même de permettre une localisation en 3D des drones en question, ou de véhicules terrestres. Les données du GIS sont utilisées afin de définir, même de manière approchée,



FIGURE 2.13 – Exemple de carte fusionnant des données cadastres et LIDAR (reproduit d’après [Brenner and Elias, 2003](#), figure 2).

les emplacements des façades verticales des bâtiments. Les données de reconstruction de la carte de points 3D, provenant de l’ajustement de faisceaux, sont ensuite appariées avec les positions des façades pré-calculées précédemment. Finalement, une texture 2D, provenant des données d’image RGB de la caméra est appliquée aux façades en question. Une telle structure de données, qui reconstruirait complètement une rue, pourrait permettre la localisation d’un drone en 3D, en se basant uniquement sur les données caméra.

D’autres techniques de localisation, reprenant au moins en partie les spécificités des articles précédents peuvent être trouvées dans ([Bonnifait et al., 2008](#); [Cappelle et al., 2008](#); [Fouque et al., 2008](#); [Qu et al., 2015](#); [Rackliffe et al., 2011](#)). La génération de telles cartographies est également possible. Dans ([Soheilian et al., 2013](#)), un modèle 3D est généré grâce à un véhicule équipé de caméras ainsi que de capteurs LIDAR, dont les données sont fusionnées avec plusieurs orthophotographies aériennes à diverses résolutions. Certains traits saillants, tels des panneaux de signalisation, ou le marquage au sol, sont extraits afin de reconstruire au mieux l’environnement dans une carte 3D virtuelle. La seule restriction à ces travaux est le fait que cette carte soit générée hors-ligne.

Ces références prouvent la pertinence de l’utilisation d’une carte géoréférencée de l’environnement au format informatique, afin de permettre la localisation de véhicules autonomes. L’agrégation d’informations de mesure obtenues durant le déplacement du véhicule autonome permettrait de mettre la carte à jour.

Avant de parler plus avant des techniques permettant à l’*ego-vehicule* de cartographier l’environnement, nous allons brièvement mentionner quelques techniques utilisées pour générer de manière *hors-ligne* les données inscrites dans de telles bases de données.

2.4.2 Techniques basées sur une acquisition orthophotographie–LIDAR vertical

Le type de technique discuté ici a pour objectif de créer des cartes de l’environnement du type GIS de manière *hors-ligne*, à partir de mesures réalisées par des moyens externes. Dans ce type de techniques, on ne tente pas d’aligner les données d’un GIS avec



FIGURE 2.14 – Exemple de détection de bâtiments utilisant des données LIDAR et orthophotographie provenant de prises de vue aériennes. Reproduction partielle d’après (Hermosilla et al., 2011, figure 8).

des données réelles, mais plutôt de déterminer à partir de plusieurs sources de données complémentaires, comme un LIDAR aérien couplé à une orthophotographie (possiblement multispectrale) l’emploi d’une zone de la surface d’une ville.

Plusieurs étapes sont nécessaires afin de réaliser cette fusion. Dans (Hermosilla et al., 2011), un filtrage par seuillage de la hauteur mesurée par LIDAR est couplé avec un algorithme de croissance de zone basé sur les données orthophotographiques. Le seuillage de l’altitude permet la classification de zones en fonction de la nature et de l’extension en hauteur sur la zone des échos LIDAR. La figure 2.14, qui reproduit partiellement (Hermosilla et al., 2011, figure 8), montre le résultat de l’une de ces détections. Dans (Rottensteiner et al., 2005), la fusion est réalisée grâce à certaines règles de fusion de données dans le cadre de la théorie de Dempster-Shaffer. Dans les deux cas, le résultat est une base de données contenant une extension en surface, une catégorisation du type de zone. Ceci pourrait permettre la génération d’une carte plus élaborée.

Ces deux méthodes sont des exemples utilisés pour séparer et identifier la nature de zones de la surface d’une ville, ou de la hauteur et de l’empreinte au sol d’un bâtiment. Les mêmes techniques pourraient servir dans le cadre de la robotique mobile à déterminer les zones carrossables (routes, parkings) des zones inaccessibles à la circulation (bâtiments, zones de végétation). Les zones carrossables pourraient être associées à un plan de ville. En fonction de la précision des données LIDAR et orthophotographiques, il pourrait être possible d’ajouter les bâtiments détectés. Néanmoins, une précision inférieure à 5 m semble être la condition nécessaire afin que ceci ait une utilité, sachant qu’un capteur GPS donne une position du véhicule avec cette précision.

Ce type de technique est dédié à la création d'une carte métrique de l'environnement. Celle-ci pourrait être convertie en GIS, lors d'une phase de génération *hors-ligne*, en général encadrée par une intervention humaine, et donc être utilisée comme base de départ par des robots terrestres autonomes. Certaines techniques utilisant un GIS ainsi que des UAV ([Majdik et al., 2014](#)) pourraient en théorie mettre à jour ce type de carte, mais elles ne sont pas actuellement dans un stade de maturité suffisant.

2.4.3 Cartographies des voies de circulation basée GPS

On vient de voir précédemment quelques techniques basées LIDAR aérien couplée à d'autres données, principalement pour identifier bâtiments et zones de végétation, qui pourraient être appliquées à l'identification des voies carrossables. Pour identifier ces dernières, ainsi que les voies de circulation, on peut également utiliser des données de capteurs GPS en grand nombre, données rendues anonymes. Ce type de données est généré par les navigateurs GPS, et téléchargées par le fabricant du dispositif lors de mises à jour du logiciel ou des données du navigateur GPS en question.

Une analyse statistique sur ces données en très grand nombre permet d'identifier les zones où le circuit de circulation change de sens, où celle-ci est stoppée, ou les axes de circulation nouvellement percés. Dans ([Ahmed et al., 2015](#)), une comparaison de tels algorithmes est faite. La figure 2.15 reproduit le résultat de l'un des algorithmes présentés dans ([Ahmed et al., 2015](#)). Un exemple d'implémentation peut être trouvé dans ([Bétaille and Toledo-Moreo, 2010](#)), dans lequel une modélisation de voies de circulation par sections de clothoïde est faite.



FIGURE 2.15 – Un exemple de reconstruction de voies de circulations dans la ville de Berlin (reproduit d'après [Ahmed et al., 2015](#), figure 1).

Cette technique est basée sur une analyse statistique des positions des véhicules se déplaçant majoritairement selon le tracé du réseau. Il est donc possible de générer au format informatique une carte *topologique-métrique* des voies de circulation, et par exemple de les intégrer dans un GIS. Une étape de traitement supervisée par un être humain est néanmoins requise. De ce fait, ce type de mise à jour est ponctuelle, plutôt qu'automatique. Elle fait de plus appel au volontariat des usagers du service de navigation GPS. On pourrait tout de même imaginer qu'à terme, des véhicules autonomes puissent générer de telle données, lorsque leur nombre le permettra.

2.4.4 Cartographies Basées SONAR

Une approche totalement différente de toutes les précédentes est présentée ici. Dans les techniques précédemment présentées, une cartographie hors-ligne était réalisée basée sur des données, soit mesurant le monde extérieur de manière externe (cartographie aérienne), soit utilisant des données obtenues à partir d'un véhicule non-robotisé. Ici, nous présentons une technique permettant de cartographier l'environnement extérieur d'intérêt pour un robot et basée sur les données de capteurs SONAR de l'*ego*-robot.

Par mesure SONAR, nous entendons un détecteur de proximité d'obstacle, fonctionnant par écho d'ondes acoustiques ultrasonores dans l'air. Ce type de détecteur a une portée allant de plusieurs dizaines de centimètres à quelques mètres. Si la résolution radiale est relativement acceptable sur ce genre de détecteurs, en raison de la diffraction des ondes acoustiques, la résolution angulaire est très mauvaise (de l'ordre de 30°) en comparaison d'autres capteurs tels une caméra ou un LIDAR. Son intérêt réside en son faible coût conjugué à une potentielle installation sur tout type de véhicule terrestre et un fonctionnement dans n'importe quelles conditions d'environnement. Ces capteurs sont utilisés dans l'industrie automobile comme éléments de sécurité sur des véhicules de série comme détecteurs d'obstacles durant les phases de recul.

La plupart des implémentations de cartes réalisées par SONAR, ainsi que certaines de celles basées sur des capteurs LIDAR (Manfredi et al., 2016, et voir figure 2.2), utilisent des cartes d'occupation discrétisées. Celles-ci consistent en un pavage régulier (carrés) du plan, et dont chaque cellule indique la probabilité que l'espace soit occupé ou vide. Les capteurs SONAR émettent un faisceau sonore d'angle de dispersion relativement large. Lorsqu'un écho est reçu, il peut avoir été généré par un ou plusieurs objets réparti sur plusieurs cellules de la grille d'occupation. Un calcul complexe de probabilités et de mises à jour bayésiennes est réalisé dans (Thrun, 2003) afin d'estimer de manière itérative la probabilité d'occupation par un obstacle de chaque cellule de la carte. Un exemple de carte réalisée par un robot mobile équipé de tels capteurs est présenté sur la figure 2.16.

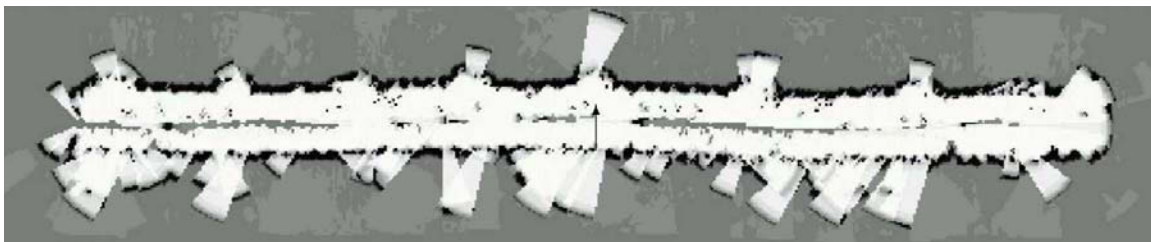


FIGURE 2.16 – Exemple de carte d'occupation générée par un robot mobile équipé de capteurs SONAR (reproduit d'après Thrun, 2003, figure 6c).

Cette technique, en utilisant des cartes de type « grille d'occupation », engendre un phénomène de discretisation. Ceci pourrait gêner un potentiel couplage avec, par exemple, une technique continue comme celles basées sur un filtre de *Kalman* (voir section 2.4.8). Cette technique fournit tout de même une information qu'aucune autre méthode ne donne, à savoir la traversabilité d'une zone. C'est de ce fait une technique importante, même si les travaux originaux dans ce sens datent du début des années 2000.

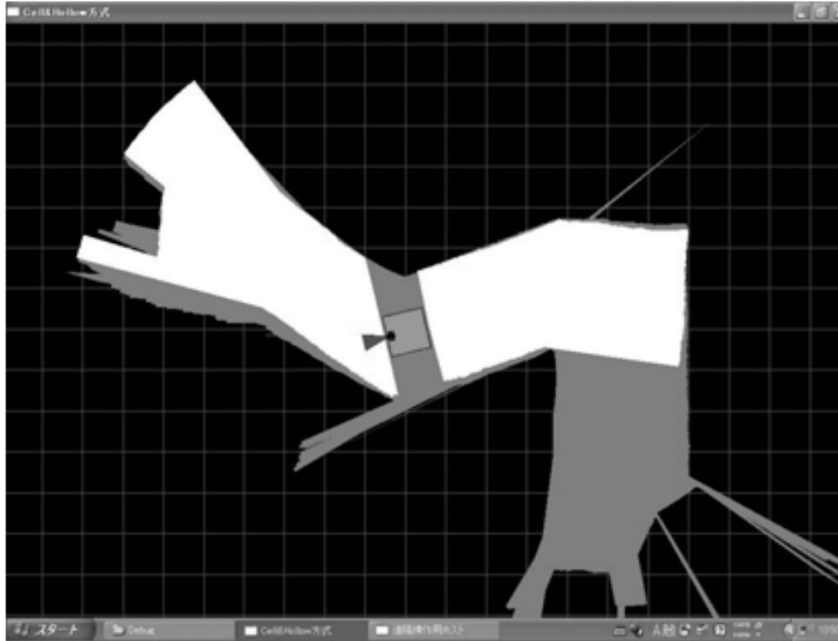


FIGURE 2.17 – Exemple de carte de type grille d’occupation générée par LIDAR, d’après (Matsumaru et al., 2009, figure 10).

2.4.5 Cartographies Basées LIDAR

Les techniques de cartographies réalisées par des robots autonomes utilisant des capteurs LIDAR sont, la plupart du temps, basées sur la reconnaissance de structures sémantiques, nommées primitives et possédant un haut degré de symétrie, comme des murs ou des colonnes. Notre approche est de ce type, et nous ne la mentionnons pas ici. D’autres sont basées sur la détection de l’état d’occupation (présence d’obstacle ou non) de chaque point du plan, et l’alignement de ces grilles d’occupation pour deux positions différentes du robot. Enfin, un alignement entre plusieurs trames LIDAR prises à diverses positions permet de reconstruire une telle carte.

Deux manières principales de cartographier l’environnement existent, l’une bidimensionnelle, l’autre tridimensionnelle, mais dans les deux cas, la carte est statique. La cartographie 2D permet la détection de structures, supposées verticales, en utilisant un LIDAR en montage plan. La cartographie 3D utilise, quant à elle, soit un LIDAR 3D, soit éventuellement un LIDAR 2D dont le support peut pivoter.

Dans (Matsumaru et al., 2009), un robot télé-opéré embarque un LIDAR 2D plan. Les données de ce LIDAR permettent, par alignement des nouvelles données LIDAR avec la carte, de mettre à jour celle-ci. La carte en question est de type grille d’occupation. Une telle carte, dans lequel la position du robot est indiquée, est visible sur la figure 2.17.

Un exemple de reconstruction 3D d’environnement est présenté dans (Iwashina et al., 2008) avec un LIDAR 2D monté sur support pivotant. Le robot en question se déplace en environnement intérieur, et acquière des données avec le LIDAR pivotant ainsi qu’un LIDAR plan. Les diverses trames du LIDAR plan et du LIDAR pivotant sur un intervalle de temps donné sont alignées entre elles grâce à une technique d’alignement de nuages de points appelée « Iterative Closest Point » (ICP). Un résultat est visible sur la figure 2.18. Cette technique est basée sur un calcul de distance minimum entre points de plusieurs trames différentes, utilisant un alignement *k-plus proches voisins*. Cette tech-

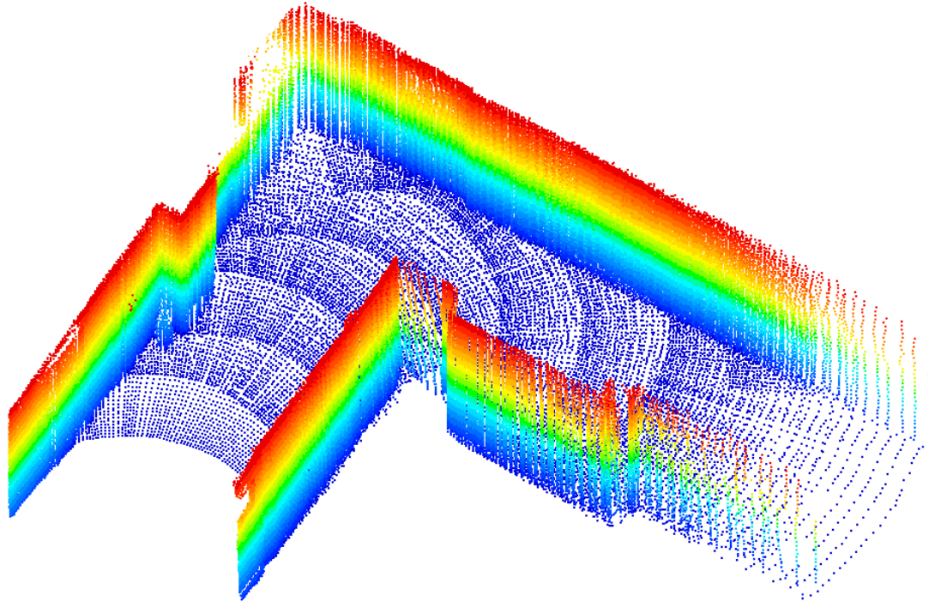


FIGURE 2.18 – Exemple d’alignement successif de séries de nappes LIDAR en pseudo-3D, d’après (Iwashina et al., 2008, figure 12b).

nique est connue comme robuste, bien qu’elle accumule de l’erreur au fur et à mesure du déplacement du robot (comme une cartographie SLAM). Par contre, elle fonctionne difficilement dans le cas où des éléments se déplacent dans le champ du LIDAR, à moins que le nombre de points appartenant à ces éléments mobiles ne reste faible en proportion. Le même résultat peut également être obtenu avec un LIDAR 3D et la même méthode ICP.

2.4.6 Cartes prenant en compte l’évolution dans le temps et les configurations

Ici, nous mentionnons deux méthodes qui ont retenu notre attention, car elles gèrent des aspects que la plupart des autres méthodes de cartographie ne prennent pas en compte. Tout d’abord, dans (Biber et al., 2005), l’objectif est de créer un système de cartographie permettant à un robot de se localiser dans un environnement fortement évolutif, à l’échelle de la journée (entrepôt contenant des marchandises par exemple). Le système stocke les échos LIDAR en fonction de l’échelle de temps. Pour chaque mesure LIDAR, le système va rechercher dans l’historique l’écho LIDAR correspondant le plus spatialement. Ceci permet de gérer le cas d’un colis stocké temporairement et occultant un mur. Les deux échos, celui du colis, et celui du mur, restent dans la base de données, mais pour des échelles de temps différentes. Cette méthode nous semble intéressante car elle prend en compte différentes échelles de temps de manière similaire, au facteur temporel près.

Dans (Avots et al., 2002), un robot se déplace dans un couloir carré, sans mesure absolue de position. Dans cette méthode, des ambiguïtés de configuration existent. Un filtrage multi-hypothèse permet au robot, d’après ses observations, d’éliminer les hypothèses erronées. Ses capteurs lui permettent de détecter des murs, mais pas de les identifier. La structure carrée ayant un degré de symétrie de 4, une quadruple-ambiguïté de localisation existe. Dans la réalité, certaines cloisons du couloir sont pleines (pas de portes), d’autres

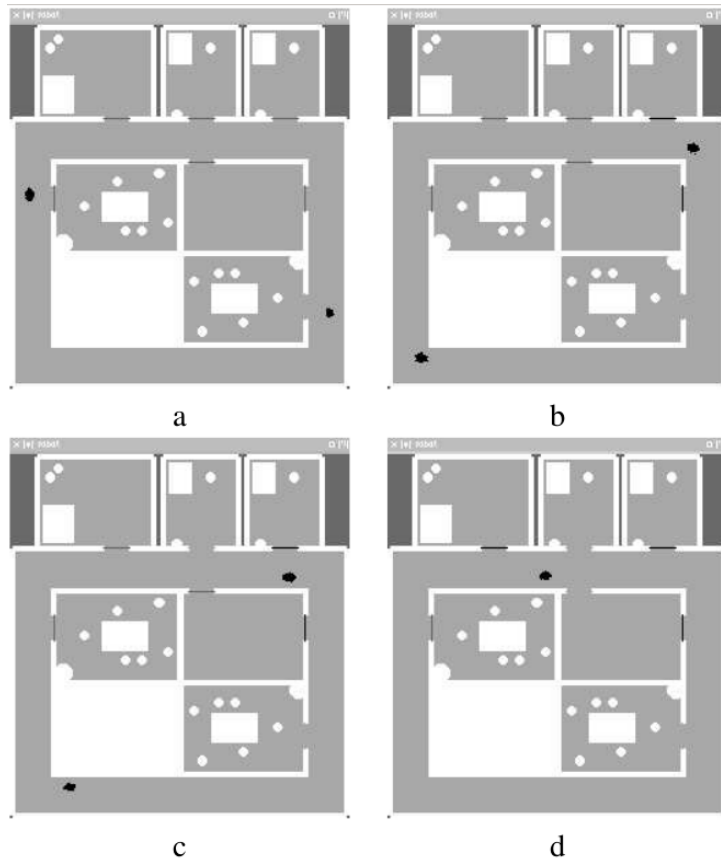


FIGURE 2.19 – Exemple de localisation multi-hypothèse avec gestion de l’ambiguïté, (d’après Avots et al., 2002, figure 3).

en ont, qui peuvent être ouvertes ou fermées. Dès qu’une porte ouverte est détectée, les zones ambiguës dans lesquelles le robot peut être situé, et en face desquelles une cloison pleine existe, sont supprimées. A la fin de la trajectoire, le robot a levé toutes les ambiguïtés de localisation, comme on peut le voir sur la figure 2.19.

2.4.7 Cartes séparant les éléments fixes des éléments mobiles

Dans toutes les approches précédentes, la carte générée ne contenait que des éléments statiques. Ceci signifie qu’il est possible qu’en présence d’objets mobiles (piétons, autres véhicules...), les méthodes précédentes pourraient prendre de manière erronée certains éléments mobiles comme référence, alors que ceux-ci sont en déplacement. Si un processus d’alignement entre des données actuelles, et une carte de l’environnement à l’étape précédente est utilisé (comme dans la section 2.4.5, ou dans 2.4.8 par exemple), la présence de tels objets peut induire un biais, dans la carte comme dans la localisation du véhicule.

Certaines techniques permettent de prendre en compte l’existence d’éléments mobiles dans les données capteurs. En règle général, la détection et séparation des objets en mouvement d’avec les éléments fixes requiert une comparaison entre les divers jeux de données, d’où l’utilisation de techniques d’optimisation globales. En effet, nous n’avons a-priori aucun moyen, dans un seul enregistrement de données à un instant t , de déterminer ce qui est statique (au même endroit $\forall t'$) de ce qui est dynamique (potentiellement à un autre endroit pour $\exists t'$). La technique de choix dans ce cas consiste à utiliser une

optimisation de type « Expectation–Maximization » (EM) sur tous les jeux de données accessibles.

Une optimisation de type EM va calculer une loi de distribution globale valable pour tous les éléments de tous les jeux de données accessibles, et choisir comme réalisation (état de convergence) un état où la valeur de chaque variable aléatoire (position de l’objet, classification de l’objet) va prendre celle qui maximise la probabilité globale. Ce type d’optimisation est de nature bayésienne, itérative, et utilise toutes les données disponibles, d’où des potentiels problèmes pour mettre à l’échelle ce type de calcul (notamment en raison du nombre variable de données sources). On peut au choix, soit réaliser une telle optimisation hors ligne en utilisant un jeu de données complet, soit utiliser en *quasi-temps réel* tous les jeux de données antérieurs à la date courante. Habituellement, le type de cartes qui s’adapte le plus à ce genre de calcul sont les grilles d’occupation, mais toute autre représentation, pourvu qu’elle soit probabiliste, est envisageable.

Nous allons discuter ici quelques exemples de réalisation. Dans (Hahnel et al., 2002; 2003), un robot mobile, équipé de capteur LIDAR (potentiellement LIDAR 2D ou 3D), mais aussi caméra, évolue dans un environnement où des piétons sont présents et se déplacent. Par une optimisation de type EM et une carte d’occupation, les auteurs parviennent à classer les données LIDAR comme obstacle toujours fixe, ou comme élément mobile potentiel pour toute trame de données. Un exemple de résultat est reproduit sur la figure 2.20. On peut observer l’erreur de dérive indiquée par les flèches rouges de la figure 2.20 lorsque la carte est reconstruite en supposant que tous les points de mesure des données LIDAR représentent des obstacles fixes. Lorsque l’on détermine le statut fixe ou mobile des obstacles détectés, la qualité de la reconstruction est bien meilleure.

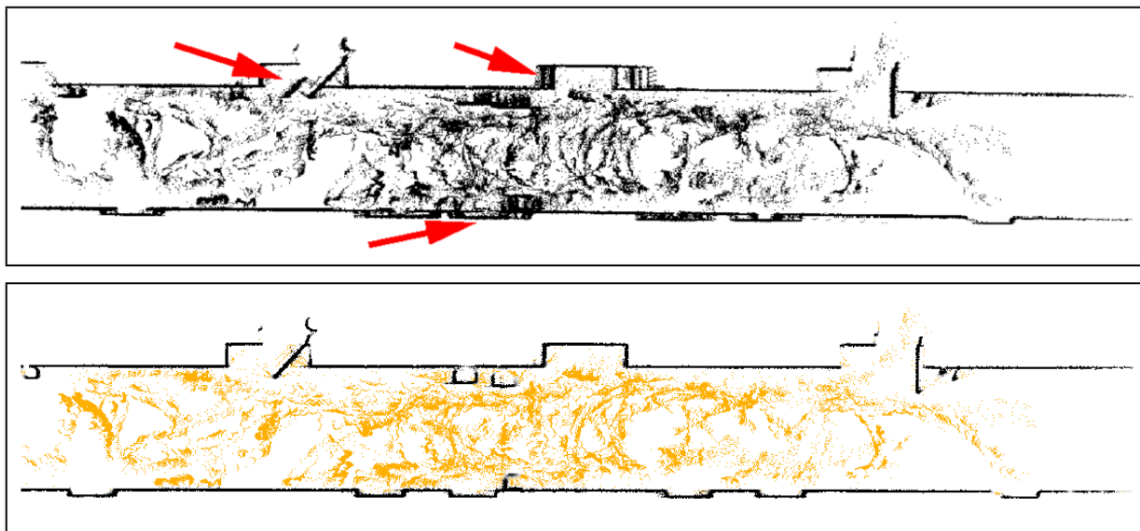


FIGURE 2.20 – Exemple de données brutes issues des capteurs LIDAR (en haut), en présence de piétons dans le champ du capteur LIDAR et à partir d’un robot mobile, d’après (Hahnel et al., 2003, figure 6), et de résultat d’optimisation EM y correspondant (en bas), d’après (Hahnel et al., 2003, figure 7). Les cellules représentées en jaune sur l’image du bas représentent les objets mobiles reconstruits, celles en noir les objets fixes (murs/portes).

Une autre technique publiée récemment afin de détecter les objets fixes ou mobiles consiste, dans un processus SLAM caméra, à utiliser une méthode d’appariement au niveau

des points d'intérêts de Harris prenant en compte des objets en mouvement. Dans (Jiang et al., 2016), une segmentation des points d'intérêt permet, par comparaison entre les points identifiés comme fixes aux étapes précédentes, et les trajectoires de ceux associés aux objets en mouvement, de séparer les deux catégories fixe/en mouvement l'une de l'autre.

2.4.8 Cartes obtenues par SLAM

Le terme SLAM recouvre différentes techniques, qui ont toutes le même objectif : la localisation d'un véhicule en évolution relativement à une carte qui se construit de manière incrémentale. Après avoir mentionné les différentes grandes familles de SLAM, nous concluons sur les possibilités qu'offrent ce type de méthodes de cartographie.

SLAM par graphes « GraphSLAM »

La technique dite du « GraphSLAM » a été présentée pour la première fois dans (Thrun and Montemerlo, 2006). Dans cette publication, on définit le problème du SLAM comme un calcul itératif de la nouvelle pose d'un véhicule à une date donnée. Cette pose est calculée et alignée à la carte créée jusqu'à cet instant. Dans cette carte, dynamique par essence, on insert de nouveaux éléments basés sur les observations du véhicule.

Chaque élément fixe de la carte est estimé par une pose relative au véhicule, celle-ci étant obtenue par les observations. On construit un graphe de toutes les relations entre chaque pose du véhicule à chaque instant, et la pose absolue de tous les éléments cartographiés, supposés fixes, et détectés par le véhicule, par le biais des observations de ce dernier. Une fusion de données globale de type *Moindres-Carrés-Non-Linéaires* permet de reconstruire l'état de tous les éléments et de toutes les poses du véhicule aux différents instants de manière globale. Les figures (Thrun and Montemerlo, 2006, figures 1 et 2) décrivent le fonctionnement du système.

Cette méthode repose sur une fusion globale de données tandis que d'autres méthodes, décrites ultérieurement (SLAM basé sur un filtre de *Kalman*) recourent à une estimation permanente de l'état du véhicule et de la carte par fusion gaussienne itérative.

Dans certains cas, certaines parties du graphe obtenu sont faiblement couplées au reste de celui-ci. Une mesure biaisée ou erronée peut alors faire diverger la carte par mauvais appariement entre ces deux sous-parties. Une variante de cette technique, présentée dans (Sünderhauf and Protzel, 2012), permet d'activer ou de désactiver de telles contraintes, afin d'optimiser la carte, en ne prenant pas en compte d'éventuelles erreurs de mesure ou d'association.

SLAM par Filtres à Particules

Avec les filtres de *Kalman*, les filtres à particules sont une technique d'estimation bayésienne, donc probabiliste, très utilisée en robotique. À la différence du filtre de Kalman, basé sur une densité de probabilité gaussienne, unimodale, continue et linéaire, le filtre à particule se base sur un échantillonnage de la densité de probabilité estimée, potentiellement multimodale, discrète et non-linéaire. Chaque densité de probabilité est modélisée dans cette méthode par un nombre, potentiellement grand, de particules. Chacune de ces particules équivaut au même poids global dans la densité de probabilités estimée, mais leur proximité dans l'espace des états confère une plus grande valeur à la densité de pro-

babilités estimée. Un des inconvénients majeurs de ce type de techniques est son coût calculatoire, augmentant exponentiellement avec le nombre de dimensions du problème.

Un exemple d'une telle méthode est présenté dans (Törnqvist et al., 2009), dans laquelle un drone aérien (plus précisément un *Unmanned Aerial Vehicle* en anglais, UAV) se localise et cartographie l'environnement. Cette méthode est particulièrement immune aux problèmes de linéarisation observés dans d'autres méthodes, par exemple dans le cas d'un capteur « caméra monoculaire ». Un autre exemple, appliqué à un robot mobile utilisant un capteur LIDAR 2D vertical sur pivot également vertical, est présenté dans (Welle et al., 2010). Certaines optimisations relatives à la méthode d'échantillonnage sont mentionnées. Une carte précise et « convaincante » obtenue par cette méthode est représentée sur (Welle et al., 2010, figure 6).

SLAM basés sur un filtre de Kalman

Les techniques de SLAM par filtre de Kalman sont les plus anciennes et les plus éprouvées de ces méthodes. Elles ont été initialement utilisées avec succès avec des capteurs LIDAR. Cette technique, prenant en compte difficilement des non-linéarités importantes, est, dans le cas de capteurs caméra monoculaire, particulièrement sensible aux non-linéarités dans le sens de l'axe de profondeur. Voir par exemple (Subbarao, 1988), qui évoque le paramétrage en distance inverse. Ce type de paramétrage est une parade classique à ce type de non-linéarités avec un filtre de Kalman, avec un exemple d'utilisation dans (Sunderhauf et al., 2007).

Une des premières implémentations d'un SLAM basé sur un filtre de Kalman est présentée dans (Dissanayake et al., 2001). Une création de carte, dans un référentiel relatif, consiste à faire évoluer le véhicule tout en agrégeant dans la carte des amers venant des données des détecteurs (ici un RADAR panoramique à courte portée). A chaque fois qu'un nouvel amer est détecté, il est ajouté à la carte, et la matrice de covariance ainsi que le vecteur d'état sont étendus d'autant.

Un exemple de carte reconstituée grâce à cette méthode est présenté sur la figure 2.21. Un unique véhicule centralise la carte de manière locale. Seuls des éléments fixes sont cartographiés. Un autre exemple d'un tel travail, (Nerurkar and Roumeliotis, 2011), implémente conceptuellement la même méthode en utilisant cette fois une caméra monoculaire.

Parmi la bibliographie récente, on pourra trouver, dans (Cadena and Neira, 2010; Sunderhauf et al., 2007) deux exemples de SLAM basés sur des variantes du filtre de Kalman, EKF et UKF, appliqués à un capteur caméra monoculaire. Des exemples d'implémentation avec un capteur LIDAR sont présentés dans (Genevois and Zielińska, 2014; Niu et al., 2017; Zhao and Farrell, 2013).

SLAM par ajustement de faisceau

Dans cette section, nous allons donner un exemple de réalisation utilisant une caméra monoculaire, présenté dans (Royer et al., 2005). Une caméra, montée sur un véhicule, acquiert des images de manière périodique sur une trajectoire. Sur chaque image, un détecteur de points d'intérêt identifie de manière reproductible et robuste des points. Ce *détecteur de Harris* identifie les bordures de motif type coin. Un exemple d'un résultat de détection est présenté sur la figure 2.22. Un processus d'ajustement de faisceau permet, de manière hors-ligne, de reconstruire simultanément la trajectoire de la caméra et la carte des points d'intérêt en 3D. Cette étape est basée sur la minimisation d'une fonction

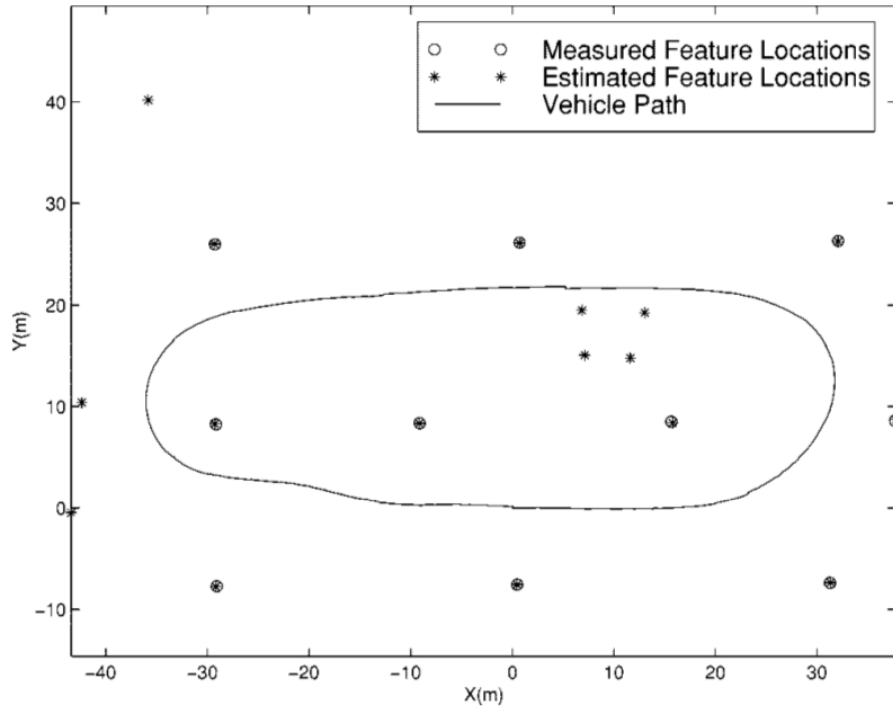


FIGURE 2.21 – Exemple de résultat d’un SLAM basé sur un filtre de Kalman mono-véhicule centralisé. La trajectoire du véhicule est symbolisée par la courbe noire. Les positions des éléments de la carte (cibles RADAR) sont indiquées par des *. D’après (Dissanayake et al., 2001, figure 7).

globale représentant la somme de l’erreur quadratique entre chaque point reconstruit en 3D de la carte, et chaque détection de point de Harris sur chaque image. Pour des raisons de simplicité, nous ne mentionnons pas le problème de l’association entre divers points de Harris de différentes images.

Finalement, cette carte, basée sur une trajectoire donnée, peut être utilisée afin de localiser le même véhicule dans la carte de points d’intérêt 3D, lors de déplacements ultérieurs, et en temps réel. Un exemple de carte reconstruite et de résultat de localisation du véhicule est présenté sur la figure 2.23. D’autres travaux, comme (Majdik et al., 2014) et (Michael et al., 2012) montrent des réalisations d’une telle méthode, basées sur des acquisitions faites par des micro-drones.

Cette méthode permet une localisation en temps réel sur une trajectoire similaire ou approchante de celle ayant permis de réaliser la carte. La génération de la carte est faite hors-ligne, une fois pour toute. Lorsque la région cartographiée change (arbre abattu ou taillé, chute du feuillage en automne), une autre carte doit être régénérée basée sur des acquisitions plus récentes. Enfin, la carte est locale (non géo-référencée) et dérive avec la longueur de la trajectoire d’acquisition initiale. Ceci génère un problème, appelé fermeture de boucle, dans lequel, lorsqu’un véhicule parcourt un circuit fermé, l’algorithme peut ne pas reconnaître le point de départ de la trajectoire, et générer une carte où la trajectoire ne reboucle pas sur elle-même.

Cette technique est conceptuellement relativement similaire dans l’esprit à la technique « GraphSLAM » qui a été précédemment présentée.

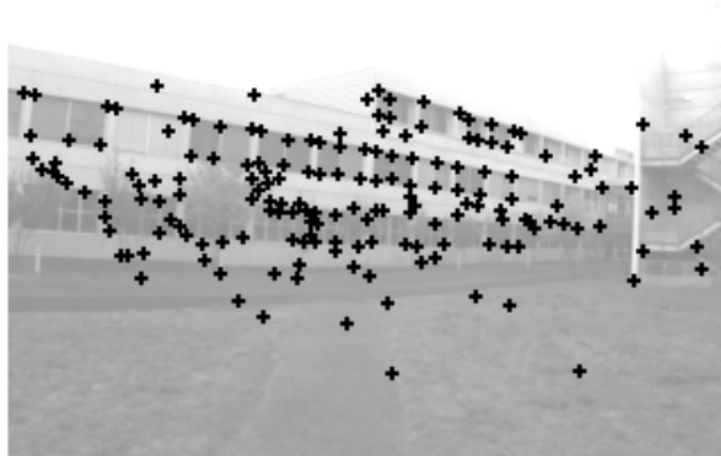


FIGURE 2.22 – Exemple de résultat d’application d’un détecteur de Harris sur une image provenant d’une caméra, d’après (Royer et al., 2005, figure 2). Les points détectés sont marqués par un symbole +.

Discussion de quelques implémentations

La plupart des techniques de SLAM se basent sur des capteurs caméra ou LIDAR, en fonction du contexte et de la finalité des travaux. La méthode historique est basée sur un algorithme de type Kalman. En raison des non-linéarités inhérentes à certains problèmes, des approches basées sur « GraphSLAM » et des *filtres à particules* ont été développées. La technique estimée la plus robuste est le SLAM basé *Filtre de Kalman Étendu* (*Extended Kalman Filter* en anglais, EKF) utilisant un capteur LIDAR plan, dont un exemple d’application industrielle peut être trouvé dans (Korpela et al., 2015).

Il est à noter que certaines méthodes non discutées, comme (Jaulin, 2009), n’emploient pas de principe bayésien, mais plutôt une analyse logique par découpage spatial de l’environnement (voir plus précisément Kieffer et al., 2000). Néanmoins, celles-ci restent actuellement cantonnées à des domaines d’application restreints.

Ces méthodes permettent à l’exception de l’*ajustement de faisceau* de construire une carte dynamique de l’environnement. Cette carte est par essence relative au véhicule, à sa trajectoire ainsi qu’à son champ de vue. Quelques implémentations tentent de fusionner cette carte avec une carte absolue de l’environnement, comme dans (Parsley and Julier, 2009), mais ces approches ne sont pas majoritaires dans le monde de la robotique.

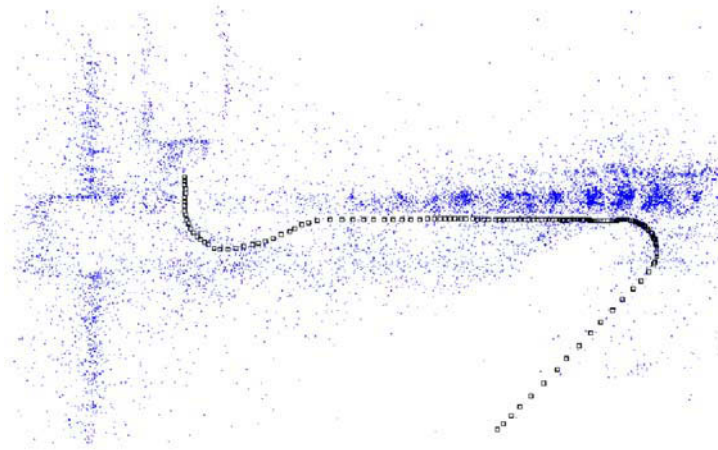


FIGURE 2.23 – Exemple d’une carte SLAM de points d’intérêt 3D représentée par des points bleus, en vue de dessus, d’après (Royer et al., 2005, figure 4). La trajectoire du véhicule est matérialisée par des carrés noirs. La structure sur la gauche de la carte correspond à un bâtiment existant.

2.4.9 Tableaux Récapitulatif

Dans cette section, deux tableaux récapitulatifs sont présentés. La table 2.2 provient de (Thrun et al., 2002, table 1) et récapitule certaines méthodes de localisation et de cartographie. La table 2.3 récapitule les différentes méthodes, basées sur les caractères présentés au début de la section 2.4.

	Kalman	Lu/Milios	EM	Incremental ML	Hybrid	Occupancy Grids	Multi-Planar Maps	Dogma
Representation	landmark locations	point obstacles	point obstacles	landmark locations or grid maps	point obstacles	occupancy grids	objects and polygons	occupancy grids
Uncertainty	posterior poses and map	posterior poses and map	maximum likelihood map	(local) maximum likelihood map	maximum likelihood map	posterior map	maximum likelihood map	posterior map
Convergence	strong	no	weak ?	no	no	strong	weak	weak
Local Minima	no	yes	yes	yes	yes	no	yes	yes
Incremental	yes	no	no	yes	yes	yes	no	no
Requires Poses	no	no	no	no	no	yes	yes	yes
Sensor Noise	Gaussian	Gaussian	any	any	any	any	Gaussian	any
Can map cycles	yes	no	yes	no	yes, but not nested	n/a	n/a	n/a
Map dimensionality	$\sim 10^3$	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited	unlimited
Correspondence	no	yes	yes	yes	yes	yes	yes	yes
Handles raw data	no	yes	yes	yes	yes	yes	yes	yes
Dynamic env's	limited	no	no	no	no	limited	no	yes

TABLE 2.2 – Comparaison des propriétés de diverses techniques de cartographie (reproduit d'après [Thrun et al., 2002](#), table 1). Texte original en anglais.

Type de méthode	Aspect temps-réel	Source de données	Évolutivité du système	Type de capteurs	Type de carte
SLAM Ajustement de faisceaux (section 2.4.8)	Temps-réel	Données véhicule	Carte statique apprise une seule fois	Caméra Monoculaire	Carte de points d'intérêt 3D
Carte basée GIS (section 2.4.1)	Hors-ligne	Données externes (GIS)	Carte statique	LIDAR aéroporté	Carte sémantique
Carte basée Photographie aérienne-LIDAR vertical (section 2.4.2)	Hors-ligne	Données externes	Carte statique	Ortho-photographie-LIDAR vertical	Carte sémantique 3D
Carte des voies de circulation (section 2.4.3)	Hors-ligne	Données véhicule (Coordonnées GPS anonymisées d'un grand nombre de véhicules)	Mises à jour périodiques de la carte	GPS	Carte sémantique-vectorielle
Cartes générées par SONAR (section 2.4.4)	Temps-réel	Données véhicule	Carte dynamique	SONAR	Carte d'occupation
Cartes générées par LIDAR (section 2.4.5)	Temps réel	Données véhicule	Carte dynamique	LIDAR	Carte d'occupation/sémantique/-nuage de points
Gestion de l'ambiguïté des configurations (Avots et al., 2002)	Temps réel	Données véhicule	Représentation dynamique	-	Carte sémantique/-suivi multi-hypothèses
Cartographie des obstacles avec information temporelle pour ajout/effacement (Biber et al., 2005)	Temps réel	Données véhicule	Représentation dynamique	LIDAR	Carte sémantique (prends la dimension temps en compte)
Carte séparant les éléments fixes et mobiles (section 2.4.7)	Hors-ligne (temps réel possible sous condition)	Données véhicule	Représentation statique	Caméra ou LIDAR	Carte d'occupation

TABLE 2.3 – Résumé des techniques de cartographie présentées.

2.5 Techniques de localisation basées sur une carte

Après avoir discuté des méthodes relatives à la génération des cartes, nous allons supposer qu’une telle carte, ainsi que décrite à la section 2.4.1 existe. Notre propos ici est d’analyser les différentes techniques de localisation basées sur une carte pré-existante. Nous analysons cette thématique sous le double angle, premièrement des capteurs et éléments à utiliser dans ce cadre, et secondement par rapport aux techniques d’alignement du véhicule sur la carte. Dans la section suivante, nous analyserons la thématique de la localisation d’un véhicule dans une carte, carte que l’on pourra enrichir. Ce thème sera vu sous l’angle des techniques d’estimation et de fusion de données à la section 2.6.

Nous introduirons tout d’abord les concepts de base relatifs à la localisation d’un robot ou véhicule dans une carte de l’environnement. Puis, nous présenterons les modalités classiques (capteurs, amers) de localisation couramment utilisées dans la communauté scientifique ainsi que le type de carte y correspondant.

2.5.1 Description des grands principes

La plupart des techniques de localisation utilisant des cartes sont basées sur un formalisme probabiliste, classiquement bayésien. Deux grandes approches générales sont prises afin de se localiser dans une carte existante et supposée exacte :

- les approches globales, qui tentent d’associer une (ou plusieurs) hypothèse(s) de localisation dans la totalité de la carte grâce aux données provenant des capteurs,
- les approches locales, qui recherchent la localisation la plus adéquate, connaissant un *a-priori* sur la localisation courante du véhicule, en fonction des données des capteurs.

Les premières sont assez souvent associées avec un formalisme permettant la prise en compte d’hypothèses multiples, de type « Probability Hypothesis Density filter » (filtres PHD). Un exemple typique de méthode multi-hypothèse est trouvée dans (Roumeliotis and Bekey, 2000). Ces méthodes permettent de résoudre le problème dit « du robot kidnappé », considéré comme un problème difficile en robotique. Celui-ci consiste à initialiser le robot sans connaissance relative à sa localisation, et à déterminer uniquement à partir des mesures capteur sa localisation exacte.

Les méthodes locales peuvent utiliser soit des méthodes paramétriques, moindres carrés par exemple, une optimisation locale de type MAP, ICP, ou bien également un formalisme bayésien, un filtre de Kalman par exemple. Elles sont généralement itératives, en utilisant le résultat de localisation obtenu à l’étape précédente comme *a-priori* de départ pour l’étape suivante. Outre les techniques indiquées précédemment on note également l’existence de méthodes *agnostiques*, n’utilisant aucun formalisme unifié et classique et consistant en l’adjonction de blocs fonctionnels de diverses natures (voir par exemple Nedevski et al., 2013, fig. 10).

Une autre séparation existe également dans les techniques de localisation :

- d’un côté, les méthodes utilisant un **ensemble** d’éléments de la carte simultanément afin que le véhicule se localise,
- d’un autre côté, celles qui n’en utilisent qu’un seul à chaque étape.

Ces premières, utilisant *plusieurs éléments*, sont dénommées **map-matching**, dans le sens où l’on tente de faire coïncider plusieurs éléments de la carte avec la totalité des données d’une trame capteur (nuage de points LIDAR, image caméra par exemple).

Les secondes sont des méthodes itératives de localisation qui corrigent ponctuellement la localisation du véhicule.

On peut également se poser la question du type de données à inclure dans la carte de l'environnement. La plupart des modalités d'acquisition utilisent la lumière visible ou infrarouge. Les deux informations retournées par exemple par une caméra ou un LIDAR, sont une information de position (dans l'espace capteur) et une information d'intensité lumineuse. L'information de position renseigne sur la position géométrique des éléments détectés par rapport au capteur ; l'information d'intensité lumineuse informe, elle, sur l'apparence des éléments d'intérêt, donc sur le processus de détection.

Bien que des implémentations puissent utiliser des capteurs comme un RADAR embarqué (voir [Ward and Folkesson, 2016](#)) ou bien un SONAR (pour [Fabrizi et al., 2000](#)), la plupart utilisent des capteurs LIDAR 2D ou 3D, ou bien des caméras, monoculaires ou binoculaires. Même si nombre de méthodes de localisation de véhicules peuvent utiliser un LIDAR 2D horizontal ainsi qu'une carte contenant l'empatement au sol des bâtiments alentours, d'autres méthodes existent également. Pour les modalités d'acquisition de type LIDAR et caméras, dans un environnement urbain, il peut être intéressant de pouvoir disposer d'une carte tridimensionnelle, notamment afin de pouvoir détecter des bâtiments en hauteur, (voir [Löwner et al., 2010](#), section 2.3 par exemple). En effet, les véhicules alentours, dans le flot de circulation ou bien garés en bord de voie, cachent fréquemment les bâtiments au niveau du véhicule. Un LIDAR 3D ou bien une caméra pourraient utiliser une telle information en hauteur. Une carte bidimensionnelle reste néanmoins exploitable, par exemple en utilisant le marquage au sol.

Nous allons maintenant passer en revue les principales modalités d'acquisition pour la localisation d'un véhicule dans une carte digitale de l'environnement, telle que décrite à la section 2.4.1.

2.5.2 Alignement de données sous forme de nuages de points

Dans la plupart des cas, les données de type « nuages de points » sont générées par des capteurs LIDAR 2D (nappe plane) ou LIDAR 3D (plusieurs nappes dans l'espace), bien que de nouvelles caméras mesurant également la **profondeur** (caméras « temps de vol » par exemple, ou bien MICROSOFT KINECT™) commencent à exister. Dans cette section, nous présentons les différentes techniques d'alignement local entre des données LIDAR et une carte de l'environnement.

Une grande quantité de méthodes d'alignement de telles collections de points sont basées sur des techniques similaires aux alignements de type ICP vus à la section 2.4.5. Dans le cadre de l'alignement entre une carte de l'environnement, une association **purement géométrique** par isométrie (translations/rotations) est faite, par exemple dans ([Haala et al., 2011](#)).

D'autres techniques d'alignement existent, basées par exemple sur la paramétrisation de primitives géométriques à partir des données (structures rectilignes, ou angles par exemple). On peut utiliser une transformée de Hough afin d'extraire les paramètres des primitives planes et linéaires comme dans ([Bernal-Marin and Bayro-Corrochano, 2011](#)), puis calculer par association dans l'espace des paramètres par rapport à la carte (ou à une trame précédente) la transformation géométrique correspondante. Une approche plus directe peut consister à extraire des primitives très informatives, telles des positions de mur, dont ([Chong et al., 2013](#)) est un exemple, des arêtes de murs, comme dans ([Im et al., 2016](#)) avec des façades de bâtiment ou bien ([Haala et al., 2011](#)) avec des angles de

murs intérieurs. Certaines approches relativement anciennes utilisent la courbure locale de la trame LIDAR, afin de détecter les points d'intérêt de celle-ci (voir [Madhavan and Durrant-Whyte, 2004](#)).

Ensuite, une association de ces primitives, via une approche par échantillonnage par exemple (filtre à particules), un filtre de Kalman, ou bien par ICP, avec des données provenant d'une carte, de la ville pour ([Im et al., 2016](#)), de l'agencement des pièces pour ([Haala et al., 2011](#)), permet de reconstruire la trajectoire.

Bien que ne traitant pas directement de nuages de points, ([Chu, 2015](#)) détecte par vision sur des images caméra des arêtes verticales d'angles de façades extérieures de bâtiment, également dans un formalisme paramétrique, et les associe à une carte urbaine de l'implantation au sol de ceux-ci. Elle se situe donc dans le même ordre de techniques que ([Im et al., 2016](#)).

Il est à remarquer que dans le cadre de véhicules en extérieur, on préfère classiquement implanter un filtre de Kalman par rapport à des primitives extraites de manière paramétrique, alors que dans des environnements intérieurs, les alignements sont généralement faits par ICP. Les primitives dans une ville évoluant lentement, on peut considérer que la carte est immuable, et dans ce cas, un filtre de Kalman est une solution classique en robotique mobile. Dans un environnement domestique, le déplacement d'un élément mobilier (chaise, fauteuil), l'ajout ou le retrait d'objets (cartons...) peuvent générer ou supprimer des primitives assez rapidement, et dans ce cas, les techniques de type ICP offrent une technique d'acquisition et d'alignement plus simple.

2.5.3 Détection du marquage au sol

Dans un environnement routier, une grande partie de la signalisation relative à son usage est au sol, seule une faible portion étant aérienne (panneaux et feux de signalisation). On peut donc penser tirer avantage, comme les conducteurs humains, de l'information au sol (limites latérales de voies, signalétique distinctive comme stop, cédez-le-passage et fléchage au sol par exemple). Nous allons détailler les quelques approches qui y sont relatives.

Plusieurs idées d'utilisation du marquage au sol sont possibles :

- corriger la position du véhicule de manière itérative par rapport au marquage au sol limitant les voies de circulation,
- utiliser la signalisation au sol **univoque** (ayant un minimum d'ambiguïté dans la zone proche, « à peu près uniques » comme le fléchage au sol, la signalisation au sol de priorité aux carrefours) pour corriger de manière plus certaine la localisation du véhicule,
- enfin, utiliser des éléments **absolument non-ambigus** (comme un carrefour complexe par exemple) dans son intégralité, de manière à pouvoir se localiser de manière **globale et certaine**.

Dans ces diverses possibilités, notons tout d'abord que le premier item, la correction de position du véhicule dans la voie correcte de circulation, a historiquement fait l'objet de recherches au cours des années 2000, en raison de son intérêt pour le guidage autonome en suivi de voie de circulation sur les routes et autoroutes. On peut citer comme exemples de telles méthodes ([Laneurit et al., 2005](#)), ([Tao et al., 2013](#)), ([Lu et al., 2014](#)) et ([Gruyer et al., 2014](#)), qui permettent par identification de la voie de circulation de se localiser de manière correcte dans le couloir de circulation emprunté effectivement. Ces méthodes détectent les limites de voies de circulation par détection globale des primitives « ligne

continue » et « ligne discontinue », et permettent au véhicule de corriger sa position de manière latérale par rapport au sens des voies. Ces techniques emploient généralement une approche bayésienne itérative, comme le filtre de Kalman ou le filtre à Particules (Particle Filter, PF).

Le second, relatif à la détection de **primitives spécifiques**, suffisamment rares à proximité pour pouvoir être identifiées de manière univoque, permet généralement de corriger la localisation du véhicule dans plus d'un degré de liberté simultanément, donc de manière plus précise. On peut citer par exemple l'emploi de cartes de **haute précision** décrivant le marquage au sol comme dans (Schreiber et al., 2013). Les éléments individuels du marquage au sol (segments de ligne discontinue) sont identifiés en tant que tels (malgré de possibles ambiguïtés), et non l'élément « global » de **séparation de voie de circulation**.

On peut également citer l'approche de (Levinson et al., 2007) basée sur un capteur LIDAR 3D utilisant la donnée réflectivité de la trame de données (de manière semblable à une caméra) afin de l'associer par **map-matching** sur $\mathbb{S}\mathbb{E}(2)$ à une carte au sol ne comprenant que la route et incluant le marquage. (Hata and Wolf, 2014) utilise également un processus d'association entre les données au sol provenant du LIDAR et une carte comprenant le marquage au sol. Un étage de filtrage particulière confirme les particules ayant une bonne correspondance avec la route, et rejette celles pour lesquelles il est mauvais. (Pink, 2008; Pink et al., 2009) utilisent une technique d'association de même nature afin d'identifier le marquage au sol et l'associer par ICP à une carte absolue de la surface de la route. Utilisant une caméra binoculaire, (Senlet and Elgammal, 2011) reconstruit l'image de la route au sol, et par association avec une vue aérienne de la route, permet en théorie une localisation absolue sans dérive.

Un exemple de méthode hybride entre les deux précédentes est présenté dans (Nedevschi et al., 2013) et permet d'identifier la voie de circulation, sa direction ainsi que la position du véhicule à une précision décimétrique.

La troisième méthode, qui n'est pas spécifiquement reliée au marquage au sol, mais plutôt à une identification **univoque** des lieux dans lesquels le véhicule est situé, est discutée dans la section suivante.

2.5.4 Localisation par association de vue caméra

Dans cette section, nous nous intéressons à l'identification de la localisation du véhicule à partir de l'apparence de l'environnement à proximité. Cette apparence est donnée par une image visuelle obtenue par caméra. Il existe deux types de telles identifications :

- sans *a-priori* sur la localisation, le résultat pouvant contenir plusieurs « hypothèses » en fonction des potentielles similitudes,
- avec un *a-priori* et se concentrant sur l'hypothèse la plus plausible.

Le premier problème est considéré comme un problème difficile, et n'a actuellement pas de solution absolue. Le second est plus accessible, mais peut potentiellement diverger si à un moment donné, une mauvaise association est réalisée.

Une technique basée sur les associations d'images par méthode « sac de mots » (Bag of Words, BOW) est présentée dans (Qiao et al., 2014). Cette méthode peut dans certains cas échouer en générant de *faux positifs*, et une étape de validation est effectuée par superposition de l'image LIDAR attendue avec les données obtenues. Ceci montre que la maturité de cette technique n'est pas encore atteinte, alors que les êtres humains la maîtrisent particulièrement bien, et indique par là même que les différences entre les

résultats obtenus par des machines et les humains peuvent être importantes. D'autres techniques d'association basées par exemple sur des descripteurs d'image type SURF (voir [Badino et al., 2011; 2012](#)) existent également.

Dans ([Wong et al., 2014](#)), on recherche dans les images de la caméra des éléments distinctifs (sommets de bâtiments, candélabres, enseignes commerciales...) visibles depuis l'emplacement estimé du véhicule. Une association de ces éléments distinctifs est réalisée à partir d'une base de données contenant des images de référence associées aux positions correspondantes.

De manière générale, les méthodes basées sur des associations entre image caméra et image de référence stockées en base de données sont avec peine aussi robustes que les techniques basées sur des capteurs LIDAR.

2.5.5 Alignement sur les bords de la route

Dans cette section, nous nous restreignons à une hypothèse extrêmement contraignante : *le véhicule se situe toujours sur une route carrossable, et dans une orientation compatible de celle de la voie sur laquelle il est localisé*. Cette hypothèse s'accorde bien à l'analyse dans le cadre d'une technique de localisation pour du suivi de voie de circulation, mais est totalement incompatible des environnements naturels (chemins forestiers, milieu agricole pour n'en citer que quelques uns). Dans le cas d'un changement non-répertorié des voies de communication routières, ceci peut également poser des problèmes de sécurité importants (non-reconnaissance de voie de circulation, changement d'axe de circulation interdit...). Pour ces raisons, ces techniques sont relativement rarement employées en dehors du contrôle d'un être humain.

Une technique simple consiste à se positionner par rapport aux bords matérialisés par une sur-élévation des trottoirs typiquement, car un pinceau LIDAR 2D vertical à l'avant du véhicule, ou un LIDAR 3D en analyse du sol permettent de le détecter aisément. Cette technique est néanmoins facilement mise en défaut à proximité des bateaux au niveau des trottoirs. Un exemple de telle détection est présenté dans ([Hata et al., 2014](#)).

La plupart du temps, une carte de type GIS va indiquer une surface carrossable pour l'axe de circulation emprunté comme dans ([Fouque et al., 2008](#)), ou tout au moins un segment de route sur lequel le véhicule est supposé se trouver (voir [Peyraud et al., 2013; Smaili et al., 2007](#)), et l'on va alors « projeter » l'état du véhicule, obtenu par intégration de l'odométrie, sur cette voie de circulation. Dans ces deux approches, l'information de position absolue comme celle d'angle de direction sont utilisées pour effectuer la correction en question.

2.5.6 Synthèse

On retiendra que dans le cas d'une localisation basée sur une carte de type GIS, les méthodes **locales** d'association de type LIDAR sont jugées robustes, particulièrement avec des primitives. Une association globale sans *a-priori* de l'ensemble des éléments vus par un LIDAR est encore actuellement un problème jugé difficile, notamment en raison de la difficulté de la mise à l'échelle de cette recherche sur la totalité d'une carte dont la taille peut augmenter, ainsi qu'en raison des ambiguïtés possibles que les données détecteurs ne permettent pas toujours de lever. L'existence d'un *a-priori* rend ce problème beaucoup plus simple. Un exemple, considéré comme robuste, consiste à prendre une détection de

bâtiments (LIDAR 2D horizontal ou bien LIDAR 3D en balayage quasi-horizontal) et de l'associer à une carte.

Les méthodes basées sur la détection du marquage au sol de manière individuelle permettent également une telle localisation, leur robustesse pouvant être impactée par le changement d'angle de vue du véhicule (le franchissement d'un ralentisseur peut par exemple provoquer un mouvement de tangage important sur un véhicule). Un intérêt des industriels semble se manifester pour de telles solutions, basées sur l'utilisation d'une carte de **haute précision** du marquage au sol, celui-ci étant présent dans quasiment toutes les situations que l'on rencontre habituellement.

Les méthodes basées sur un alignement de la position du véhicule, soit sur une voie de la route (positionnement latéral sur la chaussée grâce au marquage au sol), soit de manière moins précise sur un tronçon d'axe routier (alignement sur un axe de circulation) contraignent le problème et ne permettent pas toujours de le poser avec suffisamment de généralité pour qu'il soit résolu de manière robuste. Ces techniques peuvent nécessiter une supervision humaine.

Enfin, les techniques les moins matures de notre point de vue sont celles qui, d'un point de vue humain semblent les plus intuitives, et se focalisent sur la similitude visuelle entre une acquisition caméra et une base de données d'images clé.

Il est à noter qu'une résolution mono-hypothèse itérative, facile à réaliser, peut se focaliser sur une mauvaise solution (en raison d'une ambiguïté de détection par exemple). Bien que ceci se produise rarement, cela est difficilement récupérable une fois que l'erreur s'est produite. La solution théorique proposée classiquement dans ce cas est une localisation sans *a-priori*, qui est un problème non-résolu actuellement. De ce fait, la localisation robuste d'un véhicule est un problème qui acquiert progressivement de la maturité, mais que l'on ne peut pas encore considérer comme résolu.

Remarquons également que certaines méthodes de calcul sont spécifiques d'un type particulier de données (par exemple ICP utilise exclusivement des nuages de points, BOW des images caméra), et par conséquent sont limitées à certains types de capteurs pouvant fournir de telles données (LIDARs 2D/3D, caméra de profondeur pour ICP, caméra 2D pour BOW...). Il semble bloquant, s'agissant de techniques de localisation, d'être limité, par le choix d'une approche, à des capteurs d'une certaine classe. Dans les travaux de notre équipe, nous recherchons un formalisme plus général et pouvant englober potentiellement un nombre plus grand de capteurs, afin de lever ce type de contraintes.

2.6 Techniques de localisation et de gestion d'une carte dynamique

Dans les deux sections précédentes, nous avons analysé les méthodes de cartographie, puis celles de localisation d'un véhicule sur une carte contenant des éléments « immuables ». Dans cette section, nous nous intéressons à la thématique issue des deux précédentes, la localisation d'un véhicule dans une carte que l'on peut construire au fur-et-à-mesure. Dans ce cadre, la carte, même si elle contient des éléments fixes (bâtiments, pylones, etc...), doit alors avoir une structure qui évolue et permette de l'enrichir. De plus, nous prenons ici le parti d'être capable d'utiliser une carte du type de celle présentée à la section 2.4.1. Ce type de carte peut potentiellement contenir des informations de nature variée (position, hauteur au sol, orientation par exemple) sur des types d'éléments divers (bâtiments, pylones de feux de signalisation, ou de candélabres, panneaux de signa-

lisation, mobilier urbain tels des abris-bus...). Nous nous attachons ici à regarder cette thématique sous un angle du type *fusion de données*.

2.6.1 Localisation et fusion de données

Dans la section 2.5, nous nous sommes penchés sur le problème de la localisation d'un véhicule dans une carte existante. Nous allons ici analyser ces types de méthodes du point de vue de la technique de fusion de données utilisée. On entend par fusion de données le « processus de calcul » permettant d'incorporer de nouvelles informations à la « connaissance » obtenue jusqu'à l'instant actuel par le robot.

Si nous analysons les méthodes décrites dans la section 2.5, on peut séparer (indépendamment des modalités capteur) celles-ci en deux groupes, celles qui utilisent un *alignement local* et celles dont *l'alignement est global*. Le processus de **fusion de données** que nous allons évoquer consiste à réaliser les calculs de cette procédure d'*alignement*. En général, ce processus est indépendant du caractère *alignement local* ou *global*.

Prenons comme exemple la méthode de localisation décrite dans (Kais et al., 2004). La manière de procéder la plus répandue pour la localisation d'un robot mobile consiste à utiliser une technique d'estimation appelée *filtre de Kalman* (voir la section 4.7.2). C'est elle qui est utilisée dans cette dernière référence. La carte est constituée d'entités fixes, les *amers*, qui sont des arêtes verticales entre deux murs ou des poteaux. Le filtre de *Kalman* permet à partir de l'observation d'*un seul amer* (alignement local), ou bien de *plusieurs amers* (alignement global), de corriger la localisation estimée du véhicule. On peut trouver un exemple visuel de l'exécution de ce type d'algorithme sur la figure 2.24.

L'association simultanée de plusieurs éléments observés par un véhicule (points de Harris de type SLAM visuel monoculaire) avec le plan d'un bâtiment (murs visibles), comme dans (Parsley and Julier, 2009), est un autre exemple du même type de techniques. Un exemple est présenté sur la figure 2.25.

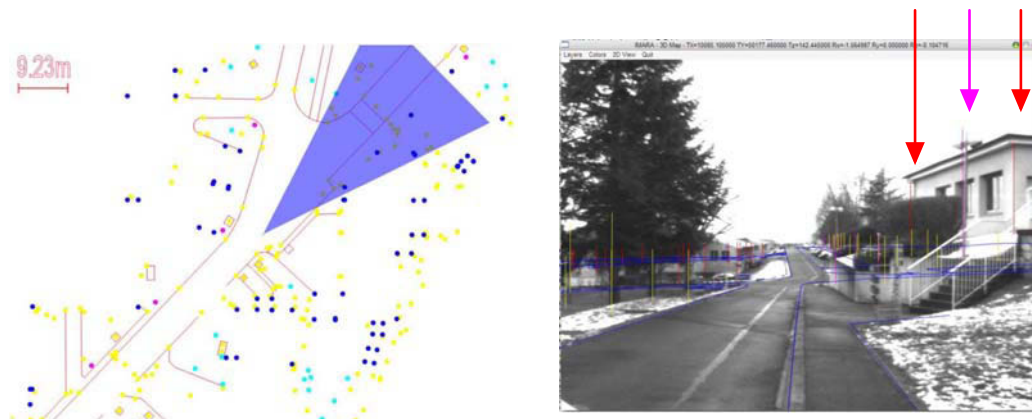


FIGURE 2.24 – À gauche, une carte GIS contenant des éléments verticaux (arêtes de murs, poteaux). À droite, une détection de ces mêmes amers par une caméra. Reproduit d'après (Kais et al., 2004, figures 5 et 6).

Dans ce qui suit, nous nous intéressons aux techniques permettant de gérer une carte pouvant contenir un nombre quelconque d'éléments de diverses natures (véhicules, amers de différents types...). Nous faisons remarquer à ce point que la plupart des méthodes dévolues à cette manière de gérer les éléments cartographiés sont orientées purement

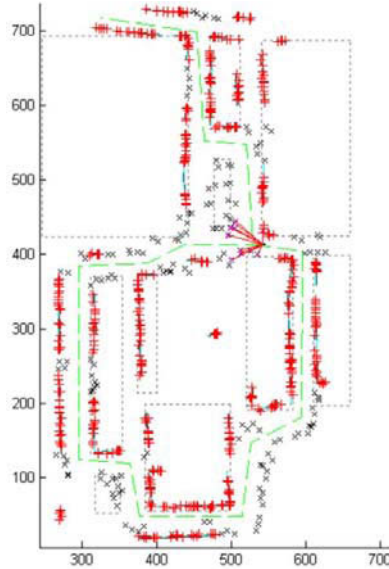


FIGURE 2.25 – Exemple de localisation et cartographie basé sur une carte *a-priori* (murs en pointillés gris). La position de la caméra est indiquée par le point de concours des segments rouges, les points d'intérêt associés à des murs sont indiqués par des + rouges, ceux qui ne le sont pas par des croix grises. La trajectoire de la caméra est indiquée par des pointillés verts. Reproduit d'après (Parsley and Julier, 2009, figure 4).

véhicule. Les méthodes utilisées dans ce cadre peuvent néanmoins être généralisées pour couvrir des amers de différents types. En effet, on peut considérer un amer comme un véhicule qui ne se déplace jamais. Certaines méthodes de localisation *multi-robot* utilisent des robots fixes à des fins de triangulation lorsque d'autres robots se déplacent, ce qui rend notre analogie plus pertinente.

Dans les techniques du type *filtre de Kalman*, qui sont des méthodes standard dans l'état de l'art, on peut se focaliser sur la gestion de la localisation d'un unique véhicule dans une carte **fixe** et **absolue** de l'environnement. Ceci est conforme aux techniques exposées à la section 2.5. On peut aussi vouloir gérer la carte comme **partiellement inconnue** ou **imprécise**, lorsque la position d'amers (ou bien d'autres véhicules) est mal connue et doit ou peut être précisée. C'est ce thème que nous allons développer.

Dans ces approches, chaque élément de la carte (véhicule, amer fixe) est modélisé par un vecteur d'état (\mathbf{x}_v pour le véhicule, \mathbf{x}_k , $k \in \{1, 2, \dots, N\}$ pour les N amers fixes de la carte). Un vecteur d'état global $\mathbf{x}_G^\top = [\mathbf{x}_v^\top, \mathbf{x}_1^\top, \dots, \mathbf{x}_N^\top]$ permet de stocker toutes les informations relatives à la carte. Associé à ce vecteur d'état, une matrice de covariance, représentant les covariances de tous les éléments du vecteur d'état, ainsi que leurs covariances croisées, modélise l'incertitude globale de la carte. Cette formulation du problème est commune au SLAM basé sur un filtre de Kalman.

Nous allons maintenant nous intéresser aux techniques permettant de gérer plusieurs éléments différents dans une carte unique. Dans notre approche, nous choisissons de gérer des amers et un (ou potentiellement des) véhicule(s). Étant donné que la plupart des approches gérant une carte de l'environnement contiennent des véhicules au détriment d'amers, nous allons les présenter en tant que telles. Mais il ne faut pas perdre de vue que l'on peut, dans ces formalismes, remplacer les véhicules ajoutés par des amers. Il suffit simplement de supposer que ceux-ci ne se déplacent pas.

2.6.2 Localisation collaborative par filtre de Kalman

La plupart des méthodes initialement envisagées en vue de permettre la localisation d'une flotte de véhicules sont basées sur filtre de Kalman, ainsi que décrit à la section 2.6.1. Des généralisations de cette technique ont été proposées, et ont pour trait commun :

1. la formalisation de l'état de chaque véhicule de la flotte dans un unique vecteur d'état,
2. l'utilisation d'une procédure centralisée de prédiction et mise à jour,
3. l'exécution de celle-ci de manière périodique.

Un exemple d'implémentation d'une telle méthode est décrite dans (Huang et al., 2009, section 2.1). Le filtre de Kalman est connu comme la méthode d'estimation optimale dans le cas de systèmes linéaires. Quoique les systèmes robotiques comportent presque-exclusivement des non-linéarités, lorsque celles-ci sont modérées, sa généralisation, appelée *filtre de Kalman étendu*–EKF, reste pertinente.

Le problème de ce genre d'approche est son coût calculatoire et en stockage, notamment au niveau des mises à jour ainsi que du stockage des matrices de covariance (se référer à Carrillo-Arce et al., 2013, section I). De plus, les problèmes de sûreté de fonctionnement sont importants avec ce type d'approche, car il suffit que le centre de fusion tombe en panne pour entraîner une panne généralisée de tous les autres véhicules.

Afin de résoudre ce type de problèmes, des approches décentralisées ont été développées, comme dans (Huang et al., 2009, section 3) et (Huang et al., 2011). Un des problèmes néanmoins mentionné est celui relatif à l'invariance par translation–rotation de l'ensemble de la flotte. En effet, lors qu'aucune mesure absolue n'est disponible (élément statique, mesure de type GPS), seules des différences de position entre robots sont possibles. Si une rotation était appliquée à l'ensemble de la flotte de robots, les mesures robot–robot resteraient inchangées. Il en est de même en translation. Ces degrés de liberté sont dans ce cas inobservables (cf. Huang et al., 2009, section 2.2.2). Une dérive globale de l'état estimé de la flotte est donc possible, en rotation comme en translation.

Une autre méthode décrite dans (Aragues et al., 2013), consiste à utiliser une généralisation multi-véhicule du filtre informationnel (une formulation alternative du filtre de Kalman), et à réaliser la fusion de manière globale sur chaque véhicule par incrément d'information. La cohérence du système est démontrée, en ce qu'il implémente sur chaque véhicule l'équivalent d'un filtre de Kalman centralisé. Les seuls problèmes potentiels consistent en la non-mise à l'échelle des tailles de matrices de covariance (évoluant comme $O(N^2)$), et par conséquent de la bande passante utilisée pour échanger les incréments sur chaque carte. Une perte de communication d'un des véhicules pourrait également être problématique pour le bon fonctionnement de l'algorithme. D'autres approches similaires utilisent des capteurs LIDAR (par exemple Fenwick et al., 2002).

Un exemple plus simple de coopération entre deux robots, dont un dépourvu de capteurs optiques, est présenté dans (Oussalah and Wilson, 2007), et montre la possibilité de comprendre dans une flotte de véhicules des configurations de capteurs différentes pour chaque robot. Ceux-ci s'échangent des données de mesure, et permettent l'emploi de méthodes de triangulation coopérative. Le principe est également basé sur l'emploi d'un filtre de Kalman ; malheureusement le formalisme de cette approche ne permet pas une généralisation à plusieurs robots, et la carte reste centralisée sur un unique ordinateur (pour des raisons d'encombrement et de capacité de calcul notamment).

2.6.3 Approches par fusion de cartes locales

Dans ce qui précède, plusieurs techniques basées sur des cartes globales et un filtre de Kalman ont été présentées. Mais d'autres approches utilisant une carte globale sont possibles. Dans (Rodriguez-Losada et al., 2004), une méthode dans laquelle une carte locale propre à chaque véhicule permet de générer une carte globale pour toute la flotte de robots.

La formulation du problème de SLAM pour chaque robot est similaire à ce qui est décrit dans la section 2.6.1. Le filtre utilisé est la version informationnelle du filtre de Kalman, l'« *Information Filter* », et dont les résultats sont rigoureusement identiques. La formulation propre du filtre informationnel est telle que chaque mise à jour de la carte est réalisée par un processus d'addition sur le vecteur d'information et la matrice d'information (l'équivalent du vecteur d'état et de la matrice de covariance dans la représentation informationnelle). Chaque carte locale est donc le résultat d'une procédure d'accumulation de toutes les innovations passées et courante du robot.

La carte globale est réalisée par accumulation sur chaque véhicule de toutes les innovations venant des autres, outre celles qui sont locales. On construit ainsi sur chaque véhicule une carte globale identique en tous points à celle générée sur les autres véhicules. Les messages sont diffusés sur tout le réseau, ce qui pourrait induire une certaine charge à ce niveau. De plus, on peut penser que la rupture de la liaison de communication entre robots pourrait dans certains cas générer une perte de fonctionnalité.

Cette approche est néanmoins intéressante car elle permet de montrer que la génération d'une carte globale n'est pas nécessairement reliée à une construction directe centralisée mais peut être générée à partir d'éléments locaux à chaque véhicule.

Une autre approche dans laquelle un environnement de grande taille est géré par la juxtaposition de cartes locales, appelée atlas, est présentée dans (Bosse et al., 2004). Les cartes locales ne sont pas ici locales au robot mais à la zone géographique. Elles sont juxtaposées exactement comme des sous-cartes de zones géographiques se juxtaposant (de manière similaire aux zones géographiques du système de cartographie UTM), et entre lesquelles on passe par un changement de repère et de coordonnées. Un filtre de Kalman global gère la carte globale, mais dans chaque tâche de navigation de chaque véhicule, ne sont utilisées que des cartes locales. Un processus assez complexe d'association de données est utilisé, par exemple entre deux cartes locales pour les aligner l'une à l'autre, ou pour associer des éléments détectés par les capteurs à ceux d'une carte, locale ou globale. Une machine à états finis est utilisée afin de gérer le niveau de validité des éléments contenus dans la carte.

Ce type d'approche serait typiquement utilisé pour se déplacer et cartographier une ville complète. Par contre, la potentielle mise à l'échelle pourrait peut-être générer des problèmes de fermeture de boucles larges, étant donné la potentielle dérive importante avec la taille de la zone cartographiée.

2.6.4 Approches par Intersection de Covariance

Dans la plupart des cas, le passage des méthodes présentées précédemment à l'échelle réelle (ville, pays) est limitée pour les approches centralisées de type « filtre de Kalman » du fait du stockage d'une matrice de covariance de taille importante. De plus, les risques de panne sur le centre de fusion rendraient potentiellement toute l'architecture inopérante, ce qui est un problème important de sûreté de fonctionnement.

De ce fait, on peut avoir l'idée de séparer les états de chaque élément et de les rendre indépendants les uns des autres. Mais un problème se pose alors avec les approches de type Kalman : les phénomènes de sur-convergence. En effet, on peut prouver que, lorsque plusieurs éléments appartiennent à une carte contenant des éléments estimés séparément, le fait d'échanger des mesures entre deux d'entre eux provoque la prise en compte un nombre non-borné de fois de la même information commune aux diverses estimations. Cet effet provoque, en règle générale, la diminution à un niveau *irréaliste* des covariances des éléments en question et par voie de conséquence l'effondrement de la carte. Pour une explication qualitative du phénomène, se référer aux sections 4.7.3 et 4.7.4.

Plusieurs approches tentent de pallier les limites précédentes, notamment au niveau de la validité des résultats générés. L'une d'elles utilise une technique de fusion appelée *filtre à intersection de covariance* (voir pour plus de détails la section 4.7.5). Elle permet de ne stocker que des états locaux pour certains nœuds, tout en permettant leur fusion ultérieure (voir Carrillo-Arce et al., 2013, section I). Cette approche permet, à la différence de filtres de Kalman séparés, de maintenir la cohérence de l'estimation.

Plusieurs exemples sont présentés dans (Julier and Uhlmann, 2007), (Carrillo-Arce et al., 2013) et (Li and Nashashibi, 2013). (Julier and Uhlmann, 2007) présente le principe du SLAM par intersection de covariance. Une partie particulièrement intéressante est celle de (Julier and Uhlmann, 2007, section 3), qui détaille les problèmes inhérents au SLAM basé sur un filtre de Kalman. Le modèle d'observation présenté dans (Julier and Uhlmann, 2007, section 4.2.1) que nous reprenons dans nos travaux, permet la gestion d'une infime partie de la carte lors d'une mise à jour avec une approche par intersection de covariance, au lieu de la totalité de la carte, comme c'est le cas avec un filtre de Kalman centralisé.

Dans (Li and Nashashibi, 2013), une variante de ce type de filtres, appelée *intersection de covariance partitionnée*, est utilisée. Elle permet de reconstruire des trajectoires d'autres véhicules jouxtant le véhicule autonome par fusion multi-capteurs.

(Carrillo-Arce et al., 2013) présente une technique de fusion dans laquelle un filtre de Kalman local à chaque véhicule permet l'estimation de la pose d'autres véhicules observés. La fusion entre l'estimation faite par le véhicule observateur, et celle stockée par le véhicule observé est opérée par un filtre à intersection de covariance.

Dans tous les cas, l'effort de stockage et de communication est réduit au minimum. Un inconvénient est la non-optimalité de l'estimation, le filtre de Kalman centralisé restant toujours la meilleure méthode en terme d'erreur, malgré ses limitations en termes de mise à l'échelle.

2.6.5 Approches de type Maximum-A-Posteriori

Une autre approche utilise une formulation du problème de localisation de type *Maximum-A-Posteriori* (MAP). Les méthodes de type MAP calculent une densité de probabilité postérieure dans l'espace d'état, puis sélectionnent comme nouvel état du système celui dont la densité de probabilité est maximale. Le processus est itéré comme un filtre de Kalman.

Un exemple est présenté dans (Nerurkar et al., 2009). Un processus d'optimisation de l'erreur quadratique par méthode *Levenberg-Marquardt* est opéré à chaque étape. Ce processus optimise, non pas la dernière séquence de mesure comme dans un filtre de Kalman, mais toute la séquence d'observations de toute la flotte de robots. La croissance du temps de calcul d'une étape en fonction de la durée de la totalité de la trajectoire à l'instant T varie en $O(T)$, et est non-bornée. Malgré le fait que le temps de calcul puisse

être faible au départ, viendra toujours un moment où celui-ci fera dépasser les exigences temps réel de l'application.

2.6.6 Approches de type Channel Filter

Une dernière approche possible est présentée schématiquement dans (Grime and Durrant Whyte, 1994), puis déclinée sous de multiples implémentations, par exemple (Ong et al., 2005; 2006a; 2008), dans le cadre du suivi de cible aérienne par des drones en coopération.

La base du système est une approche de type *filtre informationnel* décentralisée. Un filtre informationnel n'est ni plus ni moins qu'une autre formulation du filtre de Kalman. Cette approche, valable avec un seul robot, a été étendue à plusieurs robots distincts dans (Grime and Durrant Whyte, 1994), en utilisant une formulation de type filtre de Kalman. Le principe de l'approche décentralisée consiste à prendre en compte la *partie de l'estimation commune* entre plusieurs nœuds du système de manière itérative.

Cette partie commune entre deux estimations, provenant de deux robots différents, provient d'une source commune aux deux données. Afin d'éviter le double comptage d'information (voir la section 4.7.4 pour plus de détails), une fois calculée, cette quantité est déduite de l'estimation fusionnée. Cette manière de procéder évite de prendre en compte un nombre arbitraire de fois la même donnée, et donc la sur-convergence de l'estimation.

Plusieurs implémentations de cette manière de soustraire l'information commune ont été réalisées, notamment par *filtre à particules* (voir Ong et al., 2008), mais aussi par *mixtures de gaussiennes–intersection de covariance* (voir Ong et al., 2005, section 5A). La problématique de cette méthode est la potentielle lourdeur calculatoire du filtre à particules, le nombre de celles-ci variant de manière exponentielle avec le nombre de dimensions de l'espace d'état. Il est fréquent que des approches par filtre à particules utilisent de l'ordre d'un millier de particules. Ces particules doivent être calculées toutes en temps réel sur une seule machine locale, requérant alors une forte capacité de calcul.

Cette technique est par contre immune aux problèmes de linéarisation, qu'il est courant de retrouver dans les systèmes de type SLAM par caméra monoculaire basé sur un filtre de Kalman. L'implémentation de type décentralisée de type *channel filter* évite le problème d'une panne complète due à la défaillance d'un unique nœud–drone.

2.6.7 Approches Multi-Hypothèses

Une approche particulière, présentée dans (Bahr et al., 2009), a été développée dans le cadre de travaux ayant pour but la gestion d'une flotte de micro-drones sous-marins. La technique emploie des navires de surfaces équipés de capteurs GPS, donc localisés dans l'espace, qu'un ou plusieurs drones sous-marins interrogent en distance par écho SONAR. Chaque écho (distance à un navire de surface) génère un cercle de positions possibles. L'intersection de différents cercles d'hypothèses conduit à l'élimination de toutes les hypothèses non-viables, donc en définitive la sélection de la seule le restant.

Les cercles de distance aux navires de surfaces peuvent s'intersecter au même instant (plusieurs échos dans la même zone), ou nécessiter de les propager le long de la trajectoire estimée du robot. Des capteurs inertiels embarqués sur les drone sous-marins permettent cette dernière opération. Des véhicules sous-marins peuvent également procéder à de telles mesures entre eux.

Afin d'éviter la sur-convergence, une liste de filtres de Kalman générés au fur et à mesure, et estimés immune aux corrélations, est créée. La position du véhicule sous-marin est choisie par un système de vote comme la position donnée par le filtre de Kalman dont le déterminant de la covariance est le plus faible.

L'information de position absolue est fournie par les capteurs GPS des véhicules de surface, et diffusée dans la flotte de véhicules sous-marins, par rebond par exemple, dans le cas d'une flotte largement répartie.

Une autre méthode multi-hypothèse, utilisant des filtres multi-hypothèses (*Multi Hypothesis Tracking*, MHT), et présentée dans (Zhang et al., 2014), permet de suivre plusieurs véhicules à partir d'un seul capteur RADAR. La théorie sous-jacente est celle des *Random Finite Sets* (RFS), qui utilise des densités de probabilité prenant des valeurs multiples simultanément. Ce type de méthodes permet de gérer nativement les problèmes d'associations entre mesures et éléments de la carte avec le même formalisme que celui de la localisation. La raison principale qui limite l'emploi de telles méthodes est l'utilisation de méthodes d'échantillonnage probabilistes telles que Monte-Carlo, ainsi que le grand nombre d'échantillons nécessaires afin d'obtenir une estimation valable. Cela impacte directement le temps de calcul, et limite de ce fait les potentielles applications en robotique en imposant l'utilisation de fortes capacités de calcul, ce qui est toujours problématique sur des équipements embarqués.

2.6.8 Synthèse sur la cartographie et la gestion d'une carte dynamique

Nous venons de présenter les différentes méthodes classiques utilisées dans le cadre de la cartographie collaborative par une flotte de robots. Il est à noter que presque toutes les approches utilisent des cartes sémantiques-métriques, à la différence des approches présentées dans la section 2.4. On peut l'analyser comme une conséquence de la bande passante limitée accessible aux robots mobiles couplée à la forte mémoire requise par la plupart des méthodes de cartographie (nuage de points ajustement de faisceaux, grilles d'occupation...) et donc aux phénomènes limitant l'échange de données cartographiées.

Dans ce type de méthodes, le formalisme est basé en règle générale sur le filtre de Kalman et ses variantes (filtre informationnel, filtre à intersection de covariance, filtre à intersection de covariance partitionnée, *channel filter*), qui sont tous basés sur un formalisme proche du principe bayésien. Quelques méthodes comme les méthodes à estimation MAP, basées sur une optimisation globale sur la totalité des données disponibles, existent, mais restent marginales pour ce genre d'applications.

Les applications d'un principe multi-hypothèses sont prometteuses, mais se limitent pour l'instant à des véhicules d'un certain coût, encombrement, et à certaines situations, notamment des véhicules proches engendrant des problèmes d'identification et d'association, ou des systèmes dans lesquels les non-linéarités ou la méconnaissance de l'environnement est grande. Ces systèmes pourraient permettre de lever les ambiguïtés entre diverses hypothèses de manière bien plus robuste et intégrée que n'importe quel autre système, mais avec un coût calculatoire important. Les techniques typiques sont les *Probability Hypothesis Density filters* (PHD) et une de leurs variantes, les filtres à particules, que nous ne mentionnerons par outre-mesure ici.

D'autres tendances orthogonales aux méthodes précédentes existent, la décentralisation des calculs pour la flotte de robots par exemple, mais aussi la gestion d'un atlas de cartes pour une carte possédant une surface étendue.

Conclusion

Après avoir décrit les problématiques classiques pour la localisation d'un robot et la création de cartes, nous avons présenté quelques types de capteurs différents fréquemment utilisés en robotique. Nous avons ensuite présenté les méthodes de cartographie classiques dans le domaine de la robotique, en les couplant avec les capteurs avec lesquels elles sont généralement utilisées. Nous avons enfin présenté quelques méthodes permettant d'utiliser des cartes contenant des éléments en nombre variable, dont l'état peut être mis à jour en fonction des données capteurs des véhicules. Ces dernières techniques sont de même nature que celles permettant potentiellement de mettre à l'échelle le problème de la cartographie.

Le chapitre suivant va s'attacher à présenter notre vue des objectifs de ces travaux de thèse, des problématiques qui y sont associées ainsi que la manière dont nous proposons d'y répondre techniquement.

Chapitre 3

Approche Proposée

Nous allons maintenant analyser, dans ce court chapitre, et à partir de la bibliographie précédente, ce qui nous a paru pertinent, ceci afin de pouvoir développer notre approche. Comme réflexion de base, nous avons estimé que l'utilisation d'une carte afin qu'un véhicule se localise par rapport aux éléments qui y sont référencés était souhaitable. Une préférence est mise sur des cartes absolues (géoréférencées) permettant de stocker dans un formalisme unique les divers types d'éléments, notés amers. Les objectifs de ces travaux, outre l'estimation de l'état du véhicule, sont les suivants :

1. corriger la position des amers qui seraient mal positionnés dans la carte, tout en rendant possible la localisation d'un véhicule dans une telle carte,
2. supprimer des amers de la carte qui seraient impossibles à détecter, qui par exemple n'existeraient pas,
3. pouvoir ajouter des amers détectables et non encore insérés dans la carte de l'environnement.

Après un rapide survol de la bibliographie, nous exposerons les choix fondateurs qui ont permis de développer notre approche. Bien que dans la réalité, ce raisonnement ait été fait pendant la durée de ces travaux et à mesure de l'avancée du travail, nous allons tenter ici de le présenter de manière simple.

3.1 Choix de la technique de localisation sur carte

Dans l'analyse bibliographique du chapitre 2, une analyse des techniques de localisation a été faite. Parmi celles-ci, on trouve :

- les techniques paramétriques d'alignement (ICP par exemple), qui sont mono-hypothèse,
- les techniques probabilistes mono-hypothèse de type *filtre de Kalman*,
- les techniques probabilistes multi-hypothèses.

Commençons notre analyse par les techniques *multi-hypothèses*. Celles-ci sont capables de gérer à peu près tous les cas classiques dans lesquelles les autres méthodes peuvent échouer (notamment le problème de l'ambiguïté, voir plus bas). Nous allons pourtant les laisser de côté dans ce qui suit, en raison de leur fort coût calculatoire, de stockage, ainsi que des problèmes potentiels de mise à l'échelle dans le cadre multi-hypothèse (explosion combinatoire du nombre d'hypothèses à gérer). Ces problèmes sont, à l'heure actuelle, sans solution. Nous nous concentrons donc dans ce qui suit sur les techniques mono-hypothèse.

Les techniques de type paramétrique, basées par exemple sur une procédure ICP, sont connues pour être robustes dans le cas où la grande majorité des éléments dans le champ du capteur sont immobiles. Néanmoins, dans des cas d’environnements encombrés, il est possible qu’elles échouent. De plus, ces méthodes sont basées sur des données de type « nuage de points », qui sont potentiellement **volumineuses**. Une telle carte, à l’échelle d’une ville, contiendrait des milliards de points, et pourrait ralentir leur utilisation et/ou la correction des données qu’elle contient. Ce genre de localisation ne convient pas à des données de type **sémantique**, telles des données issues de cartes « riches », comme OPENSTREETMAP.

D’un autre côté, les techniques de type probabiliste sont parfaitement adaptées à des données hautement sémantiques, en raison de la faible quantité de paramètres nécessaires pour encoder l’état d’un élément de l’environnement. Dans ce cadre, une technique de stockage de type *filtre de Kalman* (distribution normale) apparaît comme un bon compromis entre la simplicité des calculs avec une distribution normale, et le fort taux de représentativité, par un vecteur d’état adéquat, des combinaisons possibles et réalisables, de l’état représentant la carte et les véhicules. On pourra modérer ces propos par le fait que, dans le cas de la prise en compte d’une hypothèse fautive au niveau de la carte, une erreur de localisation puisse se produire. C’est par exemple le cas lorsque plusieurs bâtiments offrent la même apparence et orientation, et que la méthode de localisation prend l’un pour l’autre. C’est le problème de l’**ambiguïté**, qui est un problème classique des méthodes de localisation.

On peut également analyser ces différentes méthodes selon leur caractère relatif aux méthodes d’alignement :

- alignement global, de type « map-matching »,
- alignement local, sur un seul élément.

Il est clair qu’une méthode d’alignement « globale » sera toujours plus robuste qu’une technique d’alignement *locale*, dans le cas où l’on dispose d’une **carte exacte**. Ces techniques sont généralement gourmandes en calcul, leur mise à l’échelle peut s’avérer problématiques (taille des données–temps de calcul) comparées à un *alignement local*. De plus, il est rare qu’une carte parfaite soit disponible en réalité. On peut donc imaginer séparer chacun des amers cartographiés, et ne s’aligner que sur un seul à chaque fois. Cette méthode, moins robuste que la précédente, permet néanmoins de pouvoir en théorie faire le tri entre les amers existants et ceux qui n’existent pas.

Une autre remarque peut également être faite. Dans le cas d’un alignement local, un seul élément faux dans la carte, ou bien une détection à faux d’un amer, peut faire échec au processus de localisation. En effet, dans une technique d’alignement local, toute erreur dans la chaîne des détections successives va induire une localisation erronée ultérieure. Dans ce cas, seule une modalité de localisation basée sur un capteur absolu (de type GPS par exemple) serait susceptible de récupérer d’une telle erreur.

On voit donc ici se profiler l’idée d’une solution de type *technique mono-hypothèse*, de type *Kalman*, pour la représentation d’état, notamment au niveau de la carte. Par contre, le choix entre une approche de localisation par *alignement local* ou *global* n’est pas encore évident.

3.2 Données contenues dans la carte et méthode de mise à jour

A la section 2.6, des techniques d'estimation, destinées principalement à la gestion d'une flotte de véhicules sont présentées. Nous reprenons à notre compte ces techniques dans le cadre d'une carte contenant des amers fixes, puisqu'on peut voir un amer de la carte comme « un véhicule qui ne se déplace **jamais** ». Un tel formalisme est nécessaire puisque, afin de pouvoir corriger partiellement la position d'un élément de la carte, celle-ci doit pouvoir être modifiée. Parmi les approches que nous avons mentionnées, se trouvent :

- les approches de type *filtre de Kalman centralisé*,
- la fusion de cartes locales,
- le *filtre à Intersection de Covariance*,
- les approches *Maximum-a-Posteriori*,
- le filtre *Channel filter*.

Les approches de type *filtre de Kalman centralisé* sont, en **théorie**, nonobstant les problèmes de linéarisation, les meilleures solutions possibles pour une carte sémantique gérée par des distributions normales. Malgré cela, plusieurs problèmes existent pour ce type d'approches :

- les problèmes de mise à l'échelle, que nous allons expliquer,
- la faille de sécurité que constitue la défaillance du nœud de fusion.

Tout d'abord, regardons les problèmes de mise à l'échelle. Si l'on considère un nombre N d'amers, la taille de la matrice de covariance associée croît comme $O(N^2)$. Outre le problème de stockage que cela pose, la mise à jour d'une telle matrice, indispensable pour les calculs du filtre de Kalman, est très coûteuse en temps de calcul ($O(N^3)$).

D'autre part, pour un système de transport de personnes, qui doit donc être sécuritaire, la défaillance d'un seul nœud ne doit pas mettre en péril le reste du système. Or, si un unique élément gère la carte, sa simple défaillance peut mettre le système en panne. Ceci est inadmissible dans un tel environnement.

De ce fait, d'autres solutions doivent être trouvées. L'une d'elles, la fusion de cartes locales apparaît attractive dans le cadre d'un travail de plusieurs véhicules en coopération, mais la carte elle-même reste monolithique, avec les inconvénients précédemment mentionnés.

Les approches de type MAP sont généralement gourmandes en temps de calcul, ce qui est limitant pour des applications de type temps-réel. De ce fait, nous ne les sélectionnerons pas.

Les filtres du type *Channel filter* sont des filtres décentralisés, qui opèrent généralement dans une topologie de *mise à jour en réseau* figée. On choisit généralement celle-ci de type **arbre**, afin d'éviter les cycles dans le processus de mise à jour. Il est l'équivalent d'un filtre de Kalman décentralisé.

Les approches de type *Intersection de Covariance* gèrent les états des différents éléments d'une carte (ou bien d'une flotte) comme s'ils étaient indépendants, bien qu'en réalité ce ne soit pas le cas. Une solution de type *filtre de Kalman*, dans laquelle on traiterait chaque élément comme indépendant par rapport aux autres, donnerait lieu au phénomène de **sur-convergence** décrit succinctement à la section 2.6.4, et plus en détail aux sections 4.7.3 et 4.7.4. Un filtre à *Intersection de Covariance* est démontré immune aux problèmes de sur-convergence. Il permet de gérer chaque élément comme indépendant des autres, et le coût de la mise à jour est borné dans ce cas (il n'augmente pas avec le nombre d'amers cartographiés). Le coût de stockage est linéaire au lieu d'être quadratique

comme dans le cas du filtre de Kalman. Par contre, une mise à jour avec ce type de filtre sera toujours moins efficace en terme d'apport d'information que celle liée à un filtre de Kalman centralisé (qui est « optimale »).

Les deux solutions qui ressortent de cette analyse sont le *Channel filter* et le *filtre à Intersection de Covariance*. Nous avons fait le choix du *filtre à Intersection de Covariance* en raison de sa gestion des amers. Le système peut être mis à l'échelle aisément avec un tel filtre, tant d'un point de vue stockage que du point de vue temps de calcul. Toute mise à jour n'utilise que les états des éléments qu'elle implique (**un** mur, **un** véhicule), et non **toute** la carte, comme avec un filtre de Kalman. Il est immune au phénomène de **sur-convergence**. Le *Channel filter* nous semble apporter un bénéfice dans le cas d'une flotte de véhicules, dans laquelle chaque élément (mobile) est équipé d'une capacité de calcul. Mais la transposition d'une carte d'amers nous a semblé moins bien se prêter à ce type de méthode, car la gestion des amers devra se faire sur une unité de calcul, obligatoirement embarquée sur un véhicule, ou centralisée au sol. Cette argumentation n'est pas définitive, nous jugeons notre choix réaliste et valable dans l'état actuel de nos connaissances à l'issue de ces travaux.

3.3 Méthode choisie et développée

Nous rappelons ici les trois objectifs de nos travaux :

1. corriger la position des amers qui seraient mal positionnés dans la carte, tout en rendant possible la localisation d'un véhicule dans une telle carte,
2. supprimer des amers de la carte qui seraient impossibles à détecter, qui par exemple n'existeraient pas,
3. pouvoir ajouter des amers détectables et non encore insérés dans la carte de l'environnement.

Le choix d'une solution de gestion de la carte des amers, du type *Intersection de Covariance* comme explicité à la section 3.2, nous apparaît permettre la mise à jour de celle-ci en **garantissant l'absence de phénomène de sur-convergence**. Ce formalisme est, de plus, compatible avec un formalisme de type gaussien afin de décrire les états du/des véhicules(s) et des amers. Ce type de densité de probabilités est le même que celui utilisé avec une approche par filtre de *Kalman*, et permet donc de prendre en compte une carte de type *sémantique*. Une carte de type OPENSTREETMAP est typiquement utilisable par ce type d'approche, dans le cadre de l'objectif n°1.

Dans les approches de localisation décrites à la section 3.1, il est possible, à partir d'un formalisme de type *filtre de Kalman*, d'opter soit pour une approche de localisation par alignement *local* ou *global*. Le formalisme de type *alignement local* nous apparaît apporter une plus-value, dans le sens où l'on peut identifier un élément de manière séparée des autres. On pourrait alors ajouter à l'état de cet amer, indépendamment des autres, outre le vecteur d'état continu, une probabilité d'existence. Le formalisme de type *Intersection de Covariance* permet, comme nous l'avons vu, de gérer, indépendamment des autres, l'état d'un amer de la carte. Lui adjoindre une probabilité d'existence permettrait de compléter le modèle par une probabilité, qui indiquerait la vraisemblance de l'existence de l'amer en question. Cette probabilité serait confirmée ou non par des détections de celui-ci par les capteurs du véhicule. En dessous d'un seuil donné, l'amer serait effacé de la carte de l'environnement, ce qui correspondrait à notre objectif n°2.

De ce fait, nous optons pour une stratégie de localisation de type *local*, avec un *a-priori* pour l'état précédent du véhicule, ainsi que pour les amers de la carte. Ce type de méthode, légère en termes de modèle, et rapide (généralement temps-réel) nous semble convenir particulièrement bien au problème à résoudre. Une estimation de type bayésienne au niveau de la probabilité d'existence de l'amer à détecter, dont le choix est réalisé de manière périodique, serait alors possible. La mise à jour simultanée de l'état continu du véhicule et de l'amer qu'il observe se ferait de manière cohérente. Elle permettrait de pouvoir affiner l'état de l'un ou de l'autre en fonction de celui des deux disposant de la meilleure précision sur sa localisation.

Mais même avec un tel formalisme, l'utilisation d'une technique d'alignement des données capteur sur la carte de l'environnement reste possible. Malgré un temps de calcul élevé prévisible (de l'ordre de la seconde), cette technique reste intéressante. Elle peut par exemple être lancée en « tâche de fond » sans nuire au processus de localisation, puisque l'enrichissement de la carte n'est pas prioritaire par rapport au processus de localisation du véhicule (temps-réel). Un détecteur sans *a-priori* opérant sur les données provenant du capteur pourrait alors identifier de nouveaux amers. Par exemple, un détecteur de lignes droites, afin de détecter des murs dans des données LIDAR planes, serait en mesure d'identifier des murs réels. La procédure d'*alignement global* permettrait alors d'identifier les murs détectés avec leur contrepartie dans la carte. Ceux qui ne seraient pas associés à la carte auraient alors le statut d'« amer potentiel », conformément à notre objectif n°3. La potentialité de l'existence de tels amers serait alors soumise à une confirmation provenant des détections ultérieures.

Nous tenons ici à préciser un aspect particulier de notre approche. Nous proposons l'utilisation de cartes sémantiques, en modélisant l'état de chaque amer par une information sur son type (mur, poteau ou autre), et une distribution de probabilité vectorielle-gaussienne. Une procédure d'*alignement global* utilisant des types quelconques d'amers est dans notre cas parfaitement envisageable. Dans la littérature scientifique, on note que, généralement, les stratégies d'alignement sur une carte sont tributaires des capteurs et formalismes utilisés pour décrire le monde. Nous tenons à faire remarquer que notre approche est plus ouverte, en ce sens qu'elle autorise des types d'amers variables à être gérés de manière identique. Cette généralité se décline tant du point de la stratégie de localisation « par alignement *local* » que de celui de l'ajout de nouveaux amers « par alignement *global* ».

Les choix que nous venons de présenter nous ont amené à développer la méthode décrite dans le présent mémoire. Notre approche, en se basant sur des travaux antérieurs, entre dans le cadre d'une méthode « générique », pouvant, en théorie utiliser des données de divers type de capteurs extéroceptifs (LIDAR plan, LIDAR 3D, caméra, caméra de profondeur, capteur GPS, magnétomètre...). Cette propriété multi-capteur est directement issue des travaux antérieurs sus-mentionnés, et l'auteur n'a pris aucune part directe à l'élaboration du formalisme en question. Cette généralité nous amène à aborder le thème de la localisation d'un point de vue général, et non *spécifique* de capteurs ou formalisations spécifiques (nuages de points, capteurs caméra etc...). Nous décrirons au chapitre 5 ce formalisme, avant de passer à la description de notre approche, qui en découle. Auparavant, nous allons présenter les principales propriétés de ces travaux antérieurs, sur lesquels nous basons le travail rapporté dans le présent manuscrit.

3.4 Propriétés des travaux antérieurs

Comme indiqué ci-dessus, nous reprenons des travaux antérieurs de l'équipe, décrits au chapitre 5, dans lesquels l'auteur n'a pris aucune part initialement, afin de baser notre travail comme un ajout à ces derniers. Ceux-ci consistent en une méthode de localisation d'un véhicule dans une carte supposée exacte et parfaitement connue. Les propriétés qui nous ont paru intéressantes sont les suivantes :

- un cadre probabiliste de type bayésien,
- une possibilité de fusionner des mesures provenant de diverses modalités capteur (LIDAR, potentiellement caméra) afin de mettre à jour la position estimée du véhicule,
- la sélection, pour la modalité capteur choisie, d'un amer (un mur) à observer spécifiquement,
- le choix de la modalité capteur et amer est mis à jour fréquemment en temps réel, de manière à ce que plusieurs amers différents puissent être sélectionnés successivement.

Comme, au lieu d'analyser la totalité des données capteurs, on ne cherche l'amer choisi qu'à la position à laquelle on est supposé le trouver, cette approche est appelée DESCENDANTE ou TOP-DOWN. Les avantages qu'une telle approche apportent sont une analyse temps réel, plus rapide qu'avec un alignement global de toute la trame sur la carte de l'environnement proche. Un inconvénient potentiel est la moindre robustesse de ce type d'approche.

La carte utilisée est de type sémantique, et contient des bi-points représentant des murs de la carte, dans un système de coordonnées terrestre absolu. Cette formalisation de la carte convient aux développements qui sont initialement envisagés, et présentés dans ce mémoire. Le choix de reprendre cette approche de localisation, et d'y apporter nos contributions a donc été naturel.

Conclusion

Nous avons brièvement dessiné les contours de notre méthode de développement, basée principalement sur l'évitement de la **sur-convergence**, tant du point de vue de l'existence des amers (probabilités discrètes) que leur état (densité de probabilité dans le domaine continu). Nous proposons, afin de prendre en compte ces deux informations, un formalisme permettant de gérer de manière générique des amers de type potentiellement illimité. Nous allons dans le chapitre suivant de ce mémoire nous attarder à l'analyse des solutions classiques d'estimation vis à vis de ces deux critères d'analyse. Ces techniques théoriques nous serviront ensuite lors de la déclinaison de nos contributions, qui feront l'objet des chapitres suivants.

Chapitre 4

Techniques Probabilistes et Théorie de l'Information

Dans le chapitre précédent, nous avons présenté brièvement nos options et choix de stratégies de localisation et de cartographie. Ceux-ci sont relatifs à des choix d'approches (formalisme mathématique) et de techniques de calcul (algorithmes utilisés). Dans ce chapitre, plusieurs techniques employées en robotique et basées sur le formalisme bayésien, qui correspondent aux différentes options choisies, sont présentées. La technique d'estimation de l'état d'un système par un vecteur de composantes continues appelée **Filtre de Kalman** est l'une des plus connues. Nous présenterons également le phénomène de sur-convergence, lié à la notion de consanguinité de données, ainsi que leurs implications. Nous présenterons également quelques-uns de ses dérivés, notamment le filtre à **Intersection de Covariance Partitionnée**. Nous évoquerons ensuite la thématique des réseaux bayésiens, qui permet de faire le même type de calculs que le filtre de Kalman, mais pour des variables à valeurs discrètes.

Dans ce chapitre, nous nous intéresserons également à la notion d'information contenue dans une loi de probabilité, qu'elle soit de type discrète ou continue. Nous définirons le terme générique « information » comme une notion de degré de liberté de la loi de probabilité en question. Après avoir défini numériquement cette notion, nous donnerons quelques exemples de variables liées à la théorie de l'information.

La notion d'*entropie* de la *théorie de l'information* nous intéressera particulièrement car elle définit une fonction monotone du contenu en terme de *connaissance* dans le sens de *certitude* des phénomènes probabilistes. Si l'on utilise l'*entropie de Shannon*, on a la certitude que, quel que soit le phénomène physique ou probabiliste à l'œuvre, cette fonctionnelle définira toujours une relation d'ordre stricte permettant de quantifier le « gain d'information » ou la « certitude » offerte par la prise en compte d'une source de données. Par exemple, cette fonctionnelle pourra permettre de sélectionner certaines données provenant d'un capteur choisi, afin d'économiser du temps de calcul tout en maintenant la cohérence du système.

Nous tenterons dans ce chapitre d'utiliser les concepts relatifs à la théorie de l'information nous permettant de comprendre simplement le fonctionnement des systèmes d'estimation des filtres de nature bayésienne ou similaire.

4.1 Introduction au formalisme Bayésien

4.1.1 Variable Aléatoire Normale

En robotique, toute grandeur, continue ou discrète, peut être modélisée comme une variable probabiliste. Dans ce cadre, l'état d'un système n'est pas décrit par une quantité infiniment précise, mais par une loi de probabilité.

Il est d'usage en robotique de décrire une ou plusieurs variables continues par une distribution **Normale**, dite aussi **Gaussienne** et écrite $\mathcal{N}(\mathbf{X}_0, \Sigma_{\mathbf{X}})$. Un exemple est donné dans l'équation (4.1), pour laquelle \mathbf{X} est la variable aléatoire vectorielle de taille n , \mathbf{X}_0 est l'espérance de la variable \mathbf{X} , et $\Sigma_{\mathbf{X}}$ sa covariance. Une notation abrégée, utilisant une densité de probabilité par unité de volume de l'espace du vecteur d'état est donnée dans l'équation (4.2).

$$dp(\mathbf{X}; \mathbf{X}_0, \Sigma_{\mathbf{X}}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{\mathbf{X}}|}} e^{-\frac{1}{2}(\mathbf{X}-\mathbf{X}_0)^\top \cdot \Sigma_{\mathbf{X}}^{-1} \cdot (\mathbf{X}-\mathbf{X}_0)} d|\mathbf{X}| \quad (4.1)$$

$$\begin{aligned} \underline{p}(\mathbf{X}; \mathbf{X}_0, \Sigma_{\mathbf{X}}) &= \frac{dp(\mathbf{X}; \mathbf{X}_0, \Sigma_{\mathbf{X}})}{d|\mathbf{X}|} \\ &= \frac{1}{\sqrt{(2\pi)^n |\Sigma_{\mathbf{X}}|}} e^{-\frac{1}{2}(\mathbf{X}-\mathbf{X}_0)^\top \cdot \Sigma_{\mathbf{X}}^{-1} \cdot (\mathbf{X}-\mathbf{X}_0)} \end{aligned} \quad (4.2)$$

Un exemple de loi normale en deux dimensions est présenté sur la figure 4.2. La définition générale des deux termes \mathbf{X}_0 et $\Sigma_{\mathbf{X}}$ pour une densité de probabilité quelconque peut être trouvée dans les équations (4.3) et (4.4), dans lesquelles le terme $\mathbb{E}(\dots)$ représente une espérance mathématique. La loi normale est la loi de probabilité sur un \mathbb{R} -Espace-Vectoriel la plus simple, et la moins gourmande en hypothèses, ayant ses moments d'ordre 1 et 2 fixés. Elle maximise également l'entropie, c'est à dire la méconnaissance, et reste donc compatible de toutes les autres distributions de probabilités plus contraintes.

$$\mathbf{X}_0 = \mathbb{E}(\mathbf{X}) \quad (4.3)$$

$$\Sigma_{\mathbf{X}} = \mathbb{E}\left((\mathbf{X} - \mathbf{X}_0) \cdot (\mathbf{X} - \mathbf{X}_0)^\top\right) \quad (4.4)$$

Un exemple de réalisation (de multiples tirages aléatoires) d'une telle variable peut être trouvé sur la figure 4.1.

4.1.2 Distance de Mahalanobis

On définit la **distance de Mahalanobis** comme la racine carrée du terme quadratique dans l'équation (4.1), conformément à l'équation (4.5). Dans celle-ci, la variable \mathbf{x} est un vecteur (possiblement une réalisation de $\mathcal{N}(\mathbf{X}_0, \Sigma_{\mathbf{X}})$).

$$d_M(\mathbf{X}; \mathbf{X}_0, \Sigma_{\mathbf{X}}) = \sqrt{(\mathbf{X} - \mathbf{X}_0)^\top \Sigma_{\mathbf{X}}^{-1} (\mathbf{X} - \mathbf{X}_0)} \quad (4.5)$$

Dans le cas où la matrice de covariance est la matrice identité, la distance de Mahalanobis est égale à la distance entre la moyenne \mathbf{X}_0 et la valeur testée \mathbf{X} . Lorsque $\Sigma_{\mathbf{X}}$ est quelconque (c'est à dire symétrique définie positive), les courbes de niveau où d_M a une certaine valeur sont représentées par des ellipsoïdes de même dimension que la variable

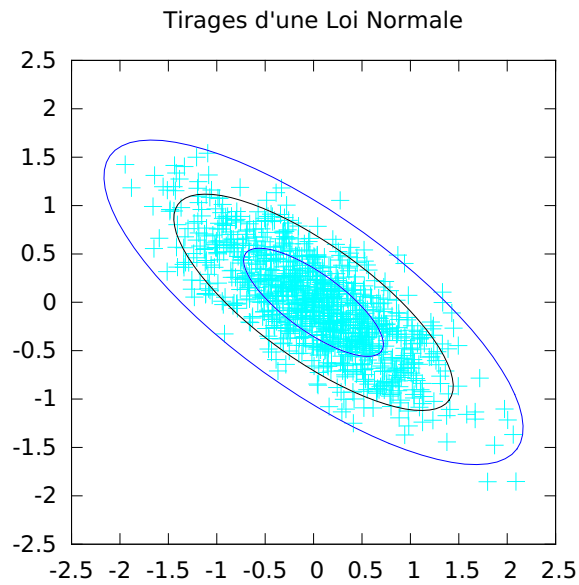


FIGURE 4.1 – Tirages d’une loi **Normale** bidimensionnelle. La distance de Mahalanobis égale à 1 est matérialisée par une ellipse noire (ellipse cyan intérieure à $\frac{1}{2}$, extérieure à $\frac{3}{2}$).

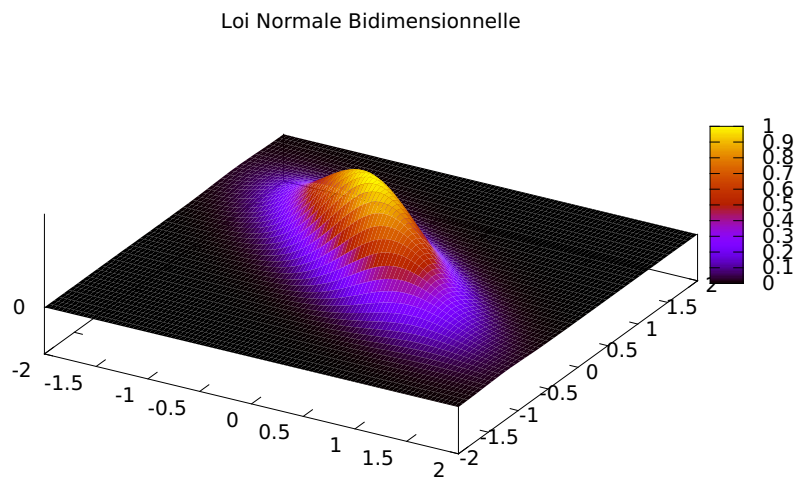


FIGURE 4.2 – Exemple de densité de probabilité associée à une distribution **Normale** bidimensionnelle.

aléatoire \mathbf{X} , comme on peut le voir d'après l'équation 4.6. La distance de Mahalanobis renseigne sur l'écart entre la valeur testée \mathbf{x} et le centre de la loi Normale \mathbf{X}_0 en termes de la densité de probabilité.

$$\underline{p}(\mathbf{X}; \mathbf{X}_0, \Sigma_{\mathbf{X}}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{\mathbf{X}}|}} e^{-\frac{1}{2} d_M(\mathbf{X}; \mathbf{X}_0, \Sigma_{\mathbf{X}})^2} \quad (4.6)$$

d_M donne la densité de probabilité de manière directe en fonction de \mathbf{X} en tout point de l'espace. Dans le cas particulier d'une variable scalaire x , on a la simplification donnée par les équations (4.7) et (4.8). La distance de Mahalanobis est une généralisation de l'équation (4.9) en plusieurs dimensions.

$$dp(x; x_0, \sigma_x) = \frac{1}{\sqrt{2\pi\sigma_x}} e^{-\frac{(x-x_0)^2}{2\sigma_x^2}} dx \quad (4.7)$$

$$\Sigma_x = \sigma_x^2 \quad (4.8)$$

$$d_M(x; x_0, \Sigma_x) = \frac{(x - x_0)}{\sigma_x} \quad (4.9)$$

$$= \sqrt{(x - x_0) \cdot \Sigma_x^{-1} \cdot (x - x_0)}$$

On voit par l'équation (4.9) que la distance de Mahalanobis est une **généralisation multidimensionnelle**, pondérée de manière **anisotrope** par $\Sigma_{\mathbf{X}}$, d'une distance **linéaire** et **scalaire** d'un vecteur de l'espace d'état à la moyenne de la distribution \mathbf{X}_0 .

4.1.3 Variable Aléatoire Binaire

Une variable discrète binaire, qui ne peut prendre que deux valeurs, 0 et 1, peut être décrite par leurs probabilités de réalisation (4.10).

$$p(A = a) = \begin{cases} p_A & \text{quand } a = 1 \\ \bar{p}_A & \text{quand } a = 0 \end{cases} \quad (4.10)$$

$$p_A + \bar{p}_A = 1 \quad (4.11)$$

Une interprétation classique dans le cas des systèmes « reproductibles » (**stochastiques**) est que la probabilité donne la fréquence des tirages lorsque l'on répète le processus. On peut prendre comme exemple le lancer d'une pièce de monnaie qui pourra retomber sur l'une quelconque de ses faces. La probabilité d'un tel événement (souvent modélisé par une probabilité de $\frac{1}{2}$ de retomber sur l'une quelconque de ses faces) ne dépend pas de l'histoire des tirages antérieurs ni du temps. La valeur $\frac{1}{2}$ représente « statistiquement » la fréquence des tirages, pris sur un « grand nombre » de tirages répétés.

4.1.4 Règle de Bayes

La règle de Bayes permet de relier deux probabilités conditionnelles conjuguées entre elles. Elle a des implications dans la règle de propagation des probabilités, qui est utilisée dans la théorie des réseaux bayésiens, continus ou discrets.

Définissons la probabilité conditionnelle $p(A|B)$, comme la probabilité d'occurrence de l'événement A dans le sous ensemble des réalisations où B est vrai. Cette probabilité est donnée par la formule (4.12).

$$p(A|B) = \frac{p(A, B)}{p(B)} \quad (4.12)$$

$$p(A, B) = p(A|B)p(B) \quad (4.13)$$

De l'équation (4.13) et de sa conjuguée permutant les événements A et B, on déduit (4.14).

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \quad (4.14)$$

On peut interpréter (4.14) comme le moyen d'obtenir $p(A|B)$ à partir de $p(A)$ et $p(B|A)$. Le facteur de normalisation $p(B)$ peut être calculé de manière indirecte, comme dans (4.15). Cette formule est à la base du calcul d'inférence et de marginalisation dans les réseaux bayésiens.

$$p(A = a|B = b) = \frac{p(B = b|A = a)p(A = a)}{\sum_{a', b' \in A \times B} p(B = b'|A = a')p(A = a')} \quad (4.15)$$

Prenons l'exemple d'une variable A que l'on tente d'estimer. Supposons que l'état de la variable B soit connu. Il s'agit d'une nouvelle « information », dont on peut vouloir tirer parti. La règle de Bayes permet, en présence de nouvelles données (observations, mesures...) de pouvoir, à partir d'un état antérieur, corriger ce dernier pour qu'il reflète mieux les éléments de mesure disponibles. On peut donc « ajouter de l'information » à une estimation *a-priori* à partir de données nouvelles grâce à la règle de Bayes.

4.2 Introduction à la Théorie de l'Information

Afin de comprendre en quoi les techniques de filtrage bayésiennes sont intéressantes, nous faisons ici une courte introduction à la théorie de l'information. Cette dernière s'inspire au moins partiellement de la notion d'entropie en thermodynamique. Depuis les travaux de Boltzmann et Gibbs, voir (Jaynes, 1965) pour une discussion des travaux originaux (XIX^{ème} siècle), on peut donner une interprétation statistique à cette variable d'état d'un système thermodynamique, qui représente habituellement le degré de « désordre » de sa population.

Prenons un exemple simple. Imaginons qu'une pièce de monnaie est lancée, et estimons l'information relative à la face sur laquelle la pièce est susceptible d'atterrir. Dans le cas habituel, les deux faces ont une probabilité égale d'être tirées. On a deux possibilités de résultat, que l'on pourrait coder en base binaire par 0 ou 1, et donc représenter par un espace de stockage d'un bit sur un ordinateur.

On peut synthétiser ainsi :

Cas d'une pièce avec une face **0** et une face **1** :

$$p(\mathbf{0}) = 0.5$$

$$p(\mathbf{1}) = 0.5$$

Information = 1 bit

Imaginons maintenant que la pièce ait deux faces identiques et indiscernables, dans ce cas, quelle que soit la face « physique » sur laquelle la pièce tombe, elle présente toujours le même symbole. Il n’y a donc pas de choix, et le résultat du lancer est toujours certain. Nous n’avons donc pas besoin de stocker le résultat de l’opération dans un emplacement mémoire d’un ordinateur. Cela donne :

Cas d’une pièce sans face **0** et avec deux faces **1** identiques de chaque côté :

$$\begin{aligned} p(\mathbf{0}) &= 0 && \text{Tirage impossible} \\ p(\mathbf{1}) &= 1 && \text{Tirage certain} \\ \text{Information} &= 0 \text{ bit} \end{aligned}$$

On observe que, lorsqu’on met en parallèle N tirages de la pièce ayant deux faces distinctes, le nombre de bits stocké augmente comme $O(N)$, mais les combinaisons de tirages possibles augmentent comme $O(2^N)$. Une première intuition dans le cas équiprobable donne alors :

$$\text{Information} = N \text{ bit} \tag{4.16}$$

$$= -\log_2 p \tag{4.17}$$

$$\text{avec } p = \frac{1}{2^N} \tag{4.18}$$

Les équations (4.16) à (4.18) correspondent à la définition de l’*entropie de Hartley*, qui correspond au cas des distributions uniformes. Dans le cas où l’on utilise la pièce avec deux faces identiques, l’information est toujours nulle ce qui traduit une certitude et le résultat est également certain (toutes les faces sont identiques).

4.3 Entropie de Shannon

4.3.1 Définition

En s’inspirant des lois de la thermodynamique statistique (notamment Gibbs), et en conservant les deux propriétés de départ, Claude Shannon établit dans (Shannon, 1948) (voir également Shannon et al., 1951) une théorie afin de calculer le degré de liberté d’une distribution ayant toutes les propriétés (convexité notamment, mais aussi extensivité au sens que lui donne la thermodynamique). La théorie de l’information qui en découle (voir par exemple MacKay, 2003) donne la valeur de l’entropie (4.19) dans laquelle p_i représente une loi de probabilité discrète.

$$H = -\sum_{i \in I} p_i \log p_i \tag{4.19}$$

$$\geq 0$$

$$0 \log 0 = 0 \quad \text{par convention}$$

$$H = \mathbb{E}_p(-\log(p)) \tag{4.20}$$

Cette quantité donne une valeur qui indique le « nombre de degrés de liberté » équivalent et comparable au cas équi-probable. Le chiffre donne alors une idée du degré de contrainte de la loi de probabilité. Cette quantité est toujours positive ou nulle, avec nullité si et seulement si l'une des probabilités p_i est égale à 1 et les autres nulles. Dans cette hypothèse, l'un des cas est certain.

H mesure donc une quantité relative à l'incertitude, ou à la méconnaissance du système, étant donné une loi de probabilités p_i donnée. Elle s'exprime dans diverses unités, en fonction de la base du logarithme utilisé. Pour une base 2, il s'agit de *bits*. Pour la base naturelle e , l'unité s'appelle *nats*. Dans le cas d'une base décimale, cette valeur s'exprime en *bans*. Une fois une base choisie, tous les calculs devront être réalisés avec la même base afin d'assurer la cohérence des calculs.

On peut prouver que cette fonctionnelle est invariante sous une transformation déterministe de la loi de probabilités à laquelle elle s'applique. Elle est également bornée au niveau supérieur par l'entropie de Hartley (4.16), qu'elle atteint pour une loi de probabilité uniforme. Une des interprétations possibles (voir Warren, 2015) consiste à dire que l'entropie, notée dans une base b , est le nombre de symboles en base b moyen (potentiellement un nombre réel) représentant le **nombre moyen d'essais-erreurs** pour estimer au hasard une variable aléatoire donnée.

4.3.2 Exemple

Donnons un exemple. Soit une loi de probabilité jointe pour deux variables X et Y :

$$\begin{aligned} p_{XY}(X = 0, Y = 0) &= 0 \\ p_{XY}(X = 0, Y = 1) &= 0.5 \\ p_{XY}(X = 1, Y = 0) &= 0.5 \\ p_{XY}(X = 1, Y = 1) &= 0 \end{aligned}$$

Dans cette loi de probabilité, $Y = \bar{X}$, et X est équi-probable. On sent intuitivement que cette loi ne contient qu'un seul bit d'information, ce qui est confirmé par le calcul (4.21).

$$\begin{aligned} H &= - 0 \log_2 0 \\ &\quad - 0.5 \log_2 0.5 \\ &\quad - 0.5 \log_2 0.5 \\ &\quad - 0 \log_2 0 \\ &= 1 \text{ bit} \end{aligned} \tag{4.21}$$

Si la loi de probabilité était équi-probable, l'entropie serait de 2 bits. Avec ce calcul de l'entropie, on capture donc bien le passage de 4 choix (2 bits) à 2 choix (1 bit).

4.4 Entropie Différentielle

Dans cette section, nous allons discuter le concept d'« entropie différentielle », qui est la généralisation au cas continu de l'entropie de Shannon.

4.4.1 Cas général

Dans le cas d'une variable continue, de densité de probabilité \underline{p} , le calcul est assez similaire, puisqu'il suffit de remplacer la sommation par une intégration. La formule générale est donnée dans l'équation (4.22).

$$H_{\text{differential}}(X) = - \int \underline{p}(x) \log(\underline{p}(x)) dx \quad (4.22)$$

À la différence de (4.19), cette fonctionnelle n'est ni positive, ni bornée. Elle peut tendre vers $-\infty$ ou $+\infty$, en fonction des paramètres de la densité de probabilités. Cette fonctionnelle est également sensible à des changements d'échelle sur la variable X , à la différence de (4.19), qui est invariante sous de telles transformations.

4.4.2 Analogie

On peut se rapprocher de l'entropie de *Shannon* en discrétisant les intervalles continus en la variable X sous lesquels l'intégration est réalisée. Supposons que, dans le cas d'une distribution monodimensionnelle, l'on subdivise l'ensemble \mathbb{R} en une série d'intervalles $[k \cdot \varepsilon, (k+1) \cdot \varepsilon]$. L'entropie de *Shannon* d'une telle loi de probabilité se calcule conformément à l'équation (4.25).

$$H_{\text{Shannon}} \approx - \sum_{k \in \mathbb{Z}} \varepsilon \cdot \underline{p}(\varepsilon \cdot k) \log(\varepsilon \cdot \underline{p}(\varepsilon \cdot k)) \quad (4.23)$$

$$\approx - \sum_{k \in \mathbb{Z}} \varepsilon \cdot \underline{p}(\varepsilon \cdot k) \log(\underline{p}(\varepsilon \cdot k)) - \sum_{k \in \mathbb{Z}} \varepsilon \cdot \underline{p}(\varepsilon \cdot k) \log(\varepsilon) \quad (4.24)$$

$$\approx H_{\text{differential}}(X) - \underbrace{\log(\varepsilon)}_{H_{\text{offset}}} \quad (4.25)$$

$$\geq 0 \quad (4.26)$$

On observe que lorsque $\varepsilon \rightarrow 0$, $H_{\text{offset}} \rightarrow +\infty$. Ceci est normal puisque le nombre de choix possibles et la dimensionnalité du problème tendent tous deux vers $+\infty$. Le terme $-\log(\varepsilon)$ fait en sorte que H_{Shannon} soit toujours positive. Par contre, on remarque que pour une valeur ε identique, l'écart entre les entropies de *Shannon* de deux distributions ne diverge pas (le terme H_{offset} est compensé). Cette différence tend vers la différence des entropies différentielles, cf. (6.32).

$$H_{\text{Shannon}}(Y) - H_{\text{Shannon}}(X) \xrightarrow{\varepsilon=0} H_{\text{differential}}(Y) - H_{\text{differential}}(X) \quad (4.27)$$

Cette entropie différentielle permet donc de comparer des valeurs d'entropie, même si elle ne jouit pas de la propriété de positivité. Ceci peut être appréciable afin de calculer un gain entre deux distributions de densité de probabilité.

4.4.3 Cas d'une distribution Normale

Dans le cas d'une distribution normale $\mathcal{N}(\mathbf{X}; \mathbf{X}_0, \Sigma_{\mathbf{X}})$ dans un espace vectoriel de dimension n , nous donnons le résultat présenté dans l'équation (4.28), et que l'on peut trouver dans (Ahmed and Gokhale, 1989, equation 1.1), sans le démontrer.

$$\begin{aligned} \underline{p}(\mathbf{X}; \mathbf{X}_0, \Sigma_{\mathbf{X}}) &= \frac{1}{\sqrt{(2\pi)^n |\Sigma_{\mathbf{X}}|}} e^{-\frac{1}{2}(\mathbf{X}-\mathbf{X}_0)^t \Sigma_{\mathbf{X}}^{-1}(\mathbf{X}-\mathbf{X}_0)} \\ H(\mathbf{X}) &= \frac{1}{2} \log(2\pi e |\Sigma_{\mathbf{X}}|) \end{aligned} \quad (4.28)$$

4.5 Divergence de Kullback-Leibler

Dans le domaine de la *théorie de l'information*, on peut vouloir comparer la qualité de l'approximation par une densité approchée \underline{q} , ou une loi de probabilité q_i , de respectivement une densité de probabilité supposée exacte \underline{p} , ou une loi de probabilité discrète p_i . La divergence de *Kullback-Leibler* permet une telle comparaison, dans les mêmes unités et avec la même signification qu'avec l'entropie.

La divergence de Kullback-Leibler est une mesure de dissimilarité entre deux lois de probabilité. Le fait que l'on n'appelle pas cette fonctionnelle une distance tient à ce qu'elle n'est pas symétrique.

4.5.1 Cas discret

Soit p une loi de probabilité de référence. Soit q une loi de probabilité tentant d'approcher p . On mesure la dissimilarité de q (loi test) par rapport à p (loi de référence) par l'équation (4.29).

$$\begin{aligned} D_{KL}(p||q) &= \sum_{i \in I} p_i \log \frac{p_i}{q_i} \\ &\neq D_{KL}(q||p) \\ &\geq 0 \end{aligned} \quad (4.29)$$

Une interprétation de cette valeur est appelée « surprise », qui se mesure en *bits*, *nats* ou en *bans*, comme une entropie. Elle représente l'écart dans le sens « erreur » ou « augmentation de méconnaissance » entre l'information réelle de la distribution exacte p , et le résultat que l'on pourrait obtenir avec la distribution approchée q . En effet, (4.31) est une autre écriture de cette divergence.

$$D_{KL}(p||q) = - \sum_{i \in I} p_i \log q_i - \left(- \sum_{i \in I} p_i \log p_i \right) \quad (4.30)$$

$$= \mathbb{E}_p(-\log q) - \underbrace{\mathbb{E}_p(-\log(p))}_{H(p)} \quad (4.31)$$

Cette valeur, qui, nous allons le voir, a un lien étroit avec l'entropie, permet de mesurer la correspondance d'une distribution q par rapport à une distribution p en termes chiffrés.

4.5.2 Cas continu

Les mêmes remarques qu'à la section 4.5.1 (positivité, contenu informationnel) sont valables pour le cas continu, puisque la divergence est l'écart entre deux termes d'entropie. On fournit la formulation dans (4.32).

$$D_{KL}(\underline{p}||\underline{q}) = \int \underline{p}(x) \log \left(\frac{\underline{p}(x)}{\underline{q}(x)} \right) dx \quad (4.32)$$

4.6 Information Mutuelle

Dans cette section, une variable, appelée « information mutuelle » donnant le degré de corrélation entre deux variables est définie. Nous présentons également quelques cas d'utilisation pratiques.

4.6.1 Entropie conditionnelle

On définit l'entropie conditionnelle de la variable X , sachant la variable Y comme indiqué dans les formules (4.33), (4.34) et (4.35) dans le cas discret. Elle représente la quantité d'information contenue en moyenne dans la variable X lorsque la valeur de la variable Y est connue (quelle qu'elle soit).

$$H(X|Y) = - \sum_{i \in I} p(X = x_i, Y = y_i) \cdot \log p(X = x_i | Y = y_i) \quad (4.33)$$

$$= - \sum_{i \in I} p(X = x_i, Y = y_i) \cdot \log \frac{p(X = x_i, Y = y_i)}{p(Y = y_i)} \quad (4.34)$$

$$= \mathbb{E}_{p_{x,y}} (\log p(x|y)) \quad (4.35)$$

4.6.2 Information mutuelle de deux variables

On définit l'information mutuelle I entre les variables aléatoires X et Y selon (4.36), (4.37) et (4.38). Cette grandeur représente la quantité d'information que l'on peut obtenir en moyenne sur une grandeur X connaissant la valeur d'une autre variable Y .

$$I(X : Y) = H(X) + H(Y) - H(X, Y) \quad (4.36)$$

$$= \underbrace{H(X)}_{\text{méconnaissance sur } X} - \underbrace{H(X|Y)}_{\text{méconnaissance sur } X \text{ sachant } Y} \quad (4.37)$$

$$= H(Y) - H(Y|X) \quad (4.38)$$

$$\geq 0 \quad \text{avec égalité si } X \perp Y$$

Lorsque X est inconnue, sa méconnaissance est égale à $H(X)$. Supposons que la valeur de la variable Y soit connue, dans ce cas, la méconnaissance n'est plus que de $H(X|Y)$. Ceci représente un *gain en information* correspondant à $I(X : Y)$, qui est l'*information* apportée à la variable X par la connaissance de la variable Y .

$I(X : Y)$ représente selon (4.37) l'information que la connaissance de Y donne sur X , avec son pendant symétrique (4.38). L'équation (4.36) indique que $H(X, Y)$ est inférieure à la somme $H(X) + H(Y)$ (entropie jointe de X et Y si indépendantes), et que la différence entre ces deux valeurs s'interprète comme le degré de corrélation entre X et Y .

On montre (4.39).

$$I(X : Y) = D_{KL}(p(X, Y) || p(X) \cdot p(Y)) \quad (4.39)$$

On en déduit donc que l'*information mutuelle* $I(X : Y)$ représente le *gain* en information apporté par la prise en compte des vraies corrélations entre X et Y (par le terme $p(X, Y)$) par rapport à une situation dans laquelle on ne les prends pas en compte (terme $p(X) \cdot p(Y)$). En effet, pour la loi de probabilités $p(X') \cdot p(Y')$, X' et Y' sont indépendantes, ce qui est noté $X' \perp Y'$. Dans ce cas, $H(X') = H(X'|Y')$.

4.6.3 Impact de la connaissance d'une variable sur une autre

Le principe du filtrage bayésien (les filtres de Kalman, les réseaux markoviens et bayésiens par exemple) est de corriger une estimation d'une variable X par la connaissance d'une variable observable Y , ainsi que de la relation entre la variable X sachant la variable Y et donnée par la probabilité $p(x|y)$, une donnée *a-priori* du modèle.

Lorsque une valeur de Y est donnée, quelle que soit cette valeur, en moyenne l'entropie de X passe de $H(X)$ à $H(X|Y)$. On sait que :

$$\begin{aligned} I(X : Y) &= H(X) - H(X|Y) \geq 0 \\ \iff H(X) &\geq H(X|Y) \end{aligned} \tag{4.40}$$

On constate dans (4.40) que la méconnaissance de X *a-priori* est supérieure à celle de X quand Y est connue. La diminution de méconnaissance de X , c'est à dire l'augmentation de connaissance sur X du fait d' Y est égale à l'information mutuelle $I(X : Y)$ entre ces deux variables, et sa valeur dépend de la probabilité conditionnelle $p(X|Y)$. Plus Y est corrélée à X , meilleure est l'amélioration donnée par la prise en compte de la valeur mesurée de la variable Y sur X .

On pourrait imaginer sélectionner une variable Y_i dans un ensemble de variables mesurables Y_k en choisissant celle qui apporte le plus d'information à la variable X que l'on souhaite estimer.

4.6.4 Inégalité relative au traitement de l'information

Supposons que nous analysions un processus causal $X \rightarrow Y \rightarrow Z$. Nous avons dans ce cas l'inégalité (4.41).

$$I(X : Y) \geq I(X : Z) \tag{4.41}$$

Supposons que l'on veuille analyser un système $X_k \rightarrow X_{k+1} \rightarrow X_{k+2} \dots$. Nous avons donc $I(X_k : X_{k+1}) \geq I(X_k : X_{k+2})$. Sachant X_k , plus n augmente, plus X_{k+n} sera décorrélié de X_k . Se baser exclusivement sur un état X_k trop vieux donnera une estimation de X_{k+n} plus imprécise. Ceci a pour conséquence qu'au fur et à mesure de l'évolution d'un système probabiliste dynamique, la corrélation entre un état antérieur et un état courant diminue.

Dans ce cas, on peut donc imaginer ajouter de l'information à la variable X_{k+n} en utilisant une mesure Y permettant de faire décroître à nouveau l'entropie, c'est à dire la méconnaissance sur X_{k+n} . C'est le principe du filtrage bayésien, qui est appliqué dans le filtre de Kalman que nous introduirons par la suite. Ce principe peut également être appliqué dans des réseaux récurrents à variables discrètes.

4.6.5 Exemples

Soient X et Y deux variables aléatoires. Si elles étaient indépendantes et équiprobables, l'information mutuelle serait 0. Prenons comme loi de probabilité pour X et Y celle indiquée à la section 4.3.2. On a :

$$\begin{aligned}H(X) &= 1 \text{ bit} \\H(Y) &= 1 \text{ bit} \\H(X, Y) &= 1 \text{ bit selon (4.21)} \\H(X|Y) &= H(X, Y) - H(Y) \\&= 1 \text{ bit} - 1 \text{ bit} \\&= 0 \text{ bit} \\I(X : Y) &= 1 \text{ bit} + 1 \text{ bit} - 1 \text{ bit} \\&= 1 \text{ bit}\end{aligned}$$

On a vu que $Y = \overline{X}$, et que X a une entropie de 1 bit. On calcule que, sachant la valeur de la variable Y , l'entropie conditionnelle $H(X|Y)$ est nulle, ce qui signifie qu'alors, la valeur de la variable X est parfaitement connue. Le gain d'*information* sur la variable X apporté par la parfaite connaissance de la variable Y est donc de 1 bit.

Il apparaît donc logique que, comme Y est totalement corrélée à X , l'information mutuelle de X et Y soit également de 1 bit.

4.6.6 Généralisations

La généralisation de la notion d'*information mutuelle* est assez complexe, et jusqu'à présent, aucune approche n'existe et ne possède les propriétés que l'on espère intuitivement. Une source discutant ce sujet de manière empirique, quoi qu'avec une certaine logique, peut être trouvée dans (Timme et al., 2011).

Afin d'illustrer ce qui vient d'être dit, nous allons supposer l'existence de trois variables aléatoires potentiellement corrélées X , Y et Z , et comparer à la situation de deux variables T et W . Dans le cas de deux variables T et W , nous allons comptabiliser les termes d'interaction existant et permettant d'identifier de manière unique la loi de probabilité $p(T, W)$, qui est :

1. T seul,
2. W seul,
3. $T \leftrightarrow W$, terme de corrélation duale .

Le terme $I(T : W)$ est une fonction dépendant du terme 3, de manière à ce que $I(T : W) = 0$ lorsque le terme 3 est nul.

Dans le cas de trois variables X , Y et Z , nous avons tous ces termes :

1. X seul,
2. Y seul,
3. Z seul,
4. $X \leftrightarrow Y$, terme de corrélation duale ,
5. $X \leftrightarrow Z$, terme de corrélation duale ,

6. $Y \leftrightarrow Z$, terme de corrélation duale ,
7. $X \leftrightarrow Y \leftrightarrow Z \leftrightarrow X$, terme de corrélation triple .

Dans ce cas, il y a, non pas un unique terme commandant la corrélation entre les variables, mais quatre (les termes 4, 5, 6 et 7). Si l'on ne considère que les variables deux à deux, on a les correspondances des équations (4.42), (4.43) et (4.44).

$$I(X : Y) \rightarrow \text{termes 4 et 7} \quad (4.42)$$

$$I(X : Z) \rightarrow \text{termes 5 et 7} \quad (4.43)$$

$$I(Y : Z) \rightarrow \text{termes 6 et 7} \quad (4.44)$$

Le terme 7 indique une corrélation multiple entre les trois variables, de type conjonctive par exemple (par exemple, celle que l'on obtiendrait par une porte AND(X, Y, Z)).

Si on choisissait pour indiquer la corrélation de trois variables le terme 7, on trouverait un concept proche de la variable notée TC (dans [Timme et al., 2011](#), section C). Dans le cas où les interactions entre deux variables parmi les trois existent et que l'on souhaite les quantifier, plusieurs concepts pourraient correspondre. Il existe de ce fait un nombre conséquent de fonctions dont le but est de mesurer les corrélations par rapport à différents mélanges des coefficients 4, 5, 6 et 7, et leurs généralisations à un nombre quelconque de variables. Voir pour cela la liste des fonctionnelles définies dans ([Timme et al., 2011](#), section III).

Mais dans toutes les interprétations connues, aucune ne conserve simultanément les propriétés de *positivité* et de *majoration absolue* que possède $I(X : Y)$. Certaines généralisations peuvent être négatives dans certains cas. D'autres peuvent ne pas être bornées de manière intuitive ou compréhensibles simplement. Dans tous les cas, aucune règle de chaînage n'existe, ce qui rend les calculs encore moins intuitifs.

Une tentative pour donner un cadre plus rationnel et intuitif à la notion de corrélation entre un nombre quelconque de variables est présentée dans ([Amari, 2001](#)). Cette méthode permet de voir l'information mutuelle comme une hiérarchie de dépendances de plus en plus corrélées, tout en la définissant grâce à une règle de chaînage par sommation.

On peut résumer la situation comme suit. La plupart des approches d'information mutuelle qui requièrent plus de deux variables, soit sont complexes ou difficiles à prouver et ne rendent compte que d'une partie de la corrélation, soit se ramènent à un chaînage d'informations mutuelles à deux termes.

4.6.7 Applications

L'*information mutuelle* est utilisée dans la plupart des cas comme une variable explicative permettant de détecter des relations de cause à effet, ou des effets d'une même cause, en termes quantifiables et ayant un sens. Une de ses limites est qu'elle n'indique pas nécessairement une relation de cause à effet.

On peut utiliser l'information mutuelle afin de détecter des corrélations significatives dans des jeux de données importants, par exemple dans des systèmes d'analyses comme dans ([Galas et al., 2014](#)) ou ([Nguyen et al., 2014](#)).

L'apprentissage bayésien est une autre application dans laquelle l'information mutuelle joue le rôle de variable à maximiser, indiquant la qualité de l'apprentissage obtenu. On pourra se référer à ([Mohamed and Rezende, 2015](#)) pour un exemple de cette application.

On utilise également l'information mutuelle pour l'alignement de données entre elles, par exemple des acquisitions bidimensionnelles provenant de moyens d'imagerie médicale entre elles, ou avec un modèle tri-dimensionnel dans les techniques les plus élaborées. Cette technique est robuste aux acquisitions obtenues par des moyens d'imagerie différentes, par exemple en comparant un cliché provenant de tomographie par rayons X avec un autre acquis par imagerie par résonance magnétique nucléaire. Quelques exemples de travaux dans cette branche sont (Besl et al., 1992) (Guyader et al., 2016) (Roche et al., 1998) (Roshni and Revathy, 2008) (Viola and Wells, 1995). Dans le domaine de la création de carte, on peut également citer (Mastin et al., 2009), dans lequel des données LIDAR et orthophotographies sont associées.

4.7 Estimation Probabiliste Continue

Dans cette section, nous allons détailler le processus d'estimation de variable continue nommé « Filtre de Kalman Étendu » (EKF), que par simplicité nous identifierons au **Filtre de Kalman**, auquel il se réduit dans le cas linéaire. Ce dernier, le **Filtre de Kalman** original (Kalman, 1960), développé dans un formalisme purement linéaire, est dû aux travaux de Rudolf Emil Kalman durant les décennies 1950 et 1960. Après avoir présenté le formalisme qui est lié aux **Filtre de Kalman** et EKF, nous décrirons brièvement certaines variantes de celui-ci, permettant d'aller au-delà de certaines limitations intrinsèques.

4.7.1 Modèle d'état

Dans tout système robotique, ou en théorie de l'automatique, une equation d'évolution est toujours à la base d'un estimateur ou d'un correcteur. Que ce soit dans le domaine discret ou continu, ce modèle permet de connaître une variable à un instant t en fonction de valeurs de la même variable aux instants précédents, ainsi que d'une variable de commande au même instant t .

Les deux équations (4.45) pour le domaine temporel discret et (4.46) pour le domaine temporel continu décrivent la formulation générale du problème. La variable notée \mathbf{U}_t , \mathbf{U}_k est appelée commande ou *mesure proprioceptive*, puisqu'elle n'informe que sur un *changement* de l'état du système. L'intégration de cette mesure du changement de l'état se nomme *prédiction*.

$$\mathbf{X}_{k+1} = \mathbf{F}(\mathbf{X}_k, \mathbf{U}_{k+1}) \quad (4.45)$$

$$\dot{\mathbf{X}}_t = \mathbf{f}(\mathbf{X}_t, \mathbf{U}_t) \quad (4.46)$$

Dans tout système réel, la commande \mathbf{U}_k , \mathbf{U}_t provient d'une mesure, ou d'une estimation, ce qui provoque, de par l'intégration permanente des erreurs, un processus de marche aléatoire provoquant à plus ou moins brève échéance la divergence du système. Afin de compenser cette dérive, il est nécessaire de réaliser une mesure d'une grandeur du système reliée avec la variable d'état \mathbf{X}_k , \mathbf{X}_t . Cette mesure, notée \mathbf{Y}_k , \mathbf{Y}_t informe sur l'état du système selon les équations (4.47) et (4.48), même si cette mesure est partielle. La prise en compte de cette mesure se nomme *mise à jour*.

$$\mathbf{Y}_k = \mathbf{H}(\mathbf{X}_k) \quad (4.47)$$

$$\mathbf{Y}_t = \mathbf{h}(\mathbf{X}_t) \quad (4.48)$$

4.7.2 Filtre de Kalman

Un filtre de **Kalman** (voir [Kalman, 1960](#)) est un algorithme constitué d'une séquence ininterrompue d'étapes *prédiction – mise à jour*. Les équations décrivant le fonctionnement de ce filtre (4.49), (4.50) sont basées d'une part sur l'intégration de l'équation (4.46), et sur l'inférence de la densité de probabilité $p(\mathbf{X}_t|\mathbf{Y}_t)$. On peut utiliser la règle de *Bayes* (4.14) afin de démontrer les formules (4.50) à (4.55). Les grandeurs sont décrites par une distribution **Normale** centrée sur \mathbf{X}_t ayant une covariance $\Sigma_t^{\mathbf{X}}$. Les suffixes $-$ et $+$ réfèrent aux valeurs *a-priori* et *a-posteriori* relatives à la variable à laquelle ils s'appliquent. t et t' sont des dates de mesure.

$$\mathbf{X}_{t'}^- = \mathbf{X}_t^+ + \int_t^{t'} \mathbf{f}(\mathbf{X}_s, \mathbf{U}_s) ds \quad (4.49)$$

$$\mathbf{X}_{t'}^+ = \mathbf{X}_{t'}^- + \mathbf{K}_{t'} \cdot \underbrace{(\mathbf{Y}_{t'} - \mathbf{h}(\mathbf{X}_{t'}^-))}_{\text{vecteur innovation}} \quad (4.50)$$

La matrice $\mathbf{K}_{t'}$ est appelée *gain de Kalman*, et est définie par l'équation (4.53). La matrice $\Sigma_{t'}^{\mathbf{Y}}$ représente la covariance du vecteur de mesure $\mathbf{Y}_{t'}$.

$$\mathbf{H}_{t'} = \left. \frac{d\mathbf{h}(\mathbf{X})}{d\mathbf{X}} \right|_{t'} \quad (4.51)$$

$$\mathbf{S}_{t'} = \mathbf{H}_{t'} \Sigma_{t'}^{\mathbf{X}^-} \mathbf{H}_{t'}^T + \Sigma_{t'}^{\mathbf{Y}} \quad (4.52)$$

$$\mathbf{K}_{t'} = \Sigma_{t'}^{\mathbf{X}^-} \cdot \mathbf{H}_{t'}^T \cdot \mathbf{S}_{t'}^{-1} \quad (4.53)$$

$$= \Sigma_{t'}^{\mathbf{X}^+} \cdot \mathbf{H}_{t'}^T \cdot \Sigma_{t'}^{\mathbf{Y}^{-1}} \quad (4.54)$$

$$\Sigma_{t'}^{\mathbf{X}^+} = (\mathbf{I} - \mathbf{K}_{t'} \mathbf{H}_{t'}) \cdot \Sigma_{t'}^{\mathbf{X}^-} \quad (4.55)$$

$$= \left\{ \begin{array}{l} (\mathbf{I} - \mathbf{K}_{t'} \mathbf{H}_{t'}) \cdot \Sigma_{t'}^{\mathbf{X}^-} \cdot (\mathbf{I} - \mathbf{K}_{t'} \mathbf{H}_{t'})^T \\ + \mathbf{K}_{t'} \cdot \Sigma_{t'}^{\mathbf{Y}} \cdot \mathbf{K}_{t'}^T \end{array} \right. \quad (4.56)$$

4.7.3 Filtres de Kalman Corrélés

Il existe des situations dans lesquelles une information unique pourra, par le jeu des mises à jour successives, être prise plusieurs fois en compte (et ce de manière non-bornée). Cet effet est à l'origine du phénomène dit « de rumeur » (rumor effect, voir [Farrell and Ganesh, 2009](#); [Ong et al., 2006b](#); [Rendas and Leitao, 2010](#)).

Afin d'illustrer cela, nous allons comparer deux systèmes basés sur le filtre de Kalman. Soient X et Y deux variables aléatoires scalaires, que l'on tente d'estimer. On dispose d'une mesure par la variable aléatoire Z qui mesure la somme $X + Y$ des deux variables précédentes. On peut utiliser un filtre de Kalman agissant sur un état constitué du vecteur $[X, Y]^T$. Nommons cette solution **Kalman Global** (en abrégé **KG**).

On peut également utiliser deux filtres de Kalman, l'un pour la variable X , en supposant la variable Y indépendante, et l'autre pour la variable Y , en supposant la variable

X à son tour indépendante. Basé sur une mesure Z , on peut donc utiliser cette dernière ainsi que la variable Y pour mettre à jour la variable X . Puis on peut réutiliser cette même variable Z avec la nouvelle valeur de X pour finalement mettre à jour Y . Nous nommerons cette solution **filtres de Kalman Corrélés (KC)**.

Notons $\Sigma_X/\Sigma_Y/\Sigma_Z$ respectivement les variances des variables X , Y et Z . Notons également Σ_{XY} la covariance croisée entre les variables X et Y . Notons avec un exposant respectivement $-/+$ les valeurs a-priori/a-posteriori pour toutes les valeurs (X et Y et leurs variances/covariances).

On note que les termes Σ_X , Σ_Y et Σ_{XY} sont reliés à la matrice $\Sigma_{\mathbf{XY}}$, représentant la covariance du vecteur $[X, Y]^T$, par l'équation (4.57).

$$\Sigma_{\mathbf{XY}} = \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{XY} & \Sigma_Y \end{pmatrix} \quad (4.57)$$

On trouve dans la table 4.1 les résultats des deux calculs, qui sont également présentés sous forme graphique sur la figure 4.3.

Variables	KG	KC
X^-	0.75	
Σ_X^-	1.00	
Y^-	0.50	
Σ_Y^-	1.00	
Σ_{XY}^-	0.00	
Z	1.00	
Σ_Z	0.01	
X^+	0.6256219	0.6256219
Σ_X^+	0.5024876	0.2499938
Y^+	0.3756219	0.3015863
Σ_Y^+	0.5024876	0.1637669
Σ_{XY}^+	-0.4975124	0.00

TABLE 4.1 – Calculs comparant un filtre de Kalman global **KG** à deux filtres de Kalman couplés **KC**.

On observe sur la figure 4.3 que les deux états mis à jour par les deux méthodes différentes ont des propriétés différentes. La mise à jour exacte crée une distribution dont la matrice de covariance est étirée sur l'axe $[1.0, -1.0]^T$, qui est l'axe sur lequel lorsque X et Y varient, la mesure Z reste identique. Utilisons ici la notion d'entropie différentielle. Nous observons qualitativement que la mesure Z n'apporte pas d'« information » dans cette direction. Afin de s'en assurer, on peut observer qu'après la mise à jour par **KC**, la valeur propre de l'ellipse de covariance jointe dans la direction $[1.0, -1.0]^T$ n'a pas changé, tandis que celle dans le sens $[1.0, 1.0]^T$ a, elle, diminué, par suite de la prise en compte de la mesure Z . Les axes propres des ellipses de covariances étant identiques, on peut donc comparer numériquement les valeurs propres de manière séparées. On peut donc écrire de manière schématique l'équation (4.58).

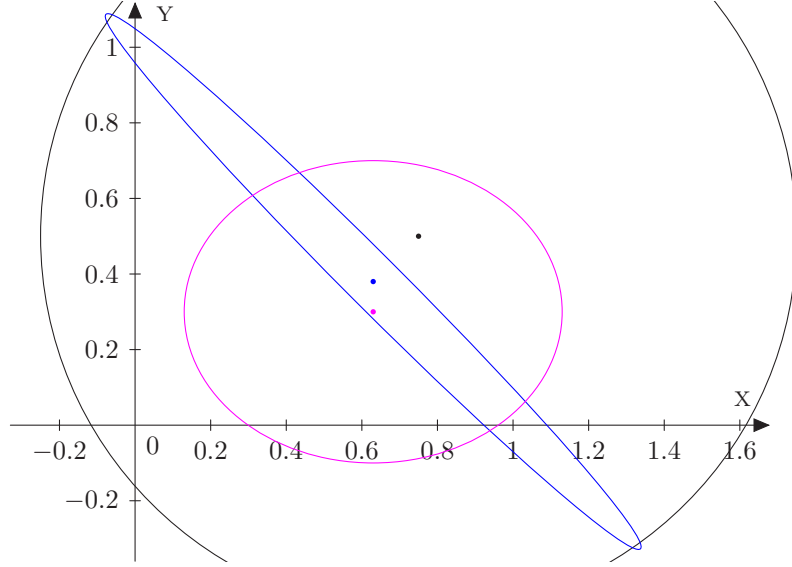


FIGURE 4.3 – En noir, l’estimation initiale des deux variables X et Y , en bleu, la mise à jour par utilisation du vecteur d’état complet (système **KG**), en magenta, la mise à jour en utilisant deux filtres de Kalman en interaction (système **KC**).

$$\begin{aligned}
 I(X, Y : Z) &= H(X, Y) - H(X, Y | Z) \\
 &= \frac{1}{2} \cdot \log (2\pi e |\Sigma_{\mathbf{XY}}^-|) - \frac{1}{2} \cdot \log (2\pi e |\Sigma_{\mathbf{XY}}^+|) \\
 &= \frac{1}{2} \cdot \log (2\pi e \sigma_{[1,-1]^\top}^{-2}) + \frac{1}{2} \cdot \log (2\pi e \sigma_{[1,1]^\top}^{-2}) \\
 &\quad - \frac{1}{2} \cdot \log (2\pi e \sigma_{[1,-1]^\top}^{-2}) - \frac{1}{2} \cdot \log (2\pi e \sigma_{[1,1]^\top}^{+2}) \\
 &= \log \sigma_{[1,1]^\top}^- - \log \sigma_{[1,1]^\top}^+ \\
 &= \log \frac{\sigma_{[1,1]^\top}^-}{\sigma_{[1,1]^\top}^+} \tag{4.58}
 \end{aligned}$$

La valeur $\sigma_{[1,1]^\top}^+$ correspond à la fusion de la mesure Z **purement dans cette direction** avec l’*a-priori* $\sigma_{[1,1]^\top}^-$. C’est donc bien la mesure Z qui améliore la connaissance du vecteur $[X, Y]^\top$. Le concept d’entropie nous permet donc de quantifier l’amélioration de connaissance sur les variables X et Y .

La mise à jour par filtres en interaction génère une distribution pour laquelle les valeurs sur X et Y sont indépendantes. On observe également que les deux dimensions de l’ellipse de covariance sont de dimension relativement similaire.

L’estimation pour le système **KC** est non-intègre. En effet, si l’estimation **KG** est la valeur exacte, on observe que, sur l’axe $[1.0, -1.0]^\top$ de l’ellipse de covariance de **KG**, celle de **KC** est plus resserrée. Ceci signifie que **KC** donne une valeur de bruit/incertitude dans cette direction plus faible que la valeur exacte. De plus, la valeur obtenue après fusion est significativement différente de la valeur exacte dans ce dernier cas.

Nous appelons ce phénomène sur-convergence. Il est provoqué par la prise en compte à deux reprises de l’information contenue dans la mesure Z , une fois pour la mise à jour

de X à partir de Z et d' Y , puis une seconde fois pour celle d' Y à partir de Z et X . Ce phénomène, s'il est reproduit un nombre conséquent de fois, mène à l'effet de rumeur, une amplification du moindre bruit jusqu'à ce que le système **KC** diverge.

La mise à jour exacte est, elle, intègre. L'information contenue dans la valeur Z sert à réduire la part de la matrice de covariance initiale dans le sens $[1.0, 1.0]^\top$. La part restante de la matrice de covariance reste inchangée. L'« information » de Z se traduit par une diminution de l'axe de la covariance dans la direction principale de Z , $[1.0, 1.0]^\top$. Nous verrons par la suite que « information » et covariance sont étroitement liées.

4.7.4 Pourquoi une fusion corrélée ne fonctionne-elle pas ?

Les raisons pour lesquelles une interaction entre plusieurs filtres de Kalman supposés indépendants, mais étant couplés en réalité, scénario décrit dans la section 4.7.3, ne fonctionne pas sont décrites dans (Julier and Uhlmann, 2007, section 3.1), notamment visible sur (Julier and Uhlmann, 2007, figure 2). Ce raisonnement, bien que parfaitement valable, est technique à décrire succinctement, et nous renvoyons le lecteur à (Julier and Uhlmann, 2007) pour plus de détails. Nous synthétiserons en disant que la suppression des termes de covariance croisée entre deux éléments de la carte fait diverger le système, et nous allons expliquer dans ce qui suit plus simplement pourquoi.

Nous allons ici reproduire un raisonnement dû à Ong *et al.* (Ong et al., 2006b). Sur la figure 4.4, on peut voir une schématisation de l'intersection de deux ensembles. Supposons que l'on tente d'estimer l'état X d'un avion à partir de deux radars au sol. L'ensemble A peut être celui des mesures (et valeurs déduites) venant d'une source donnée (le RADAR A), et l'ensemble B celui des valeurs venant d'une autre source B (le RADAR B). L'intersection $A \cap B$ est l'ensemble des valeurs découlant simultanément de A et B . L'union $A \cup B$ est l'estimation de l'état de l'avion en prenant toutes les informations disponibles en compte.

En prenant ce formalisme, on obtient le résultat de l'équation (4.59). Dans cette équation, dans l'équivalent filtre de Kalman, $\underline{p}(X|A)$ représente la covariance de A seul, et $\underline{p}(X|B)$ représente la covariance de B seul. Le terme $\underline{p}(X|A \cap B)$ représente quant à lui le terme de covariance croisée entre A et B qui permet d'estimer les corrélations entre les variables A et B .

$$\begin{aligned}
\underline{p}(X|A \cup B) &= \underline{p}(X|A \setminus B) \cdot \underline{p}(X|A \cap B) \cdot \underline{p}(X|B \setminus A) \\
&= \frac{\underline{p}(X|A \setminus B) \cdot \underline{p}(X|A \cap B) \cdot \underline{p}(X|B \setminus A) \cdot \underline{p}(X|A \cap B)}{\underline{p}(X|A \cap B)} \\
&= \frac{(\underline{p}(X|A \setminus B) \cdot \underline{p}(X|A \cap B)) \cdot (\underline{p}(X|B \setminus A) \cdot \underline{p}(X|A \cap B))}{\underline{p}(X|A \cap B)} \\
&= \frac{\underline{p}(X|A) \cdot \underline{p}(X|B)}{\underline{p}(X|A \cap B)} \tag{4.59}
\end{aligned}$$

Reprenons notre exemple de filtres de Kalman corrélés et référons nous à une présentation sous forme de diagramme des dépendances des calculs entre eux, conforme à la figure 4.5. Une mise à jour de $X \rightarrow X'$ a lieu basée sur la mesure Z et la valeur a-priori Y , puis une seconde mise à jour de $Y \rightarrow Y'$ est faite basée sur X' et Z . Dans la seconde mise à jour pour la variable Y' (celle de Y à partir de X' et de Z), on a :

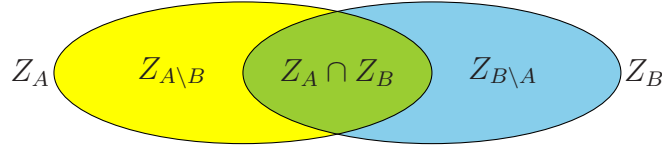


FIGURE 4.4 – Exemple d’intersection d’ensembles.

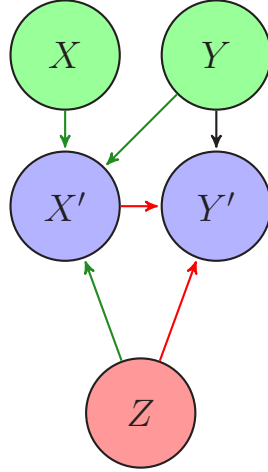


FIGURE 4.5 – Description de la mise à jour séparée de X et Y . L’information provenant de Z prise en compte de deux manières différentes (flèches rouges) conduisant à une mise à jour impropre.

$$\begin{aligned} \underline{p}(Y'|Z \cup X') &\propto \underline{p}(Y'|Z) \cdot \underline{p}(Y'|X') \\ &\propto \underline{p}(Y'|Z \setminus X') \cdot \underline{p}(Y'|X' \setminus Z) \cdot \underline{p}(Y'|X' \cap Z)^2 \end{aligned} \quad (4.60)$$

Dans le cas exact (le filtre de Kalman exact donnerait la même valeur), on aurait :

$$\begin{aligned} \underline{p}(Y'|Z \cup X') &= \frac{\underline{p}(Y'|Z) \cdot \underline{p}(Y'|X')}{\underline{p}(Y'|X' \cap Z)} \\ &= \underline{p}(Y'|Z \setminus X') \cdot \underline{p}(Y'|X' \setminus Z) \cdot \underline{p}(Y'|X' \cap Z) \end{aligned}$$

D’une manière informelle, l’exposant 2 sur le terme $p(Y'|X' \cap Z)$ de l’équation (4.60) indique que l’on prend en compte **deux fois** l’information commune entre X' et Z . On sait, puisque X' dépend de Z , que l’intersection $X' \cap Z$, qui représente une corrélation informationnelle, est non vide. On voit que l’information venant de Z à destination de Y' arrive par deux canaux, d’abord directement $Z \rightarrow Y'$, puis par le chemin $Z \rightarrow X' \rightarrow Y'$. Dans les filtres de Kalman séparés, utiliser l’équation (4.60) revient à supposer que le terme de covariance croisée entre X et Y est nul, alors qu’en réalité il ne l’est pas. Ce terme est alors comptabilisé dans les calculs effectifs deux fois. La mise à jour de l’équation (4.60) est donc impropre.

Une analyse simple, mais non dénuée d’intérêt, permet de comprendre pourquoi un exposant supérieur à 1 est problématique. Supposons que toutes nos densités de probabilité soient gaussiennes. Avec une mise à jour exacte, nous aurions :

$$\underline{p}(Y'|Z \cup X') = \mathcal{N}_{Y'|Z \setminus X'}(\dots, \Sigma_{Y'|Z \setminus X'}) \cdot \mathcal{N}_{Y'|X' \setminus Z}(\dots, \Sigma_{Y'|X' \setminus Z}) \cdot \mathcal{N}_{Y'|X' \cap Z}(\dots, \Sigma_{Y'|X' \cap Z})$$

tandis qu'avec une mise à jour sur-convergente, nous aurions :

$$\begin{aligned} \underline{p}(Y'|Z \cup X') &= \mathcal{N}_{Y'|Z \setminus X'}(\dots, \Sigma_{Y'|Z \setminus X'}) \cdot \mathcal{N}_{Y'|X' \setminus Z}(\dots, \Sigma_{Y'|X' \setminus Z}) \cdot \mathcal{N}_{Y'|X' \cap Z}(\dots, \Sigma_{Y'|X' \cap Z})^2 \\ &= \mathcal{N}_{Y'|Z \setminus X'}(\dots, \Sigma_{Y'|Z \setminus X'}) \cdot \mathcal{N}_{Y'|X' \setminus Z}(\dots, \Sigma_{Y'|X' \setminus Z}) \cdot \mathcal{N}_{Y'|X' \cap Z}(\dots, \frac{\Sigma_{Y'|X' \cap Z}}{2}) \end{aligned} \quad (4.61)$$

On observe que dans (4.61), une des distributions utilisées dans la fusion utilise une matrice de covariance de moitié plus petite que le vrai bruit associé à cette estimation. Ceci va forcément resserrer l'estimation postérieure par rapport à la valeur vraie. Ce phénomène est à la source après itérations de la sur-convergence. Au sens de l'entropie différentielle, le fait de diviser par deux la valeur de la matrice de covariance équivaut à diminuer l'entropie différentielle de la valeur (n étant la taille du vecteur \mathbf{Y}') :

$$\begin{aligned} H(\Sigma_{Y'|X' \cap Z}) - H\left(\frac{\Sigma_{Y'|X' \cap Z}}{2}\right) &= \frac{1}{2} \cdot \left(\log(2\pi e)^n - \log\left(\frac{2\pi e}{2}\right)^n \right) \\ &= \frac{n}{2} \log 2 \end{aligned}$$

D'un point de vue informationnel, estimer ainsi revient à retirer **artificiellement** de manière indûe une portion de la méconnaissance du système, et donc ajouter une **fausse information** en amplifiant artificiellement le poids du bruit par rapport au signal.

Une autre interprétation, en terme d'information mutuelle, consiste à dire qu'avec l'estimation donnée par (4.61), la partie commune $X' \cap Z$ aura une covariance deux fois plus petite que dans le cas intègre. On va alors **surestimer** l'information mutuelle entre X' et Z , c'est à dire que l'on ajoutera une pondération indûe à la corrélation entre X' et Z . Modifier l'information mutuelle correspond à augmenter artificiellement le **gain** du filtre, ce qui amènera à sa sur-convergence.

Plusieurs méthodes ont été publiées afin de résoudre ce problème :

- l'analyse des données en termes de graphe bayésien,
- l'analyse de *pedigree* des données,
- l'usage du « Channel Filter »,
- l'usage de « Covariance Union »,
- la fusion dite « de Chernoff »,
- les méthodes géométriques du type « Largest Ellipsoid »,
- les filtres « à Intersection de Covariance ».

Nous allons détailler dans ce qui suit les méthodes basées sur les filtres à Intersection de Covariance.

4.7.5 Filtre à Intersection de Covariance

Le filtre à Intersection de Covariance (Covariance Intersection filter, CI, voir [Julier and Uhlmann, 1997](#)) a pour vocation d'empêcher la sur-convergence observée précédemment à la section 4.7.3. Dans ce cas, seules les équations de mise à jour diffèrent par rapport

au filtre de Kalman. Soit \mathbf{X} une variable à estimer, et \mathbf{Z} une mesure relative à \mathbf{X} . On suppose que la mesure \mathbf{Z} possède une erreur d'estimation potentiellement corrélée à celle de la variable \mathbf{X} , mais que l'on ignore ce degré de corrélation. Ici, on supposera pour plus de simplicité que \mathbf{Z} est une mesure de \mathbf{X} et donc que les dimensions de \mathbf{X} et \mathbf{Z} sont identiques.

$$\begin{aligned}
\text{variable } \mathbf{X} & \begin{cases} \mathbf{X} & \text{valeur} \\ \Sigma_{\mathbf{X}} & \text{covariance} \end{cases} \\
\text{variable } \mathbf{Z} & \begin{cases} \mathbf{Z} & \text{valeur} \\ \Sigma_{\mathbf{Z}} & \text{covariance} \end{cases} \\
\text{Soit } \omega & \in [0, 1], \\
\Sigma_{\mathbf{X}}^{+-1}(\omega) & = \omega \cdot \Sigma_{\mathbf{X}}^{-1} + (1 - \omega) \cdot \Sigma_{\mathbf{Z}}^{-1} \\
\omega_b & = \arg \min_{\omega \in [0,1]} |\Sigma_{\mathbf{X}}^+(\omega)| \\
\Sigma_{\mathbf{X}}^+ & = \Sigma_{\mathbf{X}}^+(\omega_b) \\
\mathbf{X}^+ & = \Sigma_{\mathbf{X}}^+(\omega_b \cdot \Sigma_{\mathbf{X}}^{-1} \cdot \mathbf{X}^- + (1 - \omega_b) \cdot \Sigma_{\mathbf{Z}}^{-1} \cdot \mathbf{Z}) \quad (4.62)
\end{aligned}$$

Reprenons notre exemple (equation (4.60), figure 4.5). La mise à jour par intersection de covariance donnera le résultat de l'équation (4.63). On observe qu'aucune contribution n'a un exposant supérieur à 1 (ceci serait un indicateur fort de prise en compte multiple d'informations). Les exposants ω et $1 - \omega$, tous deux inférieurs à 1, indiquent une prise en compte moindre de l'information décorrélée provenant des termes $p(Y'|Z \setminus X')$ et $p(Y'|X' \setminus Z)$. Ce filtre n'est pas optimal dans le cas où l'information n'est pas maximale corrélée, mais il prévient efficacement la sur-convergence comme nous allons le mettre en évidence ci-dessous.

$$\begin{aligned}
\underline{p}^{\text{CI}}(Y'|Z \cup X') & \propto \underline{p}(Y'|Z)^\omega \cdot \underline{p}(Y'|X')^{1-\omega} \\
& \propto \underline{p}(Y'|Z \setminus X')^\omega \cdot \underline{p}(Y'|X' \setminus Z)^{1-\omega} \cdot \underline{p}(Y'|X' \cap Z)^{\omega+1-\omega} \\
& \propto \underline{p}(Y'|Z \setminus X')^\omega \cdot \underline{p}(Y'|X' \setminus Z)^{1-\omega} \cdot \underline{p}(Y'|X' \cap Z) \quad (4.63)
\end{aligned}$$

Supposons que nous disposons d'une estimation d'une variable X , donnée par \mathbf{X}^- et $\Sigma_{\mathbf{X}}^-$. Dans ce calcul, supposons qu'une autre donnée, la variable Z , recopie exactement en toutes circonstances la valeur de l'estimation de départ pour X . Dans un tel cas, il n'y a aucune information nouvelle, puisqu'on ne fait que copier les chiffres d'un résultat de mesure. La covariance et la valeur fusionnée entre X et Z ne doivent alors pas différer de l'estimation initiale. C'est ce que l'on observe dans le cas du filtre à intersection de covariance pour la valeur de la covariance avec l'équation (4.64). La valeur \mathbf{X}^+ ne diffère pas non plus de l'a-priori \mathbf{X}^- , voir l'équation (4.65).

$$\begin{aligned}
\Sigma_{\mathbf{X}}^{+-1}(\omega) & = \omega \Sigma_{\mathbf{X}}^{-1} + (1 - \omega) \Sigma_{\mathbf{Z}}^{-1} \\
& = \omega \Sigma_{\mathbf{X}}^{-1} + (1 - \omega) \Sigma_{\mathbf{X}}^{-1} \\
& = \Sigma_{\mathbf{X}}^{-1} \\
\Rightarrow \Sigma_{\mathbf{X}}^+ & = \Sigma_{\mathbf{X}}^- \quad (4.64)
\end{aligned}$$

$$\begin{aligned}\mathbf{X}^+ &= \Sigma_{\mathbf{X}}^-(\omega \cdot \Sigma_{\mathbf{X}}^{-1} \cdot \mathbf{X}^- + (1 - \omega) \cdot \Sigma_{\mathbf{X}}^{-1} \cdot \mathbf{X}^-) \\ &= \mathbf{X}^-\end{aligned}\tag{4.65}$$

Le filtre de Kalman suppose que les mesures sont totalement indépendantes de l'estimation de l'état du système. Comme \mathbf{X}^- est identique à elle-même, nous savons que la pseudo « mesure Z » $\mathbf{X}^-/\Sigma_{\mathbf{X}}^-$ est maximale corrélée avec l'état antérieur $\mathbf{X}^-/\Sigma_{\mathbf{X}}^-$. L'opération de fusion décrite précédemment est donc impropre pour un filtre de Kalman. Mais si le calcul était effectué, on obtiendrait le résultat de l'équation (4.66). On observe que même sans apport de nouvelle donnée, la covariance serait divisée de moitié.

$$\begin{aligned}\mathbf{S} &= \Sigma_{\mathbf{X}}^- + \Sigma_{\mathbf{Z}} \\ &= 2\Sigma_{\mathbf{X}}^- \\ \mathbf{K} &= \Sigma_{\mathbf{X}}^-(\Sigma_{\mathbf{X}}^- + \Sigma_{\mathbf{Z}})^{-1} \\ &= \Sigma_{\mathbf{X}}^-(\Sigma_{\mathbf{X}}^- + \Sigma_{\mathbf{X}}^-)^{-1} \\ &= \frac{1}{2}\mathbf{I} \\ \Sigma_{\mathbf{X}}^+ &= (\mathbf{I} - \mathbf{K})\Sigma_{\mathbf{X}}^- \\ &= \frac{1}{2}\Sigma_{\mathbf{X}}^-\end{aligned}\tag{4.66}$$

En terme d'information, le fait de diminuer artificiellement la covariance de moitié, sans utiliser la moindre information nouvelle, sans inférence avec une variable extérieure à \mathbf{X} , provoque une **diminution artificielle** de l'entropie du système, c'est à dire de la méconnaissance sur \mathbf{X} , d'une quantité égale à ($n = \dim(\mathbf{X})$) :

$$H(\Sigma_{\mathbf{X}}^-) - H\left(\frac{\Sigma_{\mathbf{X}}^-}{2}\right) = \frac{n}{2} \log 2$$

Le filtre de Kalman réduit de moitié la covariance de la variable \mathbf{X} alors qu'aucune nouvelle information n'a été acquise. Ceci est problématique dans le cas où les mesures sont des fonctions d'autres variables, ou d'états précédents de la même variable. Ces conditions sont remplies dans les systèmes en interaction, et donc par voie de conséquence dans des systèmes contenant plusieurs systèmes autonomes ne pouvant être simplement centralisés comme une flotte de véhicules.

Avec un filtre à intersection de covariance, on a vu que, contrairement à un filtre de Kalman, la covariance ne diminue pas, mais reste identique dans le cas où les deux informations sont maximale corrélées. L'entropie reste donc également inchangée. Cela est cohérent avec le fait que la prise en compte deux fois de la même information ne provoque aucune modification à l'estimation courante. Nous entendons dans la suite par le terme *conservatif* le caractère d'une loi de probabilités q qui, en remplaçant une loi exacte p , ne conduirait pas à une divergence du processus d'estimation. Dans ce cadre, on met en évidence le caractère *conservatif* de l'estimation retournée par le filtre CI.

Malheureusement, ceci vient avec un coût. En effet, si la valeur de la variable \mathbf{Z} est totalement décorrélée, le filtre CI n'a aucun moyen de le savoir, et la donnée \mathbf{Z} sera fusionnée comme si elle était maximale corrélée avec la variable \mathbf{X} . Or, cela n'est

pas le cas, et l'estimation obtenue sera fortement plus imprécise que celle obtenue par un filtre de Kalman. On peut donc ne pas gagner autant d'information avec un filtre CI que ce que donnerait un filtre de Kalman sur le vecteur d'état global. Il y a donc un manque à gagner en terme d'information, plus la mesure est décorrélée de l'état du système, plus le gain en terme d'information d'un filtre CI sera faible vis à vis du gain d'un filtre de Kalman sur le même système. Néanmoins, l'intégrité de l'estimation est conservée.

De plus, l'estimation postérieure est obtenue grâce à un processus d'optimisation sur une variable scalaire, le paramètre ω . Ce calcul permet d'obtenir la meilleure covariance possible en terme d'espace d'état. Dans une implémentation de type *unimodale*, par exemple par la **méthode du nombre d'or**, on peut obtenir en une dizaine d'itérations une précision pour le paramètre ω de l'ordre de 10^{-2} , ce qui suffit normalement à la plupart des implémentations.

Comme conclusion, un filtre de Kalman est un filtre *optimiste* car il suppose que la mesure utilisée pour mettre à jour l'état, et la variable d'état que l'on souhaite estimer sont totalement indépendantes l'une de l'autre au sens des probabilités. Un filtre à Intersection de Covariance va au contraire supposer que toutes ces valeurs sont corrélées comme si l'une était une fonction de l'autre, ce que l'on appelle *maximalement corrélées*, tendant à obtenir une estimation très conservatrice et plus imprécise. De ce fait, un filtre CI est un filtre *pessimiste*. Dans des situations réelles, le degré de corrélation entre deux variables est rarement 0% ou 100%, et afin de gérer de manière plus optimale la fusion de données partiellement corrélées, prendre en compte ce degré de corrélation améliorerait les performances.

4.7.6 Coût calculatoire d'un filtre à Intersection de Covariance

On peut légitimement se demander, au vu du chapitre précédent quel pourrait bien être l'intérêt d'un tel filtre, tout à la fois plus lent et moins précis. Nous allons faire un petit exercice de pensée et supposer que nous avons à gérer un ensemble de N éléments (des véhicules par exemple), avec $N \gg 1$. Soient leurs états les vecteurs $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$ tous de même taille. Supposons également que dans ce système, les mesures ne fassent intervenir qu'un seul, ou deux de ces éléments, mais jamais plus. Supposons également, pour avoir un ordre de grandeur simple à prendre en compte, que les mesures, sur un seul ou deux éléments, aient la même taille que chacun des éléments \mathbf{X}_i .

Une première approche (exacte) consisterait à utiliser un filtre de Kalman sur un état global $[\mathbf{X}_1^\top, \mathbf{X}_2^\top, \dots, \mathbf{X}_N^\top]$. Nous noterons cette approche **CT**. Une seconde approche consiste à estimer **séparément** chacun des éléments, de manière indépendante des autres. Ceci, nous venons de le voir, est impossible avec un filtre de Kalman séparé, mais tout à fait envisageable avec un filtre à intersection de covariance gérant chaque état \mathbf{X}_i de manière séparée. Appelons cette méthode **CI**.

La table 4.2 indique les principales grandeurs comparables entre les deux méthodes au niveau mémoire et temps de calcul. On observe que la quantité de mémoire prise pour stocker les vecteurs d'état des éléments est rigoureusement la même. Par contre, le stockage en terme de quantité de mémoire des coefficients de covariance est en faveur d'une implémentation de type **CI** avec une variation en $O(N)$ et non $O(N^2)$.

Pour les temps de calcul, deux choses sont principalement à prendre en compte : le processus d'optimisation et l'inversion des matrices de covariance pour la méthode **CI**, la mise à jour de la matrice de covariance pour le filtre de Kalman (par exemple selon la formule de *Joseph* (4.56)). Dans le cas général de matrices de taille $K \times K$, la durée du

calcul d'un produit croît en $O(K^3)$. L'optimisation pour une mise à jour est prise comme un facteur multiplicatif de 10 qui correspond à une précision de 10^{-2} sur le paramètre ω , en plus du doublement de ce temps pour prendre en compte une optimisation en deux étapes (une sur chacun des éléments considérés) pour la solution **CI**.

On suppose que la durée d'une mise à jour est principalement contrainte par des durées d'inversion des matrices de covariance pour le filtre à Intersection de covariance, et par la mise à jour par produits sur la matrice de covariance dans le cas du filtre de Kalman. Selon ce standard, le temps de calcul d'une étape de mise à jour avec la solution **CT** croît comme $O(N^3)$, tandis qu'il reste constant pour une solution de type **CI**.

De ce point de vue, la solution **CI** apparaît plus adéquate dans le cadre d'une mise à jour sur une collection de grande taille d'éléments contenant chacun relativement peu d'information.

	CT	CI
Nombre d'éléments (vecteurs)	$O(1)$	$O(N)$
Taille d'un élément	$O(N)$	$O(1)$
Taille de la covariance d'un élément	$O(N^2)$	$O(1)$
Taille cumulée des états	$O(N)$	$O(N)$
Taille cumulée des covariances	$O(N^2)$	$O(N)$
Durée d'une mise à jour élément – élément	$O(N^3)$	$O(20)$

TABLE 4.2 – Comparaison de la gestion d'une collection d'éléments distincts par un filtre de Kalman monolithique et par une collection de filtres à Intersection de covariance.

4.7.7 Filtre à Intersection de Covariance Partitionnée

Afin de pallier le problème d'estimation du degré de corrélation, le filtre à Intersection de Covariance Partitionnée (Split Covariance Intersection filter, SCI, voir [Julier and Uhlmann, 2001](#)) a été développé par les mêmes auteurs que le filtre CI. Dans le filtre de Kalman, une méthode de fusion est utilisée, la fusion de variables dont les erreurs d'estimation sont indépendantes. Dans le cas du filtre CI, une autre technique de fusion est utilisée. L'idée de base du filtre SCI est de trouver une formule combinant les deux, et qui, dans chaque cas particulier, se ramène aux deux précédents. On pourra se référer à l'annexe C, et notamment à la section C.3 pour une analyse relative à la conservativité de l'estimation.

Afin de combiner ces deux méthodes, on sépare la matrice de covariance $\Sigma_{\mathbf{X}}$ de la variable \mathbf{X} en deux parties, la partie $\Sigma_{\mathbf{X}_i}$ totalement indépendante du reste du monde (modélisant un bruit blanc totalement non-corrélé, ni avec lui-même, ni avec les erreurs d'estimation d'autres variables), et la partie $\Sigma_{\mathbf{X}_d}$ qui représente les corrélations potentielles ou avérées avec le reste de l'environnement.

$$\Sigma_{\mathbf{X}} = \Sigma_{\mathbf{X}_i} + \Sigma_{\mathbf{X}_d} \quad (4.67)$$

Les équations de mise à jour sont identiques aux équations du filtre de Kalman et du filtre CI, pour le cas où respectivement, $\Sigma_{\mathbf{X}} = \Sigma_{\mathbf{X}_i}$ et $\Sigma_{\mathbf{X}} = \Sigma_{\mathbf{X}_d}$.

$$\begin{aligned}
\omega &\in [0, 1], \\
\Sigma_{\mathbf{X}\omega}(\omega) &= \Sigma_{\mathbf{X}^-} + \frac{1}{\omega} \Sigma_{\mathbf{X}^- d} \\
\Sigma_{\mathbf{Z}\omega}(\omega) &= \Sigma_{\mathbf{X}i} + \frac{1}{1-\omega} \Sigma_{\mathbf{X}d} \\
\Sigma_{\mathbf{X}^+}(\omega)^{-1} &= \Sigma_{\mathbf{X}\omega}(\omega)^{-1} + \Sigma_{\mathbf{Z}\omega}(\omega)^{-1} \\
\omega_b &= \arg \min_{\omega \in [0,1]} |\Sigma_{\mathbf{X}^+}(\omega)| \\
\Sigma_{\mathbf{X}^+} &= \Sigma_{\mathbf{X}^+}(\omega_b) \\
\mathbf{X}^+ &= \Sigma_{\mathbf{X}^+}(\Sigma_{\mathbf{X}\omega}(\omega_b)^{-1} \mathbf{X}^- + \Sigma_{\mathbf{Z}\omega}(\omega_b)^{-1} \mathbf{Z})
\end{aligned}$$

$\Sigma_{\mathbf{X}i}$ représente la partie du modèle de bruit de la variable \mathbf{X} qui est indépendante de toute autre variable d'intérêt, et $\Sigma_{\mathbf{X}d}$ celle de ce qui dépend, de manière causale ascendante ou corrélée avec d'autres variables, ici la variable \mathbf{Z} . La réciproque est également vraie si l'on permute le rôle de \mathbf{X} et \mathbf{Z} .

Dans le cas du filtre SCI, comme les matrices du type $\Sigma_{\mathbf{X}i}$, $\Sigma_{\mathbf{Z}i}$ sont fusionnées avec une mise à jour de Kalman, on obtient toujours l'inégalité suivante :

$$\begin{aligned}
\Sigma_{\mathbf{X}^+}^{SCI}(\omega) &\leq \Sigma_{\mathbf{X}^+}^{CI}(\omega) \\
\Rightarrow H_{SCI}(\omega) &\leq H_{CI}(\omega) \\
\Rightarrow H_{SCI}(\omega_b^{SCI}) &\leq H_{CI}(\omega_b^{CI})
\end{aligned}$$

D'un point de vue informationnel, le filtre SCI est plus optimal que le filtre CI, tout en évitant la sur-convergence. Pour plus de détails, se référer à l'annexe C.

4.8 Estimation Probabiliste Discrète

Les réseaux bayésiens discrets sont généralement utilisés afin d'estimer des grandeurs probabilistes non-observables, en les déduisant de celles qui sont observables. Cette opération de calcul est réalisée grâce à un modèle de probabilités, donné par la structure du réseau et les valeurs de probabilités conditionnelles. Ce modèle probabiliste, qui est donné par hypothèse et découle d'un modèle physique, conjugue des variables observables (les « évidences ») et des variables inobservables dont on veut obtenir une loi de probabilités, même approximative. Ils sont aux variables discrètes ce que sont les filtres de Kalman et chaînes de Markov aux variables continues.

4.8.1 Formalisme des Réseaux Bayésiens Discrets

Les réseaux Bayésiens (voir [Lauritzen, 1996](#)) discrets utilisent comme le filtre de Kalman le formalisme de la section 4.1. Mais dans ce cas précis, le formalisme est plus général. En effet, des variables continues ou discrètes peuvent être insérées dans un tel réseau, tandis qu'un filtre de Kalman ne gère que des variables continues modélisées par des distributions **Normales** ayant des relations linéaires entre elles (même de manière approchée). Un exemple de travaux ayant trait à la généralisation du filtre de Kalman à de tels réseaux peut être trouvé dans ([Murphy, 2002](#)).

Dans le cas des réseaux à variables exclusivement discrètes, on le formalise comme un ensemble de probabilités conditionnelles représentant des relations de causalité. Les

réseaux markoviens sont un autre exemple de réseaux probabilistes, mais qui ne partagent pas la même représentation des lois de probabilité. Dans ceux-ci, les variables sont corrélées par des *potentiels* matérialisant des corrélations à plusieurs variables par le biais d'une représentation par une famille exponentielle. Dans les différents types de réseaux bayésiens, on a coutume de représenter les nœuds représentant des variables continues par des cercles. Les nœuds représentant des variables discrètes sont représentés par des carrés.

Dans ce qui suit, nous nous limiterons au cas des réseaux bayésiens à variables binaires, à partir duquel on pourra généraliser pour un nombre quelconque de valeurs discrètes. Un exemple de tel réseau est présenté sur la figure 4.6. Les valeurs des nœuds du réseau sont soit 1 pour *vrai*, soit 0 pour *faux*. Une table de probabilités conditionnelles lie les probabilités du nœud en question aux valeurs prises par tous ses nœuds parents. Les parents d'un nœud donné doivent être vus comme des causes à la valeur effective du nœud considéré. Un tel réseau est visible sur la figure 4.6.

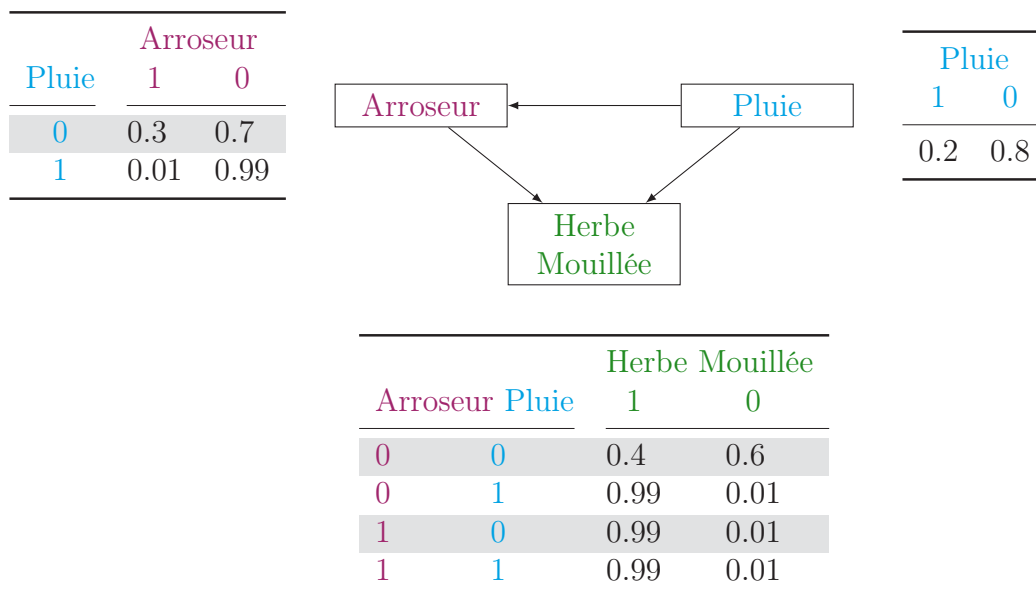


FIGURE 4.6 – Un exemple de réseau bayésien

Dans ces réseaux, plusieurs types principaux d'opérations sont possibles : le chaînage des probabilités conditionnelles, la marginalisation et l'inférence. Nous allons détailler ces opérations dans ce qui suit, en donnant d'abord un exemple illustratif, puis en donnant une expression du cas général.

4.8.2 Table de Probabilités Conditionnelles

Prenons l'exemple du nœud *Pluie* de la figure 4.6. Ce nœud n'a pas de parents, donc les probabilités qui y sont mentionnées ne sont pas conditionnées. Nous avons donc par exemple :

$$p(\textit{Pluie} = 1) = 0.2$$

Le nœud *Arroseur*, quant à lui, possède un parent, le nœud *Pluie*. Les valeurs dans la table du nœud *Arroseur* sont donc conditionnées à celles du nœud *Pluie*. On a également :

$$p(\text{Arroseur} = 0 | \text{Pluie} = 1) = 0.99$$

Dans tout réseau bayésien, chaque nœud dispose d'une table de probabilités, dite de probabilités conditionnelles, dans laquelle on trouve la probabilité de chacun des états du nœud (0 ou 1) en fonction de chaque configuration des parents directs du nœud en question.

4.8.3 Chaînage des Probabilités Conditionnelles

Selon l'équation (4.13), on a :

$$\begin{aligned} p(\text{Arroseur} = 0, \text{Pluie} = 1) &= p(\text{Arroseur} = 0 | \text{Pluie} = 1) \cdot p(\text{Pluie} = 1) \\ &= 0.99 \cdot 0.2 \\ &= 0.198 \end{aligned}$$

On peut également chaîner ces probabilités. Calculons la probabilité pour que l'herbe soit mouillée, alors qu'il fait sec, et que l'arrosage est éteint.

$$\begin{aligned} p(\text{Pluie} = 0, \text{Arroseur} = 0, \text{Herbe Mouillée} = 1) \\ &= \underbrace{p(\text{Pluie} = 0) \cdot p(\text{Arroseur} = 0 | \text{Pluie} = 0)}_{p(\text{Arroseur}=0, \text{Pluie}=0)} \cdot p(\text{Herbe Mouillée} = 1 | \text{Pluie} = 0, \text{Arroseur} = 0) \\ &= 0.8 \cdot 0.7 \cdot 0.4 \\ &= 0.224 \end{aligned}$$

Dans un réseau bayésien général, dont la totalité des nœuds, notée T_l , est associée à la valeur t_l , on a l'équation (4.68). On note $pa(S)$ les nœuds parents directs du nœud S .

$$p(T_l = t_l) = \prod_{\{S,s\} \in \{T_l=t_l\}} p(S = s | pa(S)) \quad (4.68)$$

4.8.4 Marginalisation

Une marginalisation est similaire à un chaînage de probabilités, sauf que dans ce cas précis, certains nœuds ne sont pas affectés d'une valeur, et peuvent prendre n'importe laquelle. Par exemple, la probabilité qu'« il ne pleuve pas et que l'herbe soit mouillée » est le résultat d'un processus de marginalisation, car le nœud **Arroseur** peut prendre n'importe quelle valeur. Par exemple, pour la probabilité de la proposition « il ne pleut pas et l'herbe est mouillée » on aura :

$$\begin{aligned} p(\text{Pluie} = 0, \text{Herbe Mouillée} = 1) \\ &= p(\text{Pluie} = 0) \cdot p(\text{Arroseur} = 0 | \text{Pluie} = 0) \cdot p(\text{Herbe Mouillée} = 1 | \text{Pluie} = 0, \text{Arroseur} = 0) \\ &\quad + p(\text{Pluie} = 0) \cdot p(\text{Arroseur} = 1 | \text{Pluie} = 0) \cdot p(\text{Herbe Mouillée} = 1 | \text{Pluie} = 0, \text{Arroseur} = 1) \\ &= 0.5736 \end{aligned}$$

Le principe général de ce type de calcul consiste à ne tenir aucun compte de l'état potentiel d'un certain nombre de nœuds dans l'estimation de la probabilité. Le cas général de marginalisation dans les réseaux bayésiens peut être trouvé dans l'équation (4.69).

Supposons que l'on partitionne l'ensemble des nœuds T_l en deux ensembles, V_i celui pour les nœuds duquel une valeur v_i est associée, et O_j pour ceux pouvant prendre n'importe quelle valeur possible. Dans l'équation (4.70), les termes de droite sont obtenus grâce à l'équation (4.68).

$$p(V_i = v_i) = \sum_{o_j \in \{0,1\}^{|O_j|}} \prod_{O_j \notin V_i} p(O_j = o_j \mid pa(O_j)) \prod_{V_i} p(V_i = v_i \mid pa(V_i)) \quad (4.69)$$

$$= \sum_{o_j \in \{0,1\}^{|O_j|}} p(V_i = v_i, O_j = o_j) \quad (4.70)$$

On retrouve, lorsque $O_j = \emptyset$, le cas du chaînage de l'équation 4.68. Dans le cas général où $O_j \neq \emptyset$, il s'agit d'une marginalisation.

4.8.5 Inférence

Une inférence consiste à estimer la probabilité d'un événement pour tous les cas où un autre événement est réalisé. Par exemple, la probabilité qu'« il ne pleuve pas, sachant que l'herbe est mouillée » est le résultat d'une inférence. Cet exemple est donné ci-dessous :

$$\begin{aligned} & p(\text{Herbe Mouillée} = 1) \\ &= p(\text{Pluie} = 0) \cdot p(\text{Arroseur} = 0 \mid \text{Pluie} = 0) \cdot p(\text{Herbe Mouillée} = 1 \mid \text{Pluie} = 0, \text{Arroseur} = 0) \\ &+ p(\text{Pluie} = 0) \cdot p(\text{Arroseur} = 1 \mid \text{Pluie} = 0) \cdot p(\text{Herbe Mouillée} = 1 \mid \text{Pluie} = 0, \text{Arroseur} = 1) \\ &+ p(\text{Pluie} = 1) \cdot p(\text{Arroseur} = 0 \mid \text{Pluie} = 1) \cdot p(\text{Herbe Mouillée} = 1 \mid \text{Pluie} = 1, \text{Arroseur} = 0) \\ &+ p(\text{Pluie} = 1) \cdot p(\text{Arroseur} = 1 \mid \text{Pluie} = 1) \cdot p(\text{Herbe Mouillée} = 1 \mid \text{Pluie} = 1, \text{Arroseur} = 1) \\ &= 0.7716 \end{aligned}$$

Selon (4.15), on a l'inférence « il ne pleut pas sachant que l'herbe est mouillée » :

$$\begin{aligned} p(\text{Pluie} = 0 \mid \text{Herbe Mouillée} = 1) &= \frac{p(\text{Pluie} = 0, \text{Herbe Mouillée} = 1)}{p(\text{Herbe Mouillée} = 1)} \\ &= 0.74339 \end{aligned}$$

La formule générale de l'inférence peut être trouvée dans l'équation 4.71, sachant que chacun des termes au numérateur et dénominateur du membre de droite peut être obtenu par la formule de marginalisation (4.69). Dans cette équation, les nœuds V_i sont ceux sur lesquels on veut obtenir une inférence, conditionnés aux nœuds E_j , appelés « évidences ».

$$\begin{aligned} p(V_i = v_i \mid E_j = e_j) &= \frac{p(v_i = v_i, E_j = e_j)}{p(E_j = e_j)} \\ &= \frac{p(V_i = v_i, E_j = e_j)}{\sum_{v_i' \in \{0,1\}^{|V_i|}} p(V_i = v_i', E_j = e_j)} \end{aligned} \quad (4.71)$$

Estimer une grandeur probabiliste V grâce à une inférence sur une variable E revient à ajouter de la connaissance à cette variable, et à diminuer l'entropie associée. La connaissance d'une variable E fortement corrélée avec celle que l'on veut estimer, comme par exemple un nœud parent ou successeur, permet d'estimer plus finement en ajoutant une

quantité d'information égale à l'information mutuelle $I(V : E)$, qui correspond au degré de corrélation probabiliste entre les deux variables, à l'estimation de V . On pourra se référer à la section 4.6.3.

4.8.6 Réseaux Bayésiens Chaînés

Dans cette section, nous présentons les similitudes que l'on peut avoir entre réseaux bayésiens chaînés, et les chaînes de Markov simples, tels le formalisme du filtre de Kalman. Nous allons présenter des réseaux bayésiens associés à des variables continues et discrètes.

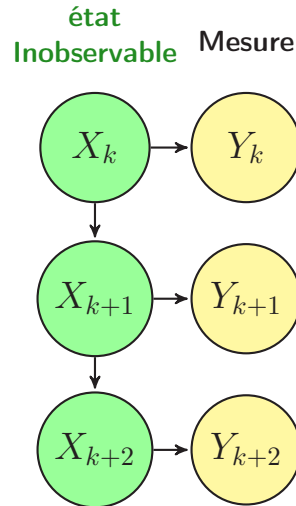


FIGURE 4.7 – Réseau bayésien type « Chaîne de Markov » représentant un filtre de Kalman.

La figure 4.7 montre qu'un filtre de Kalman peut être représenté par un réseau graphique de type chaîne de Markov. Les nœuds Y représentent la variable observable du système, qui découle directement de manière causale et linéaire de l'état inobservable du système, matérialisé par les nœuds X . Les variables continues X_k et Y_k sont représentées par des distributions **Normales**.

A chaque étape du déroulement du filtre de Kalman, une nouvelle valeur Y_k est mesurée. Les équations de 4.72 à 4.74 indiquent les étapes du calcul (hormis l'étape de prédiction).

$$p(X_k^+) = p(X_k^- | Y_k) \tag{4.72}$$

$$p(X_{k+1}^-) = p(X_k^+) \tag{4.73}$$

$$\Rightarrow p(X_k^+) = p(X_0^- | Y_0 \rightarrow k) \tag{4.74}$$

On peut voir sur la figure 4.8 un réseau bayésien à valeurs discrètes chaîné, similaire à celui de la figure 4.7, à ceci près que le réseau consiste en plusieurs nœuds en interaction (O_k^-, d_k, O_k^+) au lieu d'un seul (nœud X_k). Le chaînage ne se produit qu'en un seul point, la jonction $O_k^+ \rightarrow O_{k+1}^-$, tout comme pour le réseau de la figure 4.7, $X_k \rightarrow X_{k+1}$.

Dans ce type de réseaux, on tente d'estimer une variable unique ayant une structure temporelle causale. L'estimation d'une grandeur vectorielle dans le temps obéit à cette

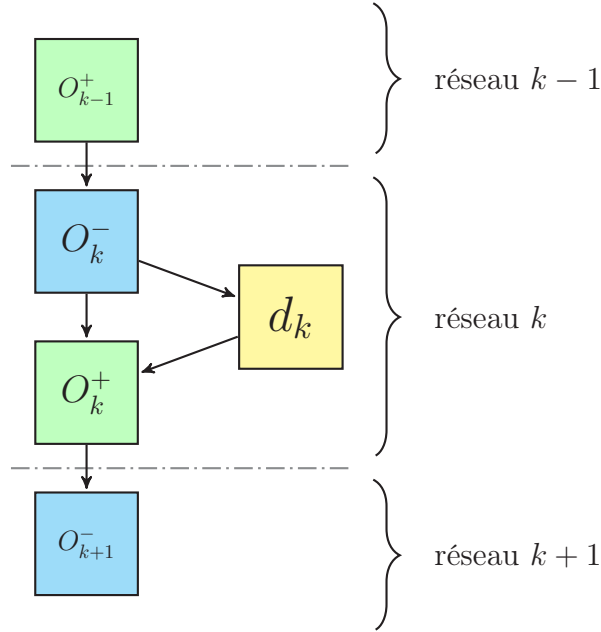


FIGURE 4.8 – Réseau bayésien discret chaîné.

logique, de même que le suivi d'un indicateur de type binaire par une probabilité entre 0 et 1, comme la probabilité d'occurrence d'une panne en fonction du temps.

Comme pour le filtre de Kalman, le réseau de la figure 4.8 peut être décomposé en unités chaînables de manière triviale. L'estimation postérieure de O_k^+ peut, comme pour les équations 4.72 et 4.73, être séparée entre les différents réseaux, ce qui fait que l'on peut réinjecter la valeur postérieure directement comme un a-priori du réseau suivant. Nous allons maintenant nous pencher sur les techniques d'approximation de probabilités discrètes.

4.9 Approximation dite de « Champ Moyen »

L'approximation de type « Champ Moyen » (Mean-Field en anglais) est une approximation classique en théorie de l'information. Elle consiste à faire l'approximation que deux ensembles de variables aléatoires ou plus sont décorrélés en supprimant les termes de corrélation entre chaque ensemble.

Dans les thématiques d'estimation statistique sur de grandes quantités de données, ou bien d'apprentissage, cette estimation permet de simplifier la représentation, et donc son coût calculatoire. Supposons que nous disposions de données corrélées, avec de fortes et de faibles corrélations. Le nombre de coefficients requis afin de modéliser exactement toutes les corrélations croît de manière exponentielle avec le nombre de variables estimées. Certaines corrélations peuvent apparaître marginales. Supprimer les termes correspondant à ces corrélations permet de simplifier grandement le modèle tout en tentant d'obtenir une estimation intégrale. Cette opération est réalisée par l'approximation « Mean-Field », que nous détaillons ci-après.

4.9.1 Exemple visuel

Utilisons le formalisme de représentation des réseaux de Markov, par cliques et corrélations non-directionnelles (c'est à dire par fonctions de potentiel). L'idée est de pouvoir passer d'une représentation conforme à la figure 4.9 à celle de la figure 4.10. L'intérêt que nous voyons à cette approche consiste à pouvoir séparer plusieurs variables, ou sous-réseaux **bayésiens** entre eux. L'usage ici de la représentation markovienne est faite ici pour simplifier le propos, mais cette approximation peut également s'appliquer aux réseaux bayésiens. Dans ce qui suit, nous allons utiliser l'approximation mean-field afin de décorréler deux variables qui initialement le sont.

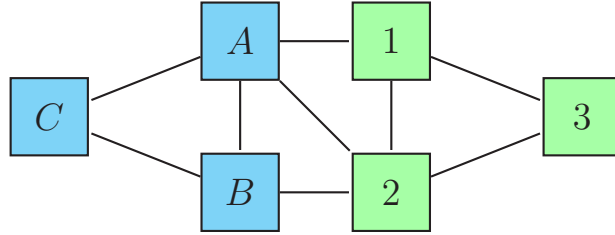


FIGURE 4.9 – Réseau markovien discret.

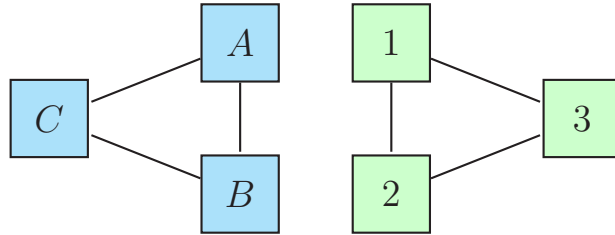


FIGURE 4.10 – Deux réseaux markovien discrets décorrés.

4.9.2 Familles Exponentielles

On désigne par le terme « famille exponentielle » toute représentation d'une loi de probabilité sous la forme de l'équation (4.75). La fonction Ψ est appelée fonction de partition. Le vecteur Θ est appelé paramètre naturel de la famille exponentielle.

$$p(X_i = x_i) = \exp \left[\sum_{x_i \in X_i} \Theta_i \cdot x_i - \Psi(\Theta_i) \right] \quad (4.75)$$

Par exemple, pour une loi de probabilité discrète à deux variables binaires, X et Y à valeurs dans $\{0, 1\}$, on a les équations (4.76) et (4.77).

$$p(X = x, Y = y) = \frac{e^{[\Theta_x \cdot x + \Theta_y \cdot y + \Theta_{xy} \cdot x \cdot y]}}{1 + e^{\Theta_x} + e^{\Theta_y} + e^{\Theta_x + \Theta_y + \Theta_{xy}}} \quad (4.76)$$

$$= e^{[\Theta_x \cdot x + \Theta_y \cdot y + \Theta_{xy} \cdot x \cdot y - \Psi(\Theta_x, \Theta_y, \Theta_{xy})]} \quad (4.77)$$

$$\Psi(\Theta_x, \Theta_y, \Theta_{xy}) = \log [1 + e^{\Theta_x} + e^{\Theta_y} + e^{\Theta_x + \Theta_y + \Theta_{xy}}] \quad (4.78)$$

Les termes Θ_x et Θ_y représentent les valeurs « indépendantes » pour chacun des potentiels des variables X et Y pris comme indépendants. Le terme Θ_{xy} quant à lui représente le terme de corrélation entre la variable X et la variable Y . Pour plus de détails sur ce sujet, se référer à la première partie de (Amari, 2001).

On peut décomposer toute loi de probabilité discrète selon (4.75). On peut prouver les équations (4.79), (4.80) et (4.81).

$$\mathbb{E}(x) = \partial\Psi\Theta_x \quad (4.79)$$

$$\mathbb{E}(y) = \partial\Psi\Theta_y \quad (4.80)$$

$$\mathbb{E}(x \cdot y) = \partial\Psi\Theta_{xy} \quad (4.81)$$

Ceci a plusieurs applications en statistique. Dans notre cas, on peut définir une sous-famille de cette famille exponentielle, dans laquelle les variables x et y sont indépendantes. Si $x \perp y$, alors :

$$\mathbb{E}(x \cdot y) = \mathbb{E}(x) \cdot \mathbb{E}(y)$$

Or, on a, selon (4.81) :

$$\begin{aligned} \mathbb{E}(x \cdot y) &= \frac{e^{\Theta_x + \Theta_y + \Theta_{xy}}}{1 + e^{\Theta_x} + e^{\Theta_y} + e^{\Theta_x + \Theta_y + \Theta_{xy}}} \\ &= \mathbb{E}(x) \cdot \mathbb{E}(y) \cdot e^{\Theta_{xy}} \end{aligned}$$

Alors :

$$x \perp y \Rightarrow \Theta_{xy} = 0 \quad (4.82)$$

Dans le cas où l'on souhaite trouver la sous-famille des lois de probabilités où les variables x et y sont indépendantes, il suffit de restreindre le paramètre générateur de la distribution Θ_{xy} à la valeur 0. Le formalisme global n'est changé en rien.

4.9.3 Approximation de « Champ Moyen »

Soit une loi de probabilité q à approximer suivant une loi indépendante. L'approximation de « Champ Moyen » consiste à prendre parmi les distributions de probabilités dans lesquelles $X \perp Y$ (soit $\Theta_{xy} = 0$ selon (4.82)), celle qui minimise la divergence de Kullback-Leibler selon (4.83). On peut prouver (c.f. Shore and Johnson, 1980) que la minimisation obtenue dans (4.83) maximise l'entropie de la solution p_{XY} , et que seule cette solution peut permettre de conserver l'« intégrité » des calculs en dépendant, par rapport aux résultats donnés par le calcul exact (basé sur la loi de probabilité q).

$$\begin{aligned} p_{XY}(x, y) &= p_X(X = x) \cdot p_Y(Y = y) \\ p_{\text{Mean-Field}} &= \arg \min_{p_{XY}/p_X, p_Y \in [0,1]^2} D_{KL}(p_X(x) \cdot p_Y(y) \| q(x, y)) \end{aligned} \quad (4.83)$$

Dans le cas général des figures 4.9 et 4.10, seuls les termes croisés correspondant à des potentiels mélangeant des nœuds des deux sous-réseaux $\{A, B, C\}$ et $\{1, 2, 3\}$ seront annulés, à l'exclusion de tous les autres. C'est la principale raison de l'utilisation de la

décomposition exponentielle de l'équation (4.75), car elle permet de séparer les contributions locales à chacun des sous-réseaux de celles faisant intervenir ces deux sous-réseaux en les couplant.

Cette manière d'estimer fait obligatoirement augmenter l'entropie, puisque certains termes probabilistes ont été éliminés, d'une valeur égale à $D_{KL}(p_X(x) \cdot p_Y(y) \| q(x, y))$. Plus l'estimation indépendante sera mauvaise par rapport à la loi exacte $q(x, y)$, plus l'entropie de l'estimation « mean-field » va augmenter par rapport à l'entropie de la loi exacte. Cela signifie que l'on retire dans ce cas une grande quantité d'information de la loi de probabilité en supprimant les termes de couplage.

4.9.4 Discussion

L'approximation « Mean-Field » est toujours définie. Mais on est en droit de se poser la question de sa pertinence. En effet, selon la valeur réelle du paramètre Θ_{xy} dans la loi de probabilités à approximer, l'approximation pourra être faite de manière plus ou moins bonne.

Il est noté dans (Brand, 1997, pages 8–9) que la qualité de l'approximation obtenue est de plus en plus médiocre au fur et à mesure que les termes de couplages augmentent. Nous en déduisons que, dans le cas où le terme Θ_{xy} est dominant par rapport aux deux autres termes Θ_x et Θ_y , l'approximation devient mauvaise.

De plus, le problème de la minimisation que l'on trouve dans l'équation (4.83) n'est pas, et ce de manière certaine, un problème convexe (la variété dans laquelle on cherche le minimum n'est pas convexe). Ce qui implique deux choses. D'une part, le minimum n'est plus garanti d'être unique, et de ce fait, les conditions d'initialisation vont avoir un impact important sur le minimum obtenu à l'issue du processus de minimisation. D'autre part, même lorsque les données du problème sont symétriques, la non-convexité du problème a pour conséquence que plusieurs optima existent. Dans ce cas, ils ne sont pas positionnés de manière à être symétriques de manière individuelle (pas sur un axe de symétrie), mais de manière globale (un optimum pourra être le symétrique d'un autre). Le résultat obtenu pourra donc ne pas être symétrique, ce qui est appelé, dans le vocabulaire de la physique, « brisure de symétrie ».

Plus la qualité de l'approximation « mean-field » est mauvaise, plus le terme $D_{KL}(p_X(x) \cdot p_Y(y) \| q(x, y))$ est mauvais. Ce terme est mauvais d'autant plus que le terme Θ_{xy} est élevé en valeur absolue. Plus ce terme est mauvais, plus les estimations obtenues auront de possibilité d'être dans un optimum non-symétrique, et de perdre certaines propriétés de la loi de probabilité exacte.

Ces éléments permettent de comprendre pour quelles raisons l'approximation « Mean-Field » peut ne pas être utilisable.

Conclusion

Dans ce chapitre, nous avons présenté le formalisme que nous utiliserons dans la suite pour les variables suivant une loi de probabilités **Gaussienne**, ainsi que pour des variables probabilistes discrètes à valeurs booléennes.

Nous avons également présenté la théorie de l'information en termes généraux. Après avoir présenté la grandeur « entropie de Shannon », ainsi que son équivalent continu, l'« entropie différentielle », nous avons brièvement présenté les concepts qui en découlent, l'*information mutuelle* et la *divergence de Kullback–Leibler*. Nous avons finalement donné

un petit aperçu des méthodes d'approximation et de projection utilisées en statistiques, telles l'*approximation de champ moyen*.

Il est à mentionner que, malgré plus de 60 ans de recherches dans ce domaine, la notion d'*entropie* reste toujours mal connue, et difficile à interpréter conceptuellement en termes concrets. De ce fait, plusieurs interprétations de cette notion existent, qui sont liées à des thématiques d'application (théorie des communications, statistiques, biologie, médecine, apprentissage...) et il n'en existe aucune qui les englobe toutes. Dans ce chapitre, nous prenons la variable entropie, ses écarts, et le terme de corrélation appelé entropie mutuelle, comme des éléments d'analyse pour éclairer le fonctionnement des estimations probabilistes.

Nous avons ensuite présenté les outils d'estimation de variables dynamiques continues, notamment le *filtre de Kalman*, mais également certaines de ses variantes. Nous avons également présenté le problème de la propagation de rumeur dans ce type de filtres. Enfin, nous avons dévolu une section à la problématique de l'estimation des variables aléatoires à valeurs binaires en utilisant le formalisme des **Réseaux Bayésiens**. Ces techniques sont basées sur un formalisme bayésien qui permet leur analyse en terme de théorie de l'information.

Nous allons maintenant passer à la partie de ce mémoire qui présente la méthode de base sur laquelle nos travaux ont été réalisés ainsi que ces travaux proprement dits et les résultats obtenus dans le cadre de cette thèse.

Chapitre 5

Localisation dans une Carte Existante

Le précédent chapitre détaillait les différentes techniques d'estimation probabilistes que nous utilisons dans les travaux décrits dans ce mémoire. Ces techniques sont utilisées dans les travaux de base, la théorie du *Triplet Perceptif*, sur lesquels nous basons nos travaux. Nous présentons tout d'abord le formalisme utilisé dans ce cadre. Puis, nous étudierons la manière dont l'estimation de la position du véhicule est faite. Enfin, nous présentons la manière dont on peut estimer l'exactitude de cette dernière estimation par une probabilité discrète. Une nouvelle version du code informatique a été réécrite, afin de permettre l'évolution et l'ajout des trois fonctions relatives aux objectifs poursuivis et qui sont :

1. permettre de préciser ou corriger un positionnement incorrect d'un élément fixe de la carte, un amer,
2. permettre la suppression d'un amer inscrit dans la carte n'ayant pas de contrepartie réelle tangible,
3. permettre l'ajout d'amers existant de manière tangible, mais non-encore insérés dans la carte.

5.1 Introduction

Dans cette introduction, nous indiquons brièvement l'historique des développements dans le cadre des thèses antérieures, avec les différentes contributions associées. Nous décrivons également la problématique que cette méthode de localisation est censée résoudre.

5.1.1 Historique des Travaux

Notre laboratoire, l'Institut PASCAL, est un laboratoire possédant de multiples centres d'intérêts, génie biologique, physique des semi-conducteurs et électromagnétisme, santé, outils de production, et robotique. Plusieurs thématiques existent dans notre axe de recherche, en robotique :

- caméras intelligentes,
- commande et guidage,
- vision par ordinateur,
- perception et systèmes.

Une des applications visées consiste en le développement d'une navette autonome. Dans ce cadre, plusieurs approches sont possibles. Une approche populaire consiste en un

SLAM caméra (voir section 2.4.8). Mais un des problèmes est que la carte est relative au véhicule, et qu'elle accumule de l'erreur au fur et à mesure du déplacement du véhicule.

Une autre idée consiste par exemple à utiliser une carte de l'environnement, contenant par exemple des positions de murs. Se repérer dans une telle carte permettrait de pouvoir limiter la dérive du véhicule de manière absolue, ainsi que de pouvoir échanger des positions entre véhicules dans un référentiel unique et absolu. C'est l'approche que nous décrivons ici.

La stratégie de localisation que nous présentons ici, est une nouvelle implémentation des travaux de (Aynaud et al., 2014a,b). Elle est basée sur plusieurs spécificités :

- la notion de fusion multi-capteur, appliquée par le formalisme de *Triplet Perceptif*,
- l'utilisation d'une approche « descendante » qui sélectionne en permanence la meilleure source d'information,
- un critère de choix de ces sources et modalités de fusion de données,
- une localisation de véhicule basée sur un filtre de Kalman,
- une estimation de l'intégrité de localisation.

Nous allons décrire les uns après les autres ces différents concepts, et nous allons également présenter la méthode de localisation qui les intègre tous.

5.2 Concept de *Triplet Perceptif*

5.2.1 Concepts de « proprioceptivité » et d'« extéroceptivité »

Dans le chapitre 2, nous avons brièvement introduit les concepts de proprioceptivité et d'extéroceptivité. Afin d'expliquer plus avant ces deux notions, nous allons les définir d'une manière légèrement différente en fonction de nos objectifs.

Supposons que nous disposions de données, sous forme d'un vecteur d'état que nous nommerons \mathbf{Z}_t , t étant la date de la mesure. Supposons que l'état de tous les éléments observables soit estimé par des vecteurs nommés $\mathbf{X}_*(t)$, expression dans laquelle t représente la date à laquelle celles-ci sont valables.

Nous définissons une mesure \mathbf{Z} **extéroceptive** comme une mesure pour laquelle une équation du type (5.1) est valable. Dans cette approche, on peut voir la mesure comme ne dépendant d'aucun changement, mais uniquement fonction de l'**état** à l'instant de mesure de chaque élément potentiellement cartographié ou modélisé.

$$\mathbf{Z}_t = \mathbf{z}(\mathbf{X}_1(t), \mathbf{X}_2(t), \dots, \mathbf{X}_n(t)) \quad (5.1)$$

Un exemple typique de ce type de situation est la mesure par un véhicule d'un amer positionné de manière absolue dans une carte. Avec $\mathbf{X}_V(t)$ représentant l'état d'un véhicule et \mathbf{X}_P celui (constant) d'un poteau par exemple, tous deux estimés à l'instant t , les coordonnées \mathbf{Z} du poteau dans le repère du véhicule (la mesure) s'exprime comme dans l'équation 5.2.

$$\mathbf{Z}_t = \mathbf{z}_1(\mathbf{X}_V(t), \mathbf{X}_P) \quad (5.2)$$

Nous définissons une mesure \mathbf{Z} **proprioceptive** par la validité d'une équation comme (5.3).

$$\dot{\mathbf{X}}(t) = \mathbf{z}(\mathbf{X}(t), \mathbf{Z}_t) \quad (5.3)$$

Par exemple, sur un véhicule automobile, un état $\mathbf{X}_M(t)$ représentant la vitesse de rotation de l'arbre moteur et une mesure \mathbf{Z}_t représentant le couple total appliqué sur celui-ci (couple moteur, couple résistant par exemple), on a l'équation (5.4).

$$\dot{\mathbf{X}}_M(t) = \mathbf{z}(\mathbf{X}_M(t), \mathbf{Z}_t) \quad (5.4)$$

Il est tout à fait envisageable que d'autres types de mesures existent, et ne puissent se réduire à ces deux types d'expressions. Cela peut être le cas dans des systèmes complexes en interaction. Dans tout ce qui suit, nous supposons que seuls ces deux types de données sont accessibles à la mesure.

Une remarque rapide sur certains types de mesures. Supposons que nous disposions d'un accéléromètre 3 axes, orienté de manière que deux des axes de mesure soient parallèles au sol. Supposons qu'il ne subisse aucune rotation, afin de rendre la discussion plus simple. Dans le cas d'un état dans lequel un véhicule est modélisé par l'équation (5.5), dans laquelle x et y représentent des coordonnées planes du véhicule, et \dot{x} et \dot{y} leurs variations dans le temps.

$$\mathbf{X}_1 = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \quad (5.5)$$

La mesure \mathbf{Z}_t correspond à la définition (5.6).

$$\mathbf{Z}_1 = \begin{pmatrix} a_x = \ddot{x} \\ a_y = \ddot{y} \end{pmatrix} \quad (5.6)$$

On vérifie que dans ce cadre, la mesure \mathbf{Z}_1 est proprioceptive par rapport à l'état \mathbf{X}_1 .

Supposons maintenant que le véhicule se déplace sur une route non-plane et que l'on tente de mesurer les défauts de la route en fonction des accélérations verticales du capteur. On pourra par exemple utiliser un état du type de celui présenté dans (5.7) (v_h étant la mesure de vitesse horizontale et a_z l'accélération mesurée) et une mesure comme celle de (5.8).

$$\mathbf{X}_2 = \begin{pmatrix} v_h \\ a_z \end{pmatrix} \quad (5.7)$$

$$\mathbf{Z}_2 = (a_z = \ddot{z}) \quad (5.8)$$

Dans ce cas, on trouve que la mesure \mathbf{Z}_2 est extéroceptive par rapport à l'état \mathbf{X}_2 . Dans ce cas, le capteur est ambivalent, car en fonction du modèle utilisé et de l'objectif visé, on peut le considérer proprioceptif ou extéroceptif. Dans tout ce qui suit, nous considérerons néanmoins, ainsi que cela avait été fait dans le chapitre 2, que ce type de capteur, tout comme les gyromètres, sont de nature proprioceptive. Nous considérerons par ailleurs que les capteurs de type magnétomètre, sont extéroceptifs, comme on le verra au chapitre 6.

Dans tout ce qui suit, nous appliquerons le concept de **triplet perceptif** à toutes les mesures de type **extéroceptives pures**, à l'exclusion de toute autre type de mesure qui ne conviennent pas à ce concept de fusion de données. Ceci exclut donc naturellement les capteurs de type proprioceptifs, comme par exemple les mesures d'odométrie sur train de véhicule automobile, de type accéléromètre et gyromètre, du concept de triplet perceptif. Ces mesures (et dans notre cas, les mesures d'odométrie) sont néanmoins utilisées pour l'étape de prédiction du filtre de Kalman.

5.2.2 Présentation du Concept de Triplet

Tout être humain a tenté au moins une fois de se localiser dans une ville inconnue. Supposons que la personne en question dispose d'un plan de la ville. Il suffirait de pouvoir identifier sur le plan un élément visible de loin (par exemple, le clocher d'une cathédrale), puis de tenter de l'observer visuellement autour de soi. Une fois cela fait, il suffit de répéter au fur et à mesure cette opération, comme on se déplace, afin de se localiser. On peut utiliser par exemple, si l'on est à une intersection, tenter de détecter les plaques indiquant le nom des rues, ou bien identifier un commerce, ou un arrêt de bus, la liste n'étant pas exhaustive, et dépendant des éléments qui sont cartographiés à proximité.

Dans le cas d'un véhicule, la méthode est sensiblement la même. On peut utiliser une carte comprenant des murs, des poteaux par exemple, cartographiés de manière supposée infiniment précise. à chaque étape de localisation, le véhicule va choisir un mur différent, tenter de l'observer, et s'il y parvient, repréciser sa localisation par rapport à ce dernier.

Mais si l'on peut faire cela avec un mur, pourquoi ne pas le faire avec un poteau, le marquage au sol identifiant les voies, ou bien les panneaux indicateurs ou de signalisation ? Cette idée se heurte au fait, que si un mur vertical est détectable avec les données provenant d'un LIDAR plan et par identification d'une ligne droite, un panneau indicateur est lui détectable avec une caméra couleur et une identification des caractères. D'où l'idée de gérer ces différentes situations grâce à un concept unique, le concept de *Triplet Perceptif*, qui comprend :

- un capteur \mathcal{S}_i , dont le rôle est de fournir des données,
- un détecteur \mathcal{D}_j , dont le rôle est, à partir des données fournies par \mathcal{S}_i , de détecter
- un amer \mathcal{L}_k , appartenant à la carte.

Donc, \mathcal{S}_i est un capteur physique (équipement transmettant des données de mesure), \mathcal{D}_j un algorithme (code informatique) et \mathcal{L}_k un élément cartographié stocké dans une « carte », par ses propriétés (position, géométrie par exemple). La carte utilisée ici est donc du double type sémantique–métrique. Un exemple parmi d'autres d'implémentation du concept de triplet perceptif est visible sur la figure 5.1. Celui-ci représente les triplets suivants :

- Capteur LIDAR–Détecteur de lignes droites–Mur,
- Capteur LIDAR–Détecteur de poteaux–Poteau,
- Capteur GPS–Détecteur de position GPS bidimensionnelle–Constellation NAVSTAR,
- Capteur magnétomètre–Détecteur d'orientation–Champ magnétique terrestre.

Ce formalisme permet de modéliser virtuellement n'importe quelle mesure extéroceptive avec cette notion de *Triplet Perceptif*. Dans ces triplets, certains amers peuvent être purement « symboliques », par exemple dans le cas de mesures absolues. Par exemple, on ne peut mettre, dans le triplet GPS, d'informations relatives à la position de la constellation NAVSTAR, puisque l'opération de prise en compte du signal est faite par le récepteur GPS. Dans ce cas, l'amer « Constellation NAVSTAR » ne contient aucune information tangible, et est donc « symbolique ». Cela peut ne pas être le cas avec le champ magnétique terrestre, puisque la direction horizontale de ce dernier n'indique pas le nord vrai, mais le nord magnétique, différence qu'il faut prendre en compte. L'écart angulaire entre ces deux directions, la déclinaison magnétique, est fonction de la localisation du véhicule, et doit pouvoir être calculée.

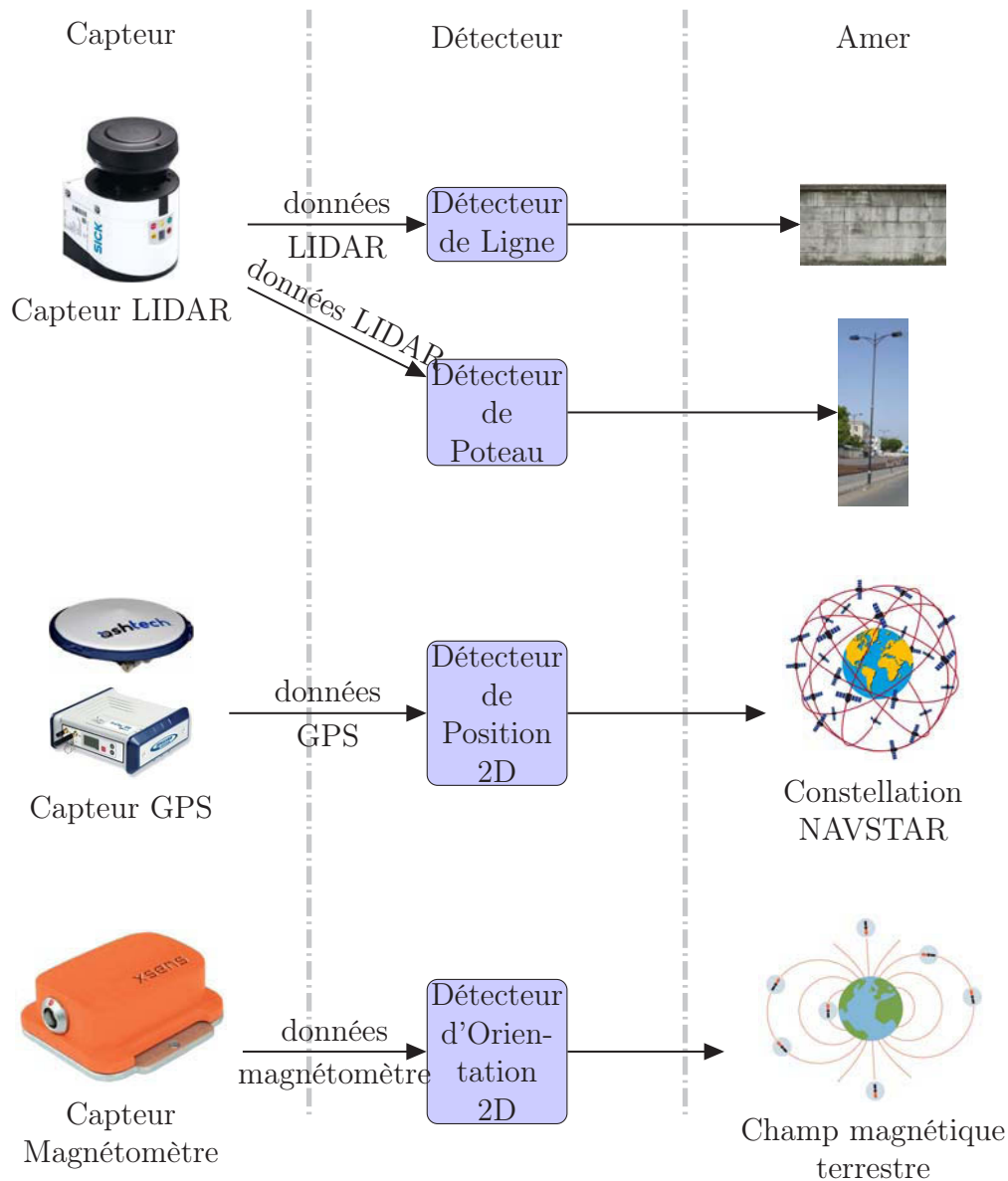


FIGURE 5.1 – Exemple de triplets perceptifs. On notera que la CONSTELLATION NAVSTAR représente la **totalité des satellites** pouvant donner des informations au récepteur GPS.

5.2.3 Choix du Triplet

Dans le déroulement du processus de localisation, des données sont reçues de manière régulière de tous les capteurs embarqués. Deux approches d'utilisation de ces données sont possibles :

1. réaliser, à chaque fois qu'une nouvelle information est reçue, une correction de la position du véhicule,
2. choisir un unique triplet pour une certaine période de temps, et pour cette durée, ne prendre en compte que les données entrantes relatives à ce triplet.

La première option revient à prendre en compte simultanément toutes les sources de données, sources dont le flux de communication peut être important. Cette approche

est gourmande en puissance de calcul, puisqu'il faut traiter tous les flux de données en temps réel, et pour tous les amers. Nous désignerons ce type d'approche par le terme **BOTTOM-UP**.

La seconde, quant à elle, consiste à limiter les analyses à un unique capteur, du type de l'amer choisi, qui doit être dans la zone que le capteur observe. Cette solution limite fortement les calculs et est de ce fait moins gourmande en puissance de calcul, mais elle a comme inconvénient de devenir aveugle aux autres capteurs, qui véhiculent également des informations importantes, en qualité et en quantité. On peut penser limiter ce problème en faisant appel au triplet offrant la meilleure mise à jour possible. Cette approche, précédemment mentionnée comme « descendante », sera désignée par le terme **TOP-DOWN**.

Dans tout le reste de ce rapport, le choix du triplet sera réalisé par l'optimisation d'une fonction de coût \mathbf{C} , conformément à l'équation (5.9). Dans ce chapitre, nous présentons à la section 5.5 le critère de départ, mais une nouvelle fonction d'optimisation sera introduite au chapitre 6.

$$\begin{aligned} \mathcal{S} &= \{\mathcal{S}_i\} \\ \mathcal{D} &= \{\mathcal{D}_j\} \\ \mathcal{L} &= \{\mathcal{L}_k\} \\ \begin{pmatrix} \mathcal{S}_{\text{best}} \\ \mathcal{D}_{\text{best}} \\ \mathcal{L}_{\text{best}} \end{pmatrix} &= \arg \max_{\begin{pmatrix} \mathcal{S} \\ \mathcal{D} \\ \mathcal{L} \end{pmatrix} \in \mathcal{S} \times \mathcal{D} \times \mathcal{L}} \mathbf{C}(\mathcal{S}, \mathcal{D}, \mathcal{L}) \end{aligned} \quad (5.9)$$

5.2.4 Représentation de l'état du Véhicule

Dans la méthode de localisation originale, l'état du véhicule est représenté par trois variables, voir l'équation (5.10).

$$\begin{cases} p_{\text{confiance } \nu} & \text{confiance dans l'exactitude de la localisation} \\ \mathbf{X}_\nu & \text{vecteur d'état du véhicule, représentant sa } \textit{position} \\ \Sigma_{\mathbf{X}_\nu} & \text{covariance reliée à l'état du véhicule} \end{cases} \quad (5.10)$$

La variable vectorielle \mathbf{X}_ν représente l'estimation de l'état du véhicule selon (5.11), conformément à la figure 5.2. La paramétrisation est du type $2D \frac{1}{2}$ plane. La variable $\Sigma_{\mathbf{X}_\nu}$ représente la matrice de covariance associée à l'estimation de l'état du véhicule.

$$\mathbf{X}_\nu = \begin{pmatrix} x \\ y \\ \varphi \end{pmatrix} \quad (5.11)$$

La variable $p_{\text{confiance } \nu}$ représente la probabilité que la variable \mathbf{X}_ν représente bien la valeur réelle de l'état du véhicule, moyennant l'incertitude donnée par $\Sigma_{\mathbf{X}_\nu}$, et en moyenne. La validité de cette estimation est en règle générale calculée selon une règle de rejet à 3σ (donnée valable si distance de *Mahalanobis* inférieure à 3), mais tout autre seuil peut être interprété par rapport au test de *Pearson- χ^2* (Pearson, 1900).

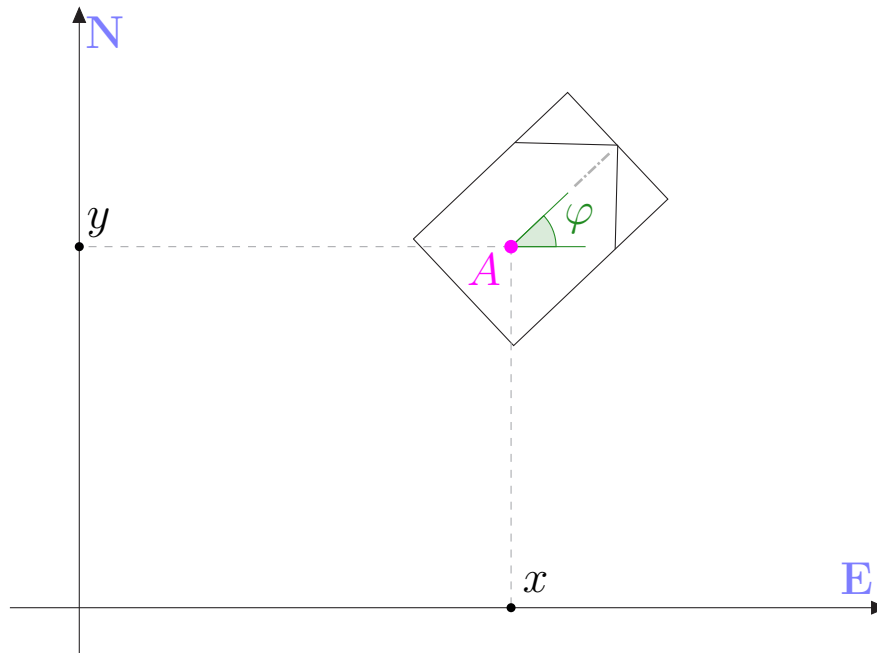


FIGURE 5.2 – Visualisation du modèle continu \mathbf{X}_v du véhicule, selon 5.11.

5.2.5 Structure de l’Algorithme de Localisation

Dans la méthode de localisation, deux processus distincts, et partiellement disjoints existent :

- le choix toutes les 100 *ms* du triplet le meilleur,
- le processus de mise à jour de l’état du véhicule.

Ces deux processus fonctionnent de manière parallèle sur deux *threads* séparés du système d’exploitation.

La mise à jour de l’état du véhicule est subdivisée elle-même en deux autres sous-parties :

- la mise à jour de l’état continu (vecteur d’état et covariance),
- la mise à jour de l’estimation de la confiance/intégrité relative à cet état continu.

Le diagramme 5.3, que nous allons commenter, décrit le fonctionnement du système de localisation qui a été réécrit et a servi de base aux développements ultérieurs.

Analysons la figure 5.3. Nous démarrons avec un véhicule positionné correctement, par l’étape *Position Estimée*, tout à gauche du diagramme. L’étape suivante est le choix du *Triplet*, qui consiste à choisir le *type de l’amer* (dans notre cas, ce sont toujours des murs), l’*amer* choisi, le *type de capteur* (ici, un LIDAR plan), le *capteur*, ainsi que le *détecteur*. Ce choix doit permettre de « se localiser au mieux », et est basé sur l’optimisation d’une fonction par le *Triplet* en question. Une fois ce choix fait, le véhicule peut alors procéder à une détection, d’abord en attendant que le capteur en question retourne des données, puis en procédant à la détection de l’amer.

Dans le cas où la détection a réussi, la position du véhicule est mise à jour par un filtre de Kalman, dont le modèle est précisé à la section 5.3.2. Que la détection réussisse ou échoue, la probabilité permettant d’estimer le degré d’exactitude (intégrité) de l’estimation de la localisation du véhicule, appelée *confiance* dans la suite de ce mémoire, est mise à jour. On revient alors à la situation de départ, et un autre cycle s’engage alors.

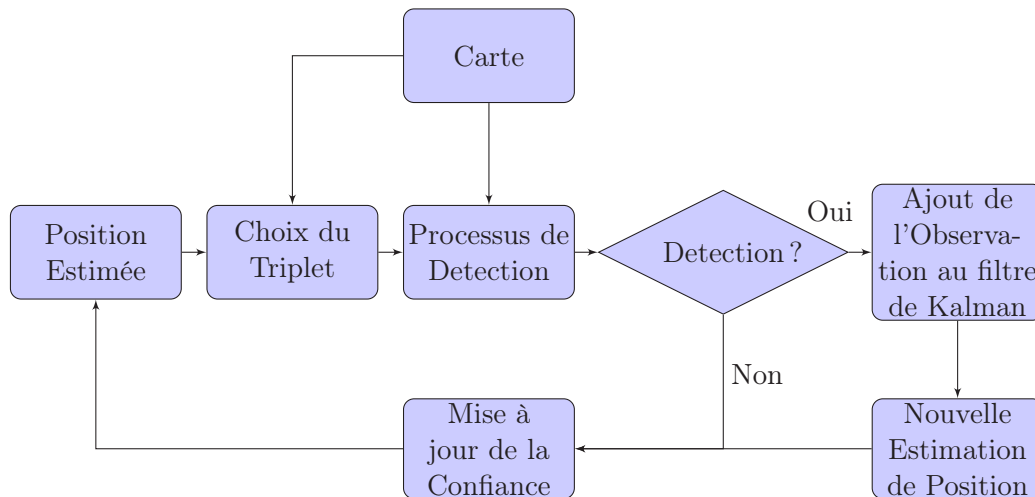


FIGURE 5.3 – Diagramme synoptique de la méthode de localisation du véhicule.

Une remarque doit ici être faite. Dans l’implémentation développée, le choix du *Triplet Perceptif* se fait dans un *thread* séparé, de manière totalement asynchrone par rapport au reste de l’algorithme, qui se déroule dans un second *thread*. Plusieurs détections avec le même triplet peuvent se produire, de manière variable en fonction de la cadence d’arrivée des données fournies par le capteur choisi. De plus, le choix du triplet est réalisé avec une fréquence de 10 Hz soit toutes les 100 ms , dans un processus cadencé par une *interruption timer*.

5.2.6 Contenu de la Carte

La base de la méthode de localisation étant le contenu de la carte, nous allons brièvement décrire celle-ci. Nos expérimentations se sont principalement déroulées sur le site d’expérimentation *Plateforme Auvergnate pour les Véhicules INtelligents-PAVIN* du campus des CÉZEAUX de l’*Université Clermont-Auvergne*.

Ce site a été conçu à l’échelle $1/2$ par rapport à la réalité de sites urbains (largeur des voies, hauteur des murs par exemple) ce qui permet de tester à minima des techniques de localisation et de commande de manière réaliste et neutre sans introduire de biais expérimental, quelle que soit la méthode de localisation (LIDAR, caméra...). Il est de plus équipé d’une station de base de type RTK permettant de calculer les corrections au signal GPS-L1, couvrant l’ensemble du campus des CÉZEAUX. On peut voir sur la figure 5.4 une vue de côté de celui-ci, côte à côte avec une vue de dessus de cette carte.

La carte numérique est une carte sémantique-métrique, comme cela a déjà été exposé. Le système de coordonnées pris comme support de l’information est le standard géographique *NTF-LambertII* de l’IGN. Cette carte contient les coordonnées planes au standard *NTF-LambertII* de chacun des sommets de chaque mur et de chaque grillage cartographiés. Celle-ci a été obtenue par numérisation au LASER-LIDAR, et alignée par rapport au système de coordonnées *NTF-LambertII* par recalage sur des bornes IGN. Sa précision estimée est de l’ordre du centimètre.

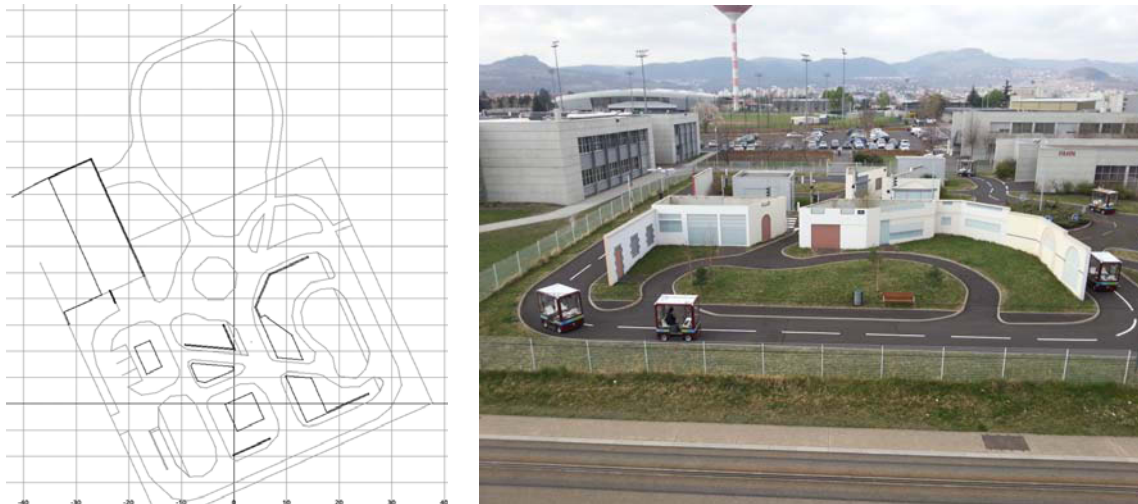


FIGURE 5.4 – À gauche, carte du site d’expérimentations PAVIN (Nord en haut), à droite, photographie prise par le côté Est du site. Sur la carte de gauche, les murs sont indiqués en noir, les bords de chemins carrossables et les grillages sont en gris.

5.3 Estimation de position d’un véhicule dans une carte précise

Dans cette section, nous présentons la méthode utilisée pour la mise à jour de la position du véhicule. À chaque fois qu’une nouvelle trame provenant du capteur correspondant au triplet choisi est mise à disposition, une nouvelle tentative de détection est faite. Dans le cas où cette dernière est couronnée de succès, la position du véhicule est mise à jour. Le triplet choisi reste valable pendant 100 ms , après quoi, une nouvelle procédure d’optimisation est réalisée. La mise à jour de l’intégrité se fait, elle, dans tous les cas et est discutée à la section 5.4.

Supposons que le véhicule soit initialement à une position connue de manière intègre. Lorsque celui-ci va se déplacer, l’erreur de positionnement, basée sur l’odométrie, va progressivement accumuler une erreur. Celle-ci a pour origine l’erreur sur les paramètres de l’odométrie (erreur sur l’angle de braquage, le nombre d’incrémentes de rotation sur l’arbre moteur), mais également l’erreur de modélisation (glissement des pneus, erreur au niveau des paramètres du train propulseur ou de la géométrie d’Ackermann).

Afin de limiter l’erreur d’estimation de la position du véhicule, on peut songer à utiliser la carte de l’environnement mentionnée ci-dessus, en la supposant parfaitement précise. Le véhicule, sur lequel un LIDAR plan, positionné à l’avant, est installé, peut voir certains murs dans le champ de mesure du capteur. Certains murs ne sont pas visibles, car ils sont positionnés derrière un autre mur, ce que nous appellerons « occultation ». Ceux qui le sont peuvent être observés dans l’espace capteur. Connaissant leur position absolue à partir de la carte, nous pouvons corriger la position du véhicule.

5.3.1 Visibilité des Murs

On suppose initialement que la carte est parfaite et exacte. Ceci signifie que l’on ne changera pas le contenu de la carte, et que l’exactitude des données est garantie. Sachant

cela, on peut donc imaginer de factoriser une fois pour toutes le calcul de visibilité d'un mur par une structure de données.

La manière choisie de faire cela a consisté à calculer une zone à partir de laquelle le capteur LIDAR embarqué sur le véhicule est susceptible de pouvoir observer le mur, et de faire appel à un calcul de recouvrement de surface entre l'ellipse d'incertitude de position du LIDAR (à 1σ) et le polygone en question. La figure 5.5 indique la manière dont le calcul est réalisé, tout comme l'équation (5.12).

$$p_{\text{visibilité}} = \frac{Z}{A} \quad (5.12)$$

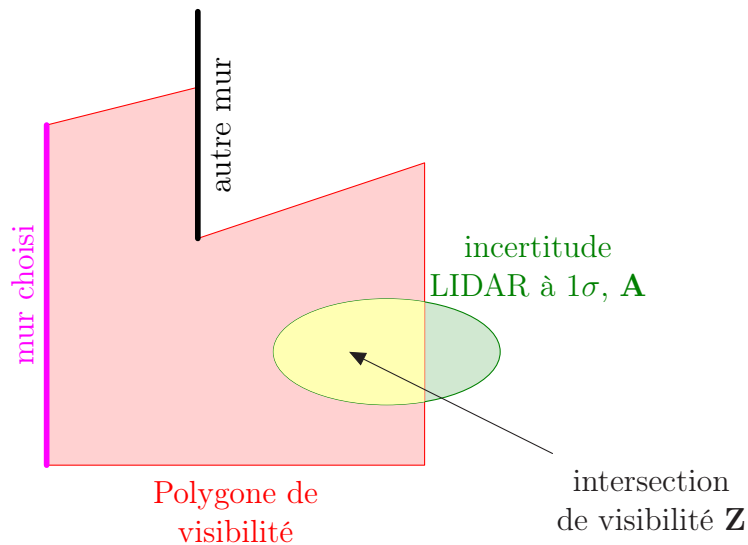


FIGURE 5.5 – Exemple de calcul de taux de visibilité d'un mur par l'intersection de l'ellipse de covariance du capteur LIDAR avec la zone de visibilité d'un mur.

Le polygone de visibilité est calculé comme la zone à partir de laquelle un pourcentage donné du mur en question est visible directement sans être occulté par un autre.

Notons tout de même deux remarques sur ce mode de calcul. Tout d'abord, on pourrait imaginer un mur très court, dont la totalité est visible à l'infini. Pour un capteur LIDAR suffisamment loin, seuls quelques points d'impact sont détectables par ce capteur, ce qui ne suffit pas à détecter un segment droit. Or, la zone telle qu'elle est calculée, pourrait aller à l'infini, et en dehors de porté du capteur. De plus, d'autres éléments occultant le mur peuvent exister sans être des murs (des végétaux, par exemple au centre du rond-point du site PAVIN). Certains murs peuvent être supposés visibles, sans l'être vraiment. Une nouvelle manière de calculer cette visibilité des murs sera introduite au chapitre 8 afin de pallier ces problèmes.

5.3.2 Modèle d'Observation des Murs Fixes

Nous avons vu dans la section 4.7.2 le fonctionnement d'un filtre de Kalman. Le modèle d'évolution utilisé pour le véhicule est le modèle vu à la section 2.3.1. Le seul élément manquant pour réaliser le filtre de Kalman en question est le modèle d'observation.

Rappelons le modèle d'observation du filtre de Kalman, dans l'équation (5.13). La fonction \mathbf{h} associe au vecteur d'état du véhicule \mathbf{X}_Y un vecteur d'observation \mathbf{Y} , qui est

fourni par le détecteur. Le vecteur d'observation \mathbf{Y} , décrit dans l'équation (5.14) est défini par les variables ρ et θ , qui sont explicitées à la figure 5.6. ρ représente la distance du capteur LIDAR à la droite définie par le mur choisi dans le repère du capteur. θ représente quant à lui le paramètre d'orientation du mur par rapport au capteur.

$$\mathbf{Y} = \mathbf{h}(\mathbf{X}_{\mathcal{V}}) \quad (5.13)$$

$$\mathbf{Y} = \begin{pmatrix} \rho \\ \theta \end{pmatrix} \quad (5.14)$$

$$\mathbf{E}_{1/2}^{\text{LIDAR}}(\mathbf{X}_{\mathcal{V}}) = \text{LIDAR} \mathcal{R}_{\text{véhicule}} \cdot \text{véhicule} \mathcal{R}_{\text{Monde}}(\mathbf{X}) \cdot \mathbf{E}_{1/2}^{\text{Monde}} \quad (5.15)$$

$$\mathbf{h}(\mathbf{X}_{\mathcal{V}}) = \text{Polar} \left(\mathbf{E}_1^{\text{LIDAR}}(\mathbf{X}_{\mathcal{V}}), \mathbf{E}_2^{\text{LIDAR}}(\mathbf{X}_{\mathcal{V}}) \right) \quad (5.16)$$

Afin de calculer ces deux valeurs, on convertit tout d'abord chacun des deux bords du mur E_1 et E_2 du système de coordonnées absolues de la carte vers les coordonnées relatives dans l'espace capteur LIDAR comme indiqué dans l'équation (5.15). Puis on applique un paramétrage polaire à la droite passant par ces deux points dans le référentiel LIDAR, ce que réalise l'équation (5.16) et correspond à la figure 5.6.

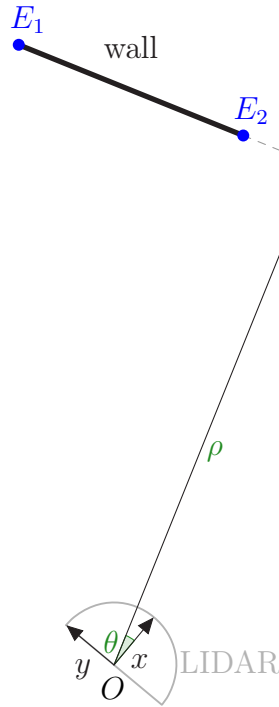


FIGURE 5.6 – Diagramme indiquant les paramètres de l'observation d'un mur faite par le capteur LIDAR. Les paramètres du vecteur \mathbf{Y} dans le repère du LIDAR sont ρ et θ .

Le paramètre $\text{LIDAR} \mathcal{R}_{\text{véhicule}}$ provient de la calibration de la position du capteur LIDAR dans le référentiel du véhicule. $\text{véhicule} \mathcal{R}_{\text{Monde}}$ provient lui de l'état du véhicule. La formule explicite de cette matrice homogène $2D1/2$ est donnée dans l'équation (5.17).

$$\text{véhicule} \mathcal{R}_{\text{Monde}}(\mathbf{X}_{\mathcal{V}}) = \begin{pmatrix} \cos \varphi & \sin \varphi & \succ \cos \varphi \cdot x \succ \sin \varphi \cdot y \\ \succ \sin \varphi & \cos \varphi & \sin \varphi \cdot x \succ \cos \varphi \cdot y \\ 0 & 0 & 1 \end{pmatrix} \quad (5.17)$$

5.3.3 Détection des Murs

Maintenant que le modèle complet du filtre de Kalman a été présenté, nous allons indiquer comment la détection effective du mur est réalisée. Nous partons des données LIDAR, reçues du capteur correspondant. Afin de pouvoir détecter le mur appartenant au triplet choisi, il est indispensable de pouvoir séparer les points de la trame LIDAR susceptibles d'appartenir au mur de ceux qui en sont trop éloignés. Ensuite, un détecteur paramétrique du type *Moindre Carrés Médiants* retourne une estimation des paramètres ρ et θ du mur détecté, ainsi qu'un indicateur de la qualité du fit, le résidu médian. La détection est validée lorsque ce dernier est suffisamment faible.

Une zone de *focalisation* \mathcal{F} , dans laquelle on espère trouver le mur en question, est calculée dans le référentiel du capteur. Les points mesurés par le LIDAR et inclus dans cette zone sont sélectionnés pour le fit moindres carrés médians, les autres étant rejetés. Cette zone est définie comme la zone convexe générée par les deux ellipses d'incertitude représentant les images $\mathbf{E}_{1/2}^{\text{LIDAR}}$ des deux bords du mur choisi, dans le référentiel capteur. La figure 5.7 indique la construction de cette zone.

Comme indiqué précédemment, un fit de type moindre-carrés médians classique est réalisé sur les points sélectionnés de la zone de recherche \mathcal{F} , comme indiqué sur la figure 5.7. A cette étape, il n'est pas certain que le fit obtenu soit correct. Il existe plusieurs cas de réjection d'un tel fit :

- un nombre de points fittés trop faible, qui indique un alignement fortuit dans des données peu nombreuses,
- un résidu médian élevé, qui indique que les points ne sont pas correctement alignés,
- un écart entre le vecteur \mathbf{Y}^- attendu, et le vecteur $\mathbf{Y}^{\text{Mesure}}$ obtenu par le fit, supérieur à 3σ .

Une fois le fit validé, on en extrait ensuite un vecteur $\mathbf{Y}^{\text{Mesure}}$ conforme à la définition 5.14. Le filtre de Kalman de localisation du véhicule utilise alors cette mesure afin de mettre à jour la position du véhicule.

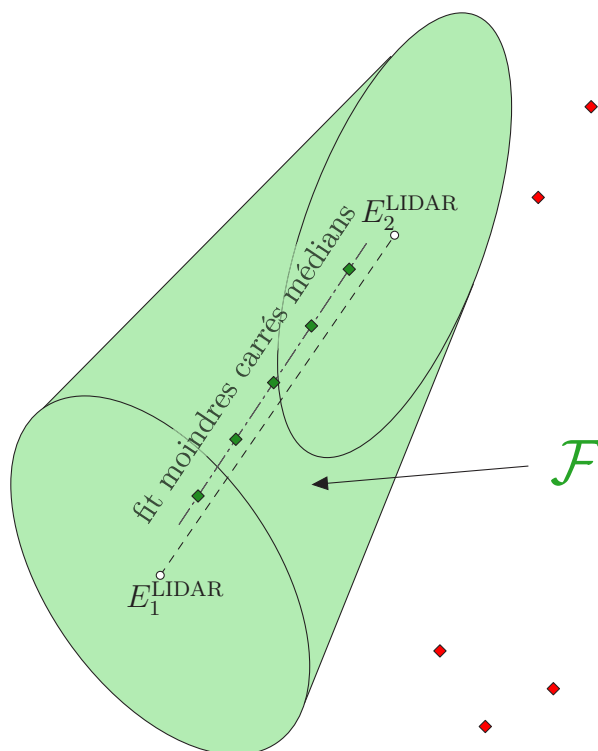


FIGURE 5.7 – Exemple de calcul de la zone de recherche \mathcal{F} associée à un mur, ainsi que de fit associé. E_1^{LIDAR} et E_2^{LIDAR} représentent les bords du mur recherché dans l'espace capteur. Les points de la trame LIDAR appartenant à la zone de recherche sont indiqués en vert, ceux n'y appartenant pas sont dessinés en rouge. En pointillés gris, la position d'une droite correspondant au fit moindres-carrés médians des points sélectionnés.

5.4 Gestion de l'Intégrité

Dans la section 5.3, nous avons présenté de quelle manière l'état du véhicule était mis à jour. Dans cette section, nous présentons par quels moyens la probabilité d'intégrité de l'estimation de position du véhicule, notée $p_{\text{confiance } \nu}$, est calculée. Cette probabilité représente la réponse à la question : *quel est le degré de confiance dans l'estimation retournée par le filtre de Kalman ?* Nous allons tout d'abord définir avec un concept imagé ce que recouvre cette variable, puis dans un second temps, comment elle est calculée.

5.4.1 Concept représenté par la probabilité d'intégrité

Expliquons tout d'abord ce que représente cette probabilité avec un schéma simple, et une formule immédiatement compréhensible. On peut présenter cette idée comme suit :

- la densité de probabilité mixte $\underline{p}^{\text{mix}}$ représente une densité de probabilité dans laquelle :
 - avec une probabilité $p_{\text{confiance } \nu}$, la densité de probabilité est celle que le filtre de Kalman estime \underline{p} ,
 - mais avec une probabilité $1 \succ p_{\text{confiance } \nu}$, nous nous trompons et ne disposons d'aucune information.

- dans le cas où nous ne disposons d’aucune information, la densité de probabilité correspondant à ce minimum d’information qu’est la distribution uniforme, notée $\mathbb{1}(x)$.

Ceci est résumé par l’équation (5.18). Sur la figure 5.8, un exemple de calcul avec $\underline{p} = \mathcal{N}(\dots)$ dans le cas monodimensionnel, et avec un support compact (un intervalle) montre les différentes courbes obtenues pour différentes valeurs de la probabilité $p_{\text{confiance } \nu}$. La zone indiquée en couleur correspond à la partie uniforme de la nouvelle densité de probabilité.

$$\underline{p}^{\text{mix}}(x) = p_{\text{confiance } \nu} \cdot \underline{p}(x) + (1 - p_{\text{confiance } \nu}) \cdot \mathbb{1}(x) \quad (5.18)$$

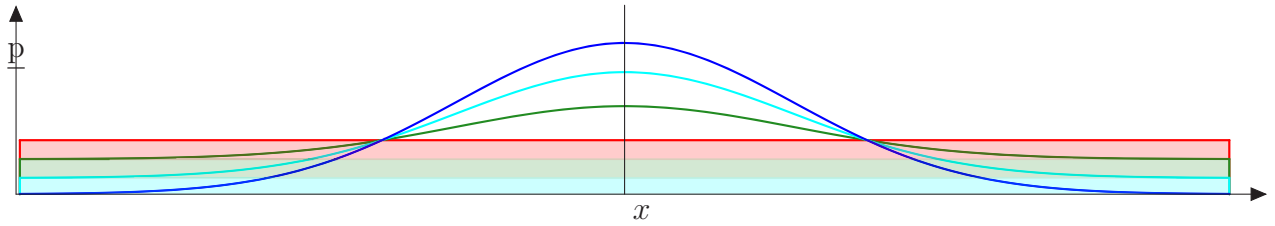


FIGURE 5.8 – Évolution de la forme de la densité de probabilité d’une gaussienne avec une probabilité de confiance variable. La zone remplie en couleur plus claire correspond à la partie densité de probabilité uniforme ajoutée par le complémentaire de la confiance. $p_{\text{confiance } \nu} = \{1, 0.7, 0.35, 0\}$.

Dans notre cas, l’état du véhicule appartient à un vecteur appartenant à $\mathbb{R}^2 \times \mathcal{S}^1$ (les coordonnées x et y appartiennent en première approximation à \mathbb{R} , et l’angle φ à \mathcal{S}^1). Le volume d’une telle variété est donc infini, et la densité uniforme revient à une fonction asymptotiquement identiquement nulle. On a alors l’expression que l’on trouve dans l’équation (5.19).

$$\underline{p}^{\text{mix}}(\mathbf{X}_\nu) = p_{\text{confiance } \nu} \cdot \underline{p}(\mathbf{X}_\nu) \quad (5.19)$$

5.4.2 Réseau Bayésien

Dans le cas du filtre de Kalman, à chaque nouvelle détection de mur, on va re-calculer la position du véhicule par le biais d’une « mise à jour ». Supposons que nous démarrons le processus de localisation avec une probabilité finie d’intégrité. Le résultat de chaque nouvelle tentative de détection d’un mur, dont on connaît le degré d’observabilité et dont on est certain et de l’existence et de la position, doit nous permettre de remettre également la probabilité d’intégrité à jour, par un procédé analogue à la mise à jour du filtre de Kalman.

Nous estimons la probabilité d’un événement booléen, qui peut être soit *vrai* soit *faux*. Le filtre de Kalman étant régi par la règle de Bayes, au niveau continu, et selon un formalisme de chaîne de *Markov* à variables cachées (Hidden Markov Model, HMM), nous cherchons à utiliser un formalisme équivalent, mais dans le domaine discret.

La section 4.8.6 nous présente une telle analogie avec les réseaux bayésiens discrets, présentés à la section 4.8. Nous reprenons ici des travaux antérieurs et en présentons le résultat.

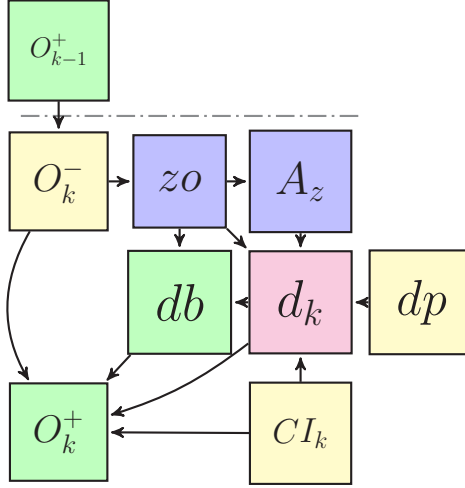


FIGURE 5.9 – Réseau bayésien utilisé pour mettre à jour la probabilité $p_{\text{confiance } \nu}$.

Nous implémentons dans notre travail le réseau bayésien présenté à la figure 5.9. Cette figure est subdivisée en deux parties :

- la partie antérieure, représentant les calculs passés effectués, qui est matérialisée par le nœud O_{k-1}^+ , située au-dessus de la ligne en pointillés,
- le réseau élémentaire, qui est situé en dessous de cette ligne, et en constitue la partie principale.

Ce réseau élémentaire a pour vocation de mettre à jour la confiance antérieure, $p_{\text{confiance } \nu}^-$, représentée par le nœud O_{k-1}^+ , en calculant la confiance postérieure, $p_{\text{confiance } \nu}^+$, représentée par le nœud O_k^+ , basé sur le résultat de détection fourni par le détecteur du triplet choisi, *vrai* ou *faux*, et représenté par le nœud d_k .

Décrivons brièvement la fonction de ces nœuds :

- O_k^- : probabilité égale à la probabilité antérieure de localisation correcte du véhicule, *a-priori*, $p_{\text{confiance } \nu}^-$,
- zo : ce nœud représente la visibilité de l’amer choisi par le détecteur du véhicule,
- A_z : représente la probabilité qu’il y ait au moins un amer (celui qui est recherché ou un autre) dans la zone de focalisation \mathcal{F} ,
- d_k : représente la probabilité de détection d’un amer dans la zone de focalisation \mathcal{F} ,
- dp : représente une valeur booléenne ($p_{dp} = 0$ si *faux*, $p_{dp} = 1$ si *vrai*) correspondant au dernier résultat retourné pour la détection du triplet en question,
- CI_k : coefficient indiquant le degré de corrélation entre le résultat espéré de d_k et le résultat de la précédente tentative de détection,
- db : ce nœud indique le degré de confiance en la détection de l’amer du triplet choisi,
- O_k^+ : représente la probabilité de confiance mise à jour $p_{\text{confiance } \nu}^+$.

Les tables de probabilité conditionnelles originales de ce réseau peuvent être trouvées dans (Aynaud et al., 2014a) et sont recopiées à l’annexe B.1. Certaines tables de probabilités conditionnelles ont été modifiées et sont présentées dans le chapitre 6.

Lors du calcul de la mise à jour, la probabilité du nœud O_{k+1}^- est calculée conformément à (5.20). En fonction du résultat de détection obtenu par le triplet, on infère la probabilité d’intégrité postérieure de O_k^+ sachant la valeur de l’évidence dk . C’est cette valeur qui sert d’*a-priori* pour l’étape $k + 1$.

$$p(O_{k+1}^- = 1) = p(O_k^+ = 1 | dk = \text{détection}) \quad (5.20)$$

L'idée de base, simplifiée, est la suivante. Supposons que notre véhicule soit positionné avec une confiance donnée. Dans le cas où notre véhicule est bien là où il est estimé être, en ayant choisi notre *amer* (un mur donné), nous avons une probabilité donnée par (5.12) de pouvoir l'observer. Mais il est possible que d'autres murs existent dans la zone de recherche, auquel cas, nous avons une plus grande probabilité d'avoir un mur observable dans la zone de recherche.

Sachant l'observabilité d'un mur quelconque dans la zone, nous déduisons la probabilité d'observer un mur, qui est la probabilité relative au résultat que retourne le *détecteur* du triplet. Ce résultat peut être relatif à la détection d'un mauvais mur. Ceci est pris en compte par une probabilité uniforme d'avoir détecté chaque mur de la zone de recherche. Finalement, dans le cas où l'on pense avoir détecté le bon mur, la confiance doit augmenter, sinon, elle est supposée diminuer.

Un dernier cas assez contre-intuitif. Si le véhicule est positionné dans une zone dans laquelle beaucoup de murs compatibles avec les hypothèses existent, et que le détecteur ne détecte rien, la confiance va diminuer. En effet, dans ce cas, même si l'on ne détecte pas le bon mur, on est censé détecter quelque chose. Si ce n'est pas le cas, une hypothèse consiste à dire que le véhicule n'est probablement pas dans une zone contenant beaucoup de murs, donc que la localisation du véhicule est fausse. Une autre hypothèse, non prise en compte dans cette méthode, consiste à expliquer les échecs de détection par la défaillance du capteur associé.

5.4.3 Coefficient de corrélation p_{CI_k}

Le coefficient de corrélation p_{CI_k} permet de ne prendre en compte comme significativement différent un résultat de détection que dans le cas où l'état du véhicule a significativement évolué. L'idée est la suivante. Si le véhicule ne se déplace pas entre deux mesures du même triplet, la visibilité de l'*amer* mur reste inchangée, tout comme la position de ce dernier. Si aucun changement de l'environnement ne se produit, alors très probablement le résultat de détection, **constitué de la détection (détecté ou non-détecté) et de la mesure de l'état détecté (dans le cas détecté)**, du même mur sera semblable. Aucune autre information nouvelle **relative à l'état détecté ou non-détecté** ne sera obtenue. **C'est ce que le coefficient p_{CI_k} permet de contrôler.**

Tandis que, si le véhicule se déplace, la visibilité du mur, sa position dans l'espace capteur vont changer. Le résultat de détection sera potentiellement différent. Et plus le véhicule aura tendance à se déplacer, plus cette corrélation pourra tendre vers zéro.

p_{CI_k} a pour objectif de moduler le résultat p_{d_k} de la manière suivante :

- si $p_{CI_k} = 0$ (aucune corrélation), le résultat p_{d_k} est intégralement nouveau (uniquement calculé en fonction de β , (B.2) et (B.3)),
- si $p_{CI_k} = 1$ (corrélation totale), p_{d_k} est identique à p_{dp} , le résultat de détection vaut 1 (détection à coup sûr) ou 0 (échec à coup sûr) puisque p_{dp} est égal à soit 0 soit 1,
- entre les deux, p_{CI_k} module de manière linéaire entre les deux précédents cas la valeur de p_{d_k} .

La formule donnant ce coefficient p_{CI_k} est donnée à l'équation (5.21). λ est un coefficient d'auto-corrélation dont l'unité est l'inverse d'une distance. Δs est la différence d'abscisse curviligne entre deux positions distinctes du véhicule. La figure 5.10 montre l'évolution de p_{CI_k} en fonction de Δs .

$$p_{CI_k} = e^{-\lambda \cdot |\Delta s|} \quad (5.21)$$

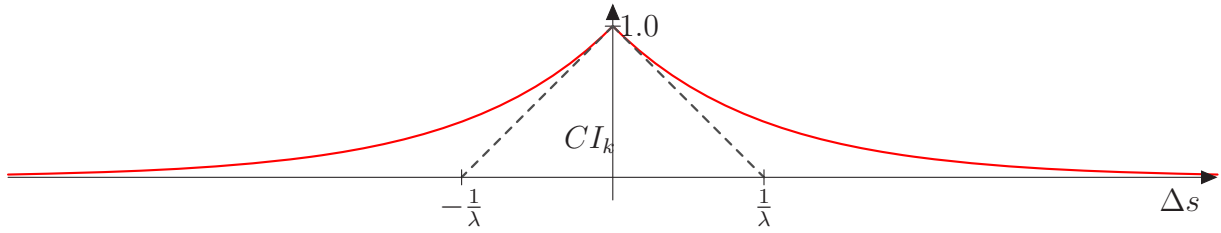


FIGURE 5.10 – Évolution de p_{CI_k} en fonction de Δs .

5.5 Critère de choix du Triplet

Dans la section 5.2.3, nous avons présenté le fonctionnement du choix du triplet. étant donné que dans cette implémentation, le type d'amer ne peut être que *mur*, et le type de capteur *LIDAR*, un seul détecteur peut être choisi.

Dans le cas où un unique capteur *LIDAR* est installé sur le véhicule, le seul élément choisi est le *mur* que l'algorithme va tenter de détecter. Si, par contre, plusieurs capteurs *LIDAR* sont installés, il est évident que l'algorithme va choisir simultanément le *capteur* et le *mur* offrant le meilleur critère.

Dans le processus d'estimation du critère, deux mises à jour « à blanc » sont réalisées. Tout d'abord, on va tenter d'estimer la probabilité que l'on a de faire une détection, *a-priori*. Ensuite, on va réaliser une mise à jour de Kalman « à blanc », en utilisant la valeur prédite du vecteur de mesure \mathbf{Y}^- , **mais en utilisant la covariance de mesure que l'on pense trouver**. En effet, la covariance $\Sigma_{\mathbf{Y}}^{\text{Mesure}}$ est aussi un résultat du processus de détection. On va donc utiliser une matrice de covariance $\Sigma_{\mathbf{Y}^-}$ calculée selon les mêmes paramètres que dans le cas de la mesure, et calculée selon le vecteur \mathbf{Y}^- .

Le critère initial utilisé dans les travaux antérieurs est présenté dans l'équation (5.23).

$$\begin{aligned} \Sigma_{\text{objectif}} & \quad \text{Covariance objectif du processus de localisation} \\ \Sigma^+ & \quad \left\{ \begin{array}{l} \text{Covariance estimée } a\text{-priori pour la mise à jour,} \\ \text{en prenant les paramètres standard de détection du mur.} \end{array} \right. \\ \mathbf{C}_{\text{covariance}} & = \frac{1}{|\Sigma_{\text{objectif}}^{-1} \cdot \Sigma^+ + \mathbf{I}|} \end{aligned} \quad (5.22)$$

$$\mathbf{C}_1(\mathcal{S}, \mathcal{D}, \mathcal{L}) = \overline{p_{CI_k}} \cdot p_{db} \cdot \mathbf{C}_{\text{covariance}} \quad (5.23)$$

Dans l'expression (5.23), contribuent trois facteurs. Tout d'abord $\overline{p_{CI_k}}$ qui est nul lorsque l'on vient immédiatement de faire une détection, et qui tend vers 1 lorsque le véhicule se déplace par rapport à l'instant de la précédente détection. La prise en compte de ce coefficient tend à favoriser les choix de triplets que l'on n'a pas utilisés récemment. Ce coefficient a pour objectif de diminuer le critère dans le cas où il n'y a pas réellement de nouvelle information. Si le véhicule ne s'est pas déplacé, on ne va pas obtenir beaucoup plus d'information par une seconde mesure, car le résultat sera probablement le même.



FIGURE 5.11 – Photographie d'un véhicule VIPALAB3 (ici le n°5, équipé de 5 LIDAR), utilisé sur la plateforme PAVIN.

Le coefficient p_{db} correspond à la probabilité *a-priori* de détecter le mur choisi. Une valeur élevée contribue à rejeter les cas où le mur détecté effectivement n'est pas le mur choisi. Il rejette également les échecs à la détection.

Le troisième critère, $C_{\text{covariance}}$, est expliqué dans (Aynaud et al., 2014a, section III.C). Il correspond à une approximation relative au calcul de la probabilité d'avoir une erreur d'estimation de position du véhicule incluse dans un ellipsoïde de *Mahalanobis* à 1σ de l'estimation faite pour l'erreur attendue, matérialisée par la matrice de covariance d'objectif Σ_{objectif} .

Ce facteur est élevé lorsque la covariance mise à jour obtenue par le test prédictif, Σ^+ , est plus petite dans toutes ses dimensions par rapport à Σ_{objectif} . Moins cela est vrai, plus le coefficient $C_{\text{covariance}}$ sera faible.

5.6 Véhicules de test

Nous présentons ici l'architecture du véhicule de type VIPALAB3 qui a été utilisé afin de valider les différents algorithmes. Ce type de véhicule, représenté sur la figure 5.11, est constitué comme suit :

- Configuration matérielle :
 - caméra PointGrey–DragonFly,
 - LIDAR SICK–LMS151 :
 - * soit 1 LIDAR, un à l'avant au centre (véhicules n°3 et n°4),
 - * soit 5 LIDAR, un à l'avant au centre et un à chaque angle du véhicule (véhicule n°5),
 - récepteur GPS RTK Thalès–Proflex800, *servant de réalité terrain*,
 - récepteur GPS standard Ublox–6T, *installé ponctuellement*,
 - capteur IMU Xsens–MTI, *installé ponctuellement*,
 - mesures de type odométrique (angle de braquage, encodeur sur arbre moteur),
 - routeur au standard IEEE 802.11p « Wifi p »
- Configuration logicielle :
 - système d'exploitation UBUNTU 12.04-32 *bits*,
 - middleware/intergiciel EFFIBOX 3.15¹.

1. <http://www.efdidence.com/>

Ces véhicules se déplacent à une vitesse de $2 \text{ m} \cdot \text{s}^{-1}$ et sont généralement utilisés sur la plateforme d'essais PAVIN. Certains tests ont néanmoins pu être conduits sur le reste du campus des CÉZEAUX.

Conclusion

Dans ce chapitre, nous avons présenté la méthode qui a servi de base à nos travaux. Celle-ci a été entièrement réécrite afin de pouvoir y apporter les modifications relatives aux objectifs décrits dans l'introduction de ce chapitre. Nous avons présenté une méthode basée sur une factorisation de l'état du véhicule en fonction d'une densité de probabilité continue représentant la position du véhicule, et d'une probabilité simple représentant la variable booléenne « confiance » dans la précédente estimation. Une approche focalisante, ne sélectionnant qu'une unique source de données à un instant donné, est utilisée afin de conduire l'estimation des deux parties continues et discrètes de l'état du véhicule. Un formalisme bayésien est utilisé pour les deux parties de l'état du véhicule (filtre de Kalman pour la partie continue, réseau bayésien discret pour la partie discrète). Nous allons maintenant présenter nos contributions principales dans les trois chapitres suivants.

Chapitre 6

Localisation dans une carte imprécise

6.1 Introduction

Dans le chapitre 5, nous avons décrit une méthode de localisation basée sur une carte existante parfaitement précise. Mais dans la réalité, il est rare qu'une carte soit parfaitement précise ou fiable. Il existe toujours un degré d'imprécision (sur la position d'un *amer* de la carte) ou de méconnaissance (l'*amer* recherché existe-t-il ? existe-t-il des amers non cartographiés ?). Ces deux dernières questions seront traitées dans les chapitres 7 et 8. Nous discutons ici la question relative à l'impact d'une incertitude de position d'un amer de la carte sur la navigation d'un véhicule autonome, et la manière d'améliorer cette imprécision.

Analysons tout d'abord l'impact d'une imprécision de chaque amer de la carte sur la localisation du véhicule. Dans le cas d'une mise à jour de l'état du véhicule, à l'incertitude de mesure, Σ_Y^{Mes} , s'ajoute l'incertitude sur la position réelle du mur observé. Ceci a deux conséquences. La première de celles-ci est l'augmentation de l'incertitude du véhicule après mise à jour, par rapport au cas dans lequel la position du mur est parfaitement connue. Cela augmente le risque pour certains murs proches du mur sélectionné d'être confondus avec ce dernier, d'où un risque d'erreur d'association et de localisation accru.

En second, l'état estimé du véhicule dépend alors de celui du mur. Tant que l'on n'utilise pas l'état du véhicule après mise à jour de manière conjointe à l'état du mur, ceci ne pose aucun problème. Mais, dans le cas où les différents amers de la carte sont gérés chacun séparément, par un filtre de Kalman par exemple, dès la première réutilisation d'un amer précédemment choisi (que ce soit pour mettre à jour l'amer, ou localiser à nouveau le véhicule), le phénomène de sur-convergence présenté aux sections 4.7.3 et 4.7.4 se produira. A terme, l'intégrité de la carte peut être mise en danger. Ces deux problèmes doivent être résolus ou contenus afin de permettre au véhicule de se localiser de manière fiable, tout en améliorant la précision de la carte sans la faire diverger.

Nous souhaitons pouvoir corriger la position des amers de la carte lorsque cela est possible, tout en maintenant la cohérence des données de celle-ci, et en évitant les deux précédents écueils. La figure 6.1 présente la nouvelle fonctionnalité que nous souhaitons ajouter au diagramme 5.3 présenté à la section 5.2.5. Lorsque le véhicule détecte un amer, en plus de corriger l'état de ce premier, nous souhaitons également pouvoir corriger l'état de l'amer observé. Sur la figure 6.1, cette fonctionnalité ajoutée correspond au nœud « Nouvelle Position de l'Amer ».

Dans ce chapitre, nous présentons le principe de notre approche de localisation multi-capteurs et de re-précision de carte de l'environnement. Tout d'abord, nous allons présen-

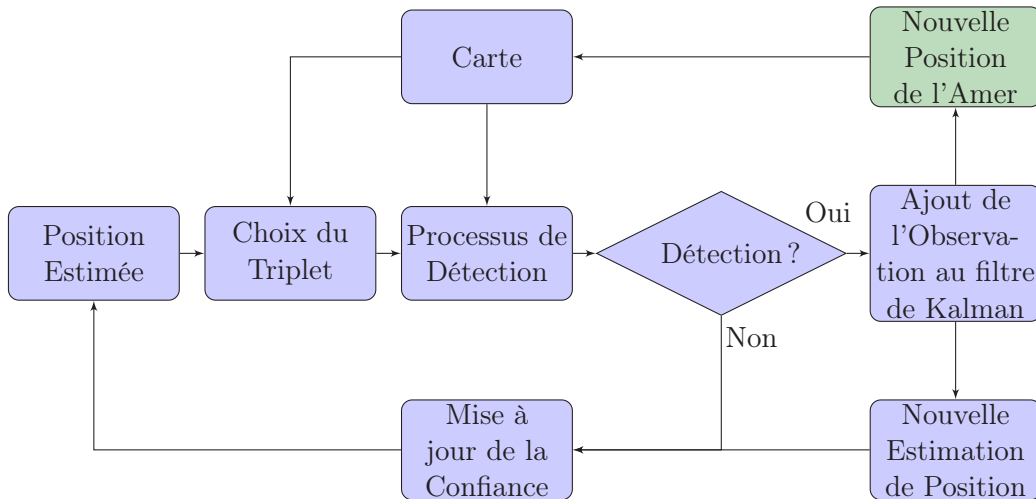


FIGURE 6.1 – Diagramme de fonctionnement de la méthode de localisation de véhicule permettant de re-préciser les amers de la carte.

ter le formalisme que nous avons utilisé afin de gérer la carte de l’environnement, avec le type d’état que nous utilisons pour modéliser les murs. Ensuite, nous expliquons de quelle manière nous prenons en compte les processus de détection des murs. Nous présentons également les simulations nous ayant permis de valider sur table notre solution.

Après avoir expliqué quelques améliorations ponctuelles afin d’optimiser la méthode de localisation, nous présentons les résultats expérimentaux de cette méthode de localisation dans une carte imprécise et d’amélioration de la précision de cette dernière.

6.2 Utilisation du filtre à intersection de covariance

6.2.1 Rappel

Analysons plusieurs types de solutions pour notre problème d’estimation de localisation d’un véhicule dans une carte imprécise. Ici, nous allons nous référer principalement aux sections 2.6, 4.7.3 et 4.7.4 afin d’argumenter sur ces différentes solutions.

Présentons tout d’abord quelques types de formats de covariances pour l’ensemble *véhicule* plus *carte*, schématisées à la figure 6.2. Sur celle-ci, la position d’un unique véhicule est estimée simultanément à celle des multiples amers d’une carte (deux sont représentés).

Sur cette figure, la couleur blanche correspond à des blocs pour lesquels les coefficients sont nuls. Les ligne/colonne indiquées \mathcal{V} correspondent aux coefficients relatifs au véhicule, celles indiquées $\mathcal{L}_1/\mathcal{L}_2$ correspondent aux amers 1 et 2. La figure 6.2a représente la structure obtenue avec un filtre de Kalman global. La figure 6.2b représente celle d’un problème où l’état du véhicule est séparé de celui d’une carte globale. La figure 6.2c représente une solution dans laquelle tous les états sont disjoints.

Nous avons vu dans la section 2.6.2 que l’un des problèmes pour l’utilisation d’un filtre de Kalman centralisé est la mise à l’échelle au niveau du stockage comme du coût d’e mise à jour de la matrice de covariance, qui croissent respectivement comme $O(N^2)$ et $O(N^3)$, N étant le nombre d’éléments estimés par le filtre de *Kalman*. Ceci est expliqué de manière plus détaillée dans (Carrillo-Arce et al., 2013, section I). De plus, il n’est pas

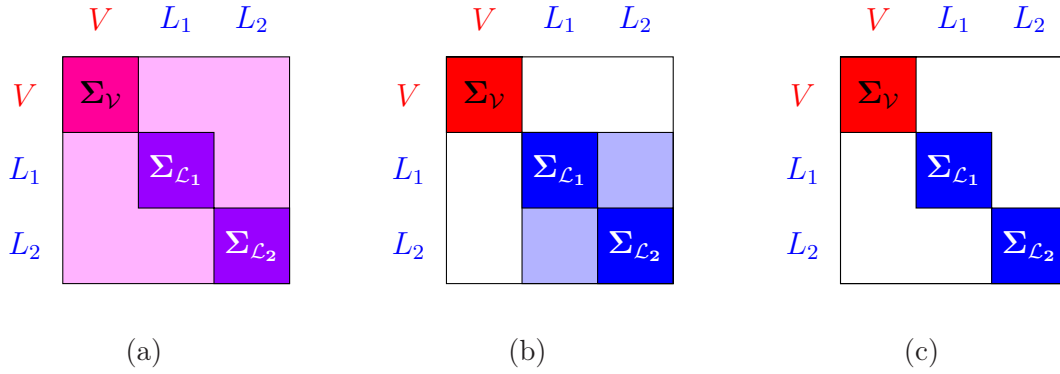


FIGURE 6.2 – Structure des matrices de covariance, pour différents types de problèmes.

souhaitable de devoir posséder l’intégralité de la carte (qui peut être, comme nous l’avons dit, conséquente), à chaque instant.

La matrice de covariance utilisée dans un filtre de Kalman centralisé est une matrice dense, ce qui signifie qu’il n’existe pas *a-priori* sur la structure de la matrice pour simplifier les calculs (voir figure 6.2a). Le coût du stockage et de la mise à jour d’une matrice de covariance pourrait en être diminué en considérant que certains blocs de celle-ci sont nuls, par exemple les blocs indiquant la corrélation entre deux sous-matrices (comme par exemple sur les figures 6.2b ou 6.2c). Dans ce cas, seule une sous-partie de cette matrice de covariance (les blocs diagonaux) a besoin d’être stockée et inversée. Dans le cas où chaque élément est considéré comme séparé, on peut appliquer l’analyse présentée en section 4.7.6.

La figure 6.2a présente une matrice de covariance *dense*, c’est-à-dire qu’il n’existe aucun *a-priori* pour que quelque partie de cette matrice soit nulle. C’est ce type de matrice de covariance qui est utilisé dans un SLAM par filtre de Kalman centralisé. Cette solution est optimale dans le cas où les non-linéarités sont faibles. Mais elle impose également de disposer de l’intégralité de la carte à chaque instant. Une solution possible, la gestion de sous-cartes, se heurte aux problèmes relatifs à leur raccordement.

La figure 6.2b considère que l’erreur liée à l’état du véhicule est décorréliée de celle liée à l’état du reste de la carte statique. En effet, les divers blocs $\Sigma_{\mathcal{V}\mathcal{L}_*}$ représentant les coefficients de corrélation entre l’état du véhicule et celui de chacun des amers sont supposés nuls. Ce type de formulation correspond à une carte dense des amers, les coefficients de corrélations entre amers sont non-nuls.

La figure 6.2c correspond à un problème dans lequel on suppose que chaque élément estimé, véhicule ou amer, est estimé avec une erreur rigoureusement indépendante des autres éléments estimés.

Comme nous l’avons vu aux sections 4.7.3 et 4.7.4, le fait de supposer certains coefficients de covariance croisée $\Sigma_{\mathbf{x}\mathbf{y}}$ dans la matrice de covariance globale nous empêche de pouvoir utiliser des filtres de Kalman en interaction, chacun estimant de manière séparée une sous-partie de la matrice de covariance. Par exemple, en transposant l’exemple de la section 4.7.3 dans le cas $\mathbf{X} = \mathbf{X}_{\mathcal{V}}$ et $\mathbf{Y} = [\mathbf{X}_{\mathcal{L}_1}^\top, \mathbf{X}_{\mathcal{L}_2}^\top]^\top$, le système de la figure 6.2b sur-converge. Un exemple de ce phénomène est présenté à la section 6.2.4.

Nous avons présenté deux paradigmes à ce problème aux sections 4.7.5 et 4.7.7 dédiées respectivement aux *filtre à intersection de covariance* (CI) et *filtre à intersection de covariance partitionnée* (SCI). Nous présenterons les résultats obtenus par chacune et prouve-

rons cette assertion. Par exemple, pour le problème relatif à la figure 6.2b, on peut utiliser comme solution au problème de sur-convergence cité précédemment, la gestion par deux filtres CI ou SCI des interactions des états séparés \mathbf{X}_V et $\mathbf{X}_{\mathcal{L}_1 \ \mathcal{L}_2}$.

De la même manière, le problème présenté sur la figure 6.2c peut être résolu par un filtre CI ou SCI pour chacun des états \mathbf{X}_V et $\mathbf{X}_{\mathcal{L}_*}$. C'est pourquoi nous orientons la résolution de ce problème vers l'utilisation de filtres CI ou SCI pour la gestion de la carte.

6.2.2 Modélisation

Nous avons fait le choix dans notre approche de prendre chaque élément estimé (véhicule, amers...) comme indépendant des autres. Nous aurons à terme à gérer une carte dans laquelle le nombre d'amers sera variable. Pour prendre en compte cette évolution de la manière la plus souple, nous avons pris un modèle dans lequel chaque amer est supposé indépendant des autres. Les suppressions et ajouts d'amers ne nécessitent donc pas de processus complexe.

Dans notre application, un unique véhicule est estimé. Supposons que nous laissons les termes de corrélations non-nuls entre chaque amer et le véhicule. Ceci impliquerait que l'état du véhicule soit corrélé à celui de la carte. Dans ce cas, toute mise à jour de l'état du véhicule modifierait l'état de **toute** la carte. Une estimation de la matrice de covariance représentant la carte complète, pour chaque mise à jour, serait nécessaire. C'est ce que nous souhaitons éviter. L'état et l'erreur liée à l'état du véhicule doivent donc être également décorréliées de celles liées aux états et erreurs du reste de la carte.

Ce raisonnement est généralisable à un nombre quelconque de véhicules, ce qui implique alors un modèle où *tout élément doit être indépendant des autres*, que ce soit un véhicule ou un amer.

Nous prenons donc une modélisation conforme à la figure 6.2c. L'état de l'unique véhicule reste modélisé conformément à l'équation (5.11). L'état de chaque mur est modélisé conformément à la figure 6.3 et à l'équation (6.1). Le mur était modélisé, à la section 5.3.2, par les coordonnées cartésiennes des ses sommets (deux doublets de coordonnées réelles dans le plan). Cette modélisation convient parfaitement à une carte fixe. Par contre, dans le cadre d'une mise à jour de l'état du mur, cette modélisation est problématique.

En effet, le détecteur de murs que nous utilisons, basé sur les données LIDAR, retourne un vecteur de deux coordonnées (le rayon-vecteur ρ , et l'angle d'inclinaison du mur θ , voir la figure 5.6). Ces coordonnées indiquent la **direction** du mur, mais pas son extension dans l'espace (qui indiqueraient les sommets de celui-ci). Nous avons donc utilisé une modélisation ne prenant en compte que la direction–position de la droite correspondant au mur, en supposant les sommets de ce dernier à peu près fixes. Ceux-ci sont projetés sur le mur paramétré pour définir son extension longitudinale. Cette représentation permet de disposer d'une jacobienne d'observation non-singulière. Si l'on avait utilisé les quatre coordonnées cartésiennes du mur dans le paramétrage, la jacobienne d'observation aurait été de rang 2, et donc singulière (l'extension longitudinale du mur étant **inobservable** avec un détecteur du type ρ – θ). Ceci aurait été problématique pour un filtre de Kalman. On définit donc :

$$\mathbf{X}_{\mathcal{L}} = \begin{pmatrix} \rho^C \\ \theta^C \end{pmatrix} \quad (6.1)$$

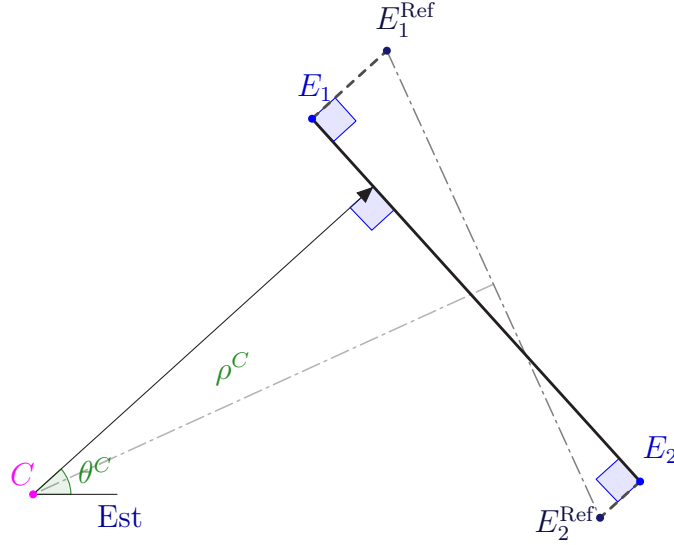


FIGURE 6.3 – Modèle de mur dynamique de la carte.

Les points E_1^{Ref} et E_2^{Ref} représentent une estimation de départ des sommets du mur en question. Les points E_1 et E_2 sont leur projection sur l'estimation courante de la droite représentant la direction du mur.

Le point C sert de référence (point *zéro*) pour les calculs de la fonction d'observation et de sa jacobienne. Cette méthode, qui équivaut à une translation, permet d'éviter les problèmes d'arrondis, tout comme les problèmes de représentation singulière, qui se produisent si E_1 , E_2 et l'origine absolue du système de coordonnées cartographiques (*NTF-LambertII*) sont approximativement alignés.

En utilisant un point C sur la médiatrice de $E_1^{\text{Ref}}-E_2^{\text{Ref}}$, et en supposant que E_1 et E_2 restent approximativement dans l'orientation des précédents, un tel alignement est impossible. Afin de limiter les non-linéarités de la fonction d'observation et les problèmes de linéarisation de la jacobienne, le point C est choisi de telle manière que $\rho \sim E_1^{\text{Ref}} E_2^{\text{Ref}}$.

La carte est alors constituée de la manière indiquée dans les équations (6.2) à (6.5). \mathcal{L}_k représente un amer (un mur), et \mathcal{L} , l'ensemble des amers. \mathcal{V} représente l'état complet du véhicule. \mathcal{M} représente la carte de l'environnement.

$$\mathcal{L} = \{\mathcal{L}_k/k \in \llbracket 1, N \rrbracket\} \quad (6.2)$$

$$\mathcal{L}_k = \{\mathbf{X}_k, \Sigma_{\mathbf{X}_k}\} \quad (6.3)$$

$$\mathcal{V} = \{\mathbf{X}_v, \Sigma_{\mathbf{X}_v}, p_{\text{confiance } v}\} \quad (6.4)$$

$$\mathcal{M} = \{\mathcal{V}, \mathcal{L}_k/k \in \llbracket 1, N \rrbracket\} \quad (6.5)$$

Nous montrerons, théoriquement, et par simulation, qu'un nombre quelconque de véhicules peuvent être pris en compte sans sur-convergence. Dans les expérimentations sur véhicule, un unique véhicule sera utilisé. Il est néanmoins possible d'ajouter un nombre quelconque de véhicules à la carte sans changer la manière dont les estimations sont faites, ni la structure de la carte.

6.2.3 Observation et mise à jour de l'état des amers

Modélisation des poteaux

Les amers de type *poteau* sont utilisés uniquement dans le cadre de simulations. Ils pourraient néanmoins être également intégrés à la méthode de localisation en situation réelle comme un autre triplet relatif au détecteur LIDAR. La modélisation utilisée pour les poteaux peut être trouvée dans les équations (6.6) à (6.9) en relation avec la figure 6.4. La variable ρ représente la distance du capteur au poteau, et θ représente l'azimut du poteau dans le repère du capteur LIDAR. La fonction Polar représente le passage de coordonnées cartésiennes à polaires.

$$\mathbf{X}_{\mathcal{L}_k} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} \quad (6.6)$$

LIDAR $\mathcal{R}_{\text{véhicule}}$ Matrice de transformation géométrique de l'espace véhicule vers l'espace capteur LIDAR.
véhicule $\mathcal{R}_{\text{Monde}}$ Matrice de transformation géométrique de l'espace carte vers l'espace véhicule.

$$\mathbf{X}_{\mathcal{L}_k}^{\text{LIDAR}}(\mathbf{X}_{\mathcal{V}}) = \text{LIDAR } \mathcal{R}_{\text{véhicule}} \cdot \text{véhicule } \mathcal{R}_{\text{Monde}}(\mathbf{X}_{\mathcal{V}}) \cdot \mathbf{X}_{\mathcal{L}_k} \quad (6.7)$$

$$\mathbf{h}(\mathbf{X}_{\mathcal{V}}, \mathbf{X}_{\mathcal{L}_k}) = \text{Polar}(\mathbf{X}_{\mathcal{L}_k}^{\text{LIDAR}}(\mathbf{X}_{\mathcal{V}})) \quad (6.8)$$

$$\mathbf{Y}^{\text{Mes}} = \begin{pmatrix} \rho \\ \theta \end{pmatrix} \quad (6.9)$$

Modélisation des murs

Les amers de type *mur* sont modélisés selon les équations (6.10) à (6.14). Les équations (6.10) et (6.11) représentent la projection de $\mathbf{E}_1^{\text{Ref}}$ et $\mathbf{E}_2^{\text{Ref}}$ sur la droite paramétrée par $\mathbf{X}_{\mathcal{L}_k}$. Les équations (6.12), (6.13) et (6.14) représentent le processus classique d'observation présentée à la figure 5.6.

$$\mathbf{X}_{\mathcal{L}_k} = \begin{pmatrix} \rho_k^C \\ \theta_k^C \end{pmatrix} \quad (6.10)$$

$$\mathbf{E}_{1/2}(\mathbf{X}_{\mathcal{L}_k}) = \text{Project}(\mathbf{E}_{1/2}^{\text{Ref}}, \mathbf{X}_{\mathcal{L}_k}) \quad (6.11)$$

$$\mathbf{E}_{1/2}^{\text{LIDAR}}(\mathbf{X}_{\mathcal{V}}, \mathbf{X}_{\mathcal{L}_k}) = \text{LIDAR } \mathcal{R}_{\text{véhicule}} \cdot \text{véhicule } \mathcal{R}_{\text{Monde}}(\mathbf{X}_{\mathcal{V}}) \cdot \mathbf{E}_{1/2}(\mathbf{X}_{\mathcal{L}_k}) \quad (6.12)$$

$$\mathbf{h}(\mathbf{X}_{\mathcal{V}}, \mathbf{X}_{\mathcal{L}_k}) = \text{Polar}(\mathbf{E}_1^{\text{LIDAR}}(\mathbf{X}_{\mathcal{V}}, \mathbf{X}_{\mathcal{L}_k}), \mathbf{E}_2^{\text{LIDAR}}(\mathbf{X}_{\mathcal{V}}, \mathbf{X}_{\mathcal{L}_k})) \quad (6.13)$$

$$\mathbf{Y}^{\text{Mes}} = \begin{pmatrix} \rho \\ \theta \end{pmatrix} \quad (6.14)$$

Principe de la mise à jour bilatérale

Dans notre manière de gérer les mises à jour, il faut bien distinguer la mise à jour de l'état du véhicule de celle de l'amer choisi. Nous avons choisi pour chaque mesure \mathbf{Y}^{Mes} de procéder à une double mise à jour :

1. d'abord, une mise à jour prenant comme données source la position du véhicule $\mathbf{X}_{\mathcal{V}}$, celle de l'amer k $\mathbf{X}_{\mathcal{L}_k}$, et mettant à jour celle du véhicule à partir de la mesure \mathbf{Y}^{Mes} , et supposant l'état de l'amer k constant, $\mathbf{Y}^{\text{Mes}} \rightarrow \mathbf{X}_{\mathcal{V}}^+$,

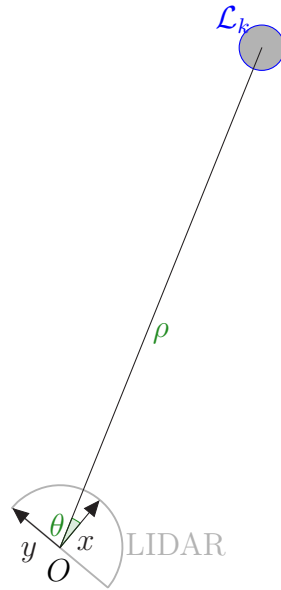


FIGURE 6.4 – Diagramme indiquant les paramètres de l’observation d’un poteau faite par le capteur LIDAR. Les paramètres du vecteur \mathbf{Y}^{Mes} dans le repère du LIDAR sont ρ et θ .

2. puis une mise à jour prenant comme données sources l’état du véhicule mis à jour précédemment $\mathbf{X}_{\mathcal{V}}^+$, celui de l’amer k $\mathbf{X}_{\mathcal{L}_k}$, et mettant à jour l’état du véhicule à partir de la mesure \mathbf{Y}^{Mes} , en supposant l’état du véhicule constant, $\mathbf{Y}^{\text{Mes}} \rightarrow \mathbf{X}_{\mathcal{L}_k}^+$.

Cette manière de procéder permet de mettre à jour, pour chaque mesure de capteur, la position du véhicule comme celle de l’amer détecté. Ainsi, deux éléments sont mis à jour par la même mesure. Cette manière de mettre à jour les mesures est prouvée théoriquement immune au phénomène de sur-convergence présenté à la section 4.7.3 lors de l’utilisation d’un filtre CI ou SCI.

Dans cette section, l’amer précédemment noté \mathcal{L}_k sera simplement noté \mathcal{L} afin d’alléger les notations. Nous allons définir ici le cadre de calcul des mises à jour « bilatérales ». Une fonction d’observation bilatérale \mathbf{h} , dépendant simultanément des valeurs des vecteurs d’état du véhicule \mathcal{V} et de l’amer \mathcal{L} , fournit un résultat de mesure, comme indiqué dans l’équation (6.15). Cette fonction d’observation possède deux jacobiniennes, l’une par rapport à l’état du véhicule (6.16), l’autre par rapport à l’état de l’amer (6.17).

$$\mathbf{Y} = \mathbf{h}(\mathbf{X}_V, \mathbf{X}_L) + \varepsilon \quad (6.15)$$

$$\mathbb{E}(\varepsilon) = \mathbf{0}$$

$$\mathbb{E}(\varepsilon \cdot \varepsilon^\top) = \Sigma_\varepsilon$$

$$\mathbf{H}_V = \left. \frac{\partial \mathbf{h}(\mathbf{Z}, \mathbf{X}_L)}{\partial \mathbf{Z}} \right|_{\mathbf{Z}=\mathbf{X}_V} \quad (6.16)$$

$$\mathbf{H}_L = \left. \frac{\partial \mathbf{h}(\mathbf{X}_V, \mathbf{Z})}{\partial \mathbf{Z}} \right|_{\mathbf{Z}=\mathbf{X}_L} \quad (6.17)$$

Dans un filtre de Kalman, afin de mettre à jour l'état du vecteur, il faut **impérativement** pouvoir formuler le problème comme l'estimation d'un état non observable directement par une mesure ne dépendant que de celui-ci. Or, notre fonction \mathbf{h} dépend de deux vecteurs. Dans le cas d'une mise à jour dans une carte monolithique contenant également l'état du véhicule, on prendrait comme vecteur unique **global** celui issu de la concaténation de l'état de chaque élément contenu dans celle-ci. Dans ce cas, \mathbf{h} ne dépend que d'un unique vecteur $\mathbf{X} \ni \mathbf{X}_V, \mathbf{X}_L \dots$. Le problème de cette stratégie est que le coût de stockage et de calcul augmente **au mieux** comme $O(N^2)$ où N est le nombre d'éléments dans la carte. Par exemple, une matrice de covariance contenant un million de murs (typiquement une grande ville) paramétré comme indiqué dans la section 6.2.2 serait stockée dans un espace mémoire d'au minimum $16 \cdot 10^{12} B = 16 TB = 16 To$. Le produit en temps réel de deux telles matrices, étape nécessaire au filtre de Kalman, serait également problématique.

C'est la raison pour laquelle nous avons choisi de séparer les éléments les uns des autres. Dans le cas de filtres de Kalman séparés, il nous faut une fonction d'observation d'une seule variable. On peut donc penser rendre la seconde variable constante afin d'utiliser à nouveau le filtre de Kalman sur chaque élément séparément. Ceci pose un autre problème. En effet, lorsque l'on utilise un résultat de mesure \mathbf{Y} , il faut être capable d'y associer une matrice de covariance contenant le niveau de bruit. Supposons que l'état de l'amer soit mis à jour, et que l'on rende fixe l'état du véhicule. Dans ce cas, l'observation a pour matrice de covariance Σ_ε . Pourtant, une cause possible pour l'incertitude de la mesure n'est pas prise en compte, la covariance de la variable muette \mathbf{X}_V , qui n'apparaît pas dans la mise à jour mais pour autant n'est pas précisément connue.

La stratégie utilisée couramment dans la communauté scientifique consiste à ajouter à la matrice Σ_ε un terme prenant en compte dans l'espace de mesure l'incertitude de la variable muette. De manière strictement vis à vis du vecteur \mathbf{Y} , ce calcul est juste. Par contre, la « pseudo-fonction monovariable $\mathbf{h}(\mathbf{X}_L)|_{\mathbf{X}_V=cte}$ » devient corrélée avec l'état du véhicule, et ceci brise l'une des hypothèses fondamentales du filtre de Kalman, celle de la blancheur du bruit.

Le filtre SCI est conçu dans le but d'éviter ce type de phénomènes. Présenté à la section 4.7.7, l'état \mathbf{X} sur lequel il opère est le suivant :

- un vecteur d'estimation \mathbf{X}_0 , représentant la moyenne de la distribution,
- une matrice de covariance $\Sigma_{\mathbf{X}_{ind.}}$, représentant la partie du bruit gaussien dont la blancheur est vérifiée,
- une matrice de covariance $\Sigma_{\mathbf{X}_{dep.}}$, représentant la partie du bruit gaussien potentiellement corrélée avec d'autres variables,
- cet ensemble représentant une densité de probabilité $\mathcal{N}(\mathbf{X}; \mathbf{X}_0, \Sigma_{\mathbf{X}_{ind.}} + \Sigma_{\mathbf{X}_{dep.}})$.

Dans la table 6.1, les calculs de trois types de mise à jour sont présentés. La partie relative au filtre de Kalman est triviale, et nous passons directement aux deux variantes des filtres à intersection de covariance. Dans nos calculs, le filtre CI a été implémenté comme un cas particulier du filtre SCI. En effet, dans ce cas, la partie **indépendante** de la matrice de covariance est nulle, la matrice de covariance étant intégralement gérée comme partie **dépendante**. Nous reprenons donc ici la formulation du filtre SCI afin de présenter le cas de mise à jour du filtre CI dans la table 6.1.

Le filtre SCI permet de séparer les parties indépendantes des parties corrélées. On trouve donc logiquement pour la partie **indépendante** la matrice de bruit relative à ε , bruit totalement décorréolé de tout autre processus. Il est donc logique de trouver pour la partie **dépendante** celle relative à la variable muette. Ce sont les trois schémas de mise à jour utilisés pour les trois types de filtres testés en simulation, et dans le cas du filtre SCI, implanté sur véhicule.

Méthode	Type de mise à jour	Valeur de Σ_Y
Kalman	$\mathcal{V} \rightarrow \mathcal{L}$	$\Sigma_\varepsilon + \mathbf{H}_V \cdot \Sigma_{X_V} \cdot \mathbf{H}_V^\top$
	$\mathcal{L} \rightarrow \mathcal{V}$	$\Sigma_\varepsilon + \mathbf{H}_L \cdot \Sigma_{X_L} \cdot \mathbf{H}_L^\top$
CI	$\mathcal{V} \rightarrow \mathcal{L}$	$\mathbf{0}$ ind. $\Sigma_\varepsilon + \mathbf{H}_V \cdot \Sigma_{X_V} \cdot \mathbf{H}_V^\top$ dep.
	$\mathcal{L} \rightarrow \mathcal{V}$	$\mathbf{0}$ ind. $\Sigma_\varepsilon + \mathbf{H}_L \cdot \Sigma_{X_L} \cdot \mathbf{H}_L^\top$ dep.
SCI	$\mathcal{V} \rightarrow \mathcal{L}$	Σ_ε ind. $\mathbf{H}_V \cdot \Sigma_{X_V} \cdot \mathbf{H}_V^\top$ dep.
	$\mathcal{L} \rightarrow \mathcal{V}$	Σ_ε ind. $\mathbf{H}_L \cdot \Sigma_{X_L} \cdot \mathbf{H}_L^\top$ dep.

TABLE 6.1 – Valeur des matrices de covariance utilisées comme *a-priori*.

6.2.4 Validation par simulation de la méthode choisie

Dans cette section, nous allons discuter de la validation de la méthode envisagée pour gérer la carte de l'environnement, au niveau de la sur-convergence et de la méthode de mise à jour choisie. Nous allons tout d'abord décrire les simulations qui nous ont permis de confirmer les propriétés des algorithmes CI et SCI. Puis nous allons présenter et discuter les résultats obtenus par les différentes variantes proposées comme techniques de mise à jour :

1. par un filtre de Kalman
2. par un filtre à intersection de covariance,
3. par un filtre à intersection de covariance partitionnée.

Principe des simulations

- Dans les simulations qui ont été conduites, on distingue deux types d'éléments :
- des véhicules se déplaçant uniformément sur des trajectoires circulaires,
 - des poteaux fixes, considérés comme des amers.

Le but de ces simulations est que des robots mobiles puissent se localiser en se basant sur une carte de l'environnement, et à partir de ce qu'ils peuvent observer. Afin de démontrer les propriétés de « localisation coopérative à plusieurs véhicules », de montrer la stabilité de l'approche à la sur-convergence, et son extensibilité à une flotte, nous avons décidé de rendre possible la simulation de plusieurs véhicules. La modélisation de plusieurs véhicules est en fait similaire à N fois celle d'un seul. Chaque véhicule est capable d'obtenir une mesure sur un élément fixe de la carte, mais aussi sur un autre robot. Ils se localisent **indépendamment** les uns des autres dans l'approche choisie.

Les simulations ont été réalisées dans un « monde plan ». La modélisation des états des divers éléments est conforme aux équation (6.18) et (6.19).

$$\mathbf{X}_v = \begin{pmatrix} x_v \\ y_v \\ \varphi \end{pmatrix} \quad (6.18)$$

$$\mathbf{X}_L = \begin{pmatrix} x_p \\ y_p \end{pmatrix} \quad (6.19)$$

Le déplacement du véhicule est estimé via un vecteur de mesures d'odométrie détaillé dans l'équation (6.20), dans laquelle \dot{s} représente la vitesse instantanée du véhicule, et ω la vitesse angulaire de ce dernier.

$$\mathbf{U}_v = \begin{pmatrix} \dot{s} \\ \omega \end{pmatrix} \quad (6.20)$$

Pour chaque véhicule, plusieurs types de mesures extéroceptives sont disponibles, au choix, et sélectionnées par l'utilisateur :

- mesure « GPS »,
- mesure « magnétomètre »,
- mesure LIDAR.

Les mesures notées « GPS » et « magnétomètre » correspondent de fait à des mesures respectivement du vecteur $[x_v, y_v]^T$ et du scalaire φ . Ces mesures sont de nature absolue. La mesure de type LIDAR est relative, elle peut être appliquée à un poteau, ou bien à un autre véhicule, afin de rendre la simulation plus réaliste.

Les mises à jour dans le cas du LIDAR sont conformes à ce qui est décrit à la section 6.2.3. Dans les mises à jour utilisées, on reprend le principe de mesure décrit dans la figure 6.4, et dont la fonction d'observation est décrite dans les équations (6.6) à (6.9). Les poteaux sont considérés fixes, il n'y a donc pas d'étape de prédiction dans ce cas.

Plusieurs véhicules peuvent se déplacer simultanément. Plusieurs poteaux peuvent être présents. Chacun des véhicules peut être équipé de capteurs différents, et avec des performances paramétrables et différentes. On peut par exemple réaliser une simulation avec :

1. un véhicule équipé d'un capteur GPS,
2. un second équipé d'un capteur magnétomètre,
3. un troisième équipé avec un capteur LIDAR.

La flotte de véhicules simulée est dans ce cas hétérogène.

Dans ces simulations, il est possible, au choix :

1. d'utiliser **toutes** les mesures possibles dans le processus de mise à jour de chaque véhicule,
2. ou bien de ne prendre que la « meilleure mesure » disponible.

Le choix 1 permet par exemple de vérifier la potentielle sur-convergence ou son absence durant la simulation. Le choix 2 permet de tester la qualité en terme de précision de la solution retenue en « ressemblant » à la méthode de choix d'amer et de triplet décrite à la section 5.2.3.

Description des configurations utilisées

Nous mentionnons ici deux simulations qui nous semblent parlantes, bien que de nombreuses autres aient été réalisées.

Dans la première simulation, appelée **simulation 1**, les éléments simulés sont :

- un unique véhicule, équipé de :
 - un capteur GPS, de type RTK ($\sigma_w = 3 \text{ cm}$), le bruit étant corrélé temporellement par un processus AR d'ordre 1 non séparable (matrice non-diagonale) opérant sur le vecteur de bruit $\varepsilon = [\varepsilon_X, \varepsilon_Y]^\top$ de dimension 2, $\varepsilon_{t+\delta t} = \mathbf{F} \cdot \varepsilon_t + \mathbf{w}_{t+\delta t}$, $\delta t = 100 \text{ ms}$, $\mathbf{w}_{t+\delta t}$ étant un bruit blanc d'écart-type σ_w , $F_{XX} = F_{YY} = 0.1$, $F_{XY} = F_{YX} = 0.03$,
 - un capteur LIDAR ($\sigma_\rho = 9 \text{ cm}$, $\sigma_\theta = 2.6^\circ$),
- 4 poteaux distincts,
- une utilisation par l'unique véhicule de toutes les mesures fournies par la détection de chacun des poteaux.

Les poteaux et le véhicule sont initialisés avec une estimation intègre, quoique imprécise.

Dans la seconde, appelée **simulation 2**, les éléments simulés sont :

- le véhicule 1, équipé de :
 - un capteur GPS, de type RTK ($\sigma_w = 3 \text{ cm}$), le bruit étant corrélé temporellement par un processus AR d'ordre 1 non séparable (matrice non-diagonale) opérant sur le vecteur de bruit $\varepsilon = [\varepsilon_X, \varepsilon_Y]^\top$ de dimension 2, $\varepsilon_{t+\delta t} = \mathbf{F} \cdot \varepsilon_t + \mathbf{w}_{t+\delta t}$, $\delta t = 100 \text{ ms}$, $\mathbf{w}_{t+\delta t}$ étant un bruit blanc d'écart-type σ_w , $F_{XX} = F_{YY} = 0.1$, $F_{XY} = F_{YX} = 0.03$,
 - un capteur LIDAR ($\sigma_\rho = 9 \text{ cm}$, $\sigma_\theta = 2.6^\circ$),
- le véhicule 2, équipé de :
 - un capteur GPS, de type EGNOS ($\sigma_w = 26 \text{ cm}$), le bruit étant corrélé temporellement par un processus AR d'ordre 1 non séparable (matrice non-diagonale) opérant sur le vecteur de bruit $\varepsilon = [\varepsilon_X, \varepsilon_Y]^\top$ de dimension 2, $\varepsilon_{t+\delta t} = \mathbf{F} \cdot \varepsilon_t + \mathbf{w}_{t+\delta t}$, $\delta t = 100 \text{ ms}$, $\mathbf{w}_{t+\delta t}$ étant un bruit blanc d'écart-type σ_w , $F_{XX} = -F_{YY} = 0.25$, $F_{XY} = -F_{YX} = -0.20$,
 - un capteur LIDAR ($\sigma_\rho = 4 \text{ cm}$, $\sigma_\theta = 0.05^\circ$),
 - un capteur de type « compas magnétique » ($\sigma_\theta = 0.5^\circ$),
- les mêmes 4 poteaux de la **simulation 1**,
- une utilisation par chacun des véhicules de toutes les mesures fournies par la détection de chacun des poteaux ainsi que de l'autre véhicule.

Le fait que l'on utilise toutes les mesures disponibles pour chaque véhicule permet de se mettre dans les conditions telles que la sur-convergence se manifeste. En effet, lorsque deux véhicules se détectent l'un-l'autre, ou détectent simultanément le même poteau, une

information commune est partagée et transmise lors de mises à jour ultérieures à d'autres éléments simulés.

D'autres simulations réalisées avec 4 véhicules ont été réalisées et confirment les résultats auxquels nous arrivons et qui sont décrites dans ce qui suit.

Résultats chiffrés

Nous avons comparé, pour les deux configurations données précédemment, les résultats donnés par trois simulations :

- avec un filtre de Kalman,
- avec un « filtre à Intersection de Covariance »-CI,
- avec un « filtre à Intersection de Covariance Partitionné »-SCI.

Un tableau de résultats synthétisant la simulation par les valeurs moyennes des erreurs est présenté ci-après, dans les tables 6.2 et 6.3.

TABLE 6.2 – Résultats de **simulation 1** avec un unique véhicule

Type de Simulation	Item	Erreur de position (m)	Distance de Mahalanobis
Kalman	Poteau 1	0.005	0.49
	Poteau 2	0.015	0.85
	Poteau 3	0.038	1.86
	Poteau 4	0.012	0.66
	Véhicule 1	0.016	0.91
CI	Poteau 1	0.020	0.10
	Poteau 2	0.023	0.11
	Poteau 3	0.033	0.13
	Poteau 4	0.074	0.43
	Véhicule 1	0.022	0.41
SCI	Poteau 1	0.015	0.24
	Poteau 2	0.021	0.38
	Poteau 3	0.019	0.26
	Poteau 4	0.036	0.47
	Véhicule 1	0.018	0.77

Gestion de la carte par filtre de Kalman

Dans le cas du filtre de Kalman, la simulation à un unique véhicule semble fonctionner. En effet, la sur-convergence ne se manifeste pas de manière évidente. En effet, les distances de Mahalanobis pour chacun des éléments cartographiés ne s'approchent pas de 3. Néanmoins, la valeur de celle du poteau n°3 est légèrement élevée.

La simulation à deux véhicules montre un profil très différent. Les distances de Mahalanobis de tous les éléments simulés, à l'exception du véhicule 1, dépassent 10, et une grande partie d'entre eux dépassent largement cette valeur. On peut estimer la probabilité résiduelle d'intégrité grâce à la probabilité associée du test du χ^2 , mais on remarquera simplement que pour un vecteur de dimension 2 ou 3, la vraisemblance correspondant à une valeur de χ^2 supérieure à 10σ est très faible ($1.929 \cdot 10^{-22}$ pour un vecteur de dimension 2, et $1.554 \cdot 10^{-21}$ pour un vecteur de dimension 3).

TABLE 6.3 – Résultats de **simulation 2** avec deux véhicules

Type de Simulation	Item	Erreur de position (m)	Distance de Mahalanobis
Kalman	Poteau 1	0.034	170
	Poteau 2	0.027	31
	Poteau 3	0.036	39
	Poteau 4	0.042	59
	Véhicule 1	0.013	1.29
	Véhicule 2	0.028	11
CI	Poteau 1	0.021	0.15
	Poteau 2	0.049	0.32
	Poteau 3	0.054	0.32
	Poteau 4	0.041	0.26
	Véhicule 1	0.022	0.42
	Véhicule 2	0.039	0.38
SCI	Poteau 1	0.005	0.15
	Poteau 2	0.010	0.17
	Poteau 3	0.012	0.18
	Poteau 4	0.006	0.18
	Véhicule 1	0.018	0.78
	Véhicule 2	0.020	0.55

Le véhicule 1 dispose de valeurs de position dont la covariance est faible en regard des autres types de mesures, tous véhicules confondus. Ceci peut efficacement limiter la sur-convergence pour ce véhicule. À l’opposé, le véhicule 2 dispose de mesures moins précises. Malgré la présence d’une mesure absolue de type GPS, la position estimée de ce dernier sur-converge.

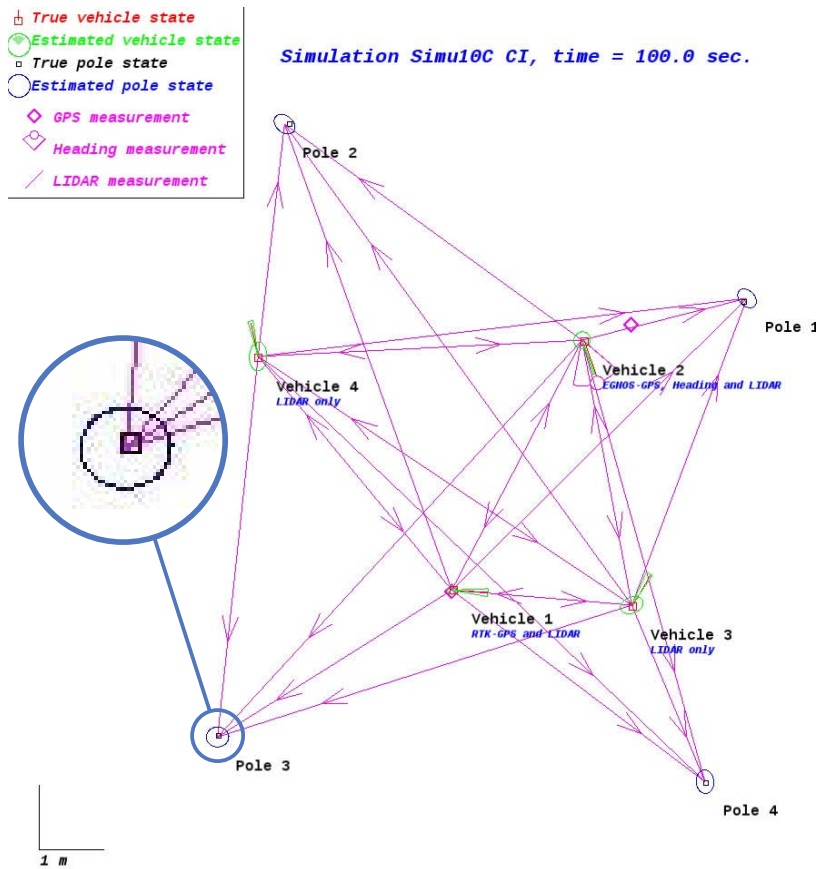
En observant les valeurs des distances de Mahalanobis pour les poteaux, elles sont toujours supérieures à 3, ce qui est hautement irréaliste. On observe sur les simulations que les matrices de covariance des poteaux tendent vers la matrice nulle au fur et à mesure du temps qui passe, alors que l’erreur de position reste finie. Ceci fait diverger la distance de Mahalanobis vers des valeurs extrêmement hautes, ce qui ne représente pas correctement la réalité (« non-physique »).

La conclusion logique de ceci est que le phénomène de sur-convergence se manifeste sur les amers de type poteau par rebond entre les mises à jour des deux véhicules. La carte devient progressivement de plus en plus fautive, et l’intégrité de la localisation des deux véhicules en pâtit en étant faussée de manière consécutive. Comme la mesure de position absolue de type GPS du véhicule 2, ce dernier ne peut corriger aussi bien que le véhicule 1 l’erreur d’estimation que la carte provoque. Ceci explique alors que l’estimation du véhicule 1 diverge moins que celle du véhicule 2.

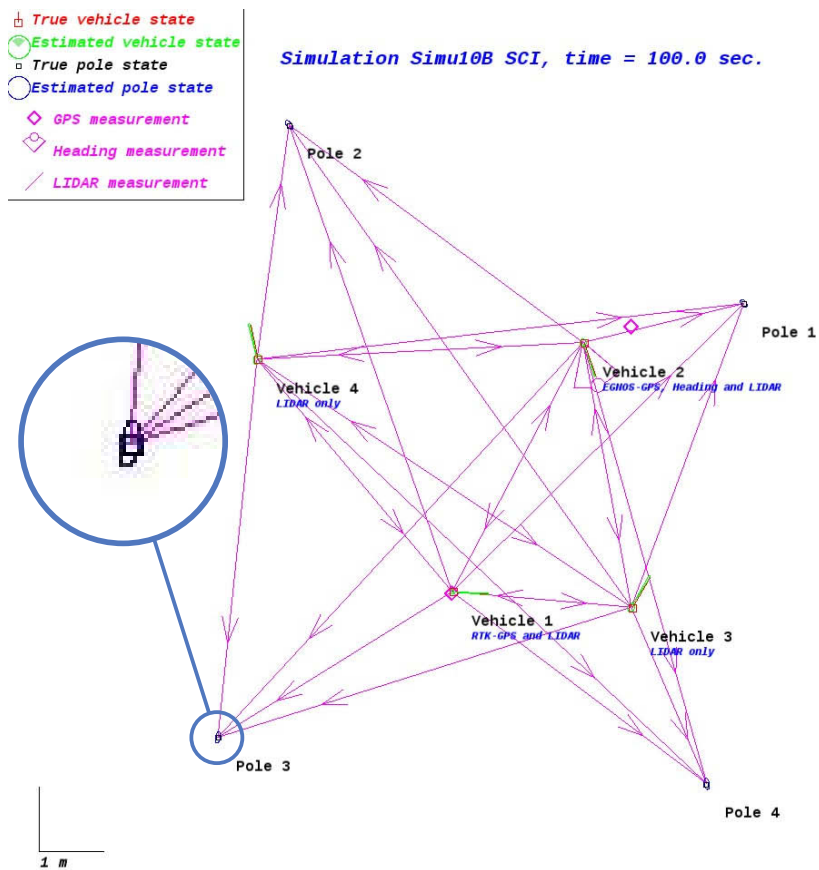
Nous montrons par ce simple exemple que l’utilisation de filtres de Kalman de manière couplée, opérés sans prendre de précautions particulières, provoque inévitablement le phénomène de sur-convergence.

Gestion de la carte par filtres CI/SCI

Dans le cas du filtre à Intersection de Covariance, on observe que la carte reste intègre. Les distances de Mahalanobis sont en moyenne toutes inférieures à l'unité, que ce soit pour un filtre CI ou SCI. De plus, qualitativement, les covariances estimées restent « stables » dans le temps dans les deux cas après que la carte aie finalement convergé. Une comparaison des résultats obtenus par les deux filtres au niveau de la carte est présentée figure 6.5 pour le cas de quatre véhicules. On observe sur la figure 6.5a, correspondant au filtre CI, que les ellipses de covariance pour les poteaux sont relativement « larges ». Sur la figure 6.5b, filtre SCI, les ellipses sont beaucoup plus petites (de la taille des symboles marquant les poteaux).



(a) avec un filtre CI



(b) avec un filtre SCI

FIGURE 6.5 – Comparaison de la carte obtenue après 100 s de temps de simulation dans le cas d’une simulation à quatre véhicules. Une ellipse de covariance prise comme exemple est agrandie sur chaque figure (zoom $\times 4$) afin de permettre la comparaison.

On observe également que la moyenne des erreurs absolues de position au niveau des poteaux est plus faible dans le cas du filtre SCI comparé au filtre CI. Ceci est logique. En effet, dans le filtre CI, l'intégralité de la matrice de covariance était fusionnée de manière conservative. Dans le cas du filtre SCI, la matrice de covariance est séparée en deux parties, dont l'une est fusionnée « d'une manière plus optimiste », plus proche d'une mise à jour de Kalman. La covariance ainsi obtenue est alors, bien que toujours conservative, plus resserrée que celle obtenue par CI.

Pour conclure sur ce point, les filtres CI/SCI apparaissent supérieurs au filtre de Kalman du point de vue de la sur-convergence, et immunes de ce problème. De plus, en termes d'erreur absolue sur les amers, comme sur l'obtention d'une covariance la plus étroite possible, le filtre SCI est meilleur que le filtre CI. C'est donc ce filtre que nous choisissons d'implanter afin de rendre la carte de l'environnement dynamique, et ainsi permettre d'affiner la position des amers tout en permettant la localisation intègre dans une carte imprécise.

6.3 Critère de sélection des amers proposé

Dans la section 5.2.3, nous avons vu que tous les triplets potentiellement détectables étaient testés selon un « critère », et qu'à chaque instant, c'est le triplet le meilleur selon ce critère qui était sélectionné pour être observé. Lorsque les données du capteur sélectionné arrivent, on tente de détecter l'amer du triplet choisi. Cette mesure sert alors à corriger l'état du véhicule.

6.3.1 Pourquoi utiliser la notion d'entropie ?

Ainsi qu'expliqué à la section 4.3, le concept d'entropie est relié à ce que l'on appelle l'*information*, en tant que concept humain. Une des propriétés de la fonction d'entropie de *Shannon*, définie dans (4.19), est qu'elle permet de définir un **ordre absolu** en terme de « quantité d'information », soit apportée, soit retirée.

On peut donc imaginer choisir un critère, comme celui défini dans l'équation (5.9), qui soit basé sur la meilleure mise à jour, en termes d'*information ajoutée*. Nous allons proposer dans ce qui suit le critère en question.

6.3.2 Remarque sur le calcul de l'entropie sur une carte

Ici, une remarque est faite concernant la manière de calculer l'entropie globalement pour toute la carte. Supposons que nous voulions prendre en compte une carte globale du type de celle mentionnée à la figure 6.2a. Le calcul de l'entropie, notée $H(\dots)$, sur la carte globale, est alors indiqué dans l'équation (6.22).

$$\Sigma_{\text{carte}} = \begin{pmatrix} \Sigma_{\mathcal{V}} & \Sigma_{\mathcal{V}\mathcal{L}_1} & \Sigma_{\mathcal{V}\mathcal{L}_2} \\ \Sigma_{\mathcal{V}\mathcal{L}_1}^\top & \Sigma_{\mathcal{L}_1} & \Sigma_{\mathcal{L}_1\mathcal{L}_2} \\ \Sigma_{\mathcal{V}\mathcal{L}_2}^\top & \Sigma_{\mathcal{L}_1\mathcal{L}_2}^\top & \Sigma_{\mathcal{L}_2} \end{pmatrix} \quad (6.21)$$

$$H(\text{carte}) = \frac{1}{2} \log |2\pi e \Sigma_{\text{carte}}| \quad (6.22)$$

$$H(\mathbf{X}_{\mathcal{V}}) = \frac{1}{2} \log |2\pi e \Sigma_{\mathcal{V}}| \quad (6.23)$$

$$H(\mathbf{X}_{\mathcal{L}_1}) = \frac{1}{2} \log |2\pi e \Sigma_{\mathcal{L}_1}| \quad (6.24)$$

$$H(\mathbf{X}_{\mathcal{L}_2}) = \frac{1}{2} \log |2\pi e \Sigma_{\mathcal{L}_2}| \quad (6.25)$$

$$H(\text{carte}) \neq H(\mathbf{X}_{\mathcal{V}}) + H(\mathbf{X}_{\mathcal{L}_1}) + H(\mathbf{X}_{\mathcal{L}_2}) \quad (6.26)$$

si les termes croisés entre $\mathbf{X}_{\mathcal{V}}$, $\mathbf{X}_{\mathcal{L}_1}$ ou $\mathbf{X}_{\mathcal{L}_2}$ sont non-nuls

Les entropies séparées de $\mathbf{X}_{\mathcal{V}}$, $\mathbf{X}_{\mathcal{L}_1}$ ou $\mathbf{X}_{\mathcal{L}_2}$ ne sont pas égales quand on les somme à l'entropie totale, dans le cas général. En effet, on peut réécrire cette somme comme indiqué dans l'équation (6.27). On observe que dans cette formule, la somme des entropies individuelles est égale à l'entropie de la carte **en mettant tous les termes de corrélation à 0**. Ce qui équivaut à supposer les divers éléments indépendants, et l'information mutuelle nulle entre deux paires quelconques de variables parmi les trois.

$$\begin{aligned} H(\mathbf{X}_{\mathcal{V}}) + H(\mathbf{X}_{\mathcal{L}_1}) + H(\mathbf{X}_{\mathcal{L}_2}) &= \frac{1}{2} \log (|2\pi e \Sigma_{\mathcal{V}}| \cdot |2\pi e \Sigma_{\mathcal{L}_1}| \cdot |2\pi e \Sigma_{\mathcal{L}_2}|) \\ &= \frac{1}{2} \log \left| 2\pi e \begin{pmatrix} \Sigma_{\mathcal{V}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\mathcal{L}_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Sigma_{\mathcal{L}_2} \end{pmatrix} \right| \end{aligned} \quad (6.27)$$

Ceci n'est évidemment pas vrai avec une modélisation par un filtre de Kalman central, dans laquelle les termes de corrélation sont en général non-nuls. Mais dans une modélisation de type CI ou SCI conforme à la figure 6.2c, l'inégalité de l'équation (6.26) devient une égalité stricte. Dans ce cas, on peut alors séparer l'entropie de la carte en une somme d'entropies de toutes les composantes de la carte.

6.3.3 Nouveau critère

Dans cette section, nous présentons le critère que nous avons utilisé pour les tests en simulation, et qui a servi de base au critère utilisé dans les expérimentations sur véhicule. Celui-ci sert à choisir l'amer que l'on va tenter de détecter, en fonction des résultats du processus de détection que l'on tente de prévoir. Ceci peut par exemple prendre en compte la probabilité *a-priori* de détecter l'amer, ainsi que l'amélioration *moyenne* de la précision de l'état décrivant les éléments (véhicule, amer...) que l'on *espère obtenir*. Ce critère est **calculé sans aucune connaissance des données du capteur**.

Avant d'analyser les données capteur, il faut en effet choisir le triplet, donc le capteur dont on va analyser les données. Afin de réaliser ce choix, on va calculer l'information H^* que l'on pense obtenir par détection à partir des paramètres liés au triplet (le capteur, le

Description	Probabilités associées	Covariance après mise à jour <i>a-priori</i>	Description des paramètres
Détection du bon amer	$p(dk = 1, db = 1)$	$\Sigma_{\mathbf{X}_V}^*$	$\tilde{\mathbf{Y}}$ tel que $\mathbf{X}_V^* = \mathbf{X}_V^-$
Détection du mauvais amer	$p(dk = 1, db = 0)$	$\Sigma_{\mathbf{X}_V}^* + \Delta\mathbf{X} \cdot \Delta\mathbf{X}^\top$	$\Delta\mathbf{X}$: écart entre \mathbf{X}_L pour l'amer choisi et \mathbf{X}_B pour l'amer détecté
Echec de détection	$p(dk = 0)$	$\Sigma_{\mathbf{X}_V}^-$	–

TABLE 6.4 – Différentes possibilités de détection *a-priori*.

détecteur et l'amer à détecter). À partir du moment où ce triplet est choisi, et seulement après, seules les données provenant du capteur choisi vont être analysées par le détecteur choisi, afin de tenter de détecter l'amer en question.

Nous nous sommes inspirés pour la définition de celui-ci (et nous n'avons aucune revendication à ce propos) de travaux tels que (Caglioti, 2001a;b), (Czarnetzki et al., 2010) et (Julian et al., 2012) bien que nous ayons appliqué ceux-ci d'une manière légèrement différente.

Lorsque le choix du triplet est opéré, on ne dispose que de prédictions afin de réaliser les calculs, puisque c'est le résultat du processus de sélection qui fournira les données et les algorithmes afin de les analyser.

On procède donc au calcul d'amélioration *a-priori* de l'information, en fonction de marginalisations elles aussi *a-priori*. L'entropie de la distribution donnée par l'estimation *a-priori* est par définition celle d'une distribution normale, présentée dans l'équation (6.28).

$$H^-(\mathbf{X}_V^-) = \frac{1}{2} \log |2\pi \cdot e \cdot \Sigma_{\mathbf{X}_V}^-| \quad (6.28)$$

Tentons d'estimer les différentes possibilités, en fonction du résultat de détection. Celles-ci sont indiquées dans la table 6.4. Rappelons le rôle des variables binaires probabilistes db et dk . dk représente le résultat de détection, 1 s'il est couronné de succès, et 0 s'il échoue. db , quant à lui, représente le fait d'avoir détecté le bon amer s'il est à 1, la détection d'un mauvais amer ou un échec de détection s'il est à 0.

Détaillons cette table. Le cas « Détection du bon amer » correspond au cas recherché. Dans ce cas, l'information apportée dépend de la mesure $\tilde{\mathbf{Y}}$ et de sa covariance. On détermine $\tilde{\mathbf{Y}}$, la mesure « à blanc » que l'on estime pouvoir obtenir, de manière que \mathbf{X}_V reste inchangé par la mise à jour. La matrice de covariance $\tilde{\Sigma}_{\mathbf{Y}}$ est la source de l'apport d'information au véhicule.

Le cas « Détection du mauvais amer » traite des possibilités d'ambiguïté. C'est à dire que, dans le processus de détection, la mesure retournée par le détecteur pour l'amer \mathcal{L} correspond en fait à celle de l'amer \mathcal{B} . Dans ce cas, la mesure donne une estimation fautive, qui provoque le décalage du vecteur d'état du véhicule d'une valeur $\Delta\mathbf{X}$. L'estimation est alors biaisée. Le terme $\Delta\mathbf{X} \cdot \Delta\mathbf{X}^\top$ correspond alors au terme d'erreur ajouté par l'ambiguïté.

Le cas « Échec de détection » n'apporte aucune information au niveau état du véhicule, et la covariance de l'estimation de celui-ci reste alors inchangée.

Le même tableau peut être présenté concernant l'état de l'amer choisi. Nous ne le ferons pas en raison de la similitude avec le tableau 6.4. Nous voyons dans ce découpage un cas intéressant dans lequel l'ambiguïté est traitée comme un problème informationnel *a-priori* et non un problème d'association *a-posteriori*.

On peut donc écrire l'entropie estimée selon la table 6.4 comme indiqué sur l'équation (6.30).

$$H_{\mathcal{V}}^* = H(p(dk, db)) + H(\mathbf{X}_{\mathcal{V}}^* | p(dk, db)) \quad (6.29)$$

$$\begin{aligned}
H_{\mathcal{V}}^* = & \\
& \left. \begin{aligned}
& -p(dk = 1, db = 1) \cdot \log(p(dk = 1, db = 1)) \\
& -p(dk = 1, db = 0) \cdot \log(p(dk = 1, db = 0)) \\
& -p(dk = 0) \cdot \log(p(dk = 0))
\end{aligned} \right\} H(p(dk, db)) \\
& \left. \begin{aligned}
& +p(dk = 1, db = 1) \cdot \frac{1}{2} \log |2\pi \cdot e \cdot \Sigma_{\mathbf{X}_{\mathcal{V}}}^*| \\
& +p(dk = 1, db = 0) \cdot \frac{1}{2} \log |2\pi \cdot e \cdot (\Sigma_{\mathbf{X}_{\mathcal{V}}}^* + \Delta \mathbf{X} \cdot \Delta \mathbf{X}^{\top})| \\
& +p(dk = 0) \cdot \frac{1}{2} \log |2\pi \cdot e \cdot \Sigma_{\mathbf{X}_{\mathcal{V}}}^-|
\end{aligned} \right\} H(\mathbf{X}_{\mathcal{V}}^* | p(dk, db))
\end{aligned} \quad (6.30)$$

L'écart entre l'entropie *a-priori* $H_{\mathcal{V}}^-$ et celle calculée par anticipation de la mise à jour $H_{\mathcal{V}}^*$ est indiqué dans la formule (6.32).

$$\begin{aligned}
H_{\text{gain pour } \mathcal{V}} &= H_{\mathcal{V}}^- - H_{\mathcal{V}}^* \\
&= -H(p(dk, db)) \\
&+ p(dk = 1, db = 1) \cdot \frac{1}{2} [\log |2\pi \cdot e \cdot \Sigma_{\mathbf{X}_{\mathcal{V}}}^-| - \log |2\pi \cdot e \cdot \Sigma_{\mathbf{X}_{\mathcal{V}}}^*|] \\
&+ p(dk = 1, db = 0) \cdot \frac{1}{2} [\log |2\pi \cdot e \cdot \Sigma_{\mathbf{X}_{\mathcal{V}}}^-| - \log |2\pi \cdot e \cdot (\Sigma_{\mathbf{X}_{\mathcal{V}}}^* + \Delta \mathbf{X} \cdot \Delta \mathbf{X}^{\top})|]
\end{aligned} \quad (6.31)$$

$$\quad (6.32)$$

Analysons ce qui se produit dans l'un des trois cas suivants :

- $p(dk = 1, db = 1) = 1$,
- $p(dk = 1, db = 0) = 1$,
- $p(dk = 0) = 1$.

Dans ces trois cas, $H(p(dk, db))$ est nul. On observe que dans le cas où $p(dk = 1, db = 1) = 1$, c'est à dire que l'on est certain de détecter le bon amer, le gain est équivalent au gain en entropie différentielle opéré sur la bonne détection.

Si par contre, $p(dk = 1, db = 0) = 1$, c'est à dire que l'on est certain de détecter un amer qui n'est pas le bon, le gain pourra être défavorable. En effet, si le paramètre $\Delta \mathbf{X}$ est de forte valeur, le déterminant de la partie *après mise à jour virtuelle* pourra être élevé, potentiellement plus grand que la partie *a-priori*. Le gain pourra être négatif. Ceci se produit si l'on détecte un amer qui n'est pas celui recherché et dont les paramètres de mesure sont très différents de ceux de l'amer recherché. Ce terme gère donc naturellement le problème de l'ambiguïté.

Il se peut que, malgré une erreur sur l'amer détecté, les paramètres mesurés soient quasiment les mêmes que ceux que l'on aurait eu avec l'amer choisi (ambiguïté entre deux

murs parfaitement alignés). Dans ce cas, le terme $\Delta \mathbf{X}$ sera faible, et la matrice $\Sigma_{\mathbf{X}_v}^*$ sera alors faiblement pénalisée.

Dans le cas où $p(dk = 0) = 1$, on est certain de ne pas détecter. Dans ce cas de figure, aucune information n'est ajoutée à la connaissance que l'on a sur le véhicule. Le même calcul peut être réalisé sur l'amer mis à jour, conformément aux équations (6.33) et (6.34).

$$H_{\mathcal{L}}^* = H(p(dk, db)) + H(\mathbf{X}_{\mathcal{L}}^* | p(dk, db)) \quad (6.33)$$

$$H_{\text{gain pour } \mathcal{L}} = H_{\mathcal{L}}^- - H_{\mathcal{L}}^* \quad (6.34)$$

On peut alors énoncer notre nouveau critère, nommé **C3** comme indiqué dans la formule (6.35). Le symbole $\mathcal{S} \in \mathcal{V}$ signifie que l'on itère sur tous les capteurs extéroceptifs installés sur le véhicule. $\mathcal{D} \sim \{\mathcal{S}; \mathcal{L}\}$ signifie que le détecteur recherche **des amers du type de \mathcal{L}** , et qu'il travaille à partir de **données fournies par \mathcal{S}** .

$$\mathbf{C3} \begin{pmatrix} \mathcal{S} \\ \mathcal{D} \\ \mathcal{L} \end{pmatrix} = \begin{cases} H_{\text{gain pour } \mathcal{L}} + H_{\text{gain pour } \mathcal{V}} & \text{si l'amer } \mathcal{L} \text{ est mis à jour (mur, poteau)} \\ H_{\text{gain pour } \mathcal{V}} & \begin{cases} \text{si l'amer } \mathcal{L} \text{ ne l'est pas} \\ \text{(champ magnétique, GPS...)} \end{cases} \end{cases} \quad (6.35)$$

Comme indiqué dans (6.35), le terme $H_{\text{gain pour } \mathcal{L}}$ n'est pris en compte que dans le cas où l'amer est dans la carte et peut être mis à jour. Ceci implique que l'amer est entré dans la carte avec un paramétrage tel que les mesures faites par le véhicule peuvent le mettre à jour. Dans le cas contraire (amer n'appartenant pas à la carte, amer virtuel...), ce terme est retiré du critère.

6.3.4 Simulation qualitative du nouveau critère

Avant de tester le critère proposé dans la formule (6.35) en vraie grandeur, un test par simulation, basé sur des données générées selon la simulation décrite à la section 6.2.4 a été réalisé.

Il compare deux modalités de prise en compte des amers :

- toutes les détections d'amers et de véhicules autres sont fusionnées avec l'état estimé du véhicule,
- seule la meilleure source d'information, au sens de **C3**, est prise en compte.

Il est précisé que dans cette simulation, ni l'ambiguïté, ni l'échec de détection ne sont simulés. On ne tient compte de l'échec de détection qu'au niveau du critère, mais les tentatives de détection sont toujours couronnées de succès. Le problème de l'ambiguïté n'est absolument pas pris en compte dans cette simulation, c'est à dire que l'on suppose $p(dk = 1, db = 0) = 0$.

Néanmoins, cette manière de procéder peut nous informer sur la perte d'information que l'on peut observer à partir d'une situation de type *bottom-up*, vers une situation du type *top-down* basée sur le critère **C3**. La table 6.5 compare l'utilisation d'une seule source de mesure sélectionnée par le critère **C3**, notée « meilleure » dans la table, et la prise en compte de toutes les informations disponibles, notée « toutes ».

On observe que l'erreur de position, notée « Erreur euclidienne », est du même ordre de grandeur lorsque l'on prend la meilleure par rapport à toutes. Le rapport, logiquement en faveur de toutes, oscille entre 1 (poteau 2) et 4.5 (poteau 4) pour le filtre CI, et 1.5 (véhicule 2) et 10 (poteau 4) pour le filtre SCI. Le filtre SCI fusionne en général mieux avec

Type de filtre	Item	Erreur euclidienne (m)		Distance de Mahalanobis	
		toutes	meilleure	toutes	meilleure
CI	Poteau 1	0.027	0.070	0.19	0.49
	Poteau 2	0.045	0.050	0.30	0.27
	Poteau 3	0.061	0.052	0.37	0.26
	Poteau 4	0.027	0.119	0.20	0.57
	Véhicule 1	0.022	0.025	0.41	0.47
	Véhicule 2	0.038	0.047	0.36	0.42
	Véhicule 3	0.027	0.089	0.21	0.41
	Véhicule 4	0.030	0.070	0.25	0.36
SCI	Poteau 1	0.007	0.040	0.17	0.43
	Poteau 2	0.008	0.069	0.18	0.40
	Poteau 3	0.010	0.057	0.15	0.44
	Poteau 4	0.006	0.060	0.15	0.89
	Véhicule 1	0.018	0.019	0.77	0.78
	Véhicule 2	0.020	0.032	0.54	0.65
	Véhicule 3	0.019	0.049	0.53	0.63
	Véhicule 4	0.020	0.061	0.53	0.62

TABLE 6.5 – Comparaison entre la prise en compte de toutes les mesures et la sélection d’une seule par le critère **C3**

toutes les informations que le filtre CI, ce qui est normal puisque le filtre SCI fusionne de manière plus optimale (prise en compte de la partie indépendante du bruit). Par contre, les résultats obtenus entre CI et SCI sont comparables pour le critère **C3**.

La distance de Mahalanobis est relativement meilleure dans le cas où toutes les mesures sont prises en compte par rapport au cas où seule la meilleure l’est, mais on n’observe aucune différence entre CI et SCI. L’intégrité est dans tous les cas conservée. Malgré une perte modérée de précision pour le filtre SCI, l’utilisation d’un critère de sélection de type **C3** permet un maintien de la stratégie de fusion comparativement à la prise en compte de toutes les informations.

6.4 Améliorations spécifiques

6.4.1 Améliorations du réseau bayésien

Les modifications au réseau bayésien sont indiquées dans l’annexe B.2, et concernent les termes $\bar{\alpha}_1$, $\bar{\alpha}_2$ et γ . Les coefficients du type α_* représentent les probabilités de faux positif, c’est à dire la probabilité de détecter quelque chose là où aucun amer n’est présent. Le coefficient γ représente la probabilité d’avoir un amer dans la zone dans laquelle le capteur en recherche un, lorsque l’amer recherché n’est pas visible (position estimée du véhicule incorrecte, amer partiellement invisible). Si x est une probabilité, la probabilité notée \bar{x} représente la valeur $1 - x$. La structure du réseau bayésien n’est pas modifiée.

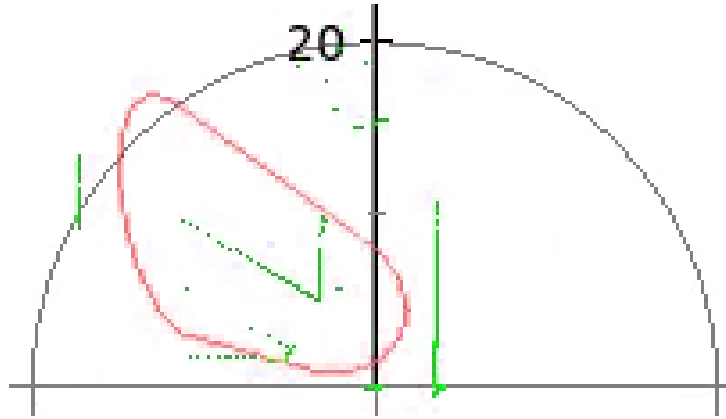


FIGURE 6.6 – Exemple de problème de détection de mur. Dans la zone de recherche en rouge, au minimum 3 murs sont présents.

6.4.2 Modification du détecteur de murs

La technique de Moindres-Carrés Médiens est basée sur une méthode de Moindres-Carrés itérée sur un grand nombre d'hypothèses. Il est capable de détecter de manière robuste des primitives, comme des droites ou des cercles en présence de points hors-distribution. Dans le cas où une unique primitive est présente dans la zone de recherche, le problème de l'ambiguïté des amers détectés ne se pose pas, et la détection se déroule généralement bien. Ceci est le cas avec la méthode décrite dans le chapitre 5.

Dans le cas de la gestion d'une carte incertaine, l'ajout d'une imprécision importante sur la position d'un amer augmente la surface de recherche dans laquelle le mur attendu est recherché, incluant possiblement d'autres murs ayant des paramètres différents. Dans ce cas, plusieurs murs peuvent, en fonction du nombre de points appartenant à chacun dans la zone de recherche, être retournés par le processus de détection. Si le mur détecté ne correspond pas, il y a échec de détection. Un exemple de tel cas est présenté sur la figure 6.6. Ceci peut provoquer des absences de détection, alors que le mur recherché est bien présent dans la zone. Cette situation est appelée « faux négatif ». Ce taux devient suffisamment élevé pour empêcher un fonctionnement normal de l'algorithme de localisation. Une amélioration des performances du détecteur de murs est donc indispensable afin de pouvoir exécuter la localisation dans une carte imprécise.

Deux modifications ont été réalisées afin d'améliorer cette situation :

1. focaliser la détection du mur sur les paramètres correspondant aux paramètres attendus du mur recherché,
2. améliorer le mode de calcul du résidu médian.

L'item n°1 consiste à ne sélectionner comme résidu médian que les jeux de paramètres compatibles, selon l'inéquation (6.36). Ce qui signifie que, non seulement on ne recherche que des segments issus de couples de point situés dans la zone de recherche (précédemment notée \mathcal{F}), mais aussi que seuls ceux dont les paramètres de position compatibles avec le mur recherché sont pris en compte comme couples valables. La validité de ces couples est estimée sur les paramètres couplés de rayon-vecteur ρ et d'orientation θ simultanément par l'intermédiaire de la matrice de covariance de l'erreur attendue, notée $\Sigma_{\mathbf{Y}_{\text{prédit}}}$. Dans la plupart des cas, cette matrice dispose d'un coefficient croisé ρ - θ non-nul, qui rend ces deux paramètres dépendants l'un de l'autre.

$$d_M(\mathbf{Y}_{\text{mesuré}}; \mathbf{Y}_{\text{prédit}}, \Sigma_{\mathbf{Y}_{\text{prédit}}}) \leq d_{Max} \quad (6.36)$$

L'item n°2, quant à lui, a pour objectif de calculer le résidu médian basé sur la distribution la plus proche des paramètres prédits, c'est à dire dont les résidus sont les plus proches de 0. Un problème se pose. Comme la zone de recherche est grande, les alignements fortuits dont les paramètres correspondent peuvent être nombreux. Cela ne gêne pas le fit primaire en lui-même, mais le seuil de sélection utilisé pour le fit secondaire peut être fortement sur-estimé en raison de la queue de distribution, et le fit échouera alors.

Un nouveau critère de calcul a donc été utilisé, et est décrit dans l'équation (6.37). On définit le nouveau résidu r_{Otsu} selon (6.37) :

$$\begin{aligned} r_k, k \in \llbracket 1 \dots N \rrbracket & \text{ résidu d'indice } k \text{ classés par ordre croissant} \\ \sigma_{th} & \text{ écart-type des résidus inférieurs au seuil } th \\ \sigma_{r_k} & \text{ écart-type des résidus inférieurs au résidu médian } k \\ r_{Otsu} & = r\left(\arg \max_{k' \in \llbracket 1 \dots N \rrbracket} \frac{k'^2}{\sigma_{r_{k'}}}\right) \end{aligned} \quad (6.37)$$

Dans le cas d'une distribution uniforme (dans le cas, par exemple d'un fond d'« outlier », lorsque l'indice k est élevé, côté droit de la figure 6.7b), on s'attend à ce que $\sigma_{r_k} \sim r_k \propto k$, le coefficient de proportionnalité étant élevé. La fonction à optimiser de (6.37) est alors faible. Dans le cas où r_k est de l'ordre du pied de la gaussienne (au niveau du trait rouge sur la figure 6.7b), lorsque le pic de résidus centrés sur 0 est significatif, $\sigma_{r_k} \sim \sigma_{pic}$, qui est notablement plus faible. La valeur de la fonction à optimiser est plus élevée. Enfin, lorsque k est proche de 0, le terme $\frac{k^2}{\sigma_{r_k}}$ tend vers 0 (comme la fonction d'erreur erfc). On s'attend donc que, lorsque le pic de résidus centré sur 0 est significatif par rapport au bruit de fond de résidus provenant d'« outliers », l'expression $\frac{k^2}{\sigma_{r_k}}$ puisse atteindre un maximum vers la base de celui-ci, typiquement au niveau du trait rouge, à gauche sur la figure 6.7b. Ce nouveau calcul est néanmoins mis en échec lorsque ce pic est insuffisamment marqué, ce qui peut correspondre à l'absence de l'amer recherché, mais pas obligatoirement.

Un exemple de distribution de résidus médians a été simulé et est présenté sur la figure 6.7. Sur la figure 6.7a, une vue des points LIDAR simulés est présentée. Un bruit de l'ordre de 30 cm a été ajouté sur toutes les structures paramétriques (murs et poteau). Un buisson a été simulé par un tirage selon une loi uniforme. Celui-ci a été positionné dans le prolongement du mur recherché (et noté « végétal »). Sur la figure 6.7b, le calcul de la distribution du résidu relatif au mur recherché est affiché. On observe bien la double distribution. Le calcul trouve un seuil au bas de la gaussienne du mode le plus proche du résidu nul, tout en éliminant la queue de distribution.

Ce nouveau détecteur a été intégré à la version opérationnelle du code. Sur le jeu de données testé en re-jeu, l'amélioration de résultat de détection, en utilisant une carte imprécise à environ 70 cm, est dans un rapport de 1 : 3, ce qui est significatif. Les performances de ce nouveau détecteur sont presque aussi bonnes que celles du détecteur de Moindres-Carrés Médiens implanté sur la version du logiciel décrite au chapitre 5, alors que la zone de recherche peut être jusqu'à 100 fois plus grande, et qu'une ambiguïté de détection existe très souvent (plusieurs murs peuvent se trouver dans la zone de recherche).

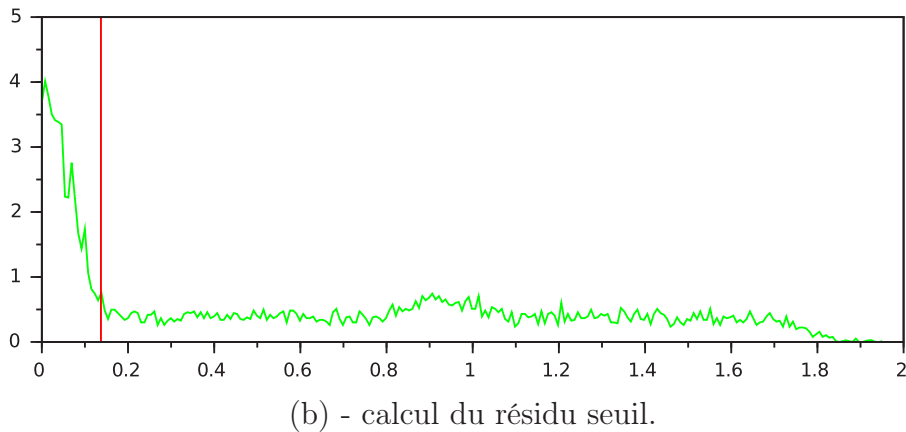
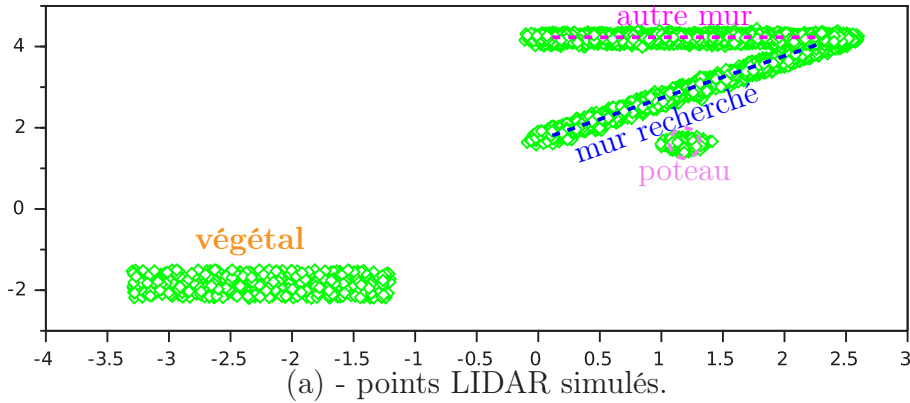


FIGURE 6.7 – Exemple de calcul de seuil selon la méthode proposée. Sur la figure (6.7b), le seuil calculé par la méthode est indiqué en rouge.

6.4.3 Modification du critère théorique proposé

Complément du critère C3 par rapport à \overline{CI}_k

Dans cette section, nous revenons au critère **C1** (vu au chapitre 5) vis-à-vis de la probabilité \overline{CI}_k . Dans ce critère, le terme \overline{CI}_k correspond à la probabilité de disposer d'une information **nouvelle** provenant d'un amer déjà observé. Or le critère **C3** ne prend pas **directement** ce terme en compte, mais indirectement, via la loi de probabilités jointe dk, db . Ce terme ayant un intérêt **conceptuel**, puisqu'il indique un caractère de **nouveauté** à l'observation potentielle, il devrait être pris en compte dans le calcul effectif.

En effet, il est clair, d'après l'expression obtenue dans (5.23) que les choix dans lesquels \overline{CI}_k est proche de zéro sont pénalisés. En effet, ces derniers n'apportent pas de nouvelle information par rapport aux précédentes détections. Or, ils ne le sont pas dans l'expression (6.34). Il est pourtant souhaitable de prendre en compte ce terme \overline{CI}_k , qui permet par ailleurs de rejeter des amers récemment choisis (dont l'information a déjà été prise en compte récemment) pour préférer des amers plus neutres (n'ayant pas été choisis récemment).

Il existe deux manières de formuler le problème :

- soit en marginalisant le gain d'entropie par rapport au nœud CI_k (au sens de l'entropie conditionnelle),

— soit en pénalisant le critère **C3** par rapport à $\overline{CI_k}$.

C'est cette dernière solution que nous avons choisi. Étant donné que l'entropie est basée sur une expression logarithmique par rapport aux probabilités, le facteur de pénalisation indiqué dans la formule (6.38) a été choisi.

$$P_{\overline{CI_k}} = \log(p_{\overline{CI_k}}) \quad (6.38)$$

Complément du critère **C3** par rapport à la dimension de l'ellipse de recherche des murs

Le critère théorique présenté à la section 6.3.3 a été modifié dans un but purement d'ingénierie. En effet, dans certains cas, l'ellipse de covariance de l'état d'un mur projeté dans l'espace capteur LIDAR pouvait être très anisotrope. L'un des axes de cette ellipse de covariance pouvait avoir une dimension dix fois moindre que l'autre.

Ceci, malgré l'amélioration précédente au détecteur de *Moindres-Carrés Médiants*, provoquait des problèmes de détection. Une des méthodes afin de limiter cet état de fait consiste à choisir une mise à jour opérée par un amer ne faisant pas augmenter de manière trop importante le grand axe d'une ellipse de covariance donnée. Celui-ci n'est autre que la valeur d'écart-type associée à la direction du grand-axe de la matrice de covariance, c'est à dire la racine carrée de la valeur propre maximale de la matrice de covariance représentant l'incertitude d'un des bords du mur recherché dans l'espace LIDAR.

Afin de rendre la mesure de ce phénomène indépendante de celle de l'amer choisi, un point test positionné à une distance de d_0 mètres dans la direction avant a été choisi. Une valeur de d_0 égale à 7 mètres a été utilisée. L'équation donnant la matrice de covariance associée à cette ellipse d'incertitude est présentée dans la formule (6.39).

$$\begin{aligned} \mathbf{T}_{\text{véhicule}} &= \begin{pmatrix} d_0 \\ 0 \end{pmatrix} \quad \begin{array}{l} \text{vers l'avant du véhicule} \\ \text{vers la gauche du véhicule} \end{array} \\ \mathbf{m}(\mathbf{X}_{\mathcal{V}}) &= {}_{\text{LIDAR}}\mathcal{R}_{\text{véhicule}} \cdot \text{véhicule} \mathcal{R}_{\text{Monde}}(\mathbf{X}_{\mathcal{V}}) \cdot \mathbf{T}_{\text{Monde}} \\ &= \mathbf{T}_{\text{LIDAR}} \\ \mathbf{M}(\mathbf{X}_{\mathcal{V}}) &= \left. \frac{d\mathbf{m}(\mathbf{X})}{d\mathbf{X}} \right|_{\mathbf{X}=\mathbf{X}_{\mathcal{V}}} \\ \Sigma_{\text{test}} &= \mathbf{M}(\mathbf{X}_{\mathcal{V}}) \cdot \Sigma_{\mathbf{X}_{\mathcal{V}}} \cdot \mathbf{M}(\mathbf{X}_{\mathcal{V}})^{\top} \end{aligned} \quad (6.39)$$

L'idée prise en compte consiste à pénaliser le critère **C3** par un terme de même dimensionnalité et comparable à ce dernier. Le terme relatif à la covariance de l'estimation de position du véhicule apparaît dans le critère **C3** comme la moitié de la somme des logarithmes des valeurs propres de la matrice de covariance, voir équation (6.40).

$$\begin{aligned}
\Sigma &= \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix} \\
H_\Sigma &= \frac{1}{2} \log \left| 2\pi \cdot e \cdot \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix} \right| \\
&= \frac{1}{2} [\log (2\pi \cdot e \cdot \sigma_1^2) + \log (2\pi \cdot e \cdot \sigma_2^2) + \log (2\pi \cdot e \cdot \sigma_3^2)] \\
&= \log (\sqrt{2\pi e} \cdot \sigma_1) + \log (\sqrt{2\pi e} \cdot \sigma_2) + \log (\sqrt{2\pi e} \cdot \sigma_3) \\
&= \log (\sigma_1) + \log (\sigma_2) + \log (\sigma_3) + \frac{3}{2} \log (2\pi e) \tag{6.40}
\end{aligned}$$

L'idée ici est de compenser le terme relatif à la plus grande valeur propre de l'ellipse de covariance d'un point donné, projetée dans l'espace capteur du LIDAR opérant la détection. Comme dans le gain en entropie **C3**, ce terme logarithmique est en produit avec chaque probabilité $p(dk = 1, db = 1)$, $p(dk = 1, db = 0)$ et $p(dk = 0)$, et que l'on veut pénaliser ce choix d'amer de la même manière dans tous les cas, on a l'équation (6.41).

$$\begin{aligned}
P_{\lambda_{\max} \text{ LIDAR}} &= -p(dk = 1, db = 1) \cdot \log \lambda_{\max \text{ LIDAR}} \\
&\quad - p(dk = 1, db = 0) \cdot \log \lambda_{\max \text{ LIDAR}} \\
&\quad - p(dk = 0) \cdot \log \lambda_{\max \text{ LIDAR}} \\
&= -\log \lambda_{\max \text{ LIDAR}} \tag{6.41}
\end{aligned}$$

Dans cette expression, le terme constant $\frac{1}{2} \log (2\pi e)$ a été éliminé.

Critère utilisé

Le critère utilisé dans les différents cas est conforme à ce qui est indiqué dans la formule (6.42).

$$\mathbf{C3}_{\text{opérationnel}} = \begin{cases} \mathbf{C3} + P_{CI_k} + P_{\lambda_{\max} \text{ LIDAR}} & \text{pour le triplet basé } \mathbf{GPS} \\ \mathbf{C3} + P_{CI_k} + P_{\lambda_{\max} \text{ LIDAR}} & \text{pour le triplet basé } \mathbf{magnétomètre} \\ \mathbf{C3} + P_{CI_k} + P_{\lambda_{\max} \text{ LIDAR}} & \text{pour le triplet basé } \mathbf{LIDAR-mur} \end{cases} \tag{6.42}$$

Il est à noter que les valeurs de CI_k sont différentes entre les différents triplets choisis. En effet, le terme λ de l'équation (5.21), rappelée ci-dessous, varie en fonction du type de mesure.

$$p_{CI_k} = e^{-\lambda|\Delta s|}$$

Par exemple, l'information donnée par une mesure **GPS** ne reste pas valable la même durée que celle relative à un mur mesuré grâce à un **LIDAR**. D'autre part, nous avons majoré la valeur maximale de la corrélation par une valeur finie $p_{CI_k}^{\text{sensor}}$ (inférieure strictement à l'unité) et paramétrable. L'utilité de cette dernière est de prendre en compte

des cas de détections successives retournant un résultat différent (*vrai*, puis *faux* ou inversement) avec une valeur de p_{CI_k} égale aux arrondis près à 1 (corrélation totale, deux résultats successifs doivent retourner le même résultat). Ce type de situations provoque des résultats incorrects (*Inf.* ou *NaN*) au niveau du réseau bayésien estimant l'intégrité. De sorte qu'on obtient l'équation (6.43) pour les coefficients CI_k utilisés dans (6.38) et (6.42). Dans cette équation (6.43), $|\Delta s|$ représente l'abscisse curviligne parcourue par le véhicule depuis la dernière sélection du même triplet que le triplet courant.

$$p_{CI_k}(\Delta s) = \begin{cases} p_{CI_k}^{\text{GPS}} \cdot \exp(-\lambda^{\text{GPS}} \cdot |\Delta s|) & \text{pour un capteur GPS} \\ p_{CI_k}^{\text{mag.}} \cdot \exp(-\lambda^{\text{mag.}} \cdot |\Delta s|) & \text{pour un capteur magnétomètre} \\ p_{CI_k}^{\text{LIDAR}} \cdot \exp(-\lambda^{\text{LIDAR}} \cdot |\Delta s|) & \text{pour un capteur LIDAR} \end{cases} \quad (6.43)$$

6.4.4 Estimation des temps de calcul *Kalman* vs *SCI*

Nous avons précisé dans la section 4.7.6 que le temps de calcul de la mise à jour pouvait être plus important pour un filtre à *Intersection de Covariance* que pour un filtre de *Kalman*. Dans ce cas, il est nécessaire de quantifier plus finement le temps de calcul utilisé par ces deux types de filtre.

La table 6.6 présente des résultats d'exécution obtenus pour une séquence de données de 60 s, avec une donnée proprioceptive et une donnée extéroceptive toutes les 100 ms. Le temps de calcul est pris brut au niveau processeur, sur un unique thread. La machine sur laquelle le test a été conduit est équipée d'un processeur CORE™i5-3210M-2.5GHz.

Le ratio entre les deux filtres est de l'ordre de 2 : 1, avec un filtre à intersection de covariance deux fois plus lent. Le nombre choisi d'étapes de « dichotomie » par la méthode du nombre d'or est ici de 10, ce qui équivaut à une précision sur le terme ω d'environ 8‰ par rapport à l'unité. Rappelons que le résultat de mise à jour d'un filtre SCI est l'optimum en termes de déterminant de la matrice de covariance fusionnée en fonction d'une variable muette ω , comprise entre 0 et 1 (se référer aux sections 4.7.5 et 4.7.7 pour plus de détails sur la variable ω). Elle représente le « degré de mélange » entre les variables fusionnées.

Il est à remarquer que même le filtre le plus lent des deux met environ 56 ms de temps d'exécution pour 60 s de données en temps réel, soit un ratio de 1 : 1000. L'intégration sur une période de 100 ms de la trajectoire d'un véhicule mettrait donc moins de 100 μs. La puissance CPU utilisée par ces deux versions de filtre n'est donc pas majoritaire dans le cumul obtenu sur l'application réelle.

Temps d'exécution (ms)	
Filtre de Kalman	Filtre à Intersection de Covariance
28.4	56.4

TABLE 6.6 – Résultats comparatifs de temps d'exécution entre le filtre de *Kalman* et le filtre à *Intersection de Covariance* pour une séquence de données de 60 s avec des mesures toutes les 100 ms.

Ces résultats sont suffisants pour permettre une exécution en temps réel sans risque de lourdeur au niveau charge CPU.

6.5 Expérimentation

Dans cette section, nous allons décrire les différentes configurations qui ont été testées dans le cadre de ces travaux. Une vue d'ensemble va d'abord être présentée, puis une analyse détaillée va être faite pour chacune des situations.

6.5.1 Différentes expérimentations

Ici, nous décrivons les données de base relatives aux essais qui sont discutés plus loin dans ce chapitre.

Statistiques

Dans le cadre des expérimentations, deux types de tests ont été réalisés :

- des tests quantitatifs en re-jeu de données,
- des tests qualitatifs en situation réelle sur véhicule.

On peut se demander pourquoi limiter l'analyse de données chiffrées aux tests de re-jeu, et non les étendre aux tests en situation réelle. La première raison tient à ce qu'il est complexe d'analyser une quantité identique de données sur véhicule réel par rapport à un re-jeu de données. En effet, avant d'avoir réalisé une centaine de kilomètres sur un circuit (ce que nous avons fait en re-jeu), il faudrait plusieurs jours, voire des semaines avec un véhicule se déplaçant à environ $2 \text{ m} \cdot \text{s}^{-1}$.

Secondement, afin de réaliser des analyses statistiques dans différentes configurations (incertitudes et biais sur les positions des murs plus ou moins importantes), il était nécessaire de pouvoir rejouer une séquence à l'identique (en terme de mesures disponibles au niveau des capteurs) avec différents paramètres de loi de probabilité pour les différents murs.

Une remarque concernant l'aspect temps-réel du système doit être faite. Nous travaillons sur une architecture utilisant le *middleware* **effibox**¹. Cette architecture est par essence *multi-thread*, c'est à dire que plusieurs processus interdépendants sont actifs simultanément. Dans ce type d'architecture, la notion qui importe est celle de *task-switching*, c'est à dire de commutation entre différentes tâches. Ces instants de commutation dépendent d'aspects purement aléatoires relatifs à la synchronisation (ou plutôt son absence) entre le système d'exploitation et l'application. En fonction de ces interruptions, certaines données capteurs pourront ou non être prises en compte en temps réel. Pour toutes ces raisons, dans un environnement *multi-threaded*, il est illusoire de chercher à obtenir deux re-jeux avec les mêmes résultats en sortie.

Il est donc nécessaire de réaliser un échantillonnage statistique des résultats obtenus grâce à la méthode utilisée, et donc de pouvoir rejouer un grand nombre de fois les jeux de données choisis. Nous avons choisi un nombre de re-jeux de 100. Ces jeux de données ont été re-joués à la vitesse du temps-réel, c'est à dire à celle des acquisitions capteur. La durée prise sur un jeu de données par exemple de $7mn\ 40s$ correspond à une durée totale de tests de 13 heures environ, par type de configuration de capteurs.

1. <http://www.efdence.com>

Lien entre capteurs utilisés et expérimentations

Comme nous avons déjà eu l’occasion de le mentionner, notre approche de localisation est, dans son principe, agnostique sur le type de capteurs utilisés. Le seul élément permettant de changer fondamentalement le comportement de la localisation est la configuration des capteurs embarqués et leurs caractéristiques.

Par exemple, un capteur GPS donne la même information, qu’il soit de type *différentiel-RTK* ou *standard*. Par contre, les propriétés de la stratégie de localisation seront très différentes, selon que l’information fournie a un écart-type de 5 centimètres ou de 5 mètres.

Ces effets se feront sentir, tout d’abord sur le type de capteur sélectionné. Par exemple, si la carte est précise à 60 *cm*, et que le véhicule a le choix entre un triplet {LIDAR, mur}, et un triplet {GPS-RTK}, il va préférentiellement choisir d’utiliser la donnée provenant du GPS-RTK. Par contre, si le GPS installé est de type *standard*, le véhicule va privilégier le capteur LIDAR, car il fournit une information plus précise que les 5 *m* d’incertitude que donne un capteur GPS *standard*.

Ensuite, lorsqu’un GPS-RTK est installé, il est possible que, par rebond, des éléments de la carte soient mis à jour, auquel cas ils le seront avec la précision de localisation que le GPS-RTK donne au véhicule. Ceci se produit à la marge avec un GPS *standard*, le système le sélectionnant principalement comme moyen de lever une ambiguïté relative à l’état du véhicule (au démarrage de la séquence par exemple).

C’est pourquoi plusieurs configurations de capteurs ont été prises en compte durant les différents tests qui ont été conduits. Ce sont ces configurations que nous décrivons ici :

1. LIDAR, GPS standard et magnétomètre dans une carte précise ($\sigma_\rho \approx 10$ *cm*),
2. LIDAR, GPS standard et magnétomètre dans une carte imprécise ($\sigma_\rho \approx 80$ *cm*),
3. LIDAR, GPS-RTK et magnétomètre dans une carte imprécise ($\sigma_\rho \approx 80$ *cm*).

Le terme σ_ρ représente ici l’incertitude de position des murs dans le sens latéral par rapport à leur direction d’orientation (d’un bord du mur à l’autre). Ces trois configurations nous semblent adéquates afin de pouvoir mettre en évidence les propriétés de notre solution, tant en localisation dans une carte incertaine, qu’en terme de re-cartographie basée sur des données provenant de capteurs plus performants ou moins biaisés.

Biais sur la carte

Afin de prendre en compte non seulement l’aspect incertain de la position de l’amer (par une matrice de covariance plus ou moins grande), mais aussi le biais sur les positions (la position estimée d’un amer réel peut différer de plusieurs dizaines de centimètres de la réalité), nous avons volontairement introduit dans la carte mentionnée à la section 5.2.6 des biais et imprécisions artificiels.

Ces biais ont été ajoutés en suivant une loi normale prenant en compte l’angle du mur et sa position latérale, sur une dizaine d’amers de type mur sur la quarantaine que comporte la carte. Les covariances ont été augmentées de manière que la carte générée ainsi reste néanmoins intègre à 2σ . Sur certains jeux de données, un décalage de plus de 2 *m* a ainsi été simulé (répartition du biais normale de 80 *cm* à 1σ). Ceci a été réalisé afin de pouvoir comparer une carte imprécise et partiellement erronée à une carte précise et intègre, l’idée étant de voir si le véhicule parvient néanmoins à utiliser la carte, et si cette dernière va être corrigée au fur et à mesure.

La figure 6.9 indique la position des murs qui ont été biaisés. Dans cette technique, l’état du mur est corrompu en lui ajoutant un terme erroné $\Delta X_{\mathcal{L}}$, conformément à l’équa-

tion (6.44). Afin de conserver à la covariance son intégrité, on ajoute un terme de compensation à la covariance originale, comme indiqué dans l'équation (6.45). Cette covariance originale est, elle, obtenue selon un calcul conforme à la figure 6.8.

$$\tilde{\mathbf{X}}_{\mathcal{L}} = \mathbf{X}_{\mathcal{L}} + \overbrace{\Delta \mathbf{X}_{\mathcal{L}}}^{\text{biais ajouté}} \quad (6.44)$$

$$\tilde{\Sigma}_{\mathbf{X}_{\mathcal{L}}} = \underbrace{\Sigma_{\mathbf{X}_{\mathcal{L}}}}_{\text{selon figure 6.8}} + \underbrace{\Delta \mathbf{X}_{\mathcal{L}} \cdot \Delta \mathbf{X}_{\mathcal{L}}^{\top}}_{\text{terme d'erreur sur la covariance}} \quad (6.45)$$

Sur la figure 6.8 est présentée la manière dont la covariance de base de chaque mur est calculée. Chaque vecteur $\mathbf{X}_{\mathcal{L}}$ est calculé à partir des positions de deux bords de mur, E_1 et E_2 . Le modèle de covariance pour chacune des positions des deux bords de mur est un modèle isotrope et indépendant ayant un écart-type donné. On propage ensuite cette incertitude sur la covariance du vecteur $\mathbf{X}_{\mathcal{L}}$ par approximation linéaire.

Les murs 74, 76, 85, 87, 88, 91, 95, 103, 106 et 114, indiqués sur la figure 6.9, sont initialisés selon les formules (6.44) et (6.45).

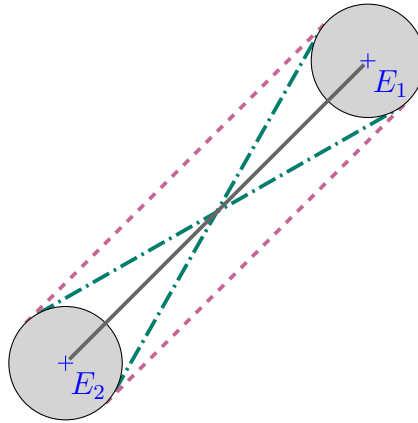


FIGURE 6.8 – Schéma détaillant le modèle d'incertitude concernant l'**incertitude angulaire** et l'**incertitude de position** relatives aux murs non-biaisés. La zone grisée correspond à une incertitude de position isotrope sur chaque bord de mur.

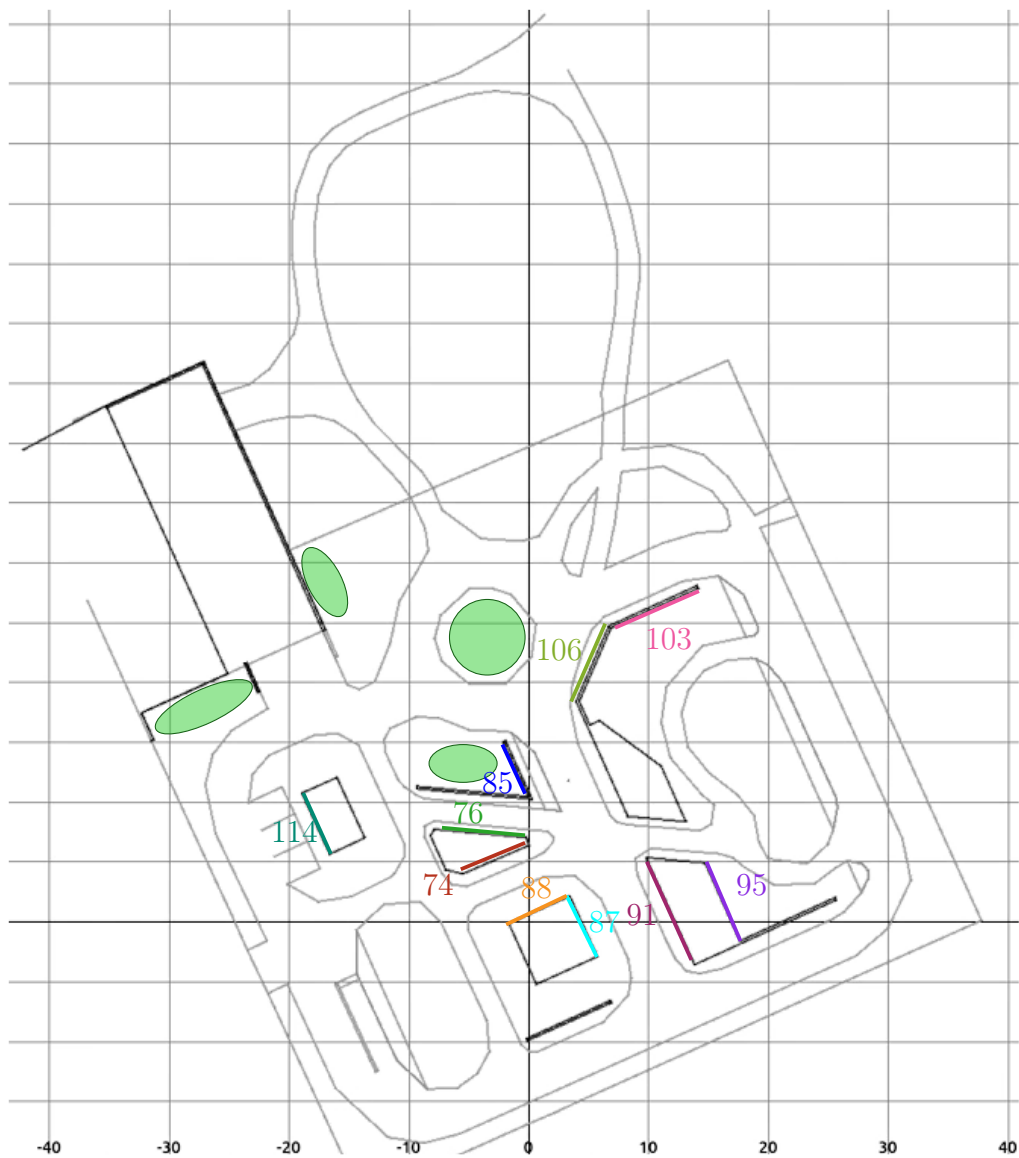


FIGURE 6.9 – Carte des positions des murs biaisés dans les expérimentations conduites. Les buissons sont indiqués par des zones vertes.

6.5.2 Test avec capteurs LIDAR, GPS et magnétomètre

Dans cette section, nous partons de la même configuration de capteurs utilisés sur le véhicule :

- un GPS standard avec une incertitude sur la valeur mesurée de 5 m à 1σ ,
- un magnétomètre ayant une incertitude de 7.5° environ à 1σ ,
- un capteur LIDAR ayant une incertitude de 0.25° en orientation (σ_θ) et de 5 cm en distance (σ_ρ).

La position initiale du véhicule est configurée dans toutes les expérimentations comme suit :

- une position du véhicule au centre du site d'expérimentations PAVIN,
- une incertitude de position de 60 m à 1σ , englobant le site urbain de la plateforme PAVIN,

- une orientation du véhicule plein EST,
- une incertitude d'orientation de 360° à 1σ , englobant complètement le cercle unité.

Deux types de cartes ont été pris comme référence :

- des éléments biaisés, toujours de manière identique :
 - une erreur angulaire ajoutée de $\pm 1.8^\circ$ à 1σ ,
 - une erreur de position latérale ajoutée de $\pm 86\text{ cm}$ à 1σ ,
 - sur les murs 74, 76, 85, 87, 88, 91, 95, 103, 106, 114, voir la figure 6.9,
 - les biais sont générés de manière différente et aléatoire suivant la loi précédente à chaque run (le biais de position peut aller jusqu'à 2 m sur la position du mur),
- dans deux types de cartes :
 - une première carte, dite « précise », contenant des murs dont les bords ont une imprécision isotrope de 7 cm à 1σ ,
 - une seconde carte, dite « imprécise », contenant des murs dont les bords ont une imprécision isotrope de 70 cm à 1σ .

Localisation et correction sur carte précise à 7 cm

Dans cette série d'expérimentations, une analyse statistique a été conduite de manière à s'assurer que la stratégie de localisation opérait convenablement. Deux éléments ont été analysés :

1. la qualité de la trajectoire du véhicule,
2. celle des murs biaisés de la carte.

Concernant le véhicule, une erreur de 15 cm en moyenne est obtenue. L'erreur de position latérale obtenue est centrée sur 0, avec un écart-type d'environ 10 cm . L'erreur angulaire sur la position du véhicule est également centrée sur 0 avec un écart-type d'environ 4° .

Concernant la gestion de la carte, on s'attend à ce que les amers biaisés (qui ont une plus forte covariance que les autres afin que leur estimation reste intègre) soient corrigés par rapport aux autres par rebond. La table 6.7 récapitule les résultats obtenus en moyenne pour les 10 murs biaisés à la fin de chaque trajectoire.

Les murs étant au départ initialisés avec un biais d'écart-type 85 cm en position et 1.8° en orientation obtiennent tous des résultats significativement meilleurs après l'exécution de la trajectoire choisie. L'erreur de direction est de l'ordre de 0.1° , avec un écart-type de l'ordre de 0.2° environ, bien qu'une certaine disparité existe dans ces résultats en fonction des murs. L'erreur de position des murs après réalisation de la trajectoire est de l'ordre de 6 cm en terme de biais résiduel, et de l'ordre de 7.5 cm , ce qui reste intègre et bien plus précis que les données de départ.

L'interprétation que nous donnons à ce phénomène est en deux temps :

1. tout d'abord, grâce aux amers « précis » de la carte, le véhicule parvient à se localiser de manière tout aussi « précise »,
2. ensuite, lorsqu'un amer plus « imprécis » est détecté, le véhicule « fournit » à cet amer une partie de la précision qu'il a lui-même acquise.

Ce qui est intéressant dans ce phénomène, c'est qu'il n'y a pas de « choix booléen » du type *amer imprécis* ou *amer précis*. Les équations de mise à jour par l'intersection de covariance, dans le cas d'une mise à jour bilatérale par mesure LIDAR entre un véhicule et un mur, transmettent l'information de l'amer le plus précis vers l'amer le moins précis.

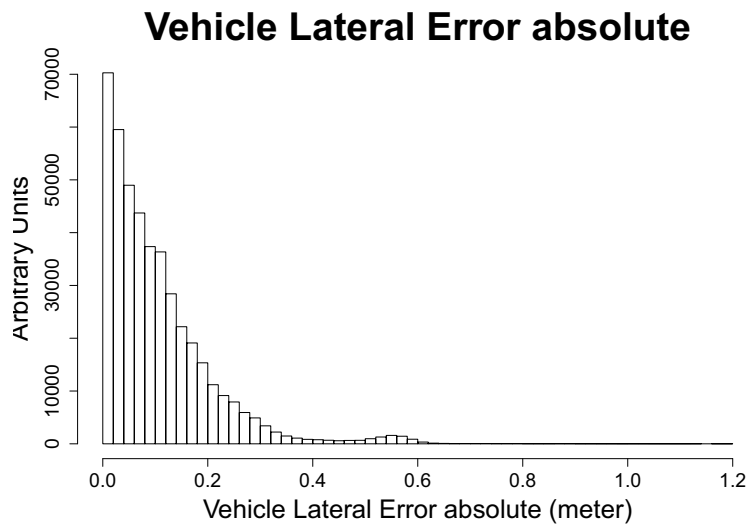


FIGURE 6.10 – Analyse statistique sur 100 trajectoires de la distribution statistique des erreurs de position latérale du véhicule. Utilisation d’une carte précise et d’un GPS UBLOX–6T.

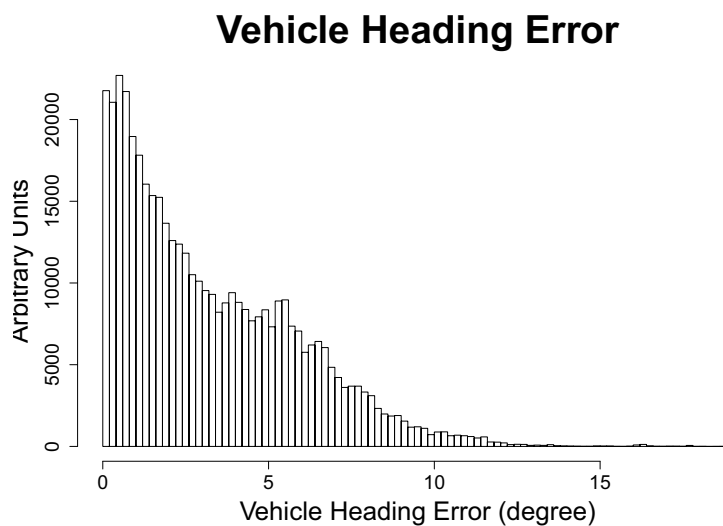


FIGURE 6.11 – Analyse statistique sur 100 trajectoires de la distribution statistique des erreurs d’orientation du véhicule. Utilisation d’une carte précise et d’un GPS UBLOX–6T.

Dans le cas où l’état du véhicule est connu avec une meilleure précision, c’est l’état de l’amer mur qui se voit « re-précisé ». Dans le cas contraire, c’est un « mur précis » qui va permettre au véhicule d’améliorer la précision de sa localisation.

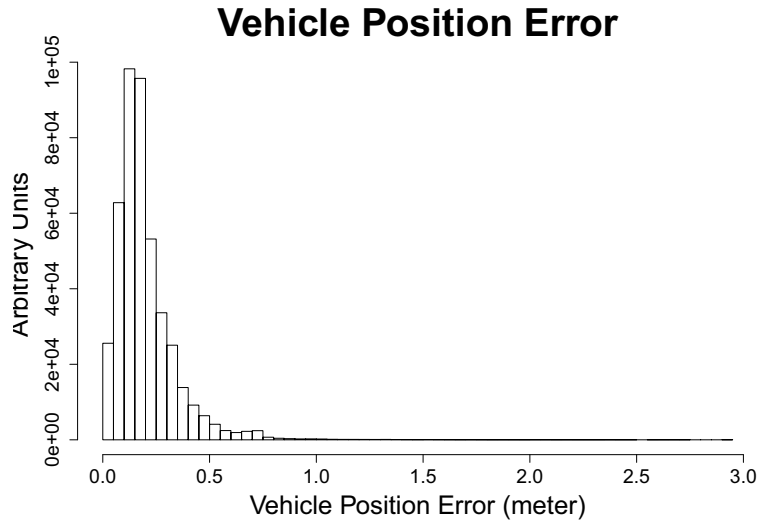


FIGURE 6.12 – Analyse statistique sur 100 trajectoires de la distribution statistique des erreurs de position du véhicule. Utilisation d’une carte précise et d’un GPS UBLOX–6T.

Mur	erreur latérale $\Delta\rho$ (m)		erreur de direction $\Delta\theta$ (°)		Hors-distribution
	μ	σ	μ	σ	
74	0.004	0.030	-0.05	0.13	0%
76	0.024	0.090	-0.01	0.08	0%
85	0.120	1.170	0.31	1.32	8%
87	0.016	0.063	0.04	0.30	0%
88	0.160	0.220	0.06	0.70	0%
91	0.073	0.076	0.16	0.40	0%
95	0.071	0.092	-0.07	0.57	0%
103	0.017	0.190	0.01	0.19	0%
106	0.064	0.033	-0.10	0.28	0%
114	0.019	0.004	0.01	0.12	0%

TABLE 6.7 – Résultats de correction d’estimation des murs biaisés après navigation dans une carte précise en utilisant un capteur GPS de type UBLOX–6T. Moyenne obtenue sur 100 runs.

Il faut bien noter ici que ces résultats ne sont possibles que parce que la matrice de covariance qui est calculée pour chacun de ces murs au départ de la trajectoire englobe le biais de manière à ce que la distance de *Mahalanobis* de l’estimation initiale de ces murs par rapport à la réalité reste entière. On a choisi comme distance de *Mahalanobis* initiale la valeur 0.5. Toute autre valeur ne dépassant pas l’unité aurait convenu tout aussi bien.

Dans le cas où un élément de la carte est intégré avec une valeur *fausse* de ses paramètres (par exemple avec une distance de *Mahalanobis* supérieure à 2 à l’initialisation), cela pourrait avoir des conséquences négatives sur l’intégrité de la trajectoire du véhicule

si cet amer venait à être détecté. Dans le cas où il ne le serait pas (en raison de ses paramètres biaisés), il serait judicieux de pouvoir le supprimer de la carte utilisée. Cette thématique sera l'objet du chapitre 7.

Nous n'avons pas discuté encore les résultats relatifs au mur 85. Ce dernier est caché par un buisson sur la quasi-totalité de sa longueur, excepté les 20 *cm* les plus proches de la voie de circulation. Des détections à faux, en raison de la présence du buisson, font dériver l'estimation de position de ce mur, notamment en raison du faible nombre d'échos LIDAR en provenant effectivement (de l'ordre de 5). Dans les autres cas, l'absence de détection du mur a pour conséquence que les paramètres initiaux du biais sont conservés. Ce mur est détecté correctement de manière « accidentelle », et de ce fait les résultats sur ce mur sont bien plus mauvais que ce à quoi l'on est en droit de s'attendre. Néanmoins, ces résultats sont cohérents, puisque ce mur n'est, sauf exception, jamais correctement détecté.

Localisation et correction sur carte imprécise

Dans cette série d'expérimentations, la seule différence d'avec la section précédente est le fait que la carte utilisée est beaucoup plus incertaine. En effet, l'incertitude relative au positionnement des bords de murs passe de 7 *cm* à 70 *cm*. Les biais sont eux statistiquement identiques, même si leurs valeurs sont différentes, ce qui est dû aux tirages.

Nous allons ici encore analyser tout d'abord les performances de la localisation du véhicule, puis analyser celle des murs biaisés. On observe ici encore une valeur d'erreur de position latérale centrée sur 0 avec un écart-type d'environ 15 *cm*. L'erreur d'orientation est également centrée sur 0 avec un écart-type de l'ordre de 3°. L'erreur de position moyenne est de l'ordre de 25 *cm* avec un écart-type de l'ordre de 10 *cm*.

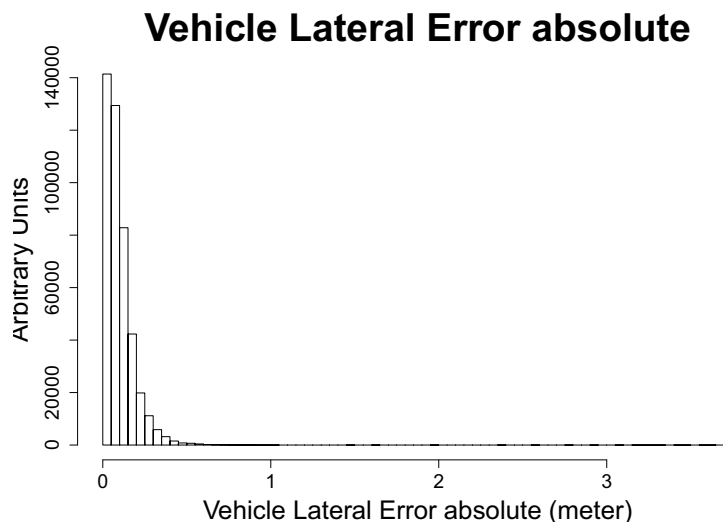


FIGURE 6.13 – Analyse statistique sur 100 trajectoires de la distribution statistique des erreurs de position latérale du véhicule. Utilisation d'une carte imprécise et d'un GPS UBLOX-6T.

Vehicle Heading Error

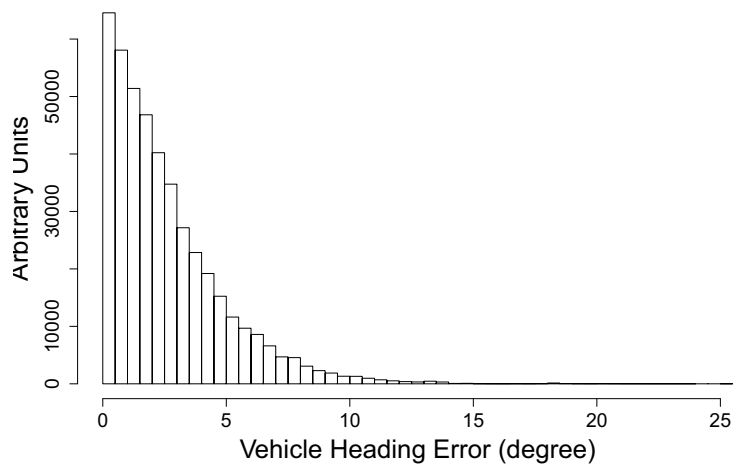


FIGURE 6.14 – Analyse statistique sur 100 trajectoires de la distribution statistique des erreurs d’orientation du véhicule. Utilisation d’une carte imprécise et d’un GPS UBLOX–6T.

Vehicle Position Error

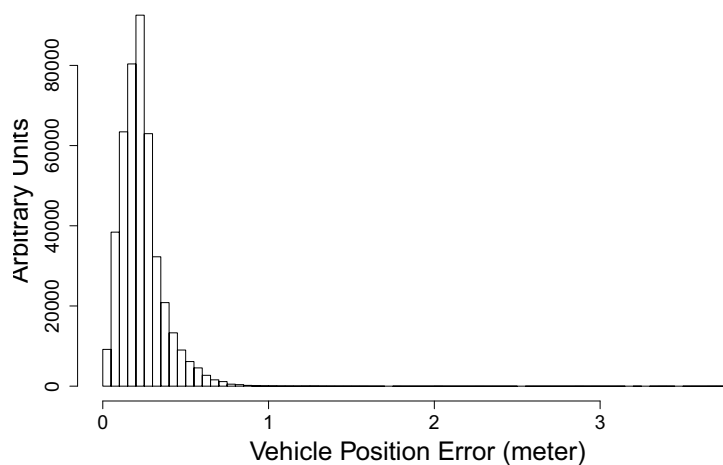


FIGURE 6.15 – Analyse statistique sur 100 trajectoires de la distribution statistique des erreurs de position du véhicule. Utilisation d’une carte imprécise et d’un GPS UBLOX–6T.

Concernant les murs biaisés, la statistique initiale des biais est identique sur toutes les expérimentations (1.8° à 1σ pour l’orientation, 86 cm à 1σ pour la position). Les résultats obtenus pour les murs biaisés après chacune des 100 trajectoires rejouées sont présentés dans la table 6.8.

Mur	erreur latérale $\Delta\rho$ (m)		erreur de direction $\Delta\theta$ ($^\circ$)		Hors-distribution
	μ	σ	μ	σ	
74	-0.030	0.250	-0.44	0.74	0%
76	-0.025	0.290	-0.13	0.51	1%
85	0.075	0.820	-0.11	1.70	10% , mur caché
87	-0.030	0.140	0.30	0.34	1%
88	0.046	0.082	0.51	0.28	0%
91	0.143	0.109	0.25	0.34	0%
95	0.100	0.058	1.36	0.40	0%
103	0.085	0.040	0.11	0.14	0%
106	-0.050	0.510	-0.30	0.93	7%
114	-0.020	0.040	0.35	0.29	0%

TABLE 6.8 – Résultats de correction d’estimation des murs biaisés après navigation dans une carte imprécise en utilisant un capteur GPS de type UBLOX–6T. Moyenne obtenue sur 100 runs.

On observe tout d’abord un plus grand nombre de murs « hors-distribution » à la fin de l’exécution. Par exemple, pour le mur 76, dans l’une des 100 séquences rejouées, ce mur est initialement décalé de 1.76 *m*. La première (et unique) détection de celui-ci provoque une mise à jour à faux. On observe que le mur est, suite à cette détection à faux, décalé de plus de 2.7 *m* en position latérale, et de 1.6° en orientation. Par rapport à notre environnement à échelle $\frac{1}{2}$, cette erreur serait l’équivalent pour une échelle 1 d’un décalage de 5.4 *m*. Ces mauvais paramètres mis à jour du mur entraînent par voie de conséquence une impossibilité d’être à nouveau détecté afin d’être corrigé. Le même phénomène se produit pour les murs 87 et 106.

On voit donc que l’augmentation de l’incertitude relative aux murs peut provoquer des mises à jour délétères. Néanmoins, le phénomène apparaît marginal, et il serait envisageable d’améliorer le mécanisme de détection afin de limiter encore plus ce phénomène. Les valeurs « hors-distribution » du mur 85 passent à 10%, cette valeur étant de 8% sur une carte précise. On peut donc déduire que pour les cas d’occultation des murs, l’imprécision de la carte ne dégrade pas outre mesure les performances par rapport aux résultats obtenus avec une carte précise.

Pour les autres murs, les biais sont de l’ordre de 7 *cm* et 0.5°, et les écarts-types correspondants sont d’environ 10 *cm* et 0.4°. Le mur 95 apparaît avoir une orientation légèrement décalée par rapport à la position enregistrée dans la base de données. Étant donné que la zone d’où ce mur est observable est en pente d’environ 5°, il est dans l’ordre du possible que le modèle $2D\frac{1}{2}$ se révèle inadéquat dans ce type de situations, ce qui pourrait expliquer le léger décalage observé (environ 1°).

On observe que, malgré la plus grande imprécision de la carte (70 *cm* par rapport à 7 *cm*), les valeurs et écarts-types observés en fin de séquence sont faiblement biaisés et d’ordre de grandeur comparable à ceux obtenus pour une carte précise (rapport entre 1 : 1 et 1 : 4). On conclut donc que le biais est corrigé dans la carte, pour autant que celui-ci soit estimé de manière cohérente par l’incertitude de position. Ceci est réalisé par le biais de la valeur de la matrice de covariance, qui indique par une forte valeur l’identification d’une méconnaissance au niveau de la carte. Si cette valeur de covariance ne reflétait pas

cette méconnaissance, on serait certain au niveau de la carte des coordonnées d'un mur, mais cette information serait fausse.

6.5.3 Test avec capteurs LIDAR, GPS-RTK et magnétomètre dans une carte imprécise

Dans cette section, nous nous intéressons à ce qui se produit lorsque le capteur GPS standard (ayant une incertitude de position de 5 m) est remplacé par un capteur de type GPS-RTK (ayant 5 cm d'incertitude de position). On peut légitimement estimer qu'utiliser un tel capteur dans une carte précise n'offre qu'un intérêt limité, étant donné que la carte est déjà précise et ne va être reprécisée qu'à la marge. Nous avons donc choisi de ne faire qu'une unique expérimentation avec ce type de configuration, en utilisant une carte imprécise.

La configuration totale du système est la suivante :

- configuration des capteurs comme suit :
 - un capteur GPS-RTK avec une incertitude sur la valeur mesurée de 5 cm à 1σ ,
 - propriétés du capteur magnétomètre inchangées,
 - propriétés du capteur LIDAR inchangées,
- estimation de la situation de départ du véhicule inchangée,
- configuration de la carte comme suit :
 - statistique des biais initiaux sur murs choisis inchangés (murs biaisés),
 - utilisation d'une carte, dite « imprécise », contenant des murs dont les bords ont une imprécision isotrope de 70 cm à 1σ (tous les murs),

L'idée est de pouvoir analyser dans quelle mesure l'adjonction au véhicule d'un capteur donnant une faible incertitude (donc une grande quantité d'information) permet à la carte d'acquérir elle-même une faible incertitude et par conséquent d'accumuler de l'information qui lui faisait défaut au démarrage de la séquence. La séquence choisie a été rejouée 100 fois, comme dans les deux précédentes analyses.

Comme le capteur GPS-RTK utilisé ici comme source de données est celui qui est dévolu à l'obtention de la réalité terrain du véhicule, il est impossible de mesurer correctement la dérive de ce dernier. De plus, le terrain du site expérimental de PAVIN étant en pente jusqu'à 7° , une erreur de parallaxe existe (de l'ordre de 10 à 15 cm) sur la mesure du GPS-RTK dont l'antenne est positionnée sur le toit du véhicule à 1.80 m du sol.

Nous nous sommes focalisés ici sur l'agrégation par la carte de l'information de localisation absolue fournie par le GPS-RTK, et spécialement sur les murs dont les estimations initiales ont été biaisées.

La table 6.9 récapitule les résultats obtenus pour les 100 re-jeux en question.

On peut observer que les murs initialement biaisés sont repositionnés à des valeurs assez proches de celles entrées dans la carte absolue du site de PAVIN. Il semble exister un biais d'orientation d'environ 0.5° sur les murs 87 et 88, qui appartiennent au même bâtiment léger. L'écart-type obtenu pour cette mesure étant dans les deux cas de l'ordre de 0.15° , la valeur de biais semble significative.

De la même manière, le mur 91 apparaît décalé d'environ 5 cm et de 0.5° . Étant donné que ce mur a des structures gravées (structure en forme d'escalier gravée par paliers graduels afin de donner du relief au mur), cela pourrait expliquer tout ou partie de ces valeurs.

Nous ne pouvons expliquer avec certitude chacun des écarts relevés, mais la carte obtenue semble cohérente. Les écarts-type latéraux et angulaires sont respectivement de

Mur	erreur latérale $\Delta\rho$ (m)		erreur de direction $\Delta\theta$ ($^\circ$)		Hors-distribution
	μ	σ	μ	σ	
74	-0.080	0.017	-0.17	0.14	0%
76	0.024	0.014	-0.17	0.14	0%
85	0.130	0.810	0.14	1.85	6% mur caché
87	-0.030	0.012	0.50	0.10	0%
88	-0.035	0.032	0.57	0.16	0%
91	0.054	0.005	0.47	0.05	0%
95	-0.037	0.028	0.80	0.51	0%
103	-0.035	0.007	-0.30	0.06	0%
106	-0.003	0.167	-0.08	0.93	3%
114	-0.050	0.015	0.15	0.31	0%

TABLE 6.9 – Statistique des positions et incertitudes sur les murs biaisés à la fin des trajectoires du véhicule équipé d’un GPS–RTK dans une carte imprécise.

l’ordre de 1.6 *cm* et 0.2°. Ils indiquent que la carte obtenue est de précision centimétrique en position (ainsi que la mesure de position de type RTK). La précision angulaire obtenue est cohérente, 3.5 *mrad* soit 0.2°, puisqu’elle correspond à une précision spatiale de 1.6 *cm* pour un mur d’environ 4 à 5 *m*, ce qui est la longueur moyenne des murs biaisés analysés.

Nous concluons de cette analyse que lorsque le véhicule dispose d’une source de données de qualité centimétrique, une carte imprécise (70 *cm* d’imprécision) peut être re-précisée de manière également centimétrique.

6.5.4 Comparaison au filtre de Kalman

Dans cette section, nous comparons la gestion d’une carte précise grâce à un GPS standard, carte comportant des murs biaisés comme dans les expérimentations précédentes, en utilisant un filtre à *intersection de covariance* et un filtre de Kalman. Les expérimentations relatives à la gestion d’une carte imprécise grâce à un GPS standard ou à un GPS–RTK ne sont pas présentées, mais concluent dans le même sens.

Nous nous focalisons ici sur la performance de repositionnement des murs biaisés à l’issue de l’exécution de la séquence de localisation, en comparant la distance de Mahalanobis dans les deux cas. Ces données sont présentées dans la table 6.10.

Les valeurs de distance de Mahalanobis pour le filtre SCI sont en moyenne inférieures à 0.5σ . à l’opposé, pour le filtre de Kalman, celles-ci sont toutes au-delà de 2σ , sauf celle du mur 114, et les écarts-type sont tous strictement supérieurs à 2σ . Ces deux éléments indiquent :

- la sur-convergence d’une carte gérée par un filtre de Kalman,
- une grande dispersion dans les résultats obtenus par le filtre de Kalman, donc une instabilité de la carte.

Un exemple typique de sur-convergence est présenté sur la figure 6.16. On observe que la zone dans laquelle on estime le mur (en bleu dans les cadres zoomés) est à une grande distance de la valeur estimée vraie dans la carte absolue de l’environnement. Ceci montre un exemple typique de cas de sur-convergence que l’on peut obtenir avec un filtre

Mur	Kalman $d_M(\sigma)$		SCI $d_M(\sigma)$	
	μ	σ	μ	σ
74	4.47	5.32	0.11	0.12
76	4.34	6.85	0.11	0.06
85	5.31	11.37	0.27	0.37
87	14.76	12.20	0.18	0.24
88	22.63	15.67	0.17	0.07
91	27.94	26.86	0.20	0.11
95	3.51	2.91	0.36	0.09
103	6.94	10.48	0.10	0.04
106	5.45	4.25	0.33	0.77
114	1.83	2.22	0.08	0.04

TABLE 6.10 – Distance de Mahalanobis moyenne à la fin des séquences pour un filtre de Kalman et un filtre SCI dans une carte précise à 7 cm.

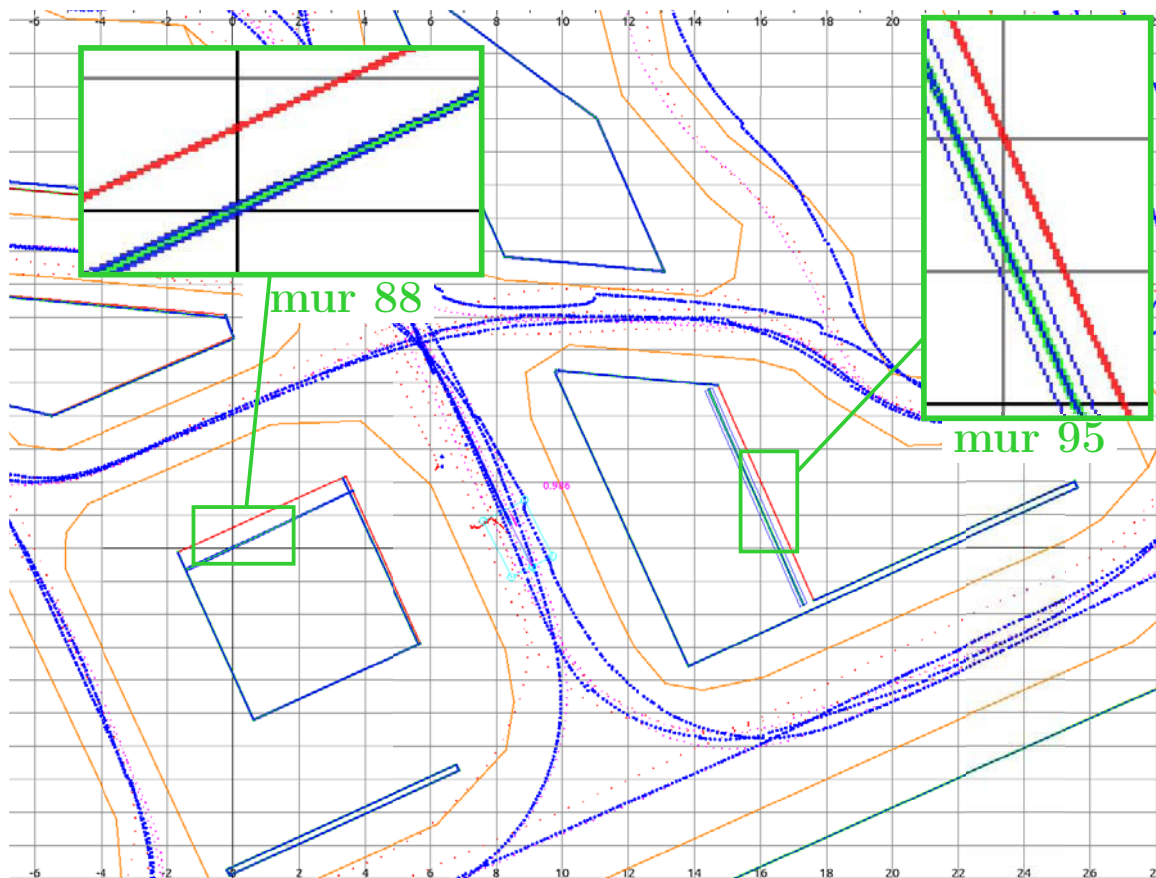


FIGURE 6.16 – Exemple de sur-convergence obtenue avec un filtre de Kalman. Les zones d'incertitude des murs sont indiquées en bleu, autour de l'estimation courante du mur, qui elle est en vert. La vérité au niveau carte est indiquée en rouge.

de Kalman, qui est de même nature que le problème mentionné à la section 4.7.3. C'est

le problème que les filtres CI et SCI résolvent, ce qui explique que les résultats montrés dans la table 6.10 soient corrects, à contrario de la situation précédente.

On peut donc conclure que l'utilisation d'un filtre de Kalman, afin de gérer une carte contenant des éléments dont les estimations sont séparées, est totalement inappropriée en raison de la sur-convergence de la carte obtenue. Le filtre SCI, que nous avons utilisé avec succès, résout efficacement le problème de sur-convergence.

6.5.5 Test du « robot kidnappé »

Dans cette expérimentation, le véhicule est équipé du GPS standard. Les paramètres du magnétomètre et du LIDAR sont toujours inchangés. Le principe est ici d'analyser en combien de temps, à partir d'une position estimée conforme aux précédentes simulations, et d'une position réelle quelconque, le véhicule met à se positionner à 30 *cm* près.

On utilise toujours le même jeu de données que précédemment, avec la configuration suivante :

- capteurs :
 - un GPS standard avec une incertitude sur la valeur mesurée de 5 *m* à 1σ ,
 - un magnétomètre ayant une incertitude de 7.5° environ à 1σ ,
 - un capteur LIDAR ayant une incertitude de 0.25° en orientation (σ_θ) et de 5 *cm* en distance (σ_ρ).
- position initiale du véhicule :
 - une position du véhicule au centre du site d'expérimentations PAVIN,
 - une incertitude de position de 60 *m* à 1σ , englobant le site urbain de la plateforme PAVIN,
 - une orientation du véhicule plein EST,
 - une incertitude d'orientation de 360° à 1σ , englobant complètement le cercle unité.
- une carte :
 - sans murs biaisés,
 - avec une incertitude de position des murs de 14 *cm* à 1σ .

Les instants de démarrage seuls changent, par pas de 10 *s*. Pour chaque date de départ, 10 re-jeux de 20 *s* ont été réalisés. La durée totale est de 7mn 40s, soit environ 43 dates de démarrage différentes.

Dans cette expérimentation, aucun mur n'a été biaisé, afin d'éviter d'avoir un effet localisé à l'environnement de l'un d'entre eux, et donc pour obtenir une image la plus fidèle du phénomène. Le jeu de données choisi a la particularité de parcourir toutes les voies de circulation de notre site PAVIN, et la plupart d'entre elles dans les deux sens. Les résultats obtenus sont résumés dans la table 6.11. Dans cette table, d_M représente la distance de Mahalanobis, μ et σ respectivement la moyenne et l'écart-type de la variable analysée.

On observe sur cette table :

- que le temps d'accrochage à 30 *cm* est de l'ordre de 4 secondes,
- que la distance de Mahalanobis n'est pas entière pendant 3% du temps soit 0.6 secondes en moyenne,
- que la valeur de la distance de Mahalanobis en fin de séquence de 20 secondes est quasiment toujours entière (inférieure à 1.7σ),

	μ	σ
Nombre de re-jeux	428	
Durée où l'erreur de position > 30 cm (sec)	3.9	2.7
Durée où $d_M > 1.7 \sigma$ (sec)	0.6	1.0
Durée où la confiance > 50 % (sec)	55 %	16 %
Proportion de jeux où $d_M(t = 20sec) < 1.7 \sigma$	99 %	
Erreur moyenne à $t = 20sec$ (cm)	26	12

TABLE 6.11 – Résultats obtenus pour le test du « robot kidnappé ».

- que l'erreur absolue de position en fin de séquence de 20 secondes est de l'ordre de 26 cm, ce qui est proche d'être acceptable pour la conduite autonome.

Dans le processus de démarrage, l'algorithme peut tester plusieurs murs qu'il peut ne pas parvenir à détecter. Ceci peut faire chuter la probabilité de confiance de telle sorte que celle-ci est la moitié du temps en dessous de 50%. Néanmoins le démarrage « sans connaissance préalable » de la localisation du véhicule en quatre secondes à une erreur de localisation de 25 cm en moyenne nous semble satisfaisant.

Conclusion

Dans ce chapitre, nous avons présenté plusieurs modifications apportées à la méthode de base décrite au chapitre 5, qui permettent :

- la localisation dans une carte précise,
- la localisation dans une carte imprécise,
- la correction des biais des éléments imprécis et faux,
- l'amélioration de précision de la carte lorsqu'une donnée capteur permet au véhicule de se localiser précisément.

Après avoir présenté la gestion de la carte de l'environnement par un filtre SCI, nous avons vérifié que cette technique était applicable dans le cadre de simulations. Celles-ci, outre la gestion d'une carte de l'environnement, valident également la possibilité de la gestion d'une flotte de véhicules sans sur-convergence. Un critère de sélection des sources de données provenant d'observations choisies à l'avance, basé sur des notions de théorie de l'information, a été présenté et testé.

Afin de porter cette méthode sur véhicule, certaines variations ont été apportées, qui sont présentées également. Des expérimentations qualitatives sur le site PAVIN ont été réalisées afin de valider la fonctionnalité des différentes configurations. Une analyse statistique en mode re-jeu a été faite afin de jauger les performances de ce nouvel algorithme dans différentes situations, et de valider son principe. Le problème du « robot kidnappé » a été résolu ici à partir d'une estimation très imprécise de l'état initial du véhicule. Les performances d'accrochage à chaud de la localisation du véhicule à partir d'un grand

nombre de positions réelles de départ du véhicule ont été investiguées et nous jugeons qu'elles sont satisfaisantes.

Deux points relatifs à la gestion d'une carte d'environnement incertain n'ont pas été évoqués ici :

1. la gestion d'un amer de la carte dont l'état, estimé de manière précise, est faux,
2. l'ajout d'un amer non intégré à la carte à partir d'observations des capteurs.

L'item n°1 est traité dans le chapitre suivant, et le n°2 fait l'objet du chapitre 8.

Chapitre 7

Suppression d'Amers de la Carte

Nous avons traité dans le chapitre précédent le cas d'amers imprécis de la carte, de manière à permettre à un véhicule de se localiser malgré l'imprécision de celle-ci. Une possibilité d'améliorer la carte imprécise en fonction de données plus précises permet de propager à la carte elle-même l'information fournie par les capteurs.

Mais qu'en est-il du cas où un amer est pourvu d'un état erroné? Par exemple, un amer positionné à une distance de Mahalanobis très élevée (un mur à plus de 4σ) peut être considéré comme faux, puisqu'on le recherchera au mauvais endroit. De la même manière, un amer enregistré dans la carte, mais totalement absent du terrain, sera également introuvable, donc considéré comme faux.

La gestion de ce type de cas pourrait permettre de mettre à jour une carte de l'environnement, en supprimant les amers introuvables ou indétectables, au fur et à mesure du temps qui passe et de l'évolution du terrain. Nous allons montrer comment nous envisageons de réaliser cette fonctionnalité après avoir passé en revue les solutions potentielles. Nous présenterons également quelques résultats expérimentaux montrant l'intérêt de pouvoir supprimer des amers qui, parce qu'inobservables, ne peuvent fournir la moindre information au véhicule.

7.1 Introduction

La méthode de base, présentée au chapitre 5, permet à un véhicule de se positionner en corrigeant itérativement l'estimation de son état à partir de l'observation d'amers référencés de manière précise et absolue dans une carte. Dans le chapitre 6, nous avons ajouté aux murs de la carte une imprécision sur leur état estimé. Ceci a deux implications :

1. la localisation du véhicule devient plus imprécise (que dans le cas d'une carte infiniment précise), en fonction de l'imprécision des murs observables,
2. l'état des murs peut être corrigé si le véhicule dispose d'une estimation plus précise que celle de ces derniers.

Le diagramme équivalent à celui de la figure 5.3, dans le cas du repositionnement des murs, est présenté sur la figure 6.1.

Voici ce que nous proposons de faire afin de pouvoir nettoyer la carte des éléments erronés qui n'ont rien à y faire, car inexistantes ou totalement biaisés :

- estimer un « ordre de grandeur » de la vraisemblance de l'existence de l'amer,
- supprimer les amers de la carte lorsque ceux-ci obtiennent une vraisemblance d'existence inférieure à un seuil donné.

Le formalisme utilisé pour le véhicule, et présenté aux sections 5.2.4 et 5.4.1, peut également être utilisé pour estimer l'état d'un mur. Dans le chapitre 6, nous avons estimé un mur grâce à une densité de probabilité gaussienne, ce qui représente la partie continue du formalisme utilisé pour le véhicule. Afin de compléter celui-ci, l'ajout d'une probabilité d'existence est alors nécessaire et c'est ici notre objectif. Une telle probabilité proche de zéro indique alors un amer vraisemblablement manquant.

Notre diagramme est alors modifié ainsi que présenté sur la figure 7.1.

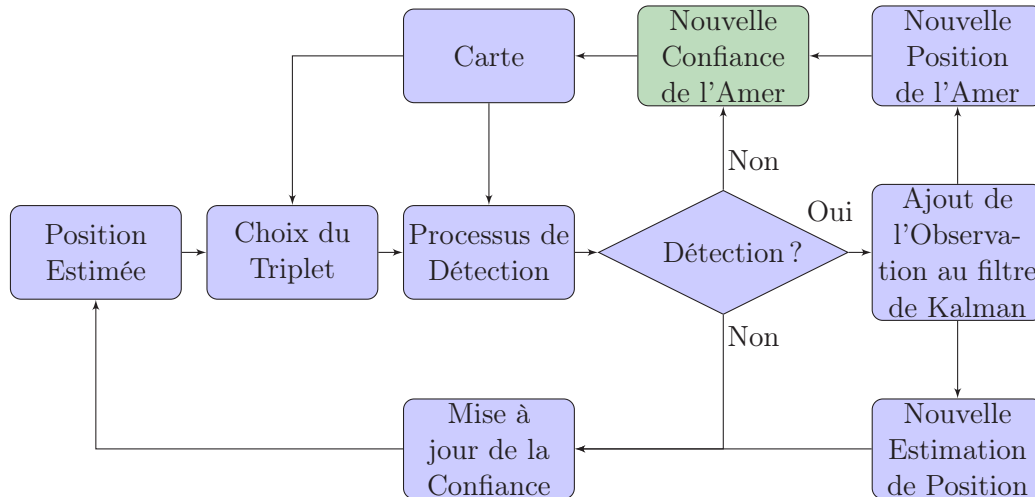


FIGURE 7.1 – Diagramme de fonctionnement de la méthode de localisation de véhicule permettant d'estimer la vraisemblance de l'existence des amers.

Le bloc de la figure 7.1 indiqué en vert estime la vraisemblance de l'existence de l'amer. C'est cette fonction que nous allons présenter dans le reste de ce chapitre.

7.2 Problématique des Réseaux Couplés

Nous allons faire dans cette section une petite digression relative aux techniques d'approximation de lois de probabilités de deux variables ou plus corrélées, par des lois de probabilité non-corrélées, c'est à dire indépendantes l'une de l'autre. L'idée est de présenter ici les raisons qui nous imposent le recours à une approximation probabiliste plutôt qu'une analyse exacte du problème. De plus, les approximations classiques peuvent poser certains problèmes que nous souhaitons mettre en lumière avant d'analyser les différentes solutions testées.

7.2.1 Estimation exacte et explosion combinatoire

Dans cette section, nous allons expliquer simplement pourquoi une estimation exacte de la loi de probabilité d'existence de chaque amer dans une carte unique est hors de portée, tant de manière théorique que pratique.

Supposons qu'un véhicule \mathcal{V} , et N amers \mathcal{L}_k soient gérés au niveau de leur intégrité ou vraisemblance par une carte unique. Dans ce formalisme, V représente la valeur booléenne *intégrité de la localisation du véhicule*, L_k représente la valeur booléenne *existence de l'amer k*. Supposons qu'au départ des estimations, la probabilité *a-priori* soit un produit cartésien de probabilités indépendantes, comme sur la formule (7.1).

$$p(\mathcal{V}, \mathcal{L}_k) = p(V) \cdot \prod_k p(L_k) \quad (7.1)$$

Le nombre de termes indépendants pour estimer un tel produit (étant donné que la somme des probabilités sur chaque variable est unitaire) est $N + 1$, comme détaillé dans les formules (7.2) et (7.3).

$$p(V) \rightarrow p(V = 1), \quad p(V = 0) \text{ sachant } p(V = 0) + p(V = 1) = 1 \quad (7.2)$$

$$p(L_k) \rightarrow p(L_k = 1), \quad p(L_k = 0) \text{ sachant } p(L_k = 0) + p(L_k = 1) = 1 \quad (7.3)$$

Supposons que l'on observe une variable O_1 qui dépend de deux autres variables V et \mathcal{L}_{n_1} . Nous obtenons la loi de probabilité postérieure indiquée en (7.4).

$$p(V, L_k | O_1) \propto p(O_1 | V, L_{n_1}) \cdot p(V) \cdot \prod_k p(L_k) \quad (7.4)$$

Comme $p(O_1 | V, L_{n_1})$ dépend de chaque terme, et que dans le cas général, cette loi ne se factorise pas en fonction de V et L_{n_1} , les probabilités de ces variables deviennent corrélées, et le nombre de termes nécessaires devient $N + 2$. Pour une seconde inférence, on obtient la formule (7.5).

$$p(V, L_k | O_1, O_2) \propto p(O_2 | V, L_{n_2}) \cdot p(O_1 | V, L_{n_1}) \cdot p(V) \cdot \prod_k p(L_k) \quad (7.5)$$

Le nombre de termes indépendants de la loi de probabilités est alors de $N + 4$ dans le cas où $n_1 \neq n_2$. Dans ce cas là, la table de probabilités devient celle indiquée aux équations (7.6), (7.7).

$$p(V, L_{n_1}, L_{n_2}) \rightarrow p(V = 0/1, L_{n_1} = 0/1, L_{n_2} = 0/1) \\ \text{sachant } \sum_{\{v, l_{n_1}, l_{n_2}\} \in \{0,1\}^3} p(V = v, L_{n_1} = l_{n_1}, L_{n_2} = l_{n_2}) = 1 \quad 7 \text{ termes} \quad (7.6)$$

$$p(L_k) \rightarrow p(L_k = 0/1) \\ \text{sachant } p(L_k = 0) + p(L_k = 1) = 1 \quad 1 \text{ terme} \quad (7.7)$$

Lorsque L amers deviennent corrélés, le nombre de termes devient $N - L + 2^{L+1} - 1$, et lorsque $L = N$, on obtient un nombre de termes égal à $2^{N+1} - 1$. Pour 40 amers et un véhicule, le nombre de termes indépendants de la loi de probabilités est égal à environ $2.2 \cdot 10^{12}$. La quantité de mémoire nécessaire pour stocker un tel nombre de composantes est de l'ordre de $8.8 \text{ To} = 8.8 \text{ TB}$, ce qui est gigantesque pour une application de type embarquée. Comme ce nombre de composantes croît de manière exponentielle, pour un ordre de grandeur de 1 000 000 de murs (une grande ville), une telle loi de probabilités n'est pas stockable dans l'état actuel des techniques de l'ingénierie (de l'ordre de 10^{301030} octets seraient nécessaires, ce qui est infiniment plus que la taille de stockage actuellement disponible sur la planète entière, estimée à $\sim 10^{18}-10^{20}$ octets), et n'est même pas représentable de manière humaine en base décimale.

Ce phénomène d'« explosion combinatoire » en $O(2^N)$ pour ce problème de lois de probabilités complètement corrélées est conceptuellement équivalent au problème d'augmentation en $O(N^2)$ de la matrice de covariance de la solution exacte d'un système vectoriel

résolu de manière exacte par un filtre de Kalman. Dans ces deux cas, ce sont les termes de couplage de la loi de probabilité exacte qui font augmenter la complexité de celle-ci au point de ne pouvoir être mise à l'échelle.

Une solution exacte au problème d'estimation de la vraisemblance d'un amer dans une carte est actuellement hors de portée. Il est donc indispensable de pouvoir estimer de manière cohérente cette probabilité, par exemple par le biais d'approximations, qui restent à trouver. On pourrait imaginer utiliser, comme pour un ensemble de systèmes corrélés en interaction un filtre équivalent au filtre à intersection de covariance pour le cas discret. Ce type de filtre ne faisant pas partie de l'état de l'art actuel, d'autres solutions doivent être proposées.

7.2.2 Problème théorique et entropie

Supposons que nous voulions rendre indépendante une loi de probabilités totalement corrélée $p(X, Y)$. Par exemple, imaginons une table de contingence comme celle indiquée dans la table 7.1.

$p(X, Y)$	$X = 0$	$X = 1$
$Y = 0$	0	$\frac{1}{2}$
$Y = 1$	$\frac{1}{2}$	0

TABLE 7.1 – Exemple de loi de probabilités maximalement corrélée.

Si l'on veut préserver la symétrie de cette distribution, tout en la factorisant en écrivant la formule (7.8), la seule solution est donnée par la loi de probabilité indiquée à la table 7.2.

$$q(X, Y) = q_X(X) \cdot q_Y(Y) \quad (7.8)$$

Si l'on calcule les entropies des deux lois de probabilité, on obtient les résultats (7.9) et (7.10).

$$H(p) = 1 \text{ bit} \quad (7.9)$$

$$H(q) = 2 \text{ bits} \quad (7.10)$$

Le processus de factorisation fait perdre un bit d'information, c'est à dire le degré de liberté relatif à un choix binaire. C'est une perte **importante** d'information, puisque **la moitié de l'information a été détruite**. Cette approximation est donc bien plus mauvaise que la loi de départ, et l'information perdue peut être cruciale.

$q(X, Y)$	$X = 0$	$X = 1$
$Y = 0$	$\frac{1}{4}$	$\frac{1}{4}$
$Y = 1$	$\frac{1}{4}$	$\frac{1}{4}$

TABLE 7.2 – Exemple de loi de probabilités indépendante.

7.2.3 Factorisation et sur-convergence

Imaginons que nous prenions comme loi de probabilité approximée de $p(x, y)$ la loi $q(x, y)$. Supposons que la variable Z soit définie comme dépendant de X et de Y , selon la table de probabilités conditionnelles présentée en table 7.3.

$p(Z X, Y)$		$Z = 0$	$Z = 1$
$X = 0$	$Y = 0$	0	1
$X = 0$	$Y = 1$	$\frac{1}{2}$	$\frac{1}{2}$
$X = 1$	$Y = 0$	$\frac{1}{2}$	$\frac{1}{2}$
$X = 1$	$Y = 1$	0	1

TABLE 7.3 – Table de probabilité conditionnelle $p(z|x, y)$.

Si l'on prend la loi exacte $p(x, y)$, et que l'on marginalise $p(x, y, z)$ selon x et y en $p(z)$, on obtient le résultat indiqué dans (7.11) et (7.12), avec une entropie comme présenté dans (7.13).

$$p(Z = 0) = \frac{1}{2} \quad (7.11)$$

$$p(Z = 1) = \frac{1}{2} \quad (7.12)$$

$$H_{\text{exact}}(Z) = 1 \text{ bit} \quad (7.13)$$

Par contre, si l'on prend la loi approchée $q(x, y) = q_X(x) \cdot q_Y(y)$, on obtient les résultats (7.14) et (7.15) et (7.16).

$$p(Z = 0) = \frac{1}{4} \quad (7.14)$$

$$p(Z = 1) = \frac{3}{4} \quad (7.15)$$

$$H_{\text{approx}}(Z) \approx 0.8112 \text{ bits} \quad (7.16)$$

On observe qu'avec l'approximation (7.16), l'entropie est plus faible qu'avec la valeur exacte trouvée dans (7.13). L'approximation q est celle obtenue par le maximum d'entropie; pourtant l'estimation de Z qui en découle possède une entropie plus faible que la valeur obtenue avec p . Ceci est non-conservatif, et provoque sur-convergence et erreur d'estimation.

Lorsque l'on propage de manière **bayésienne** une estimation de type **factorisation**, même dans le cas où cette dernière est un maximum d'entropie, la valeur obtenue peut ne pas être entière. *On ne peut donc pas propager une approximation de manière bayésienne.* Ceci correspond au cas de l'intersection de covariance dans laquelle la fusion est obtenue de manière non-linéaire par processus d'optimisation, à la différence d'une fusion **bayésienne exacte** du type **Kalman**.

Aucunes méthodes connues de l'auteur ne permettent de réaliser une telle fusion en conservant le caractère conservatif de la loi de probabilités. Nous devons donc trouver une autre approche afin de corriger simultanément les problèmes de consanguinité et d'explosion combinatoire.

7.2.4 Réseaux bayésiens couplés et approximations

Dans la littérature, (cf. Brand, 1997, pages 8, 9 et figure 4), le problème des couplages entre réseaux est investigué. Plusieurs structures de réseaux itératifs sont présentées :

- les « Linked Hidden Markov Model » (LHMM), des séquences ordonnées de variables corrélées, que nous ne discuterons pas,
- les « Hidden Markov Decision Trees » (HMDT), des réseaux hiérarchiques de variables,
- les « Coupled Hidden Markov Model » (CHMM), des réseaux en interaction croisée.

Notre cas d'emploi correspond directement aux réseaux *CHMM*, qui sont en interaction croisée à chaque étape du processus d'estimation. On peut voir un réseau de ce type sur la figure 7.2a. Un réseau hiérarchique du type *HMDT* correspond, lui, à la figure 7.2b. L'étape schématisée en pointillés est une étape de « factorisation approximative » que nous souhaitons ajouter afin de garantir l'indépendance des variables manipulées, et donc leur séparabilité.

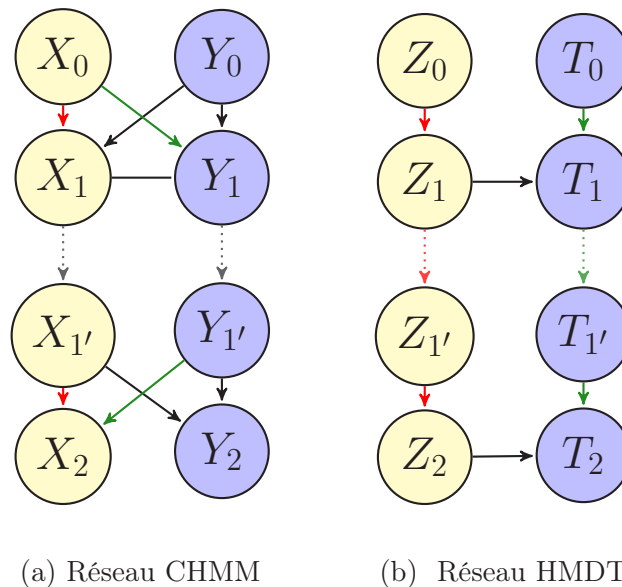


FIGURE 7.2 – Réseau corrélé (CHMM, à gauche) et réseau hiérarchique (HMDT, à droite) avec une étape de décorrélation (flèches en pointillés).

Les réseaux corrélés du type *CHMM* ont un inconvénient majeur. Si, lorsque les coefficients de couplage entre les variables X et Y sont faibles, l'approximation factorisée, pour une seule étape, est relativement satisfaisante, lorsque le degré de corrélation augmente, la qualité de l'approximation se dégrade. Lorsque plusieurs estimations de mauvaise qualité se succèdent, il est possible que l'estimation diverge.

Les réseaux du type *HMDT* sont constitués d'un réseau parent principal, le réseau de la variable Z dans notre exemple, et d'un réseau esclave constitué par la variable T (ou potentiellement plusieurs réseaux esclaves, chacun dépendant de tous les précédents dans l'ordre). Ces réseaux ont la propriété que le réseau parent est vierge de toute corrélation causale venant des autres réseaux ; ce qui implique que celui-ci convergera de la même manière que si les réseaux esclaves n'étaient pas présents. Les approximations d'indépendance dans ce type de réseau sont, en règle générale, meilleures que celles des réseaux

couplés. La variable T peut, en fonction du type d'approximation, sur-converger ou non, mais la variable principale Z reste stable.

On peut donc envisager, afin de **contenir** ou **limiter** l'ampleur du problème de sur-convergence, de formuler le problème en un réseau hiérarchique plutôt qu'en un réseau corrélé. Nous allons passer maintenant à la partie décrivant les diverses tentatives pour obtenir une solution viable au problème de l'estimation simultanée de l'*intégrité de localisation du véhicule* et de la *vraisemblance de l'amer* que l'on tente de détecter.

7.3 Test des solutions proposées

Dans cette section, nous rapportons les résultats des tests des diverses solutions que nous avons simulées de manière aussi réaliste que possible.

7.3.1 Description des simulations réalisées

Un « véhicule » se déplaçant en cercles est simulé, dans un espace limité dans lequel 4 poteaux existants sont détectables, et un cinquième, inscrit dans la carte, n'a pas de contrepartie physique (n'existe pas). Une vue de ce système de simulations est présentée sur la figure 7.3.

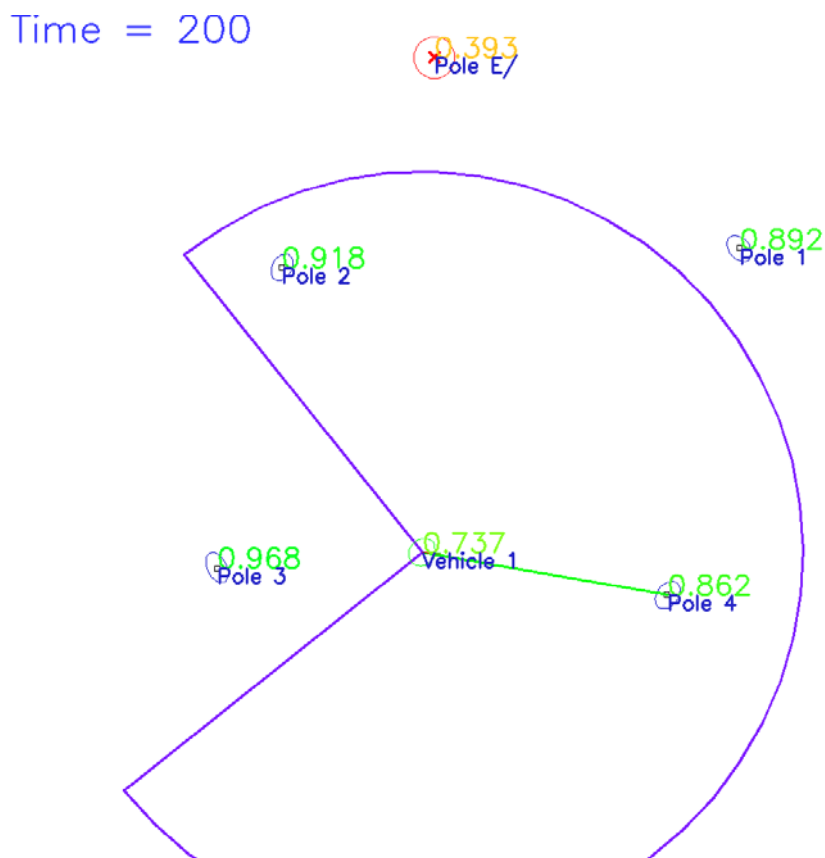


FIGURE 7.3 – Vue d'une simulation. Les poteaux existants sont désignés par « Pole n°1... n°4 », le poteau inexistant (« Pole E/ ») est matérialisé par une croix et un cercle rouges. Le véhicule est désigné par « Vehicule 1 ».

Sur cette figure, un véhicule (au centre) se déplace en cercle à proximité de 4 poteaux et d'un cinquième inexistant et indiqué en rouge. La zone dessinée en violet correspond à la zone du champ LIDAR du véhicule, c'est à dire la zone dans laquelle un poteau doit se trouver afin d'être détectable par le véhicule. Le trait, vert dans ce cas, indique une détection de poteau réussie par le véhicule. Lorsqu'il est rouge, il indique un échec du processus de détection.

Le chiffre coloré à proximité de chaque élément (poteau ou véhicule) indique sa « probabilité d'intégrité », c'est à dire, pour le véhicule, la justesse de sa localisation, et pour les poteaux, la vraisemblance de leur existence. Proche de 1, il est vert. En diminuant, il passe à l'orange, puis proche de 0, au rouge.

Chaque élément est simulé au niveau état continu (état du véhicule ou du poteau), comme au niveau discret (probabilité d'intégrité). Le bruit de mesure (mesure LIDAR d'un poteau) est simulé par un bruit gaussien. Les processus de détection sont simulés par des tirages selon une loi de Bernoulli. L'estimation continue est réalisée par un filtre à intersection de covariance partitionnée.

Pour chaque solution testée, un réseau gérant simultanément la probabilité de localisation correcte du véhicule comme celle indiquant l'existence de l'amer poteau recherché est développé. C'est cette partie d'estimation de probabilité discrète qui est la partie sous test et dont les résultats sont analysés ci-après dans le reste de la section 7.3.

7.3.2 Réseau modifié n°1 avec marginalisation

Le réseau que nous testons ici est présenté à la figure 7.4. Le nœud parent \exists , qui caractérise la présence de l'amer choisi, a été ajouté au nœud enfant z_0 . La nouvelle table de probabilités conditionnelles du nœud z_0 est présentée à la table 7.4.

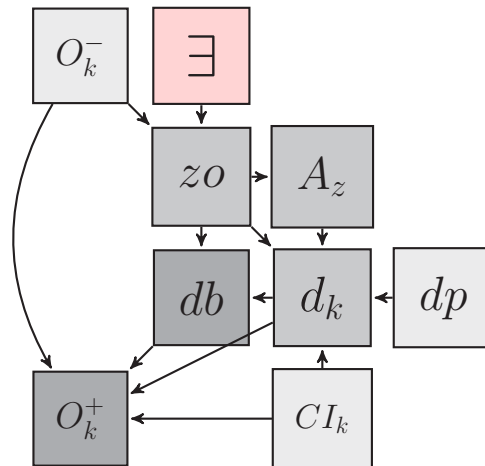


FIGURE 7.4 – Modification du réseau original prenant en compte l'existence de l'amer recherché. Le nœud ajouté est le nœud « \exists ».

- On peut interpréter cette dernière table de probabilités conditionnelles ainsi :
- si l'amer n'existe pas et que le véhicule est au mauvais endroit, de toutes façons, le véhicule ne peut détecter l'amer en question,
 - même si l'amer existe, si le véhicule n'est pas au bon endroit, l'amer ne peut être dans la zone capteur attendue,

O_k^-	\exists	p_{zo}
0	0	0
0	1	0
1	0	0
1	1	$p_{\text{visibilité}}$ selon (5.12)

TABLE 7.4 – Table de loi de probabilité conjonctive pour le nœud zo de la figure 7.4.

- si l’amer n’existe pas, même si le véhicule est au bon endroit, l’amer ne peut se situer dans le champ du capteur,
- par contre, si l’amer existe bien, et que le véhicule est au bon endroit, la probabilité d’être visible dans la zone de recherche du capteur est celle de la méthode de base, $p_{\text{visibilité}}$, selon (5.12).

On observe un couplage **fort** entre le nœud O_k^- et le nœud \exists , par le biais du nœud zo . Lors d’une inférence sur l’évidence d_k , O_k^- et \exists seront fortement corrélés, d’où une forte corrélation entre \exists et O_k^+ .

Comme indiqué à la section 7.2.4, nous allons rendre, après le processus d’inférence, qui va fortement coupler \exists et O_k^+ , ces deux variables indépendantes. Ceci sera fait à chaque étape où le réseau sera mis à jour par une observation sur d_k . Ici, nous choisissons de rendre les deux variables O_k^+ et \exists indépendantes avec l’ensemble d’équations (7.17), (7.18) et (7.19).

$$q(O_k^+, \exists) = q_{O_k^+}(O_k^+) \cdot q_{\exists}(\exists) \quad (7.17)$$

$$q_{O_k^+}(O_k^+ = 1) = p(O_k^+ = 1, \exists = 0) + p(O_k^+ = 1, \exists = 1) \quad (7.18)$$

$$q_{\exists}(\exists = 1) = p(O_k^+ = 0, \exists = 1) + p(O_k^+ = 1, \exists = 1) \quad (7.19)$$

Une simulation de ce réseau bayésien donne le résultat visible sur la figure 7.5.

On observe plusieurs choses sur la figure 7.5 :

1. la probabilité d’existence des poteaux de 1 à 4 passe très vite à 1 (moins de 10 sec),
2. la probabilité d’intégrité de la localisation du véhicule reste proche de 1 ; les fluctuations de celle-ci vers le bas sont provoquées par des échecs de détection (simulés selon une loi de Bernoulli) ;
3. **bien que le processus de détection du poteau inexistant échoue systématiquement, sa probabilité d’existence tend (à l’échelle de 100 sec) vers 1.**

L’item n°3 est particulièrement problématique, puisque, que l’amer que l’on tente de détecter existe ou non, sa probabilité d’existence tendra vers 1. C’est à dire que **l’on sera certain de l’existence des amers, même s’ils ne sont jamais détectés**. On ne peut donc pas utiliser un tel processus d’estimation.

On peut analyser l’origine de ce problème comme suit :

- l’inférence sur d_k corréle O_k^+ et \exists (spécialement lorsque $d_k = 1$), ce qui correspond à l’ajout d’information par une mesure dans un processus de filtrage,
- mais cette corrélation est **enlevée** par le processus qui rend \exists indépendante de O_k^+ , on revient donc à une estimation dans laquelle on a enlevé l’information ajoutée par la mesure ; comme ce processus n’est pas rigoureusement exact, une dérive de l’estimation peut se produire (dérive lente à la marge, qui est observée),

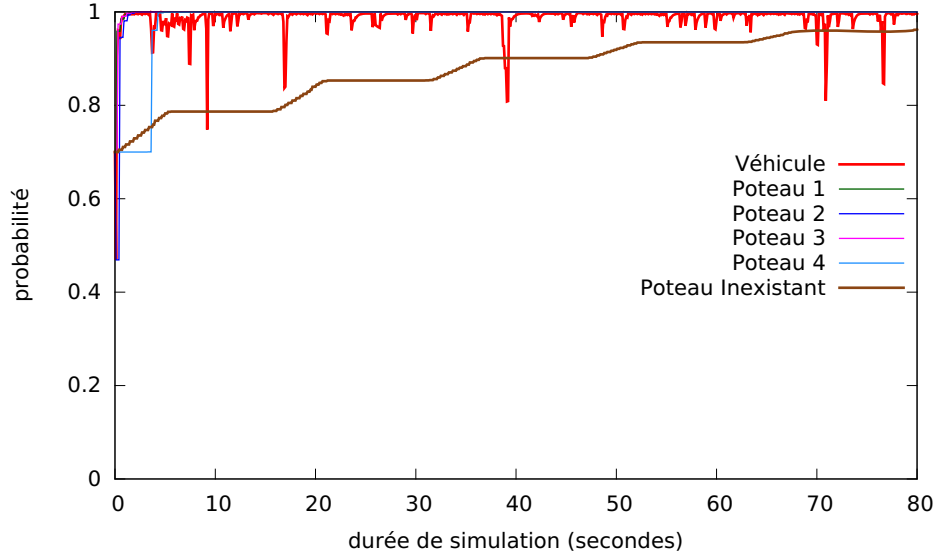


FIGURE 7.5 – Résultat de simulation obtenu pour le réseau de la figure 7.4 avec les équations de marginalisation (7.17), (7.18) et (7.19). La probabilité d’existence des poteaux de 1 à 4 est quasiment confondue avec l’axe $y = 1$, et de ce fait difficile à observer.

- enfin, on peut penser que l’approximation par marginalisation est mauvaise, car elle ne reflète pas le principe du maximum d’entropie.

Les deux premiers éléments sont intrinsèques du réseau choisi et de la manière de gérer les probabilités comme des objets séparés. Par contre, on peut penser améliorer le point n°3 en passant d’une approximation par marginalisation à une approximation de type « mean-field ». Celle-ci maximise l’entropie de la solution dans laquelle les variables sont indépendantes. C’est cette seconde solution que nous allons tester dans la section suivante.

7.3.3 Réseau modifié n°1 avec approximation « mean-field »

Dans cette section on reprend les éléments de la section 7.3.2, mais en remplaçant les équations (7.17), (7.18) et (7.19) par la formule d’approximation « Mean-Field » (4.83). On obtient le résultat indiqué sur la figure 7.6.

L’approximation de type « mean-field », va avoir tendance à factoriser une loi de probabilité composite par un produit de probabilités indépendantes, chaque probabilité étant la plus proche possible de $\frac{1}{2}$. Cela va donc avoir tendance à rapprocher les lois de probabilités vers l’état de méconnaissance maximale.

Bien que l’on n’affiche que les 15 premières secondes de résultats obtenus avec cette méthode sur la figure 7.6, l’intégralité de la simulation reflète le même comportement. On observe sur la figure 7.6 que :

- la probabilité d’existence des poteaux qui existent fluctue fortement autour de 0.8, ce qui peut indiquer que l’approximation « mean-field » les estime de la manière la plus conservatrice (éloignée de 1 et proche de 0.5),
- la probabilité d’intégrité de localisation du véhicule fluctue entre 0.1 et 0.9 de manière uniforme, sans position de préférence, alors même que l’état du véhicule est correctement estimé,

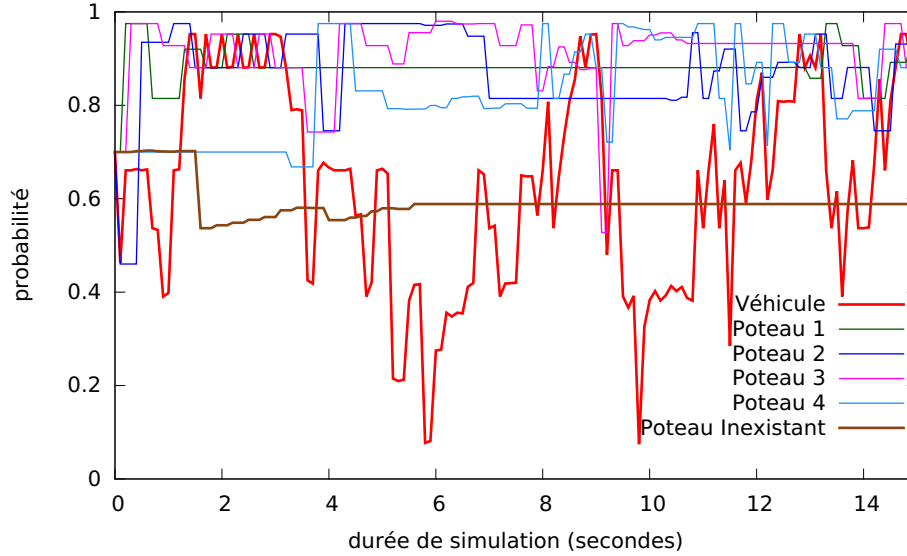


FIGURE 7.6 – Résultat obtenu avec le réseau n°1 et une approximation de type « mean-field ».

- la probabilité d’existence du poteau inexistant se stabilise entre 0.4 et 0.6, c’est à dire dans l’état d’entropie maximale, et de méconnaissance maximale, alors même que ce « poteau » n’est jamais détecté et que l’on peut raisonnablement estimer qu’il n’existe pas.

Cette approximation est bien plus conservatrice que la précédente. Au lieu de stabiliser le système, en retirant plus d’information que la précédente, tout en le conservant cohérent, elle supprime l’information importante et le met dans un état de méconnaissance maximale.

Une autre approche doit donc être trouvée afin de mieux résoudre ce problème. C’est ce que nous allons présenter dans la section suivante.

7.3.4 Réseau hiérarchique n°2

Dans cette section, nous rappelons les éléments de la section 7.2.4. Deux éléments nous apparaissent importants dans celle-ci :

- les réseaux hiérarchiques semblent être plus immunes aux problèmes de sur-convergence que les réseaux couplés,
- lorsque les corrélations sont faibles, les approximations factorisées (indépendantes) sont meilleures que lorsque les corrélations sont fortes.

Dans le réseau de la figure 7.4, la structure de corrélation est de type couplée. Ensuite, les corrélations entre les nœuds \exists et O_k^- sont fortes lorsque la probabilité de visibilité l’est.

On peut donc imaginer tester maintenant un système :

- avec un réseau de type hiérarchique,
- dans lequel le coefficient de couplage est faible.

Une idée qui vient en s’inspirant de la structure d’un intégrateur en automatique consiste à se rapprocher de cette structure. On peut, avec un intégrateur, utiliser de petits effets (coefficient de couplage faible), qui, sur une plus longue période de temps, font augmenter la réaction intégrale.

En décalant légèrement les coefficients de la valeur zéro, on peut transformer ce type de système en passe bas avec un temps caractéristique d'un certain ordre de grandeur. L'équation d'un tel système serait celle indiquée dans (7.20). La valeur δ est égale à l'inverse d'un temps caractéristique dans celle-ci.

$$\frac{dx}{dt} = \begin{cases} \delta \cdot (1 - x) & \text{si probabilité source à 1} \\ \delta \cdot (0 - x) & \text{si probabilité source à 0} \end{cases} \quad (7.20)$$

On peut imaginer utiliser comme variable x la probabilité Ξ , dans le cadre d'une discrétisation, en remplaçant la variable t représentant le temps par un pas d'itération entre une estimation *a-priori* et une estimation *a-posteriori*. Il faut également une probabilité source que Ξ devra suivre. On peut penser au nœud db , qui représente la probabilité d'une détection avec association de l'élément détecté à l'amer recherché. Quand $db = 1$, le bon amer est détecté, et l'on voudrait faire augmenter Ξ vers 1. Quand $db = 0$, soit il n'y a pas eu de détection, soit il est certain que c'est un mauvais élément qui a été détecté, et dans ce cas on veut faire diminuer Ξ vers 0.

Le réseau présenté à la figure 7.7 représente cette idée.

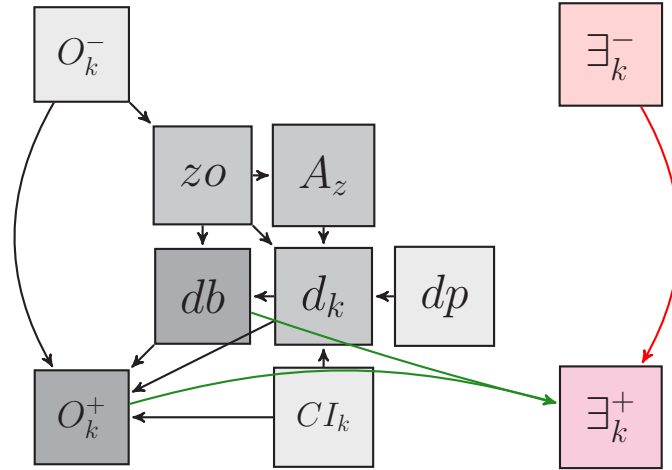


FIGURE 7.7 – Réseau n°2 de type hiérarchique. Le réseau relatif à la variable Ξ_k , à droite en couleurs chaudes, représentant l'existence de l'amer choisi, est esclave du réseau principal relatif à l'intégrité de localisation du véhicule. La structure globale est similaire au réseau Z_*-T_* de la figure 7.2.

Dans la structure hiérarchique représentée sur la figure 7.7, on observe que :

1. la structure du réseau est similaire à celle du réseau Z_*-T_* de la figure 7.2,
2. le réseau relatif à l'intégrité de localisation du véhicule est inchangé par rapport au réseau décrit à l'annexe B,
3. le réseau relatif à l'existence de l'amer choisi est en « aval » du précédent, en ce qu'il reçoit des informations venant de ce dernier (flèches vertes), mais que ce dernier ne dépend à aucun moment de la variable Ξ ;
4. bien que ce ne soit pas visible, chacune des estimations pour O_k^+ et Ξ_k^+ , une fois rendues indépendantes l'une de l'autre, sont reprises comme *a-priori* respectifs pour O_{k+1}^- et Ξ_{k+1}^- .

Ce qui est intéressant avec cette structure, c'est que la probabilité de « double comptage » d'information est limitée. Reprenons tout d'abord l'exemple de la section 4.7.4.

Lorsqu'une estimation de deux variables en interaction, X et Y , traitées séparément par deux filtres de Kalman ignorant délibérément les corrélations induites, le système sur-converge. En fait, une partie de l'information relative à la mesure est transmise à la variable X en premier. Ensuite, la même information est transmise à la variable Y . Or cette nouvelle estimation de Y dépend de la nouvelle valeur X' de X . On transmet donc plus d'une fois l'information commune de mesure.

Dans le réseau couplé de la figure 7.2a, il existe de tels chemins d'information double. Par exemple, supposons que l'information du nœud X_0 passe aux nœuds X_1 (flèche rouge) et Y_1 (flèche verte). Tant que la corrélation entre ces deux derniers nœuds existe, il n'y a pas de double comptage. Lorsque la corrélation est supprimée, l'information est présente en double dans les nœuds $X_{1'}$ et $Y_{1'}$. Lorsque le nœud X_2 est calculé, la prise en compte de manière double de cette information peut le faire sur-converger.

Dans le réseau hiérarchique de la figure 7.2b, le réseau maître Z_* ne peut absolument pas sur-converger, car il ne reçoit à aucun moment d'informations venant d'une seconde source. Le réseau T_* pourrait, lui, sur-converger, bien que ce soit plus modéré que pour le réseau couplé précédent (X et Y). En effet, il n'y a pas de double comptage (aucun chemin en \mathbf{X}) de manière directe. Le seul double comptage possible intervient avec la factorisation $Z_1-T_1 \rightarrow Z_{1'}-T_{1'}$. Pour limiter cela, un faible couplage avec la variable Z_* limite cet effet.

La table de probabilités pour le nœud Ξ_k^+ est donnée à la table 7.5. Celle-ci s'analyse ainsi :

- pour les lignes 1 et 2 de la table ($O_k^+ = 0$), puisque le véhicule n'est pas intègre en localisation, Ξ_k est inchangé, c'est à dire que $p(\Xi_k^+)$ est inchangée par rapport à $p(\Xi_k^-)$, puisque $p(\Xi_k^+ = 1 | O_k^-, db, \Xi_k^-)$ est égale à la valeur du nœud parent associé Ξ_k^- ,
- pour les lignes restantes, $O_k^+ = 1$, donc l'estimation du véhicule est intègre ; on peut donc mettre à jour le nœud Ξ_k^+ de la manière suivante :
 - si $db = 0$ et $\Xi_k^- = 0$, on n'a pas détecté le bon amer et l'existence *a-priori* est nulle, il n'y a aucune raison de faire augmenter Ξ_k^+ , elle reste donc à 0,
 - si $db = 0$ et $\Xi_k^- = 1$, on n'a pas détecté le bon amer, il faut donc faire diminuer (légèrement) Ξ_k^+ ; on utilise la probabilité δ proche de 1 par borne inférieure,
 - si $db = 1$ et $\Xi_k^- = 0$, on a détecté le bon amer, il faut donc faire augmenter (légèrement) Ξ_k^+ ; on utilise donc $\bar{\delta}$ qui est proche de 0 par borne supérieure,
 - si $db = 1$ et $\Xi_k^- = 1$, on a détecté le bon amer et Ξ_k^- est déjà maximale, on laisse donc Ξ_k^+ à 1 inchangée par rapport à Ξ_k^- .

Le coefficient de corrélation entre le sous-réseau relatif au véhicule, de la figure 7.7, et le sous-réseau esclave Ξ , est égal à $\bar{\delta} = 1 - \delta$, que l'on choisit proche de $\mathbf{0}^+$.

O_k^+	db	Ξ_k^-	$p(\Xi_k^+ = 1 O_k^-, db, \Xi_k^-)$
0	ϕ	0	0
0	ϕ	1	1
1	0	0	0
1	0	1	δ
1	1	0	$\bar{\delta} = 1 - \delta$
1	1	1	1

TABLE 7.5 – Table de probabilités conditionnelles du nœud Ξ_k^+ .

Le résultat obtenu pour ce réseau bayésien hiérarchique dans les simulations est présenté sur la figure 7.8. Sur cette figure, on observe que :

- la probabilité d’existence des poteaux n°1 à n°4 reste stable autour de 0.9, ce qui est proche de la probabilité complémentaire de *faux positifs* (c’est à dire la probabilité de détection),
- la probabilité d’existence du poteau inexistant n’augmente jamais et tend vers 0 au fur et à mesure des échecs de détection,
- la probabilité de localisation correcte reste la plupart du temps très proche de 1 ; sauf lorsque plusieurs fois d’affilée les tentatives de détection des poteaux échouent, ce qui fait baisser la probabilité en question, mais elle remonte à 1 dès que des succès de détection se produisent.

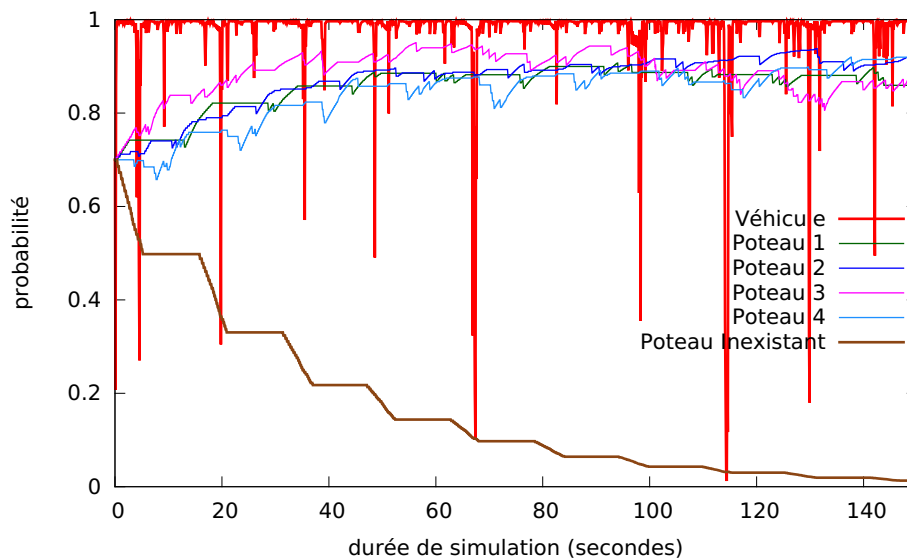


FIGURE 7.8 – Résultats de simulation obtenus pour le réseau hiérarchique n°2.

Il est à noter que le phénomène de détection est simulé de manière assez pessimiste dans le cadre de la simulation. Alors que dans la réalité le taux α de *faux négatifs* est là pour pallier des manques au niveau de l’estimation de visibilité (polygone de visibilité ne rendant pas compte totalement de celle-ci, buissons occultants), dans la simulation, c’est une vraie loi de Bernoulli qui est utilisée. Ceci rend possible, bien que peu fréquent, des séries successives d’échecs de détection, tandis que dans la réalité, cela est très rare, voire inexistant, les échecs étant quasi-toujours générés par des problèmes de visibilité (buisson cachant le mur, véhicule trop éloigné ou visibilité mal estimée).

À l’heure actuelle (où ce rapport est rédigé) nous ne disposons d’**aucune solution fiable permettant de connaître**, avant de tenter de le détecter, la **visibilité d’un mur**, ou de pouvoir l’**apprendre** par « estimation successive » de quelque manière que ce soit. Nous pensons qu’il s’agit d’un **point bloquant majeur** pour les éventuels travaux à venir. Nous estimons que cette réserve est importante.

La visibilité calculée *a-priori* ne tient aucun compte des éléments occultants, comme les végétaux. Par exemple, au NORD du site d’expérimentation de PAVIN, un rond-point est pourvu de buissons, ce qui empêche un faisceau LIDAR de pouvoir le traverser. Or, la manière de gérer la visibilité, conforme à la section 5.3.1, ne prend pas en compte cet élément. Autre exemple, un mouvement de terrain existe sur le versant EST du site,

provoquant des impacts LIDAR au sol lorsque le véhicule est situé sur la voie la plus à l'EST. Étant donné que le capteur LIDAR est positionné, sur le véhicule, à seulement 40 centimètres au-dessus du sol, tout mouvement de roulis, tangage du véhicule, comme également les dénivellations du terrain, vont provoquer des impacts au sol. Ces mouvements de terrain peuvent rendre inobservables les murs à l'OUEST (qui sont alors surélevés et hors de porté LIDAR), vus depuis la voie la plus à l'EST du site (en aval). Ces deux phénomènes ne sont actuellement pas pris en compte et seraient un intéressant sujet pour de futurs travaux.

Nous estimons donc que cette manière de calculer l'existence des amers observés permet de pallier les problèmes observés dans les techniques précédentes, tout en permettant une estimation raisonnable de celle-ci. Il est à mentionner que le paramètre $\bar{\delta}$ doit être le plus proche de zéro afin de limiter la sur-convergence. Nous allons maintenant passer à la partie expérimentale, qui implémente cette technique.

7.4 Expérimentation sur véhicule

7.4.1 Description

Nous avons implémenté dans l'état de chaque amer de type mur une valeur stockant la vraisemblance de l'existence de celui-ci. Leur probabilité d'existence est initialisée à 70%, soit 0.7. Le coefficient $\bar{\delta}$ est mis à la valeur de 4% soit 0.04. Lorsque la probabilité d'existence d'un mur passe en dessous de la valeur de 5% soit 0.05, il est **supprimé de la carte**. Un réseau conforme à la section 7.3.4, figure 7.7 a été implémenté, et est mis à jour à chaque tentative de détection de l'un d'entre eux.

Nous avons fait circuler un véhicule VIPALAB-3 à proximité d'un mur totalement caché par les buissons (mur 119). D'autres murs étaient parfaitement visibles dans le voisinage (mur 114 par exemple).

7.4.2 Résultats

Nous présentons ici l'évaluation de l'existence d'un mur. Avant de démarrer les explications, nous précisons que le véhicule utilisé (VIPALAB 3 n°4) est équipé d'un LIDAR plan à l'avant du véhicule. De ce fait, les murs ne sont observables que lorsque positionnés dans une portion de demi-plan à l'avant du véhicule et suffisamment proches de ce dernier. On note expérimentalement que la distance de laquelle un mur peut être vu avec le capteur LIDAR installé sur le véhicule est de l'ordre de 20 à 30 mètres. Celle-ci dépend également de l'incidence sous laquelle celui-ci est vu ainsi que du coefficient de réflectivité de la surface de ce dernier pour la longueur d'onde utilisée par le LIDAR (la source lumineuse de celui-ci est un LASER à semi-conducteurs émettant dans le proche infra-rouge).

Nous nous intéressons aux murs 114 et 119, représentés sur la carte en haut à gauche de la figure 7.10, et représentés respectivement en vert et en rouge, pour lesquels la visibilité théorique peut être correctement estimée. Le mur 119 est caché en totalité par un buisson, mais existe bien. De ce fait, si le mur en question n'est pas détecté, sa probabilité d'existence va probablement chuter. Étant donné que l'on ne peut, dans la configuration actuelle, ni ajouter un mur inexistant à la base de données gérant la carte, sans la corrompre, ni ajouter à l'initialisation un mur excédentaire (non prévu au départ des développements), nous nous sommes rabattus sur cette solution. Celle-ci permet, de notre

point de vue, de valider tout de même que, lorsqu'un mur qui doit être visible n'est pas détecté, sa probabilité d'existence doit tendre vers 0 au fur et à mesure des tentatives de détections. Le mur 114, quant à lui, est bien visible de manière maximale.

Il est à noter que la probabilité d'existence d'un mur est modifiée en fonction de sa **visibilité estimée**, qui est issue d'un modèle statique déterministe (modèle théorique **a-priori** de type géométrique, et non appris). Plus un mur aura une probabilité (théorique) de visibilité élevée, plus le résultat de détection de ce dernier aura un impact élevé sur l'évolution de sa probabilité d'existence.

Sur la figure 7.9, sont comparées deux vues de dessus de la carte obtenues par l'exécution sur véhicule, l'une avant et l'autre après la suppression du mur 119. Le mur 119 n'est jamais détecté; lorsqu'il passe en dessous de 5% de probabilité d'existence, il est supprimé, à l'instant indiqué ⑫ sur la figure 7.10.

Le mur 114, quant à lui, est quasiment certain d'exister (88.4% de probabilité d'existence). A la fin du jeu de données, il frôle la valeur de 100%. Sur la figure 7.10, on peut voir la courbe des probabilités des deux murs. Cette figure détaille (au centre) la probabilité d'existence des deux murs 114 et 119. De plus, elle indique certains points temporels particuliers de la trajectoire, durant lesquels les murs en question sont théoriquement observables. Ceux-ci sont indiqués par des traits verticaux, reliés à des bulles contenant une référence chiffrée. Sur les quatre cartes (deux au-dessus, deux au-dessous), les portions de trajectoires relatives à ces références temporelles sont indiquées, et la position correspondante estimée du véhicule est également tracée.

On observe que la probabilité d'existence du mur 119 ne fait que décroître. En fait, ce dernier n'est jamais détecté, à aucun des quatre passages du véhicule en visibilité de celui-ci. La diminution de sa probabilité d'existence est donc conforme aux simulations réalisées, décrites à la section 7.3.4.

Examinons plus en détail l'évolution de la probabilité d'existence du mur 114. Lors du premier passage, entre les instants ② et ③, carte en haut à gauche, on observe une augmentation de la probabilité d'existence de ce dernier, qui passe de 70% à 81.4%. Durant son second passage, instants ⑤ à ⑥, carte en haut à droite, sa probabilité d'existence chute à 74.7% car on échoue à détecter le mur en question à plusieurs reprises. En regardant en détail les données et la configuration géométrique, on observe que

- le nombre de spots LIDAR sur ce mur est relativement faible, en raison :
 - d'une incidence quasi-rasante du LIDAR sur ce dernier (le secteur angulaire sous lequel il est vu est matérialisé sur la carte en haut à droite de la figure 7.10, celui correspondant au mur 119 est ajouté pour comparaison),
 - de l'occultation partielle de celui-ci par un boîtier technique, indiqué par un carré noir à proximité, (même carte que précédemment);
- il y a un nombre important de spots LIDAR dans la zone de recherche, correspondant à d'autres éléments (le boîtier précédemment mentionné, le mur 115 au sud).

On peut comprendre que dans ces conditions d'incidence rasante, cause principale de l'échec de détection, la probabilité d'existence de ce mur puisse diminuer.

Les troisième et quatrième passages du véhicule en visibilité de ce mur (instants ⑨ à ⑩, puis ⑬) font à nouveau remonter la probabilité d'existence de ce mur suite aux multiples détections franches de celui-ci. La probabilité d'existence du mur 114 à la fin de la trajectoire est de 98.2%. Ceci est l'effet recherché, et montre que le principe de base de la méthode est valable.

Néanmoins, on peut songer à quelques améliorations :

1. le calcul de la visibilité pourrait être amélioré, notamment :
 - en estimant « mieux » la probabilité de visibilité des murs,
 - en prenant en compte les occultations possibles ;
2. les éléments ou phénomènes occultants pourraient être cartographiés (végétaux et autres, mouvements de terrain).

Pour l'élément n°1, une idée serait de calculer en temps réel le nombre théorique de points d'impact LIDAR prévus correspondant au mur dans les données LIDAR. Ceci correspond de manière directe au secteur angulaire sous lequel le mur est vu. Dans ces expérimentations, cette probabilité de visibilité est calculée conformément à ce qui est expliqué à la section 5.3.1, ce qui pose problème dans certains cas.

Pour l'élément n°2, on pourrait tenir compte d'éléments occultants comme les autres murs de la carte, ainsi que des autres occultants (végétaux comme au centre du rond-point de la plateforme d'expérimentation). Ces deux améliorations seront intégrées dans les contributions du chapitre 8.

La cartographie des éléments occultants, bien qu'importante, a été considérée hors du périmètre des travaux rapportés dans ce mémoire. Mais elle revêt un intérêt particulier, étant donné que cette notion de calcul de visibilité, primordiale au fonctionnement de cette approche de localisation et de cartographie, en dépend directement.

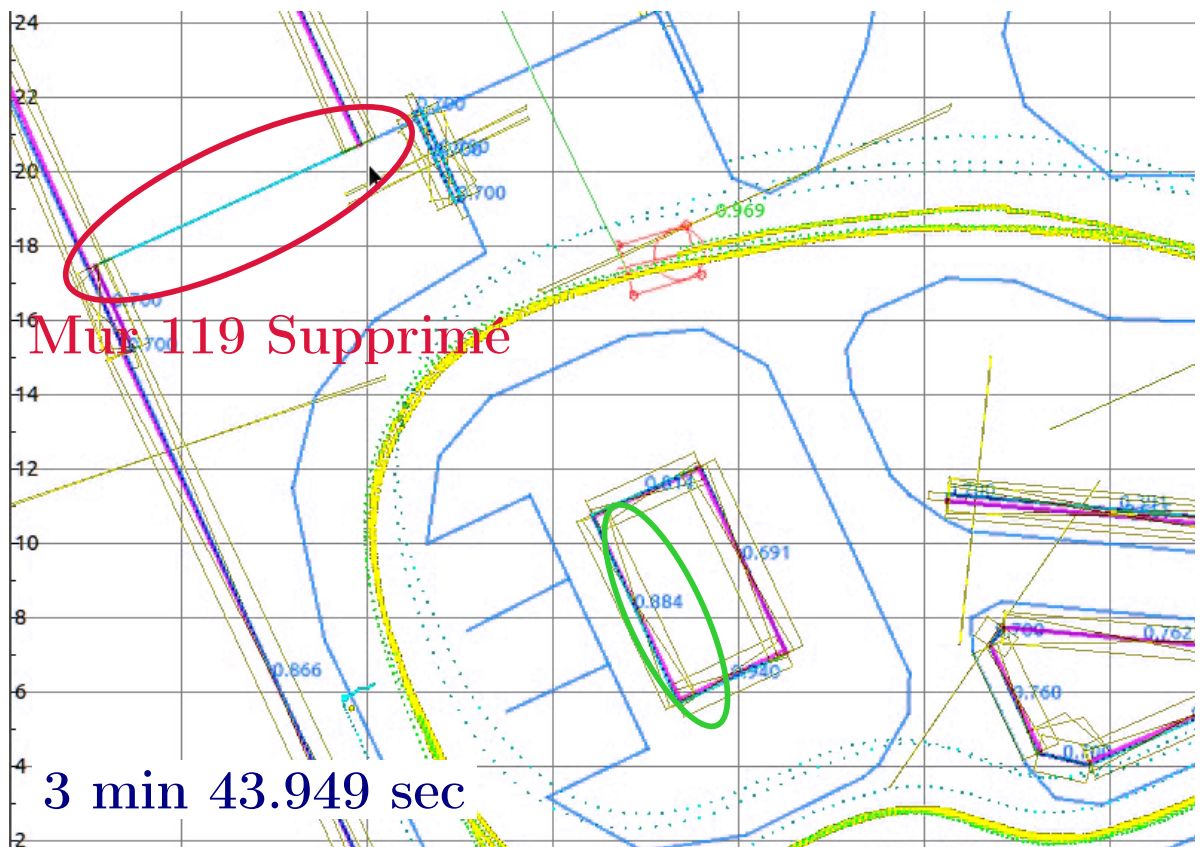
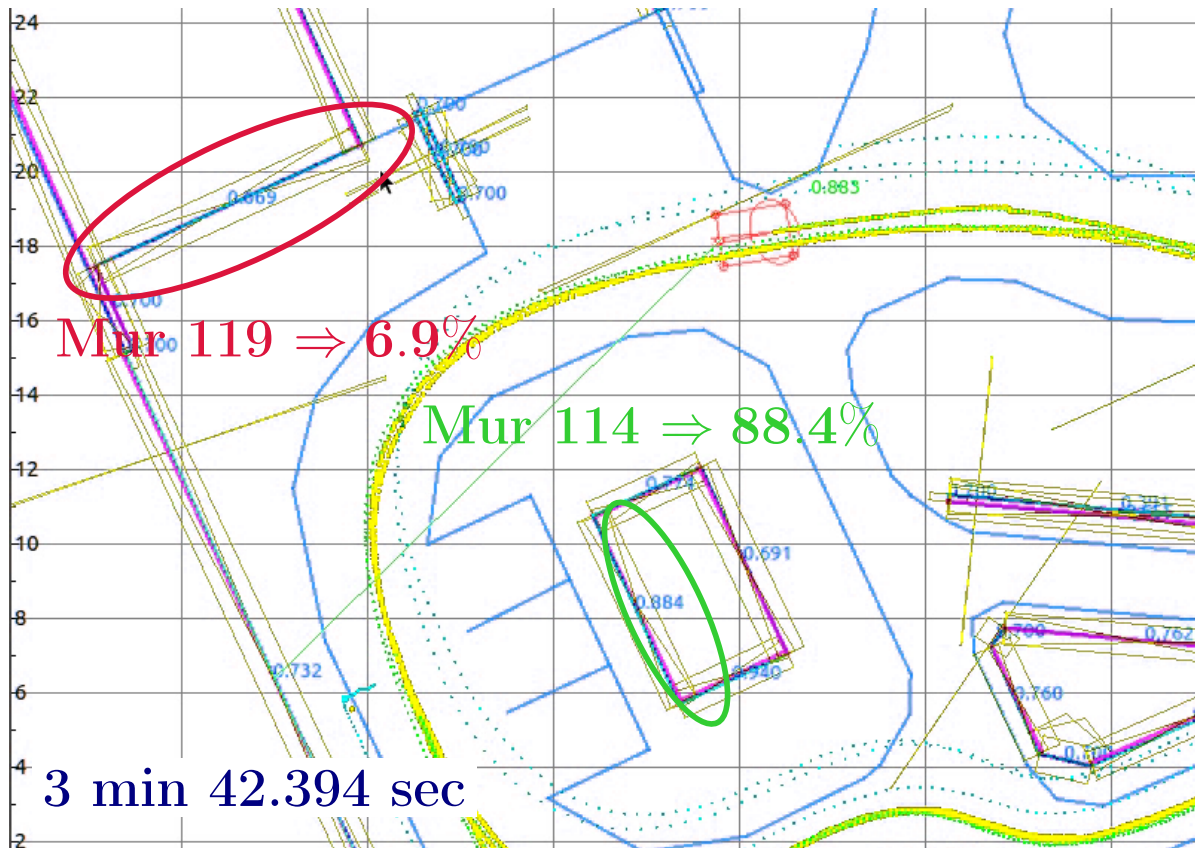


FIGURE 7.9 – Suppression du mur 119. En haut, vue de la carte avant la suppression, en bas après.

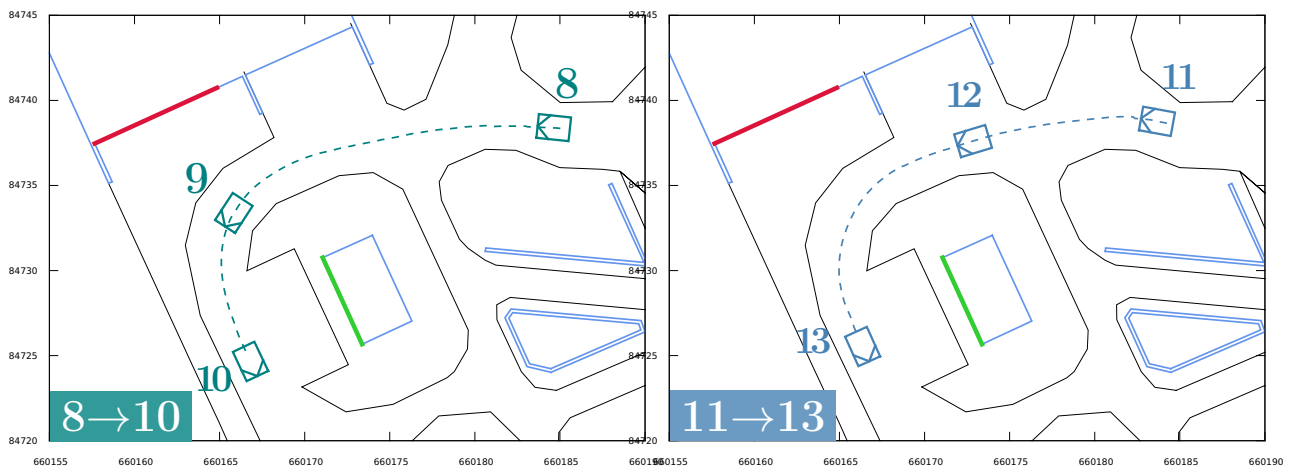
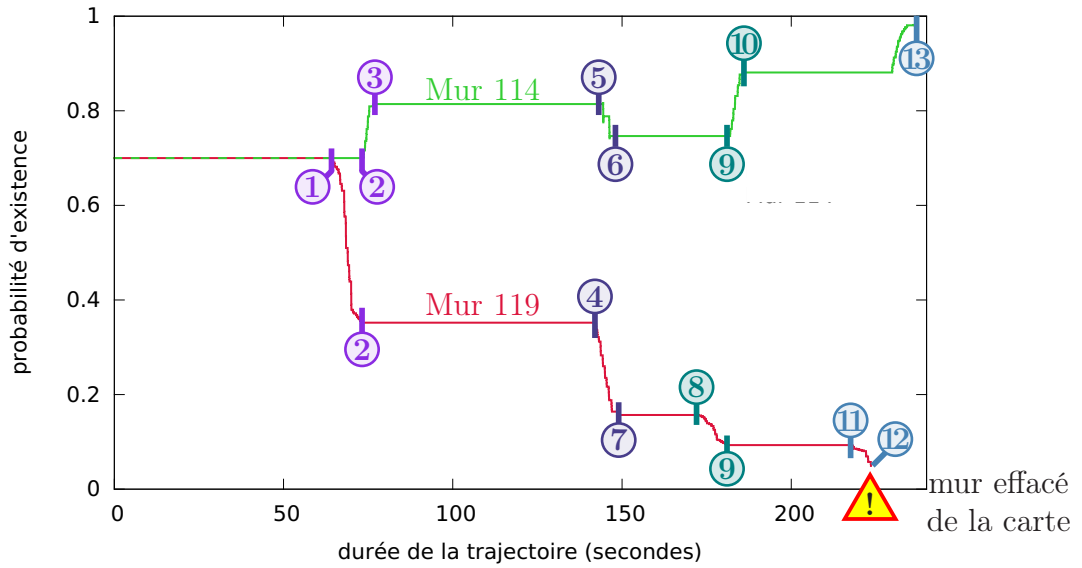
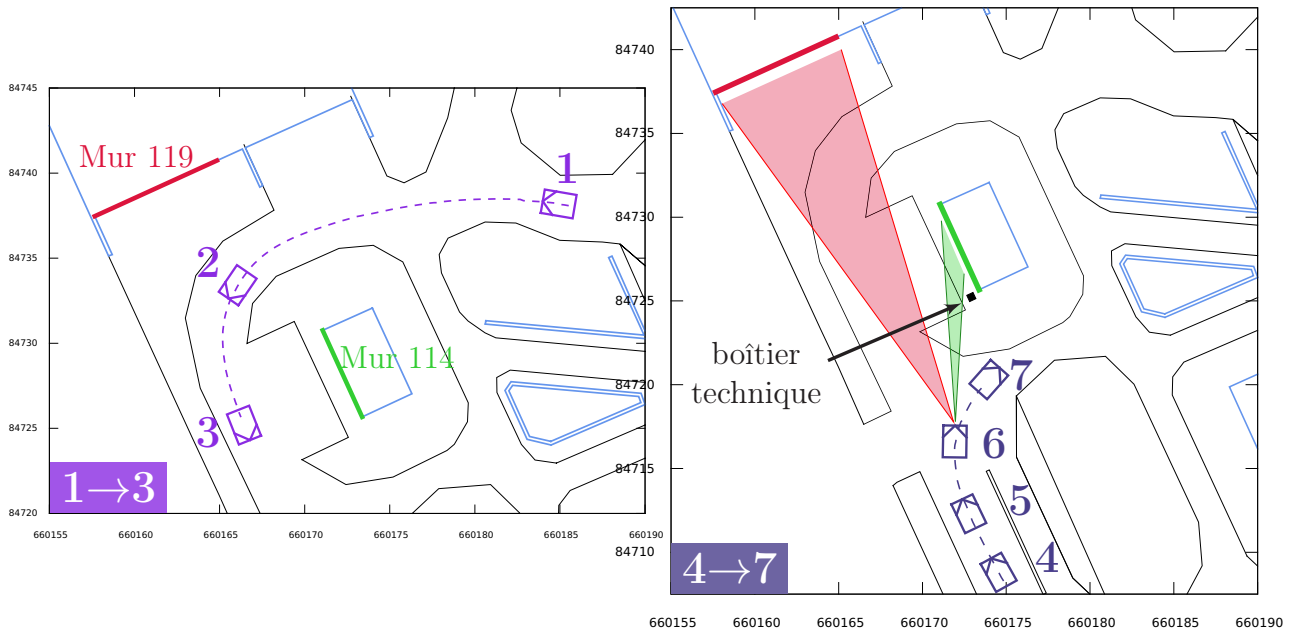


FIGURE 7.10 – Tracé de la probabilité d'existence des murs 114 et 119 au cours du temps. Quatre cartes, deux au-dessus et deux au-dessous du tracé principal, montrent les trajectoires des passages du véhicule à proximité des murs.

Conclusion

Nous avons dans ce chapitre tenté de montrer pourquoi l'estimation de probabilités d'existence de murs pour une carte complète est problématique, l'explosion du nombre de termes correspondant à une analyse exacte. Notre objectif ici est de trouver une formulation de ce problème qui puisse s'abstraire de cette limite, tout en permettant de résoudre, même de manière approchée, celui-ci. Après avoir montré les différents problèmes potentiels de manière théorique, nous avons conduit des simulations mettant en évidence ceux-ci. Nous avons proposé une solution pour minimiser les potentiels problèmes de sur-convergence qui peuvent se produire.

Cette solution, testée en simulation, a montré de bonnes performances. En test réel sur véhicule, ces propriétés ont été confirmées. Nous estimons donc que, malgré le fait qu'en théorie, la sur-convergence reste possible, nous avons fortement réduit son occurrence de manière à pouvoir disposer d'une estimation la plus robuste possible, à notre niveau.

Jusqu'à présent, nous avons travaillé sur la correction d'une carte imprécise ou fausse. Nous allons maintenant passer au volet relatif à l'enrichissement de la carte, qui fait l'objet de notre prochain chapitre.

Chapitre 8

Ajout d'Amers dans la Carte

Les deux précédents chapitres étaient dévolus à la correction d'informations erronées inscrites dans la carte de l'environnement. Par exemple en corrigeant la position de murs positionnés de manière légèrement erronée, ou bien en supprimant les amers totalement absents ou introuvables.

Dans ce chapitre, nous allons nous attaquer au pendant additif des deux précédents items :

1. ajouter des éléments existants dans l'environnement, éléments que la carte ne possède pas,
2. le faire à un positionnement cohérent, c'est à dire tel que l'estimation soit conforme à la réalité.

Nous allons d'abord introduire brièvement le concept, puis détailler comment nous avons implémenté cette fonction. Finalement, deux tests en grandeur réelle seront commentés.

8.1 Introduction

Dans ce chapitre, nous allons nous attacher à ajouter de l'information inexistante dans la carte de l'environnement. On peut tout à fait imaginer, à partir des éléments mesurés par le LIDAR, de reconstruire une carte dans son ensemble.

Nous utilisons pour localiser un véhicule et corriger les paramètres erronés des murs une méthode TOP-DOWN. Cette méthode choisit à *l'aveugle, sans les données de mesure*, et À L'AVANCE, quel amer (mur) elle va tenter de détecter. Le processus de détection ne s'effectue qu'une fois cet amer choisi, et que les données de mesure, arrivées postérieurement à ce choix, sont fournies par le capteur.

Or, pour ajouter des éléments inexistants dans une carte, on ne peut bien évidemment pas choisir de se focaliser sur un élément dont la carte n'a aucune connaissance. Il nous faut donc employer une méthode du type BOTTOM-UP, une méthode qui analyse dans son ensemble toutes les données venant du capteur, et non uniquement celles auxquelles l'on s'attend.

Une remarque ici sur la méthode TOP-DOWN utilisée jusqu'à présent. Cette méthode a un *a-priori* fort sur le fait que les éléments cartographiés dans la carte existent. Ce choix, même valable, n'est pas neutre : on ne cherche pas à savoir *si l'amer cartographié existe*, mais uniquement *si ce dernier n'existe pas*. Ceci crée donc une asymétrie entre CARTOGRAPHIÉ-EXISTE et NON CARTOGRAPHIÉ-N'EXISTE PAS d'une part, et

CARTOGRAPHIÉ–N’EXISTE PAS et NON CARTOGRAPHIÉ–EXISTE d’autre part, ce qui peut, dans certaines situations observationnelles créer un **biais fort**. Un exemple typique serait une carte trop ancienne totalement fausse. Un autre exemple serait le démarrage à partir d’une carte vierge.

Le processus TOP–DOWN a pour *a-priori* les éléments CARTOGRAPHIÉ–EXISTE et NON CARTOGRAPHIÉ–N’EXISTE PAS, et le processus décrit au chapitre 7 va analyser l’hypothèse CARTOGRAPHIÉ–N’EXISTE PAS. Le processus BOTTOM–UP s’attaque au couple de choix NON CARTOGRAPHIÉ–EXISTE. Nous remarquons l’extrême asymétrie qui existe entre l’approche focalisante de type TOP–DOWN d’une part, et de l’autre l’approche BOTTOM–UP, qui ne visent pas les mêmes objectifs, ni ne fonctionnent de manière semblable. Ceci nous semble problématique, selon le point de vue qu’aucun biais ni *a-priori* ne devrait exister du point de vue de la carte. C’est à dire, que la carte soit exacte ou non, les différents algorithmes devraient fonctionner de manière identique pour confirmer ou infirmer les hypothèses, et non de manière asymétrique (un algorithme pour confirmer, un autre pour infirmer).

Notre manière de prendre cette disparité en compte consiste à rajouter un processus qui va détecter les éléments existants dans la réalité, mais non encore dans la carte. Comme ce processus doit analyser la totalité de ce qui existe dans le champ des capteurs, nous allons réaliser cette opération par un processus BOTTOM–UP. Celui-ci sera exécuté de temps à autres, quand cela est possible, et donc ne donnera pas systématiquement lieu à analyses et ajout de murs.

Dans la suite de ce chapitre, nous allons décrire les différentes opérations nécessaires pour extraire les murs à partir d’un processus permanent de recherche systématique de ceux-ci dans **toutes** les données LIDAR accessibles. Celui-ci fonctionnera en *tâche de fond*, tandis que le processus TOP–DOWN fonctionnera en *tâche principale* sans modification d’avec ce qui a été décrit précédemment.

8.2 Idée de base de l’ajout de murs dans la carte

Étant donné que la technique de localisation est basée sur un processus sélectif supposant que l’on connaît tout (la « carte »), pour ajouter un mur, l’idée est de partir d’un processus qui ne connaît **rien** et va tenter d’agrèger à la carte **tout** ce qu’il peut observer. Nous nous limitons ici à l’analyse de cette problématique dans le cadre des données issues de capteurs LIDAR 2D plans montés sur véhicule.

Nous allons donc nous baser sur une technique de type paramétrique, opérant sur toutes les données LIDAR et non sur une unique zone définie à l’avance. Plusieurs de ces techniques sont classiques en robotique, par exemple des appariements de type RANSAC ou JCBB, un fit MOINDRE-CARRÉS-MÉDIANS, ou bien une transformée de HOUGH. Nous verrons dans ce qui suit que chacune de ces méthodes sera utilisée à une étape différente.

Ensuite, si l’on analyse la totalité des données LIDAR, cela signifie qu’**aucun** *a-priori* n’est défini, ce qui indique l’usage d’une méthode de type BOTTOM–UP, potentiellement gourmande en temps de calcul. Ceci implique alors un fonctionnement en tâche de fond plutôt qu’activé à chaque fois que des données sont émises par les capteurs.

Enfin, il nous faut un mécanisme permettant de rejeter les **fits erronés**, tout en associant ceux qui sont corrects à la carte de l’environnement. Un processus d’association de type récursif sera présenté et permettra de réaliser ces deux fonctions simultanément.

Nous proposons donc de réaliser ces opérations de la manière suivante :

1. un étage de détection paramétrique pour des *lignes droites* ou ce qui en tient lieu,
2. un étage de rejet de fits à faux (qui ne correspondent pas à de vrais murs),
3. un étage d'appariement avec la carte de l'environnement à l'instant t , de type « fusion de données »
4. un étage de séparation des contributions de chaque mur (modélisation *vecteur d'état-covariance*).

L'étage de détection paramétrique a pour objectif de détecter les segments de mur dans les données LIDAR. Celui de réjection de « fits à faux » doit rejeter par exemple les impacts au sol tout comme les structures non-verticales. Celui de fusion de données indique quels murs détectés existent réellement dans la carte et positionne ceux qui n'existent pas encore dans la carte par rapport à ces premiers. Enfin, afin de pouvoir utiliser ces données concernant les murs à ajouter dans un algorithme de gestion de carte, nous avons besoin de rendre les paramètres obtenus indépendants de ceux de chacune des autres entités de la carte. C'est le processus dit improprement de « bloc-diagonalisation ». Nous allons dans ce qui suit détailler ces différents éléments.

8.3 Principe du processus BOTTOM-UP

Ce processus, fonctionnant en tâche de fond, est constitué de plusieurs étapes :

1. une **transformée de Hough** modifiée, appliquée sur les données LIDAR,
2. un étage d'appariement entre deux scans LIDAR distincts pour rejeter les détections à faux,
3. un étage d'appariement entre les murs confirmés et la carte,
4. un étage calculant pour chaque élément nouveau de la carte les covariances associées.

8.3.1 Transformée de Hough

Nous utilisons ici une variante développée spécifiquement dans le cadre de notre application de l'algorithme de la transformée de Hough (voir par exemple [Galamhos et al., 1999](#)) qui est un classique de la robotique.

Dans l'algorithme 1, la transformée de Hough modifiée que nous avons utilisée est décrite. Elle consiste en trois étapes :

1. transformée de Hough classique en paramétrage $[\rho, \theta]^T$,
2. segmentation par proximité des points appartenant à des maxima locaux de la transformée de Hough,
3. rejet des segments dont les paramètres de Hough ne sont pas compatibles avec le fit Moindre Carrés Médiants associé.

Un exemple d'une transformée de Hough standard est visible sur la figure 8.1, et correspond à l'étape n°1 de la liste précédente. On observe que les fits obtenus peuvent être totalement fortuits, en associant des groupes de point séparés par une dizaine de mètres. La solution que nous proposons est de sélectionner des groupes de points par proximité géographique (segmentation). Enfin, un fit Moindres Carrés Médiants est utilisé pour :

- rejeter les points aberrants,

— valider le fit si les paramètres Moindres Carrés et Hough correspondent.
Si les paramètres ne correspondent pas, il est clair que le fit est fortuit.

Le résultat de cette opération est présenté sur la figure 8.2. On peut remarquer que, s'il reste des alignements fortuits (en haut), ils sont beaucoup moins nombreux que sur la figure 8.1.

```

Data: LIDAR points D
Result: list of linear fits F

1  Initialisation de la matrice de résultats de la transformée de Hough;
2  HoughMatrix[ $\forall i, \forall j$ ]  $\leftarrow$  0;
3  F = [ ];
4  Itération sur toutes les paires de points LIDAR;
5  for  $A \neq B \in D$  do
6  |    $\rho, \theta = \text{LineParameters}(A, B)$ ;
7  |   HoughMatrix [ $\rho, \theta$ ]  $\leftarrow$  HoughMatrix [ $\rho, \theta$ ] + 1;
8  end
9  Rejet des ensembles de points dont le fit Moindres Carrés Médiants est trop éloigné
   des paramètres de Hough;
10 for  $\rho', \theta' \in \text{HoughMatrix}$  do
11 |   Sur chaque maximum;
12 |   if HoughMatrix [ $\rho', \theta'$ ] is a local maximum then
13 | |   Segmenter les points par proximité;
14 | |   VotingPoints  $\leftarrow$  PointsVotingFor(HoughMatrix [ $\rho', \theta'$ ]);
15 | |   ArrayOfSequencesOfPoints  $\leftarrow$  SpatialClustering (VotingPoints);
16 | |   Vérification de compatibilité entre paramètres Moindres Carrés Médiants et
   Hough;
17 | |   for SequenceOfPoints  $\in$  ArrayOfSequencesOfPoints do
18 | | |   ClusterParameters  $\leftarrow$  LeastSquareOfMedianParameters
   (SequenceOfPoints);
19 | | |   if ClusterParameters close to ( $\rho', \theta'$ ) then
20 | | | |   F  $\leftarrow$  [F, SequenceOfPoints ]
21 | | |   end
22 | |   end
23 |   end
24 end

```

Algorithm 1: Détecteur de segments linéaires.

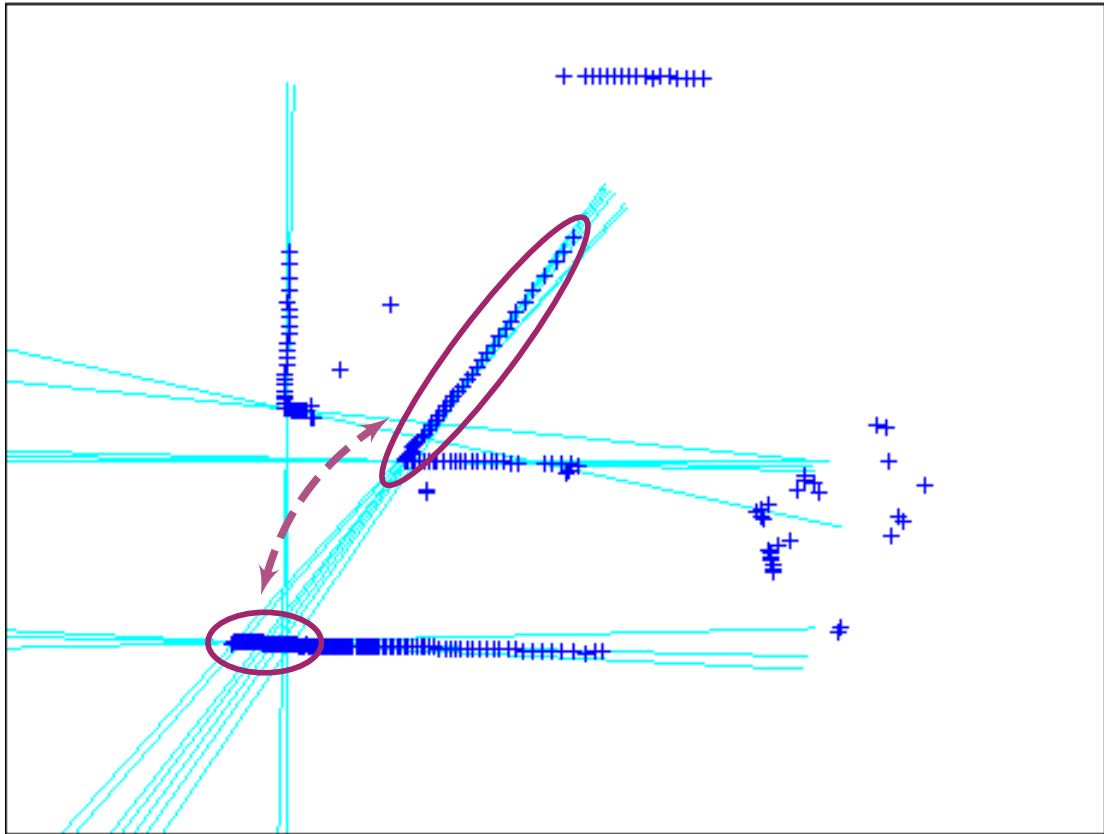


FIGURE 8.1 – Transformée de Hough standard. On observe de multiples fits correspondant à des points distants (indiqués en violet).

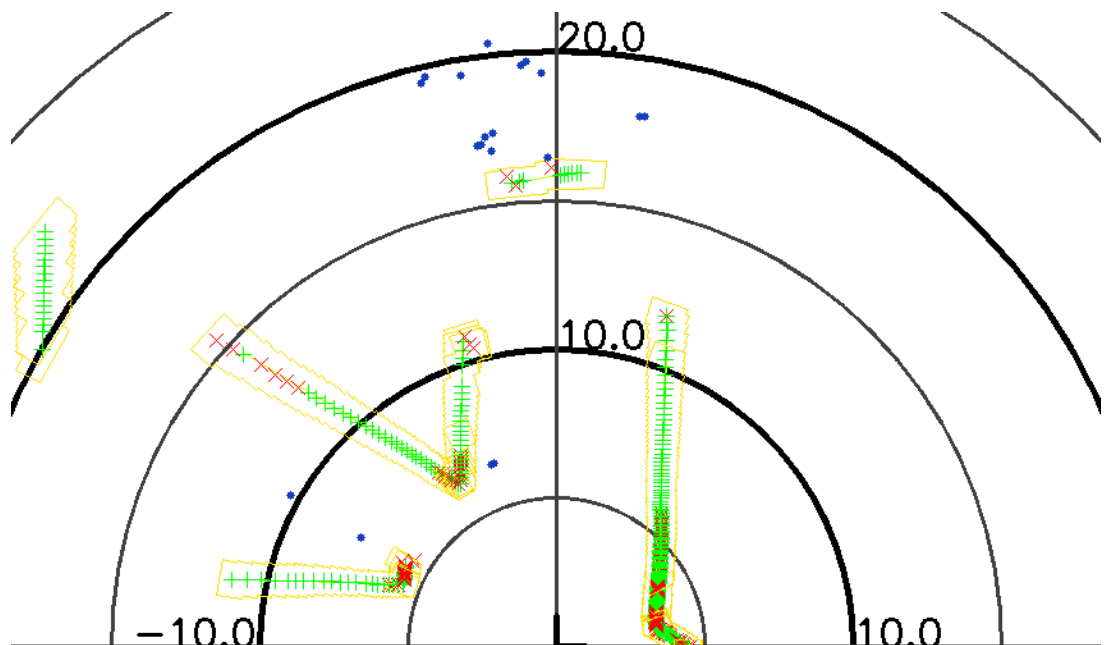


FIGURE 8.2 – Transformée de Hough proposée, selon l'algorithme 1, pour les mêmes données que pour la figure 8.1. On observe que les fits (zones entourées en jaune) ne concernent pas de points distants et correspondent à des segments de droite.
Noter la rotation de $+90^\circ$ par rapport à la figure 8.1.

8.3.2 Processus d’alignement entre scans LIDAR

Comme on peut le voir sur la figure 8.2, des fit « aberrants » peuvent se produire (en haut de la figure). Dans le cas présent, la zone correspond au rond point visible sur la figure 6.9 et indiqué comme comportant une zone végétale au centre. Il est possible que les points LIDAR fittés puissent correspondre à la bordure physique délimitant celui-ci. Le but de la procédure d’alignement que nous proposons est de rejeter les détections à faux de segments rectilignes (non-associé à des murs), comme par exemple des impacts LIDAR au sol. Un exemple de tels impacts au sol est visible sur la figure 8.3.

Afin d’éliminer les fits aberrants, plusieurs idées viennent à l’esprit :

- utiliser un second LIDAR à une hauteur significativement différente, ou, mieux, un LIDAR 3D,
- comparer deux fits associés à des points de vue différents.

Dans notre cas, la première option n’est pas souhaitable (en raison de la nécessité d’installer un autre capteur sur les véhicules). Par contre, quand le véhicule se déplace, il peut potentiellement prendre un scan LIDAR à partir d’un autre point de vue, ce qui correspond à la seconde possibilité. Il nous faut alors, dans ce cas, un moyen d’associer les éléments d’un scan LIDAR avec un autre, pris d’un autre point de vue mais à proximité tout de même. Il est très probable dans ce cas que l’on n’observe pas les mêmes impacts au sol à partir de cet autre point de vue. On peut donc imaginer rejeter les fits LIDAR aberrants en tentant d’aligner deux trames LIDAR prises selon des points de vue légèrement différents, et valider uniquement des « couples » de fits LIDAR entre eux, à l’exclusion de tout le reste.

Une seconde raison pour laquelle on peut vouloir aligner les deux images LIDAR est que la résolution angulaire sur un véhicule est trop mauvaise pour permettre une superposition directe des trames LIDAR à partir des paramètres estimés du véhicule. On peut en voir un exemple sur la figure 8.3, qui superpose de manière approximative des scans LIDAR pris de différents points de vue. En effet, la précision d’orientation du véhicule est de l’ordre de 5° à 10° . Ceci génère par parallaxe une erreur à 20 m de l’ordre de 1.7 à 3.5 mètres, voire plus si l’estimation angulaire est mauvaise.

Un exemple d’une telle association est visible sur la figure 8.4. Nous avons résolu le problème de l’alignement de deux scans LIDAR de la manière suivante :

1. le principe de base est du type Joint Compatibility Branch and Bound (JCBB), conformément à (Neira and Tardós, 2001), ce type de test étant récursif (parcours d’un arbre de possibilités),
2. pour chaque association potentielle entre un ensemble de segments du fit du scan LIDAR n°1 et un ensemble d’autant de fits provenant du scan LIDAR n°2, on calcule selon une loi donnée la compatibilité de l’hypothèse proposée selon un test du χ^2 de probabilité donnée p_{χ^2} ,
3. si le fit est valable (distance de Mahalanobis compatible du test précédent), on itère une récursion supplémentaire par rapport à l’hypothèse courante, si cela est possible (s’il reste des amers à associer),
4. sinon, dans le cas où les items n°2 et n°3 échouent ou sont impossibles à réaliser, on remonte d’un niveau.

Une fois une liste des possibilités générée selon l’algorithme mentionné précédemment, il reste encore à trouver la meilleure d’entre elles. Ce choix est opéré par un tri, en choisissant l’association ayant la meilleure « information mutuelle ».

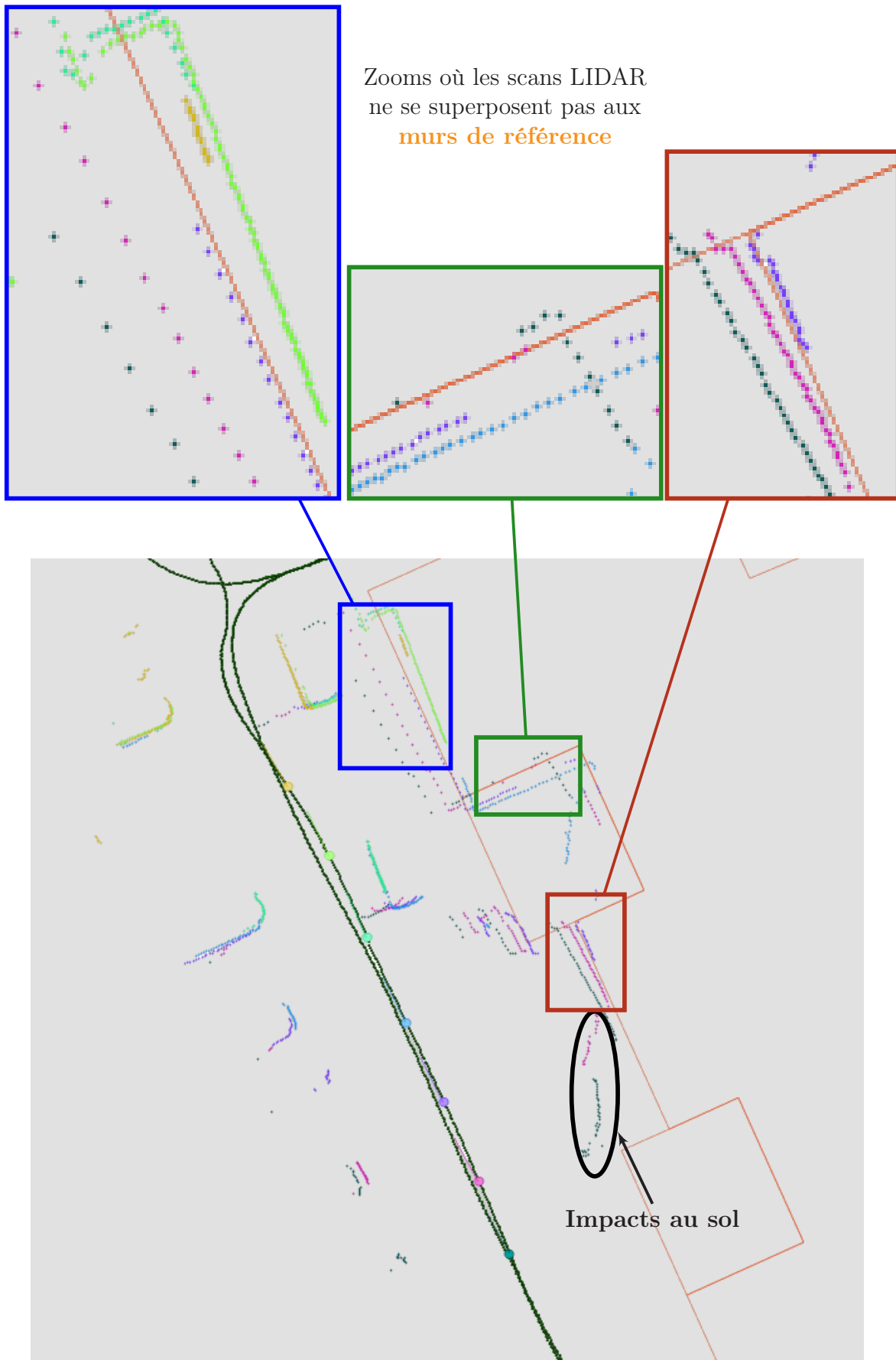


FIGURE 8.3 – Scans LIDAR superposés uniquement à partir de l'estimation d'orientation et de position du véhicule. Les murs de référence sont indiqués en orange.

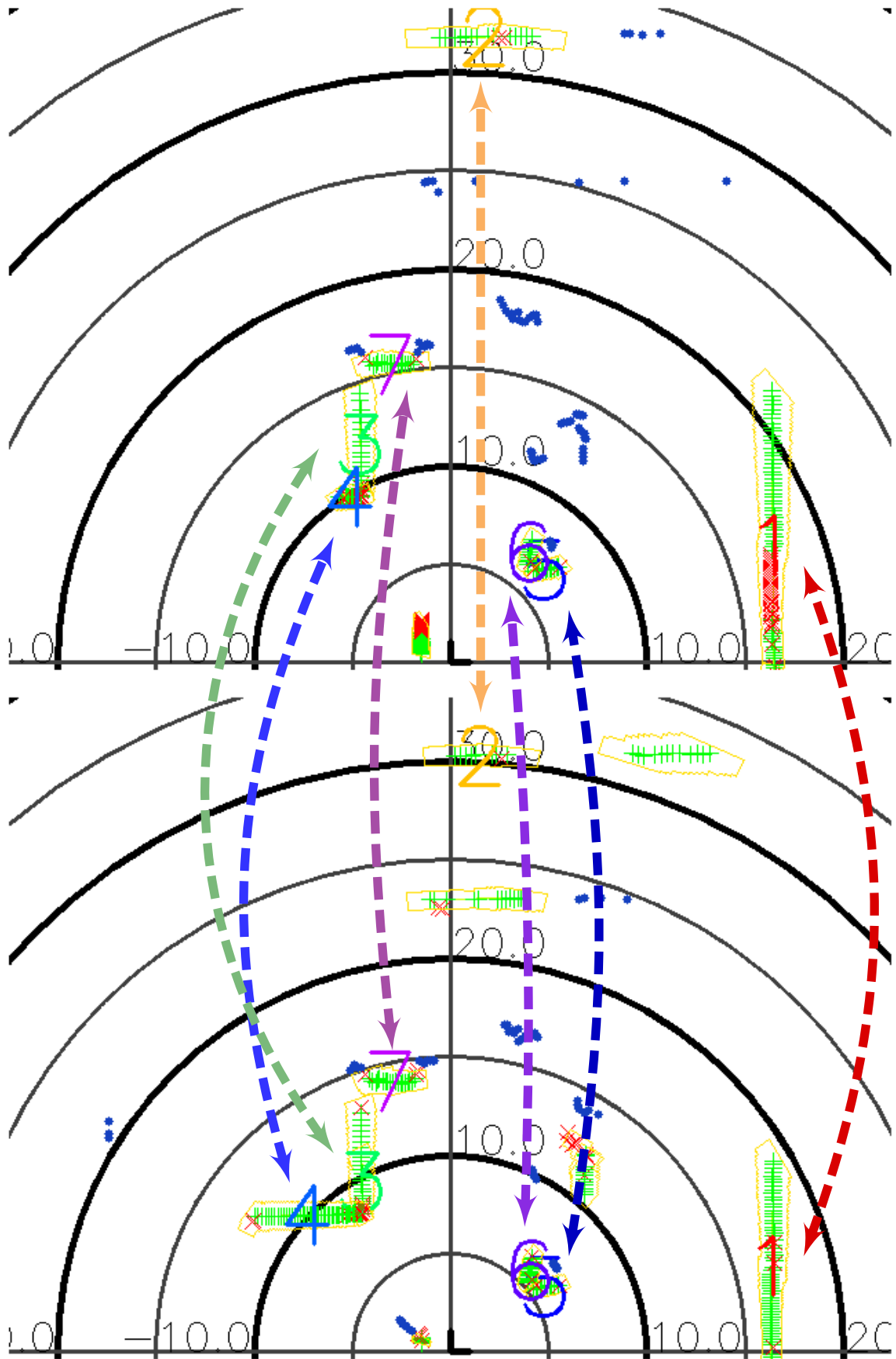


FIGURE 8.4 – Deux scans LIDAR associés. Noter les associations de fit LIDAR pris des deux points de vue.

Nous allons décrire brièvement dans ce qui suit les deux éléments techniques relatifs, à la fusion de données d'une part, et au calcul de l'information mutuelle d'autre part.

Fusion de trames LIDAR

Afin de pouvoir aligner deux scans LIDAR l'un sur l'autre, il nous faut obtenir une fusion de données cohérente entre les deux scans LIDAR. Par exemple, si l'on a les deux équations (8.1) et (8.2), on veut obtenir une fusion de données selon (8.3). Dans celles-ci, le suffixe V est relatif à un état de véhicule, $LIDAR$ est relatif aux trames LIDAR analysées depuis chaque position du véhicule. Les suffixes 1 et 2 sont relatifs aux trames LIDAR acquises, 1 étant temporellement antérieur à 2. Le suffixe Fit est relatif à un fit donné.

$$\mathbf{X}_{LIDAR\ 1} = \mathbf{f}(\mathbf{X}_{V\ 1}, \mathbf{X}_{Fit\ k}) \quad (8.1)$$

$$\mathbf{X}_{LIDAR\ 2} = \mathbf{f}(\mathbf{X}_{V\ 2}, \mathbf{X}_{Fit\ k}) \quad (8.2)$$

$$(\mathbf{X}_{Fit\ k}) = \mathbf{g}(\mathbf{X}_{V\ 1}, \mathbf{X}_{LIDAR\ 1}, \mathbf{X}_{V\ 2}, \mathbf{X}_{LIDAR\ 2},) \quad (8.3)$$

Nous avons utilisé une technique de fusion basée sur l'Intersection de Covariance Partitionnée. En effet, l'état du véhicule et celui de chaque mur est estimé, dans la partie localisation, par un tel formalisme, comprenant une partie *indépendante* et une partie *corrélée* (matrice de covariance). Il est donc indispensable de l'utiliser également dans le cas d'une telle fusion, afin d'éviter toute sur-convergence de l'estimation. Supposons que l'on puisse écrire le système observé sous la forme (8.4)···(8.8).

$$\mathbf{X}_{\mathcal{F}ull} = \begin{pmatrix} \mathbf{X}_{V\ 1} \\ \mathbf{X}_{V\ 2} \\ \mathbf{X}_{Fit\ 1} \\ \vdots \\ \mathbf{X}_{Fit\ k} \end{pmatrix} \quad (8.4)$$

$$\mathbf{X}_{LIDAR\ 1} = \mathbf{H}_{LIDAR\ 1} \cdot \mathbf{X}_{\mathcal{F}ull} \quad (8.5)$$

$$\mathbf{X}_{V\ 1} = \mathbf{H}_{V\ 1} \cdot \mathbf{X}_{\mathcal{F}ull} \quad (8.6)$$

$$\mathbf{X}_{LIDAR\ 2} = \mathbf{H}_{LIDAR\ 2} \cdot \mathbf{X}_{\mathcal{F}ull} \quad (8.7)$$

$$\mathbf{X}_{V\ 2} = \mathbf{H}_{V\ 2} \cdot \mathbf{X}_{\mathcal{F}ull} \quad (8.8)$$

$$(8.9)$$

On suppose que $\mathbf{X}_{Fit\ k}$ représente les coordonnées **absolues** d'un mur quelconque. Supposons que pour un scan LIDAR global, on ait une covariance de mesure indépendante $\Sigma_{\mathbf{X}_{LIDAR\ 1}}$ et $\Sigma_{\mathbf{X}_{LIDAR\ 2}}$. On peut alors formuler le système global comme une fusion à n termes d'un filtre SCI, comme indiqué dans (8.10) et (8.11), dans lesquelles ω_1 et ω_2 doivent être choisis pour minimiser $|\Sigma_{\mathbf{X}_{\mathcal{F}ull}}(\omega_i)|$.

$$\begin{aligned}
& \omega_1, \omega_2 \in [0, 1], \quad \omega_1 + \omega_2 = 1 \\
\Sigma_{\mathbf{X}_{\mathcal{F}\text{ull}}}(\omega_i)^{-1} &= \mathbf{H}_{\mathbf{V}\ 1}^\top \cdot \Sigma_{\mathbf{X}_{\mathbf{V}\ 1}}^{-1}(\omega_1) \cdot \mathbf{H}_{\mathbf{V}\ 1} \\
&+ \mathbf{H}_{\mathbf{V}\ 2}^\top \cdot \Sigma_{\mathbf{X}_{\mathbf{V}\ 2}}^{-1}(\omega_2) \cdot \mathbf{H}_{\mathbf{V}\ 2} \\
&+ \mathbf{H}_{\text{LIDAR}\ 1}^\top \cdot \Sigma_{\text{X}_{\text{LIDAR}\ 1}}^{-1} \cdot \mathbf{H}_{\text{LIDAR}\ 1} \\
&+ \mathbf{H}_{\text{LIDAR}\ 2}^\top \cdot \Sigma_{\text{X}_{\text{LIDAR}\ 2}}^{-1} \cdot \mathbf{H}_{\text{LIDAR}\ 2} \tag{8.10} \\
\mathbf{X}_{\mathcal{F}\text{ull}}(\omega_i) &= \Sigma_{\mathbf{X}_{\mathcal{F}\text{ull}}}(\omega_i) \cdot \\
&[\mathbf{H}_{\mathbf{V}\ 1}^\top \cdot \Sigma_{\mathbf{X}_{\mathbf{V}\ 1}}^{-1}(\omega_1) \cdot \mathbf{X}_{\mathbf{V}\ 1} \\
&+ \mathbf{H}_{\mathbf{V}\ 2}^\top \cdot \Sigma_{\mathbf{X}_{\mathbf{V}\ 2}}^{-1}(\omega_2) \cdot \mathbf{X}_{\mathbf{V}\ 2} \\
&+ \mathbf{H}_{\text{LIDAR}\ 1} \cdot \Sigma_{\text{X}_{\text{LIDAR}\ 1}}^{-1} \cdot \mathbf{X}_{\text{LIDAR}\ 1} \\
&+ \mathbf{H}_{\text{LIDAR}\ 2} \cdot \Sigma_{\text{X}_{\text{LIDAR}\ 2}}^{-1} \cdot \mathbf{X}_{\text{LIDAR}\ 2}] \tag{8.11}
\end{aligned}$$

Le test du χ^2 est alors réalisé entre les valeurs mesurées $\mathbf{X}_{\text{LIDAR}\ 1}$ et $\mathbf{X}_{\text{LIDAR}\ 2}$ d'une part, et les valeurs fusionnées $\mathbf{X}_{\mathcal{F}\text{ull}}$ et $\Sigma_{\mathbf{X}_{\mathcal{F}\text{ull}}}$ d'autre part. Le même test du χ^2 est également réalisé avec les valeurs estimées $\mathbf{X}_{\mathbf{V}\ 1}$ et $\mathbf{X}_{\mathbf{V}\ 2}$ et les valeurs fusionnées.

Calcul de simili-information mutuelle

Ici, nous tentons de trouver une analogie avec l'information mutuelle discrète décrite à la section 4.6. Nous rappelons sa définition dans (8.12).

$$I(X : Y) = H(X) - H(X|Y) \tag{8.12}$$

Nous avons ici employé les définitions (8.13) et (8.14).

$$I(\mathbf{X}_{\mathcal{F}\text{its}} : \mathbf{X}_{\text{LIDAR}\ 1}) = H(\mathbf{X}_{\mathcal{F}\text{its}\ 1} \text{ avant fusion}) - H(\mathbf{X}_{\mathcal{F}\text{its}\ 1} \text{ après fusion}) \tag{8.13}$$

$$I(\mathbf{X}_{\mathcal{F}\text{its}} : \mathbf{X}_{\text{LIDAR}\ 2}) = H(\mathbf{X}_{\mathcal{F}\text{its}\ 2} \text{ avant fusion}) - H(\mathbf{X}_{\mathcal{F}\text{its}\ 2} \text{ après fusion}) \tag{8.14}$$

On se rend compte numériquement que ces deux définitions ne sont pas égales et qu'il y a donc une asymétrie. Celle-ci a été corrigée en moyennant les éléments précédents dans l'équation (8.15).

$$I(\mathbf{X}_{\mathcal{F}\text{its}} : \mathbf{X}_{\text{LIDAR}}) = \frac{1}{2}I(\mathbf{X}_{\mathcal{F}\text{its}} : \mathbf{X}_{\text{LIDAR}\ 1}) + \frac{1}{2}I(\mathbf{X}_{\mathcal{F}\text{its}} : \mathbf{X}_{\text{LIDAR}\ 2}) \tag{8.15}$$

La séquence d'association ayant la plus forte valeur donnée par (8.15) est choisie comme fit maximal. Les éléments non associés sont supprimés. Les éléments associés sont pris en compte dans la suite comme paire.

8.3.3 Association avec la carte existante

Le processus de fusion est ici du même type que celui de la section 8.3.2. La différence en est que l'on associe ici des paires de fits LIDAR avec des murs ayant un état donné par un filtre SCI. Dans ce cas, la formulation globale est conforme aux équations (8.4)... (8.8) plus (8.16) pour les fits associés à un mur.

Association du fit \mathbf{k} avec le mur \mathbf{l} :

$$\mathbf{X}_{\text{Fit } \mathbf{k}} = \mathbf{H}_{\text{Mur } \mathbf{l}} \cdot \mathbf{X}_{\text{Mur } \mathbf{l}} \quad (8.16)$$

Le processus d'association des murs à la carte de l'environnement est de nature récursive. Il utilise ainsi que précédemment une sélection par test du χ^2 . Le critère de sélection du meilleur fit est basé sur un nombre minimal de fits non-associés à un mur (nouveaux murs potentiels).

Un exemple de telle association est présentée sur la figure 8.5. On observe sur celle-ci quatre nouveaux murs :

- l'un au sud de l'image #1, correspondant à un bâtiment extérieur à notre site d'expérimentation,
- un autre près d'un mur #6, correspondant à une haie végétale,
- deux proches l'une de l'autre, #4 et #5, correspondant à des tables volontairement rajoutées pour être détectées, et positionnées couchées sur le côté.

Les murs associés à des détections LIDAR sont les murs 88, 89 et 133, qui sont bien positionnés.

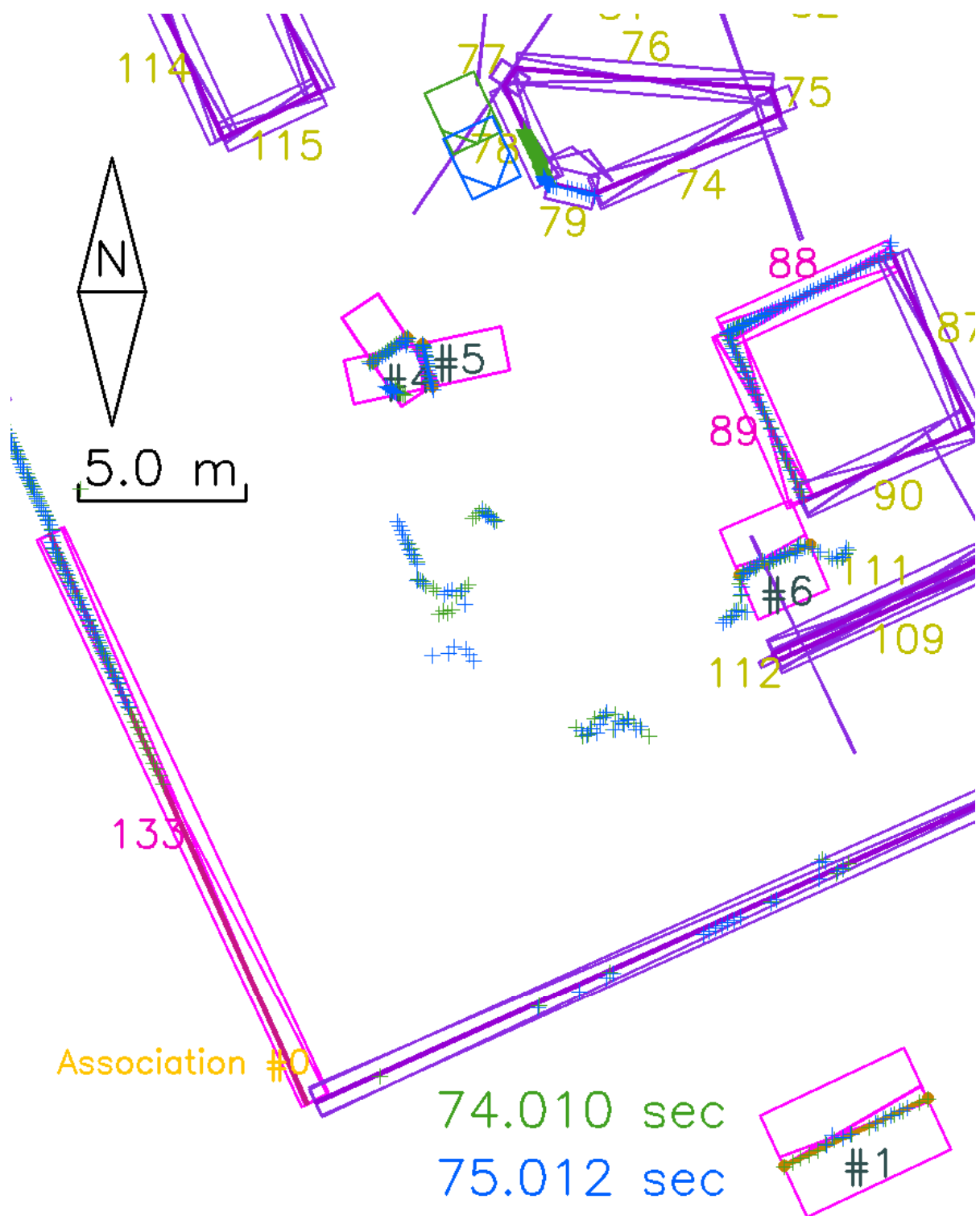


FIGURE 8.5 – Association carte avec fits LIDAR provenant d'un véhicule. La position du véhicule ainsi que les données LIDAR correspondant aux deux instants sont indiqués en bleu et vert. Les murs associés à des fits comme les nouveaux murs sont en magenta.

8.3.4 Processus de « bloc-diagonalisation »

À ce point, nous disposons d'une fusion entre les scans LIDAR et la carte de l'environnement. Dans cette fusion, les murs qui ne sont pas encore inscrits dans la carte sont estimés de manière conjointe avec les murs de la carte. En effet, la procédure de fusion **corrèle** l'état de ces murs à celui des autres. Or, afin de pouvoir gérer chacun de ces nouveaux murs de manière indépendante de tous les autres, il est **indispensable** de rendre leur état estimé indépendant de celui des autres. Pour une densité de probabilité gaussienne, la corrélation se manifeste par le biais de la structure de la matrice de covariance. L'idée est de passer d'une structure de matrice de covariance pleine à une matrice de covariance dans laquelle chaque nouveau mur est modélisé par un bloc diagonal unique et un ensemble de coefficients croisés nuls avec tous les autres éléments. Cette matrice de covariance devra de plus, pour obtenir une estimation conservative, être supérieure, au sens de la positivité, à la matrice issue du processus de fusion.

Ceci est ce que nous appelons ici « bloc-diagonalisation » et elle est réalisée par un processus de minimisation. Il s'agit d'une minimisation sous contrainte définie dans (8.17). Chaque bloc i correspond à l'état soit d'un mur, soit d'une situation du véhicule. On peut, à la place du déterminant de la matrice Σ_{bloc} reprendre uniquement le produit de celui des nouveaux murs (puisque la partie relative aux autres murs ne sera pas fusionnée aux états antérieurs).

Soit Σ de taille $n \times n$,

Soient k_i tels que $\sum_i k_i = n$,

Soient Σ_i symétrique définie positive de taille $k_i \times k_i$

Soit $\Sigma_{\text{bloc}} = \begin{pmatrix} \Sigma_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Sigma_k \end{pmatrix}$

On cherche $\arg \min_{\Sigma_{\text{bloc}}} |\Sigma_{\text{bloc}}|$ sachant $\Sigma_{\text{bloc}} - \Sigma \succeq \mathbf{0}$ (8.17)

L'estimation ainsi obtenue sur les éléments blocs est alors conservative et peut être réutilisée dans une procédure ultérieure d'alignement de carte, ou de recherche d'amer de la carte (processus TOP-DOWN).

8.4 Résultats de simulation

Dans cette section, nous allons décrire les différentes étapes de la simulation que nous avons conduit :

1. acquisition de trames de données LIDAR et GPS/Magnétomètre, destinées à pouvoir détecter des « murs supplémentaires »,
2. localisation *off-line* sur table du véhicule grâce à la localisation TOP-DOWN, et qui sert de données d'entrée aux étapes suivantes,
3. découpage des données de trames LIDAR toutes les secondes,
4. alignement des données LIDAR sur la carte, basées sur la localisation donnée à l'étape n°2,

5. enrichissement de la carte avec les nouveaux éléments détectés.

Les jeux de données qui ont permis de réaliser la simulation en question ont été obtenu dans le site de PAVIN. La configuration utilisé inclus, outre les murs du site, comprends sur le terre-plein SUD-OUEST du site trois tables d'environ 2 mètres positionnées de manière triangulaire. On notera également qu'au SUD du site, se trouve un bâtiment extérieur (PÔLE COMMUN POLYTECH-ISIMA) à portée de capteur LIDAR.

On peut obtenir une image de la carte au démarrage sur la figure 8.6 à 10 secondes, et celle obtenue après 83 secondes sur la figure 8.7. On peut voir qu'à 83 *sec* de la simulation, plusieurs éléments ont été ajoutés :

- la porte coulissante côté ouest du site de PAVIN,
- les trois tables dont nous venons de parler,
- un morceau de mur au SUD appartenant au bâtiment extérieur mentionné ci-dessus.

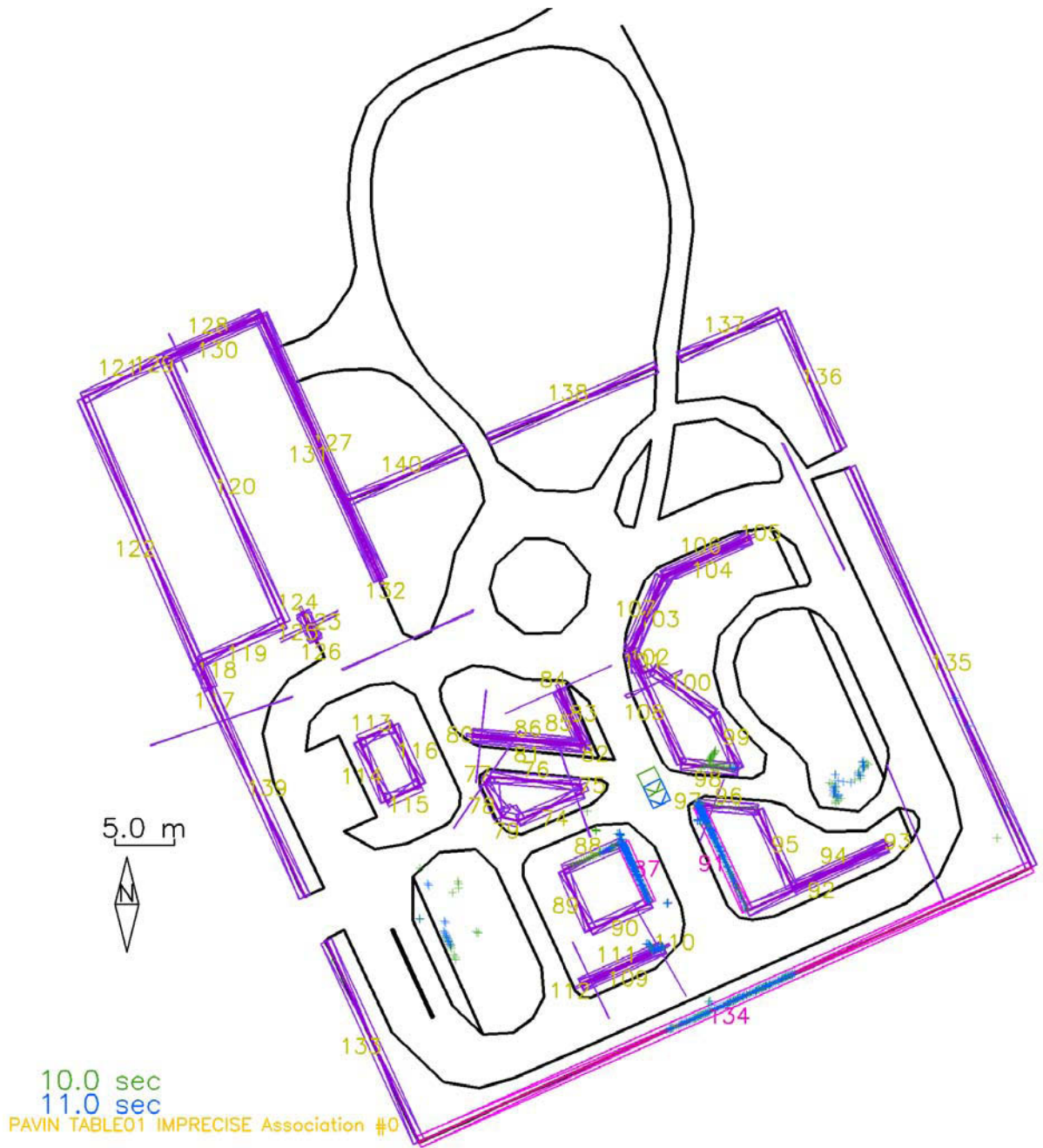


FIGURE 8.6 – Carte au démarrage de la simulation (10 sec).

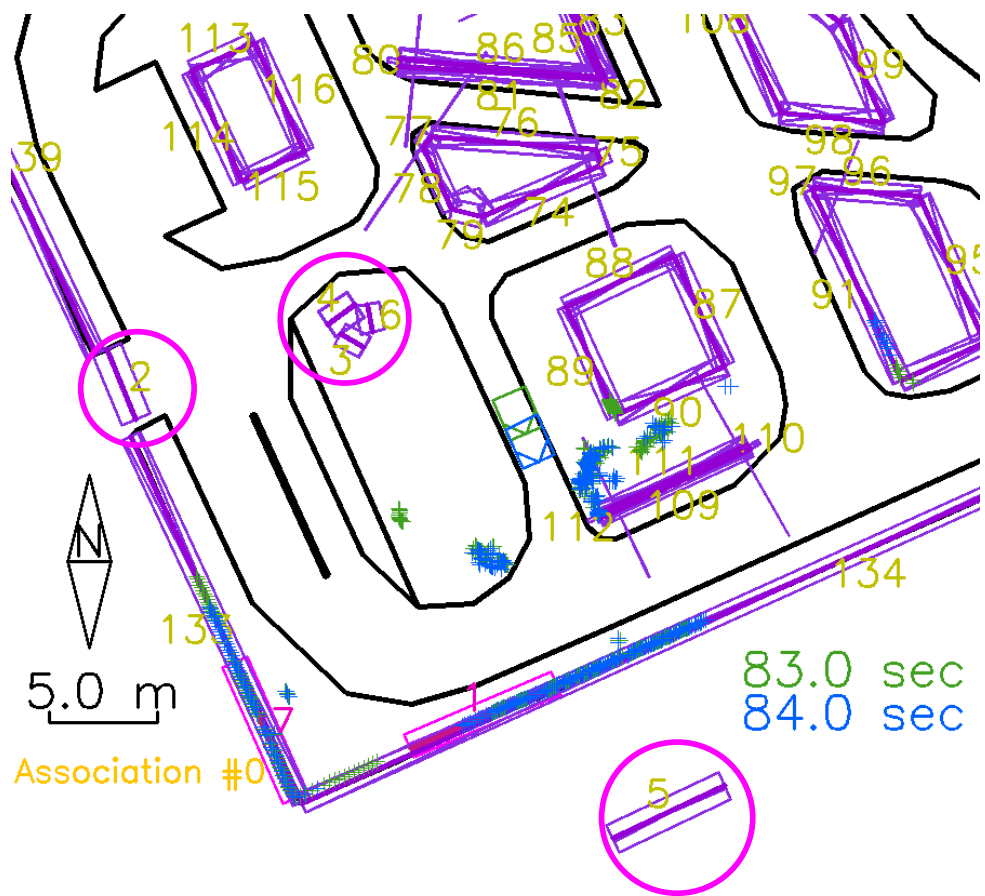


FIGURE 8.7 – Carte après 83 sec de simulation. Les murs ajoutés sont entourés en magenta.

8.5 Résultats expérimentaux

Dans cette section, nous présentons des résultats qualitatifs relatifs à l'ajout de « murs » à la carte de l'environnement. Certaines modifications ont été réalisées sur le code véhicule afin d'implanter cette fonctionnalité en temps réel sur véhicule :

- le calcul de visibilité des murs, qui était basé sur un calcul de recouvrement de surface avec un *polygone de visibilité* calculé *hors-ligne* une fois pour toutes, a été remplacé par un calcul *en ligne* basé sur le secteur angulaire sous lequel un mur est vu par le capteur LIDAR ; afin de permettre aux murs nouvellement ajoutés de pouvoir être utilisés dans l'algorithme de localisation,
- le rond-point au NORD du site de PAVIN a été intégré comme élément occultant, en raison du buisson qui occupe totalement son centre, et bloque les nappes LIDAR, ce qui n'était pas pris en compte dans le système précédent du calcul de la visibilité,
- la partie *inobservable* du mur choisi, trop éloignée du capteur LIDAR, est supprimée de l'estimation de visibilité du mur en question (ceci s'applique particulièrement aux murs longs, comme les grillages du bord du site de PAVIN, ou éloignés du capteur LIDAR),
- une distance de visibilité différente pour les grillages (plus faible, les échos LIDAR étant moins marqués) et pour les murs en « dur » (plus élevée) est introduite,
- la zone de focalisation du capteur LIDAR est spécifique du secteur angulaire principal visible du mur, ceci a pour conséquence de ne se focaliser que sur la partie réellement visible de chaque mur choisi (selon le secteur angulaire, la distance sous lequel il est vu et selon les occultants),
- les éléments au niveau de la section 8.3 ont été codés en C++11 dans un processus TÂCHE DE FOND s'exécutant en permanence en fonction des données disponibles à tout instant.

L'amélioration produite par le nouveau mode de calcul de la visibilité d'un mur, et de sa zone de focalisation, nous semblent avoir apporté une amélioration *significative* à l'algorithme de manière générale, malgré le fait qu'aucune analyse chiffrée n'aie été conduite. Ce calcul impacte directement et améliore :

- la sélection (triplet perceptif) de murs observables en réalité mais qui avec le calcul de visibilité antérieur pouvaient être estimés inobservables, et réciproquement,
- la détection effective du mur, lorsque choisi (taille de la zone de focalisation et détecteur par moindres-carrés médians),
- le calcul d'intégrité du véhicule basé sur le résultat de cette détection (calcul de la probabilité de visibilité),
- le calcul de l'existence du mur sélectionné, toujours basé sur ce résultat de détection (calcul de la probabilité de visibilité également).

Une trajectoire, comme celle présentée sur la figure 8.8 exécutée sur véhicule réel permet de se rendre compte du fonctionnement de l'algorithme. On voit sur celle-ci les murs correspondant aux tables couchées être ajoutés les uns après les autres. Dès que les murs sont ajoutés, ils sont disponibles au niveau de la carte pour être utilisés comme amers que l'on tente de détecter (images 4, 6 et 8, colonne de droite). Leur probabilité d'existence augmente au-delà de 80% pour chacune des trois tables, étant donné qu'elles sont presque toujours détectées lorsque recherchées (trait vert du véhicule au mur sur les vignettes).

D'autres expérimentations utilisant une carte moins précise ou démarrant sans carte ont été conduites et ont montré qualitativement la pertinence de cette approche.

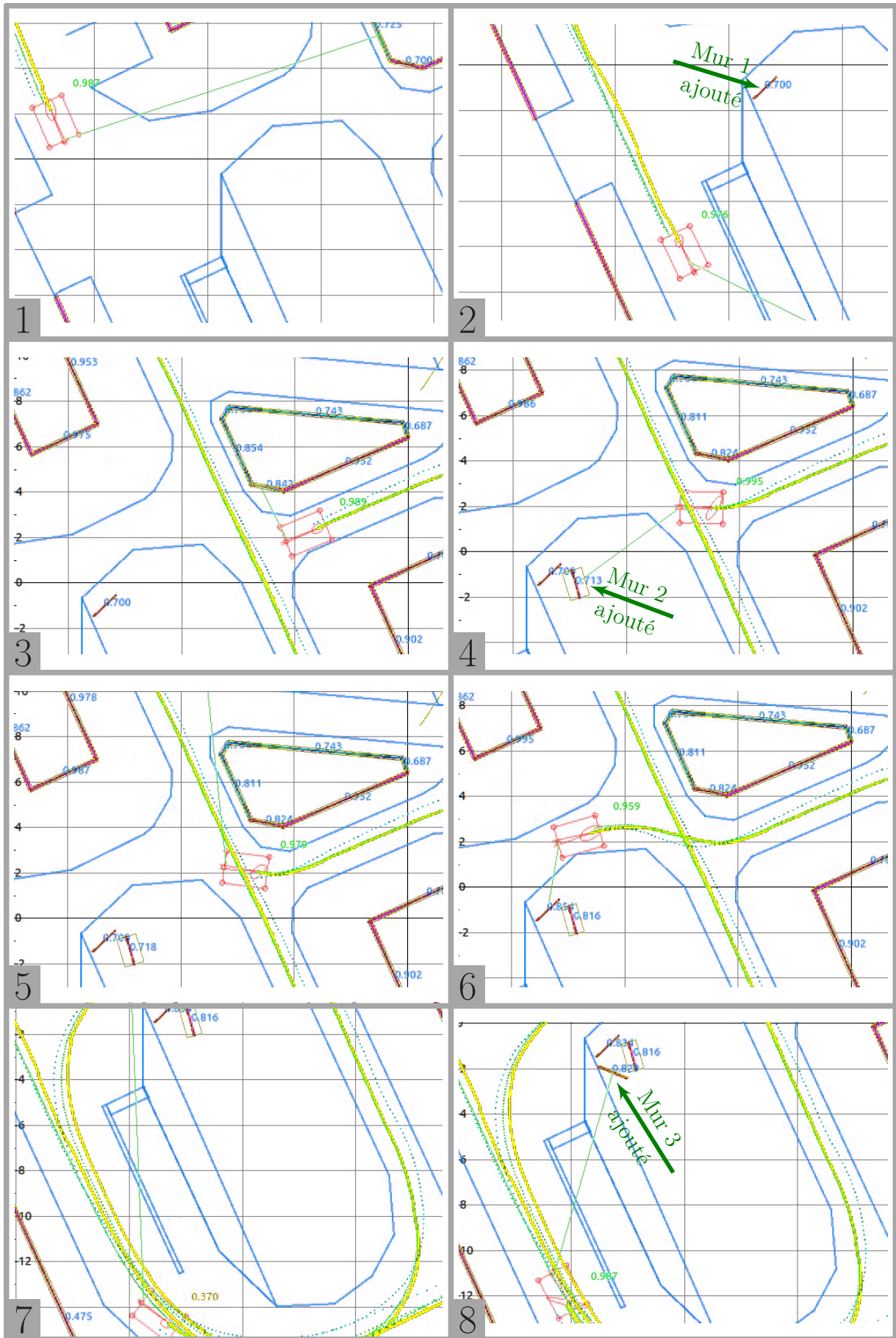


FIGURE 8.8 – Exemple d'ajout de murs sur une séquence sur véhicule réel. De gauche à droite puis de haut en bas.

Ceci montre le fonctionnement nominal de notre algorithme, qui permet d'ajouter des amers à une carte. L'incertitude d'état des amers ajoutés en question peut diminuer si une information de localisation plus précise est disponible pour le véhicule. La probabilité d'existence peut être modifiée en fonction de la détection des amers en question. La carte est ici totalement dynamique, dans son ensemble.

Conclusion

Nous avons dans ce court chapitre montré comment nous pouvons ajouter des murs non cartographiés à partir d'une carte incomplète de l'environnement. Après avoir expliqué le principe de notre analyse, nous avons présenté des résultats de simulations réalisées en temps différé. Nous avons finalement montré les résultats qualitatifs obtenus par notre système en temps réel sur véhicule, montrant la complémentarité de l'approche TOP-DOWN de départ avec un étage BOTTOM-UP ajouté.

Cette fonctionnalité étant la dernière obtenue dans le cadre de ces développements, seuls deux tests fonctionnels sur véhicule de quelques minutes ont été réalisés. De ce fait, nous n'avons pas effectué d'analyse détaillée des performances de celle-ci. Bien que nous manquions d'éléments afin de l'investiguer en détail, ces deux tests nous ont semblé donner des résultats cohérents.

Conclusion et Perspectives

Afin de clôturer ce rapport, nous allons brièvement rappeler les *objectifs de base*, les *objectifs spécifiques* et les *objectifs atteints*, puis nous évoquerons les possibles pistes de recherches que cela ouvre pour conclure. Les *objectifs de base* peuvent être compris comme le cadre dans lequel nos travaux s'incrivent. Il s'agit de permettre la localisation d'un véhicule (ou de plusieurs d'entre eux), de manière **robuste** et **précise**. Nous avons conclu au chapitre 3 qu'il serait souhaitable de travailler à partir d'une carte de type sémantique. Ce type de carte a les avantages suivants :

- fort degré d'*information* fourni par les primitives cartographiées,
- caractère potentiellement générique, *multi-amer* de la carte, permettant potentiellement d'intégrer des types quelconques d'information, d'une manière **unifiée**.

Un formalisme de nature **probabiliste** a été choisi, car il offre un cadre unique se prêtant à de multiples approches et formalisations diverses. Nous estimons que c'est, en quelque sorte, un *langage unique* pour gérer des données de nature variée.

Nous avons repris, à partir de travaux antérieurs (voir chapitre 5), une méthode de localisation de ce type, basée sur une carte sémantique et absolue de l'environnement, contenant des murs. Nos *objectifs spécifiques* étaient la généralisation de cette approche afin de rendre possible, sans dégradation **catastrophique** du principe de localisation, l'utilisation de sources d'information cartographiques absolues **imprécises** et potentiellement **erronées**.

Cette question en amène d'autres. Par exemple, si un amer est localisé de manière imprécise et un véhicule l'est précisément, où bien l'inverse, comment rendre le formalisme compatible de ces deux situations de manière **similaire**. Lorsqu'un amer existe dans la carte, sans être tangible dans le monde réel, comment maintenir l'intégrité présente et future de la localisation du véhicule. Enfin, lorsque des amers existent sans avoir été insérés dans la carte, comment faire en sorte que leur présence puisse aider, dans le futur, la localisation d'un véhicule amené à être à portée de détection de ceux-ci.

Dans ce mémoire, nous avons décrit les résultats obtenus par une recherche conduite de manière pragmatique, en aller-retour entre la théorie et la pratique. Nous nous sommes attachés à reprendre un cadre d'estimation de nature bayésienne ainsi qu'il est d'usage en robotique.

Nos *objectifs atteints*, au nombre de trois ont été :

1. permettre de préciser ou corriger un positionnement incorrect d'un élément fixe de la carte, un amer,
2. permettre la suppression d'un amer inscrit dans la carte n'ayant pas de contrepartie réelle tangible,
3. permettre l'ajout d'amers existant de manière tangible, mais non-encore insérés dans la carte.

Nous avons traité l'objectif 1 tout d'abord de manière théorique, puis, après la validation de la technique de filtrage Intersection de Covariance Partitionnée (SCI), nous sommes passés à la mise à l'échelle sur véhicule. Notre **première contribution**, l'utilisation d'un filtre SCI dans le cadre de la gestion d'une carte contenant des amers paramétrés par un vecteur d'état, permet d'éviter la sur-convergence, tout en favorisant fortement la mise à l'échelle du système, comparée à d'autres solutions (filtre de *Kalman*). Nous estimons que cet objectif a été traité de manière robuste.

L'objectif 2 a été tout d'abord analysé d'un point de vue pratique, en tentant de trouver un équivalent du filtre SCI pour les réseaux discrets (qui, selon nos recherches, ne fait pas partie de l'état de l'art actuel). L'échec de cette phase de recherche nous a incité à rechercher une autre formulation du problème en question. Notre **seconde contribution**, une reformulation du problème par intégration de probabilités dans un réseau bayésien « hiérarchique » nous a permis de nous abstraire suffisamment du problème de sur-convergence (relié à la notion de consanguinité de données). Nous estimons qu'avec cette nouvelle formulation il ne se produit plus, ou suffisamment peu pour pouvoir être ignoré.

L'objectif 3 a été traité de manière plus qualitative. Notre **troisième contribution**, une procédure de fusion de données multiples basée toujours sur un filtre SCI, a été utilisée conjointement à une transformée de Hough couplée à un étage de segmentation pour ajouter les murs inexistantes à une carte incomplète de l'environnement. Dans son principe, cet étage d'analyse sans *a-priori* pourrait être porté sur n'importe quel type d'amer, et non uniquement aux amers *mur* associés à un capteur *LIDAR*.

Les trois contributions mentionnées ci-dessus ont toutes un trait commun, le phénomène de **sur-convergence**, qu'il soit relatif à des densités de probabilité continues (objectifs n°1 et n°3), ou des lois de probabilité discrètes (objectif n°2). Ce phénomène se produit lorsque des variables aléatoires sont couplées entre elles dans la logique, mais que leur estimation à chacune est séparée. Ce type de problème se pose pour tous les systèmes complexes que l'on ne sait pas, ou que l'on ne peut estimer de manière exacte. Dans ces trois cas, une méthode d'estimation adéquate doit être trouvée, en préservant l'intégrité des données estimées, c'est à dire leur représentativité de la variable aléatoire en question.

Concernant les améliorations, elles sont de deux ordres :

1. rendre compte de la visibilité des amers de manière plus fine (mouvements de terrain, occultants à proximité de ceux-ci),
2. améliorer la procédure dite de *bloc-diagonalisation*, qui reste lente sur un processeur standard, et potentiellement insuffisamment précise.

Pour le premier item (une meilleure estimation de la visibilité des amers), le problème de la visibilité des amers de type mur a toujours été problématique. Dans certains cas, ceux-ci sont partiellement ou totalement occultés par des végétaux. Dans d'autres cas, des mouvements de terrain génèrent des tirs *LIDAR* atteignant le sol ou le ciel sans toucher d'amer (roulis/tangage). Ces problèmes peuvent être abordés de deux manières :

- par la cartographie des éléments occultants positionnés de telle manière à écranter les nappes *LIDAR*,
- par une modélisation de type $6D$ et non $2D\frac{1}{2}$ de l'état du véhicule comme de la scène cartographiée.

Nous pensons que ces deux éléments seraient de nature à fortement améliorer la qualité des estimations de visibilité des amers que l'on tente de détecter. Ceci permettrait également de cartographier de manière dynamique, outre les amers (murs par exemple), les occultants qui en diminuent la visibilité, ce qui dans ce dernier cas reste à faire.

Pour le second item (le processus de bloc-diagonalisation), une expertise en optimisation dans le cadre de l'**optimisation semi-définie** pour des matrices permettraient certainement d'améliorer sérieusement la qualité de l'optimisation effectuée en temps réel, tout en pouvant diminuer son temps de calcul. En effet, une sous-branche de l'**optimisation non-linéaire**, une discipline nommée « optimisation semi-définie », consiste à trouver des matrices (semi-) définies—positives respectant un certain nombre de contraintes mathématiques que l'on peut spécifier. Le problème indiqué à la section 8.3.4 pourrait être avantageusement reformulé en termes d'**optimisation semi-définie**, ce qui permettrait très certainement de résoudre plus rapidement et de manière numériquement plus précise ce problème.

Au-delà de ces contributions, présentons maintenant les perspectives que ces travaux permettent d'envisager. Rappelons que la détection des éléments de la carte (murs) est faite par des mesures **extéroceptives**. On peut imaginer que d'autres types d'amers puissent être détectés par de nouvelles modalités extéroceptives, dans le cadre du formalisme de **triplet perceptif**, introduit par les travaux antérieurs (Aynaud et al., 2014a;b) et présenté à la section 5.2.

Bien que nos travaux se soient orientés vers la détection des murs par modalité de type capteur LIDAR 2D, l'algorithme utilisé pour détecter ces amers est transposable à d'autres types d'amers. On pourrait imaginer utiliser par exemple les **triplets** suivants :

- Capteur LIDAR 2D plan – détecteur d'angle entre deux droites – Arrête verticale entre deux murs,
- Capteur LIDAR 2D plan – détecteur de cercle – Poteau-colonne,
- Capteur LIDAR 3D – ... – Murs, Arrête de murs, Poteau,
- Capteur Caméra – détecteur de type Harris – Marquage au sol,
- Capteur Caméra – détecteur de Panneaux de signalisation – Panneaux de signalisation,
- Capteur Caméra – détecteur d'enseignes publicitaires – Enseignes publicitaires,
- Capteur Caméra – détecteur de segment vertical – Arrête verticale entre deux murs (cf. Kais et al., 2004),
- Capteur Caméra – détecteur de segment vertical – Poteau-colonne (cf. également Kais et al., 2004),
- Capteur Caméra de profondeur – ... – Arrête verticale entre deux murs, Poteau-colonne...

Ces amers pourraient appartenir à la carte de l'environnement. Prenons l'exemple d'un panneau de signalisation, un panneau « cédez-le passage » par exemple. Il existe dans l'état de l'art actuel des détecteurs ayant de très bonnes performances et utilisant des images caméra. On pourrait améliorer leur localisation, dans le cas où le véhicule est porteur de plus d'information que ces amers. Ceci pourrait être fait en choisissant d'exécuter le processus de détection dans une sous-partie de l'image caméra. L'aspect du triangle permet de remonter à son orientation, sa position dans la fenêtre de recherche, à sa position par rapport à l'axe de vue, et sa taille apparente à sa distance.

L'existence de ce panneau pourrait être confirmée ou infirmée, à partir du résultat du processus de détection utilisant le détecteur précédent. Il pourrait, dans le cas d'absences de détections répétées, être supprimé de la carte. Un processus s'exécutant en tâche de fond pourrait également être utilisé pour rajouter ceux-ci lorsqu'ils sont détectés sans correspondre à un élément cartographié. On pourrait par exemple utiliser le détecteur de panneaux mentionné ci-dessus sur la totalité de l'image, et donc rajouter le panneau en question à la carte de l'environnement. Des expérimentations réalisées à la fin de ces

travaux ont montré la pertinence et l'intérêt de la construction d'une carte de l'environnement **sans a-priori** et sans données de départ (« from scratch »).

L'approche que nous avons développée, à partir des travaux antérieurs décrits au chapitre 5, nous permet d'envisager des généralisations similaires aux travaux (Pink, 2008; Pink et al., 2009) tout en conservant un formalisme plus général que celui de ces derniers. Dans ceux-ci, une carte sémantique et géoréférencée des marquages au sol, est générée à partir d'images aériennes de haute précision, de manière *hors-ligne*. Sur le véhicule, en temps réel, une caméra binoculaire détecte ces primitives de marquage au sol, et permet, par alignement de plusieurs de ces primitives, grâce à un filtre de Kalman, de localiser l'ego-véhicule à 10 cm près. Dans notre approche, en se basant sur des informations de marquage au sol, un véhicule pourrait choisir, en fonction du contexte, un ou plusieurs éléments au sol, afin de corriger l'estimation de sa pose. Dans notre cas, un capteur caméra monoculaire pourrait être utilisé (évitant une calibration, complexe pour une caméra binoculaire). En outre, lorsque le marquage au sol évolue (route re-asphaltée), l'ancien marquage au sol pourrait être effacé de la carte, puis le nouveau ajouté, lorsque peint au sol. Des éléments observables de référence seraient néanmoins nécessaires afin de procéder à l'ajout avec intégrité.

Ce dernier exemple indique que, conformément à la présentation des propriétés du concept de *Triplet perceptif*, proposé antérieurement à nos travaux, on pourrait intégrer, non pas uniquement des triplets dits « simples » (LIDAR–mur, par exemple), mais des triplets élaborés, tels que le bloc de « map-matching » au sol décrit dans (Pink, 2008; Pink et al., 2009). Ce serait une alternative plus générique, et crédible, que l'adjonction de blocs algorithmiques divers et variés dans des structures « ad-hoc ». Ce type de méthodes est généralement rapporté dans les publications faisant état de modalités d'acquisition multi-capteur, et l'un de ses points faibles est l'absence de formalisme de haut niveau. Pour chaque choix de modalité de capteur différente, une structure « spécifique » de ce choix doit être réalisée. Bien que nos contributions n'aient pas porté sur ce principe, nous avons utilisé le concept de *Triplet perceptif*, dans les travaux rapportés dans ce mémoire. Nous pensons que des travaux intégrant des modalités d'acquisition plus complexes, en termes de *Triplet perceptif*, seraient intéressants à évaluer, afin de connaître le degré de pertinence de cette proposition.

Dans le même ordre d'idées, on peut songer à ajouter à la carte de l'environnement plus d'un type d'objet à la fois. Supposons que nous utilisions un capteur LIDAR 2D plan sur un véhicule, comme c'est le cas dans nos travaux. Si l'on dispose, outre du détecteur de murs utilisé présentement, d'un détecteur de poteaux, fonctionnant tous deux à partir d'une unique trame LIDAR, on peut songer à ajouter de nouveaux éléments de chacun de ces deux types à la carte, en une seule étape. On peut également songer à aligner les données des deux types **simultanément**, afin de parvenir à l'objectif précédent (par exemple aligner une trame LIDAR sur une carte ne contenant localement qu'un mur et un poteau). La structure de **carte sémantique** que nous avons utilisée permet en effet, sans changer le formalisme utilisé dans nos travaux, d'aligner des objets de différents types (poteaux, murs), détectés au préalable, **en une seule étape de fusion de données**.

L'enrichissement de la carte de plusieurs types d'objets (ajout de plusieurs murs et poteaux) **en une passe** rendrait **possible** et **plus efficace** la construction d'une carte contenant des **types d'amers quelconques**. Comme l'on ne peut pas garantir que toutes les cartes accessibles contiendront tous les types d'amers possibles, on peut imaginer **enrichir** une carte extérieure, provenant de sources quelconques. Ceci permettrait, en abordant une zone dans laquelle aucun véhicule ne s'est encore aventuré, d'utiliser une carte

publique incomplète qui, par exemple, contiendrait des murs, mais ne posséderait aucune information sur d’hypothétiques poteaux. Basé sur la position des murs stockée dans la carte, on pourrait alors ajouter les poteaux nouvellement détectés à celle-ci. Notre approche de localisation permettrait alors d’utiliser ces poteaux nouvellement cartographiés tout comme les murs contenus dans la carte ayant servi de base à leur ajout.

Diverses sources de données cartographiées pourraient être utilisées. Nous avons présenté le GIS OPENSTREETMAP au chapitre 2. Mais d’autres sources de nature différente existent. Notons l’existence des bases de données cadastrales de certaines villes, ou bien des sources de données de *haute précision* ou de *haute sémantique* telles que des cartes IGN (voir [Soheilian et al., 2013](#)). Des travaux orientés de manière à pouvoir utiliser diverses modalités sources de données, dans un contexte unifié, seraient à même de permettre la **robustification** de la méthode.

On pourrait ainsi prendre une telle carte « imparfaite », et tenter de l’utiliser comme base de départ d’un processus de correction basé sur l’emploi d’un véhicule. On remarque que dans le cadre de la reprise d’un plan de cadastre, l’orientation des murs d’une parcelle est **très bien définie**, typiquement au degré près ; mais que par contre, le plan obtenu doit très souvent être **translaté de manière horizontale**. Il serait donc nécessaire de modifier le profil de bruit utilisé, en rendant la position horizontal du mur dans une parcelle tributaire d’un décalage horizontal, tout en utilisant une très faible incertitude angulaire ; ce modèle n’est pas compatible de celui que nous avons utilisé dans nos travaux (l’incertitude de position provenant d’une imprecision de chacun des sommets du mur indépendamment de l’autre), mais peut parfaitement être pris en compte par une approche de type *Intersection de Covariance Partitionnée*. Nous estimons que la modification du profil de bruit est relativement mineure, et pourrait être intégrée au système d’importation de la carte en question. Une autre thématique qu’ouvrent ces travaux consiste à tenter de mélanger le concept de **carte dans un référentiel absolu**, que nous avons utilisé dans ceux-ci, à la notion de **carte dans un référentiel relatif local** comme celle utilisée dans les processus de type **SLAM par ajustement de faisceaux**. Ce serait l’union du meilleur des deux mondes et permettrait d’utiliser des informations de position relative tout comme absolue.

Une troisième voie d’amélioration du processus consiste à permettre la calibration automatique du véhicule, tout comme celle des capteurs embarqués sur ce dernier. Il est reconnu qu’un défaut de calibration des paramètres régissant l’odométrie d’un véhicule implique potentiellement une dérive de l’estimation de position de ce dernier avec son déplacement (cf. section 2.2.1). Une solution classique de ce problème consiste à incorporer dans le vecteur d’état du véhicule les paramètres de calibration en question. Ceci permet à toute mise à jour de type **extéroceptive**, faite par le véhicule, de corriger et affiner ceux-ci. Cette solution, considérée comme faisable au démarrage de ces travaux, n’a pas été retenue pour des raisons techniques. De la même manière, on pourrait intégrer la calibration de chaque capteur dans un vecteur d’état propre à ce dernier. Ceci permettrait alors de découpler la partie calibration, propre à chaque capteur, du vecteur d’état du véhicule. Ceci permettrait donc d’ajouter dynamiquement d’autres capteurs sans modifier la représentation d’état des capteurs déjà pris en compte ou bien celle du véhicule.

Dans cette troisième amélioration proposée, une vérification de bon fonctionnement de chaque capteur pourrait également être réalisée en ligne, rendant chaque capteur modélisable de la même manière que véhicule et amers, par une loi normale pour la partie calibration, ainsi que par une probabilité discrète de bon fonctionnement. Lorsque, par exemple, sur un nombre conséquent de détections successives, opérées sur différents amers,

le processus de détection échoue systématiquement, on peut considérer que le capteur n'est plus fonctionnel. On peut penser utiliser à cette fin les travaux du chapitre 7 pour estimer cette probabilité de bon fonctionnement pour chaque capteur indépendamment des autres.

Dans le cadre d'applications de type sécuritaire, on pourrait avoir tendance à vouloir mettre la priorité de l'action d'un véhicule sur la cartographie de l'environnement ; ou bien au contraire, sur la sûreté de navigation. Au vu du critère **C3** utilisé, qui est conçu d'un point de vue universaliste, cela semble en contradiction avec les objectifs poursuivis. On peut néanmoins imaginer donner plus de poids à une partie de l'état global, par rapport à une autre. Imaginons que l'on puisse concevoir une représentation bayésienne du style :

$$p''_{XY} = p_\alpha \cdot p_X + \bar{p}_\alpha \cdot p'_X + p_\beta \cdot p_Y + \bar{p}_\beta \cdot p'_Y \quad (9.18)$$

On serait capable de mélanger deux représentations p et p' pour deux objets X et Y . En choisissant les paramètres de manière judicieuse, on se ramènerait à une recherche d'optimum d'« information », mais pondérée par des probabilités p_α et p_β . Ces coefficients permettraient alors de « choisir » la partie de la loi de probabilités obtenue, en l'optimisant par exemple, soit pour la variable X , soit pour Y . X pourrait représenter l'état du **véhicule**, et Y l'état de la **carte**. Ce critère composite serait alors utilisé en lieu et place du critère **C3**. On pourrait alors mettre l'accent sur l'amélioration prioritaire de la connaissance pour le véhicule, ou bien sur la correction de la carte.

Les diverses contributions mentionnées dans le corps de ce mémoire, ainsi que le contexte de la robotique actuel nous semblent particulièrement propices à pouvoir amener à moyen ou long terme à améliorer les capacités des véhicules. Ils pourraient être rendus plus autonomes et aidant potentiellement des millions de personnes dans leurs tâches quotidiennes, en évitant l'emploi outre mesure de véhicules émetteurs de gaz nocifs, de poussières et de dioxyde de carbone en centre ville, et améliorant la qualité de vie de nos concitoyens.

Annexe A

Articles Publiés dans le cadre de ces travaux

- *Delobel, L., Aynaud, C., Aufrère, R., Debain, C., Chapuis, R., Chateau, T., & Bernay-Angeletti, C.* (2015, December), « **Robust localization using a top-down approach with several lidar sensors** », In *Robotics and Biomimetics (ROBIO)*, 2015 IEEE International Conference on (pp. 2371-2376). IEEE.
- *Delobel, L., Aufrère, R., Chapuis, R., & Chateau, T.* (2016, June–July), « **Efficient Fleet Absolute Localization and Environment Re-Mapping** », in *International Autonomous Vehicles Conference (IAV)*, 2016, IFAC-PapersOnLine, 49(15), 236-241.
- *Delobel, L., Aufrère, R., Chapuis, R., Debain, C., & Chateau, T.* (2017, June), « **Towards automated map updating for mobile robot localization** », In *Intelligent Vehicles Symposium (IV)*, 2017 IEEE (pp. 1342-1347). IEEE.
- *Aufrère, R., Delobel, L., Alizon, S., Lelong, G., & Chapuis, R.* (2017, June), « **Multi-vehicles localization by using a georeferenced map and a top-down approach for automatic guidance** », In *Intelligent Vehicles Symposium (IV)*, 2017 IEEE (pp. 505-510). IEEE.

Annexe B

Tables de probabilité conditionnelles

B.1 Tables de probabilités originelles

Dans cette section, nous trouvons les tables originales utilisées dans la méthode de localisation par carte décrite au chapitre 5. Ici, le chiffre 0 représente la valeur booléenne *faux*, le chiffre 1 représente la valeur booléenne *vrai*.

O_k^-	p_{zo}
0	0
1	$p_{\text{visibilité}}$ selon (5.12)

TABLE B.1 – Table de probabilités conditionnelles relative au nœud z_o .

z_o	P_{A_z}
0	γ voir (B.1)
1	1

TABLE B.2 – Table de probabilités conditionnelles relative au nœud A_z .

- λ densité d'amers dans la zone de navigation
- \mathcal{F} zone de recherche, et par extension, sa surface, conforme à la figure 5.7.
- $\gamma = \min(1, \mathcal{F} \cdot \lambda)$ (B.1)

CI_k	dp	z_o	A_z	p_{d_k}
0	ϕ	0	0	β taux de détections réussies à faux
0	ϕ	0	1	$\bar{\alpha}_1$ selon (B.2)
0	ϕ	1	ϕ	$\bar{\alpha}_2$ selon (B.3)
1	0	ϕ	ϕ	0
1	1	ϕ	ϕ	1

TABLE B.3 – Table de probabilités conditionnelles relative au nœud d_k .

- α taux d'échec de détection
 λ densité d'amers dans la zone de navigation
 \mathcal{F} zone de recherche, et par extension, sa surface, conforme à la figure 5.7.
 $n_{amers}^{\mathcal{F}}$ Nombre d'amers dans la zone \mathcal{F}
 $\bar{\alpha}_1 = 1 - \alpha^{\mathcal{F} \cdot \lambda}$ (B.2)
 $\bar{\alpha}_2 = 1 - \alpha^{\max(\mathcal{F} \cdot \lambda, n_{amers}^{\mathcal{F}})}$ (B.3)

zO	d_k	p_{db}
0	ϕ	0
1	0	0
1	1	$\frac{1}{n_{amers}^{\mathcal{F}}}$

TABLE B.4 – Table de probabilités conditionnelles relative au nœud db .

O_k^-	d_k	db	$p_{O_k^+}$
0	ϕ	ϕ	0.001
1	0	ϕ	0.999
1	1	0	0.001
1	1	1	0.999

TABLE B.5 – Table de probabilités conditionnelles relative au nœud O_k^+ .

B.2 Tables de probabilités modifiées

Des modifications ultérieures ont été réalisées concernant le réseau bayésien dont les tables de probabilités ont été présentées dans B.1. Trois grandes modifications ont été réalisées, sur les variables $\bar{\alpha}_1$, $\bar{\alpha}_2$ et γ , dont les nouvelles expressions sont présentées dans les formules (B.4), (B.5) et (B.6).

$$\bar{\alpha}_1 = \frac{1}{1 + \alpha \cdot \mathcal{F} \cdot \lambda} \quad (\text{B.4})$$

$$\bar{\alpha}_2 = \frac{1}{1 + \alpha \cdot n_{amers}^{\mathcal{F}}} \quad (\text{B.5})$$

$$\gamma = \frac{\mathcal{F} \cdot \lambda}{1 + \mathcal{F} \cdot \lambda} \quad (\text{B.6})$$

Cette modification des probabilités a pour origine le calcul du nombre d'amers moyen dans une zone de surface donnée \mathcal{F} et avec une densité d'amers λ .

Annexe C

Estimation continue et théorie de l'information

Dans cette annexe, nous indiquons quelques résultats sur l'optimalité du filtre de Kalman, ainsi que sur les propriétés d'immunité aux surconvergences des filtres CI et SCI.

C.1 Représentation d'Etat et Dérivation du Filtre de Kalman

C.1.1 Développement des équations

Le filtre de Kalman est basé sur une estimation itérative. Supposons que la distribution de probabilité d'une variable \mathbf{X} soit donnée par :

$$\underline{p}(\mathbf{X}_k^- | \mathbf{Y}_{0\dots k-1}) = \mathcal{N}(\hat{\mathbf{X}}_k^-, \boldsymbol{\Sigma}_{\mathbf{X}_k^-}) \quad (\text{C.1})$$

Supposons que la probabilité conditionnelle d'une observation \mathbf{Y}_k soit donnée par :

$$\underline{p}(\mathbf{Y}_k | \mathbf{X}_k^-) = \mathcal{N}(\mathbf{H} \cdot \mathbf{X}_k^-, \boldsymbol{\Sigma}_{\mathbf{Y}_k}) \quad (\text{C.2})$$

Selon le théorème de Bayes, on a :

$$\underline{p}(\mathbf{X}_k^- | \mathbf{Y}_k) = \frac{\underline{p}(\mathbf{X}_k^-) \cdot \underline{p}(\mathbf{Y}_k | \mathbf{X}_k^-)}{\underline{p}(\mathbf{Y}_k)} \quad (\text{C.3})$$

$$\underline{p}(\mathbf{X}_k^- | \mathbf{Y}_k) \propto \underline{p}(\mathbf{Y}_k | \mathbf{X}_k^-) \quad (\text{C.4})$$

L'estimation postérieure est donnée par :

$$\underline{p}(\mathbf{X}_k^+) = \underline{p}(\mathbf{X}_k^- | \mathbf{Y}_k) \cdot \underline{p}(\mathbf{X}_k^- | \mathbf{Y}_{0\dots k-1}) \quad (\text{C.5})$$

On obtient enfin :

$$\underline{p}(\mathbf{X}_k^+) \propto \exp\left(-\frac{1}{2}d_M^2(\mathbf{Y}_k - \mathbf{H} \cdot \mathbf{X}_k^+, \boldsymbol{\Sigma}_{\mathbf{Y}_k}) - \frac{1}{2}d_M^2(\mathbf{X}_k^+, \boldsymbol{\Sigma}_{\mathbf{X}_k^-})\right) \quad (\text{C.6})$$

La fonction d_M^2 représente le carré de la distance de Mahalanobis :

$$d_M^2(\mathbf{Z}, \Sigma_{\mathbf{Z}}) = \mathbf{Z}^\top \cdot \Sigma_{\mathbf{Z}}^{-1} \cdot \mathbf{Z} \quad (\text{C.7})$$

En développant terme à terme la fonction :

$$d_M^2(\mathbf{Y}_{\mathbf{k}} - \mathbf{H} \cdot \mathbf{X}_{\mathbf{k}}^+, \Sigma_{\mathbf{Y}_{\mathbf{k}}}) + d_M^2(\mathbf{X}_{\mathbf{k}}^+, \Sigma_{\mathbf{X}_{\mathbf{k}}^-}) \quad (\text{C.8})$$

et en identifiant terme à terme, on retrouve les équations classiques du filtre de Kalman pour la mise à jour.

C.1.2 Optimalité du filtre de Kalman

On peut prouver que si les hypothèses suivantes sont vérifiées :

- l'estimation $\Sigma_{\mathbf{X}}$ utilisée dans les calculs est supérieure au sens définie positive au bruit vrai sur la variable \mathbf{X} ;
- l'estimation $\Sigma_{\mathbf{Y}}$ utilisée dans les calculs est supérieure au sens définie positive au bruit vrai sur la variable \mathbf{Y} ;

alors l'estimation donnée par un filtre de Kalman reste intègre, et qu'elle est optimale dans le sens où, en cas d'égalité stricte des bruits estimés et réels :

$$I(\mathbf{X}^+, \mathbf{Y} - \mathbf{H} \cdot \mathbf{X}^+) = 0 \quad (\text{C.9})$$

Le filtre de Kalman sur-converge lorsque l'un des bruits ou l'une de ses composantes est estimée trop faible (la matrice de covariance estimée n'est pas supérieure à celle du bruit vrai). Il est donc important, si l'on veut disposer d'une estimation cohérente, de minimiser les bruits estimés tout en les maintenant suffisamment élevés pour que la sur-convergence ne se produise pas.

C.2 Conservativité du filtre à Intersection de Covariance

C.2.1 Estimation gaussienne conservative

Nous postulons dans la suite de cette annexe le résultat que nous annonçons dans cette section.

Supposons que nous disposions d'une variable \mathbf{X} dont la distribution est $\mathcal{N}(\mathbf{X}_0, \Sigma_{\mathbf{X}_0})$. Soit une autre estimation de cette même variable selon la loi $\mathcal{N}(\mathbf{X}_0, \Sigma'_{\mathbf{X}_0})$. La seconde estimation est dite conservative si et seulement si :

$$\begin{aligned} & \Sigma'_{\mathbf{X}_0} \succeq \Sigma_{\mathbf{X}_0} \\ \iff & \Sigma'_{\mathbf{X}_0} - \Sigma_{\mathbf{X}_0} \text{ symétrique semi-définie positive} \end{aligned} \quad (\text{C.10})$$

Qualitativement, cet argument équivaut à dire que le bruit dédié à l'estimation ne doit pas sous-estimer l'étalement du bruit. Si cela se produisait, ce phénomène provoquerait la « sur-convergence » de l'estimation donnée par le filtre de Kalman. Ce phénomène se produit typiquement lorsqu'une ou plusieurs réalisations des variables dont le bruit est sous-estimé apparaissent **hors-distribution** pour la covariance de la variable estimée. De ce fait, la pondération associée à ces tirages est supérieure à ce qu'elle devrait être, provoquant la convergence du filtre sur de fausses valeurs.

C.2.2 Principe du filtre à Intersection de Covariance

Si l'on reprend les équations de la section C.1.1, on se rend compte que l'une des hypothèses dans la simplification de (C.4) est l'indépendance des lois de probabilités entre la variable \mathbf{Y} et la variable \mathbf{X} . La matrice globale de covariance dans ce cas est donnée par :

$$\Sigma_{\mathbf{Y}}^{\mathbf{X}} = \begin{pmatrix} \Sigma_{\mathbf{X}} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\mathbf{Y}} \end{pmatrix} \quad (\text{C.11})$$

Dans le cas d'une corrélation entre la mesure \mathbf{Y} et l'estimation *a-prior* \mathbf{X} , on obtient :

$$\Sigma_{\mathbf{Y}}^{\mathbf{X}} = \begin{pmatrix} \Sigma_{\mathbf{X}} & \Sigma_{\mathbf{XY}} \\ \Sigma_{\mathbf{XY}}^{\top} & \Sigma_{\mathbf{Y}} \end{pmatrix} \quad (\text{C.12})$$

dans laquelle le terme $\Sigma_{\mathbf{XY}}$ représente le terme de corrélations croisées :

$$\Sigma_{\mathbf{XY}} = \mathbb{E}(\mathbf{X} \cdot \mathbf{Y}^{\top}) \quad (\text{C.13})$$

Lorsque ce terme est nul, les variables \mathbf{X} et \mathbf{Y} sont indépendantes. Généralement, deux cas de figure se produisent dans la réalité :

1. soit les termes de corrélation sont estimés, dans le cadre d'une estimation « stochastique » d'un état unique contenant toutes les variables, c'est une modélisation de type Kalman,
2. soit les termes de corrélation sont inconnus, et dans ce cas, une modélisation par filtre de Kalman est impossible ; c'est ce cas que résout le filtre à intersection de covariance.

Les informations supposées disponibles sur $\Sigma_{\mathbf{Y}}^{\mathbf{X}}$ sont :

- $\Sigma_{\mathbf{X}}$,
- $\Sigma_{\mathbf{Y}}$,
- le fait que la matrice $\Sigma_{\mathbf{Y}}^{\mathbf{X}}$ est symétrique définie positive.

Ceci implique que les variables \mathbf{X} et \mathbf{Y} ne soient pas maximales corrélées. On pourra remarquer néanmoins à partir des équations du filtre CI que celles-ci résolvent néanmoins le cas singulier où $\Sigma_{\mathbf{XY}}$ correspond à un couplage maximal. Dans ce cas, la matrice $\Sigma_{\mathbf{Y}}^{\mathbf{X}}$ est singulière.

Ce fait indique alors un ensemble compact fermé pour toutes les valeurs possibles de la matrice $\Sigma_{\mathbf{XY}}$ compatibles avec les matrices $\Sigma_{\mathbf{X}}$ et $\Sigma_{\mathbf{Y}}$, ainsi que l'hypothèse selon laquelle $\Sigma_{\mathbf{Y}}^{\mathbf{X}}$ est symétrique définie positive.

En reprenant les équations du filtre de Kalman, et en prenant en compte la dépendance entre les variables \mathbf{X} et \mathbf{Y} , on obtient :

$$\underline{p}(\mathbf{X}^+ | \mathbf{Y}_0, \mathbf{X}^-) \propto \exp \left(-\frac{1}{2} d_M^2 \left(\begin{pmatrix} \mathbf{X}^+ - \mathbf{X}^- \\ \mathbf{Y}_0 - \mathbf{H} \cdot \mathbf{X}^+ \end{pmatrix}, \Sigma_{\mathbf{Y}}^{\mathbf{X}} \right) \right) \quad (\text{C.14})$$

En développant le terme $d_M^2 \dots$, on obtient :

$$d_M^2 \left(\begin{pmatrix} \mathbf{X}^+ - \mathbf{X}^- \\ \mathbf{Y}_0 - \mathbf{H} \cdot \mathbf{X}^+ \end{pmatrix}, \Sigma_{\mathbf{Y}}^{\mathbf{X}} \right) = \mathbf{X}^{+\top} \cdot (\mathbf{I}_{\mathbf{X}} \quad -\mathbf{H}^{\top}) \cdot \Sigma_{\mathbf{Y}}^{\mathbf{X}-1} \cdot \begin{pmatrix} \mathbf{I}_{\mathbf{X}} \\ -\mathbf{H} \end{pmatrix} \cdot \mathbf{X}^+ + \text{termes croisés...} \quad (\text{C.15})$$

Le terme quadratique :

$$(\mathbf{I}_X \quad -\mathbf{H}^\top) \cdot \Sigma_Y^{\mathbf{X}^{-1}} \cdot \begin{pmatrix} \mathbf{I}_X \\ -\mathbf{H} \end{pmatrix} \quad (\text{C.16})$$

représente la covariance de l'estimation postérieure \mathbf{X}^+ . C'est cette matrice qui doit être estimée de manière « conservative », en l'absence de connaissance du terme $\Sigma_{\mathbf{X}\mathbf{Y}}$.

Selon Julier, une estimation conservative de la matrice $\Sigma_Y^{\mathbf{X}}$ est donnée par :

$$\forall \omega \in [0, 1], \forall \Sigma_{\mathbf{X}\mathbf{Y}} t.q. \Sigma_Y^{\mathbf{X}} \text{ symétrique définie positive,}$$

$$\Sigma_Y^{\mathbf{X}}(\omega) = \begin{pmatrix} \frac{1}{\omega} \Sigma_X & \mathbf{0} \\ \mathbf{0} & \frac{1}{1-\omega} \Sigma_Y \end{pmatrix} \succeq \Sigma_Y^{\mathbf{X}}$$

Toujours selon Julier, on démontre ceci en écrivant :

$$\Sigma_Y^{\mathbf{X}}(\omega) - \Sigma_Y^{\mathbf{X}} = \begin{pmatrix} \frac{1-\omega}{\omega} \Sigma_X & -\Sigma_{\mathbf{X}\mathbf{Y}} \\ -\Sigma_{\mathbf{X}\mathbf{Y}} & \frac{\omega}{1-\omega} \Sigma_Y \end{pmatrix} \quad (\text{C.17})$$

et en remarquant que, puisque $\Sigma_Y^{\mathbf{X}}$ est symétrique définie positive, on peut écrire :

$$\Sigma_{\mathbf{X}\mathbf{Y}} = \Sigma_X^{\frac{1}{2}} \cdot \Omega \cdot \Sigma_Y^{\frac{\top}{2}}$$

avec Ω n'ayant que des valeurs singulières $\sigma < 1$

selon le critère de la matrice de contraction, qui caractérise les termes non-diagonaux des matrices symétriques définies positives. On peut alors également écrire :

$$\begin{aligned} & \left(\frac{1-\omega}{\omega} \Sigma_X \right)^{\frac{1}{2}} \cdot -\Omega \cdot \left(\frac{\omega}{1-\omega} \Sigma_Y \right)^{\frac{\top}{2}} \\ &= \Sigma_X^{\frac{1}{2}} \cdot -\Omega \cdot \Sigma_Y^{\frac{\top}{2}} \\ &= -\Sigma_{\mathbf{X}\mathbf{Y}} \end{aligned}$$

On peut donc également associer à $-\Sigma_{\mathbf{X}\mathbf{Y}}$ dans la matrice $\Sigma_Y^{\mathbf{X}}(\omega) - \Sigma_Y^{\mathbf{X}}$ à une matrice de contraction, $-\Omega$, qui ne possède également que des valeurs singulières inférieures à l'unité. Ce qui prouve donc que $\Sigma_Y^{\mathbf{X}}(\omega) - \Sigma_Y^{\mathbf{X}}$ est symétrique définie positive, et donc la conservativité de $\Sigma_Y^{\mathbf{X}}(\omega)$ par rapport à $\Sigma_Y^{\mathbf{X}}$.

On dispose donc du choix de ω pour générer la matrice conservative $\Sigma_Y^{\mathbf{X}}(\omega)$. On a alors :

$$\begin{aligned} & \Sigma_{\mathbf{X}\mathbf{Y}}(\omega) \succeq \Sigma_{\mathbf{X}\mathbf{Y}} \\ \Rightarrow & (\mathbf{I}_X \quad -\mathbf{H}^\top) \cdot \Sigma_Y^{\mathbf{X}^{-1}}(\omega) \cdot \begin{pmatrix} \mathbf{I}_X \\ -\mathbf{H} \end{pmatrix} \preceq (\mathbf{I}_X \quad -\mathbf{H}^\top) \cdot \Sigma_Y^{\mathbf{X}^{-1}} \cdot \begin{pmatrix} \mathbf{I}_X \\ -\mathbf{H} \end{pmatrix} \\ \Rightarrow & \Sigma_X^+(\omega) = (\omega \Sigma_X^{-1} + (1-\omega) \cdot \mathbf{H}^\top \cdot \Sigma_Y^{-1} \cdot \mathbf{H})^{-1} \succeq \Sigma_{\mathbf{X}\text{Kalman}}^+ \end{aligned}$$

où $\Sigma_{\mathbf{X}\text{Kalman}}^+$ est l'hypothétique matrice de fusion sur la variable \mathbf{X} obtenue grâce à $\Sigma_Y^{\mathbf{X}}$ par un filtre de Kalman, qui est l'estimation optimale. Ceci prouve le caractère conservatif de l'estimation donnée par un filtre à Intersection de Covariance.

C.2.3 Choix de ω

Selon la théorie de l'information, la quantité de « méconnaissance », reliée à l'entropie, que l'on a dans la distribution postérieure de \mathbf{X} est donnée par :

$$H_{\mathbf{X}^+} = \frac{1}{2} \log |2\pi e \Sigma_{\mathbf{X}}^+(\omega)| \quad (\text{C.18})$$

L'optimum en terme d'information de cette estimation conservative est donc celle dont le déterminant de la matrice $\Sigma_{\mathbf{X}}^+(\omega)$ est le plus faible, c'est à dire celle pour laquelle le volume d'incertitude correspondant dans l'espace des phases est le plus faible.

Il est à remarquer que cet ω optimum ne varie pas lorsque au lieu de prendre en compte la variable \mathbf{X} , on prendrait la matrice $\mathbf{M} \cdot \mathbf{X}$, dans laquelle \mathbf{M} est constante, carrée de dimension identique à \mathbf{X} , et de rang maximal. C'est à dire que l'on mappe différemment la variable \mathbf{X} , mais en conservant les données de celle-ci. Ceci équivaut à une transformation linéaire de l'espace des phases.

En effet,

$$\mathbf{T} = \mathbf{M} \cdot \mathbf{X} \quad (\text{C.19})$$

$$|\Sigma_{\mathbf{T}}(\omega)| = |\mathbf{M}|^2 \cdot |\Sigma_{\mathbf{X}}| \quad (\text{C.20})$$

$$\propto |\Sigma_{\mathbf{X}}(\omega)| \quad (\text{C.21})$$

et donc l'optimum en ω est le même. Celui-ci est donc invariant quel que soit le paramétrage utilisé.

C.3 Filtre à Intersection de Covariance Partitionnée

Dans la section C.2, l'hypothèse de base consiste à dire que le coefficient de couplage $\Sigma_{\mathbf{X}\mathbf{Y}}$ est inconnu, et qu'il **peut** être maximalelement corrélé. Mais supposons que ce coefficient de couplage puisse être borné d'une manière quelconque. Par exemple, supposons que l'on puisse écrire :

$$\begin{aligned} \Sigma'_{\mathbf{X}} &\preceq \Sigma_{\mathbf{X}} \\ \Sigma'_{\mathbf{Y}} &\preceq \Sigma_{\mathbf{Y}} \\ \begin{pmatrix} \Sigma'_{\mathbf{X}} & \Sigma_{\mathbf{X}\mathbf{Y}} \\ \Sigma_{\mathbf{X}\mathbf{Y}}^\top & \Sigma'_{\mathbf{Y}} \end{pmatrix} &\text{ est symétrique définie positive.} \end{aligned}$$

Cela signifie que le terme $\Sigma_{\mathbf{X}\mathbf{Y}}$ est plus faible que le terme de couplage maximal, car :

$$\begin{aligned} \Sigma_{\mathbf{X}\mathbf{Y}} &= \Sigma'_{\mathbf{X}}^{\frac{1}{2}} \cdot \Omega'' \cdot \Sigma'_{\mathbf{Y}}^{\frac{\top}{2}} \\ \Omega'' &\text{ a toutes ses valeurs singulières } \sigma \leq 1. \end{aligned} \quad (\text{C.22})$$

Comme $\Sigma'_{\mathbf{X}}$ et $\Sigma'_{\mathbf{Y}}$ sont plus faibles que les valeurs block-diagonales $\Sigma_{\mathbf{X}}$ et $\Sigma_{\mathbf{Y}}$, cela signifie que le terme de couplage est plus faible d'autant.

On peut alors écrire :

$$\begin{pmatrix} \Sigma_{\mathbf{X}} & \Sigma_{\mathbf{X}\mathbf{Y}} \\ \Sigma_{\mathbf{X}\mathbf{Y}}^\top & \Sigma_{\mathbf{Y}} \end{pmatrix} = \underbrace{\begin{pmatrix} \Sigma'_{\mathbf{X}} & \Sigma_{\mathbf{X}\mathbf{Y}} \\ \Sigma_{\mathbf{X}\mathbf{Y}}^\top & \Sigma'_{\mathbf{Y}} \end{pmatrix}}_{\Sigma_d \succeq 0} + \underbrace{\begin{pmatrix} \Sigma_{\mathbf{X}} - \Sigma'_{\mathbf{X}} & 0 \\ 0 & \Sigma_{\mathbf{Y}} - \Sigma'_{\mathbf{Y}} \end{pmatrix}}_{\Sigma_i \succeq 0}$$

Dans ce cas, la matrice Σ_i représente une partie totalement vierge de corrélations, et la matrice Σ_d représenter le terme de corrélation totale maximale. On peut alors écrire :

$$\Sigma_Y^X \preceq \Sigma_i + \underbrace{\begin{pmatrix} \frac{1}{\omega} \Sigma'_X & \mathbf{0} \\ \mathbf{0} & \frac{1}{1-\omega} \Sigma'_Y \end{pmatrix}}_{\text{mise à jour SCI}} \preceq \underbrace{\begin{pmatrix} \frac{1}{\omega} \Sigma_X & \mathbf{0} \\ \mathbf{0} & \frac{1}{1-\omega} \Sigma_Y \end{pmatrix}}_{\text{mise à jour CI}}$$

$$\Rightarrow H_{SCI}(\omega) \leq H_{CI}(\omega)$$

La stratégie de fusion pour Σ_d est alors du type de celle qui est utilisée pour l'intersection de covariance. Comme Σ_i possède des termes non-diagonaux nuls, une fusion de type Kalman est appropriée. On obtient alors les équations de mise à jour présentées à la section 4.7.7, qui sont plus optimales que celles de l'Intersection de Covariance de base.

Bibliographie

- Mahmuda Ahmed, Sophia Karagiorgou, Dieter Pfoser, and Carola Wenk. A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica*, 19(3) :601–632, 2015.
- Nabil Ali Ahmed and DV Gokhale. Entropy expressions and their estimators for multivariate distributions. *IEEE Transactions on Information Theory*, 35(3) :688–692, 1989.
- S-I Amari. Information geometry on hierarchy of probability distributions. *IEEE transactions on information theory*, 47(5) :1701–1711, 2001.
- Rosario Aragues, Carlos Sagues, and Youcef Mezouar. Feature-based map merging with dynamic consensus on information increments. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2725–2730. IEEE, 2013.
- Jamal Jokar Arsanjani, Christopher Barron, Mohammed Bakillah, and Marco Helbich. Assessing the quality of OpenStreetMap contributors together with their contributions. In *Proceedings of the AGILE*, pages 14–17, 2013.
- Aamer Ather. A quality analysis of OpenStreetMap data. *ME Thesis, University College London, London, UK*, 22, 2009.
- Dzintars Avots, Edward Lim, Romain Thibaux, and Sebastian Thrun. A probabilistic technique for simultaneous localization and door state estimation with mobile robots in dynamic environments. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 521–526. IEEE, 2002.
- Claude Aynaud, Coralie Bernay-Angeletti, Roland Chapuis, Romuald Aufrere, Christophe Debain, and Nadir Karam. Real-time vehicle localization by using a top-down process. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–6. IEEE, 2014a.
- Claude Aynaud, Coralie Bernay-Angeletti, Roland Chapuis, Romuald Aufrere, and Christophe Debain. Vehicle localization by using a multi-modality top down approach. In *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*, pages 1415–1420. IEEE, 2014b.
- Hernán Badino, D Huber, and Takeo Kanade. Visual topometric localization. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 794–799. IEEE, 2011.
- Hernán Badino, Daniel Huber, and Takeo Kanade. Real-time topometric localization. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1635–1642. IEEE, 2012.

- Alexander Bahr, John J Leonard, and Maurice F Fallon. Cooperative localization for autonomous underwater vehicles. *The International Journal of Robotics Research*, 28(6) :714–728, 2009.
- Miguel Bernal-Marin and Eduardo Bayro-Corrochano. Integration of Hough transform of lines and planes in the framework of conformal geometric algebra for 2D and 3D robot vision. *Pattern Recognition Letters*, 32(16) :2213–2223, 2011.
- Paul J Besl, Neil D McKay, et al. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2) :239–256, 1992.
- David Bétaille and Rafael Toledo-Moreo. Creating enhanced maps for lane-level vehicle navigation. *Intelligent Transportation Systems, IEEE Transactions on*, 11(4) :786–798, 2010.
- Peter Biber, Tom Duckett, et al. Dynamic maps for long-term operation of mobile service robots. In *Robotics : science and systems*, pages 17–24, 2005.
- Philippe Bonnifait, Maged Jabbour, and Véronique Cherfaoui. Autonomous navigation in urban areas using GIS-managed information. *International Journal of Vehicle Autonomous Systems*, 6(1-2) :83–103, 2008.
- Michael Bosse, Paul Newman, John Leonard, and Seth Teller. Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *The International Journal of Robotics Research*, 23(12) :1113–1139, 2004.
- Matthew Brand. Coupled hidden Markov models for modeling interacting processes. Technical Report 405, MIT Media Lab Perceptual Computing, Cambridge, MA, USA, June 1997.
- C Brenner and B Elias. Extracting landmarks for car navigation systems using existing GIS databases and laser scanning. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(3/W8) :131–138, 2003.
- Claus Brenner. Vehicle localization using landmarks obtained by a LIDAR mobile mapping system. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences :[PCV 2010-Photogrammetric Computer Vision And Image Analysis, Pt I] 38 (2010), Nr. Part 3A*, 38(Part 3A) :139–144, 2010.
- César Cadena and José Neira. SLAM in $O(\log n)$ with the Combined Kalman-Information filter. *Robotics and Autonomous Systems*, 58(11) :1207–1219, 2010.
- Vincenzo Caglioti. An entropic criterion for minimum uncertainty sensing in recognition and localization. I. theoretical and conceptual aspects. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(2) :187–196, 2001a.
- Vincenzo Caglioti. An entropic criterion for minimum uncertainty sensing in recognition and localization. II. a case study on directional distance measurements. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(2) :197–214, 2001b.

- Cindy Cappelle, Maan E El Najjar, Denis Pomorski, and François Charpillet. Multi-sensors data fusion using dynamic bayesian network for robotised vehicle geolocalisation. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE, 2008.
- Luis C Carrillo-Arce, Esha D Nerurkar, José L Gordillo, and Stergios I Roumeliotis. Decentralized multi-robot cooperative localization using covariance intersection. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1412–1417. IEEE, 2013.
- ZJ Chong, Baoxing Qin, Tirthankar Bandyopadhyay, Marcelo H Ang, Emilio Frazzoli, and Daniela Rus. Synthetic 2D lidar for precise vehicle localization in 3D urban environment. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1554–1559. IEEE, 2013.
- Hang Chu. Vision-based localization with map information. Master of science, Cornell University, Cornell University, Ithaca, NY, USA, August 2015.
- Cornell University. GIS data & maps : Europe. <https://olinuris.library.cornell.edu/collections/maps/links/gis-data/europe>. accédé le 01/09/2017.
- Stefan Czarnetzki, Sören Kerner, and Michael Kruse. Real-time active vision by entropy minimization applied to localization. In *Robot Soccer World Cup*, pages 266–277. Springer, 2010.
- MWM Gamini Dissanayake, Paul Newman, Steve Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on robotics and automation*, 17(3) :229–241, 2001.
- Ghina El Natour. *Towards 3D reconstruction of outdoor scenes by mmw radar and a vision sensor fusion*. PhD thesis, Université Blaise Pascal-Clermont-Ferrand II, 2016.
- E Fabrizi, G Oriolo, S Panzieri, and G Ulivi. Mobile robot localization via fusion of ultrasonic and inertial sensor data. In *Proceedings of the 8th International Symposium on Robotics with Applications, Maui, USA*, 2000.
- Hongchao Fan, Bisheng Yang, Alexander Zipf, and Adam Rousell. A polygon-based approach for matching OpenStreetMap road networks with regional transit authority data. *International Journal of Geographical Information Science*, 30(4) :748–764, 2016.
- William J Farrell and Chidambar Ganesh. Generalized Chernoff fusion approximation for practical distributed data fusion. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pages 555–562. IEEE, 2009.
- L F Fehlner. Precision aircraft navigation based on LORAN-C and digital altimetry. Technical Memorandum TG1184, John Hopkins University, January 1972. obtained from <http://www.loran-history.info/Loran-C/GetTRDoc.pdf>.
- John W Fenwick, Paul M Newman, and John J Leonard. Cooperative concurrent mapping and localization. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1810–1817. IEEE, 2002.

- Clément Fouque, Philippe Bonnifait, and David Bétaille. Enhancement of global vehicle localization using navigable road maps and dead-reckoning. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1286–1291. IEEE, 2008.
- C Galamhos, Jose Matas, and Josef Kittler. Progressive probabilistic Hough transform for line detection. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 554–560. IEEE, 1999.
- David J Galas, Nikita A Sakhanenko, Alexander Skupin, and Tomasz Ignac. Describing the complexity of systems : multivariable « set complexity » and the information basis of systems biology. *Journal of Computational Biology*, 21(2) :118–140, 2014.
- Thomas Genevois and Teresa Zielińska. A simple and efficient implementation of EKF-based SLAM relying on laser scanner in complex indoor environment. *Journal of Automation Mobile Robotics and Intelligent Systems*, 8, 2014.
- Jean-François Girres and Guillaume Touya. Quality assessment of the french OpenStreet-Map dataset. *Transactions in GIS*, 14(4) :435–459, 2010.
- S Grime and Hugh F Durrant Whyte. Data fusion in decentralized sensor networks. *Control engineering practice*, 2(5) :849–863, 1994.
- Dominique Gruyer, Rachid Belaroussi, and Marc Revilloud. Map-aided localization with lateral perception. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 674–680. IEEE, 2014.
- Jean-Marie Guyader, Wyke Huizinga, Valerio Fortunati, Dirk HJ Poot, Matthijs van Kraenburg, Jifke F Veenland, Margarethus M Paulides, Wiro J Niessen, and Stefan Klein. Total correlation-based groupwise image registration for quantitative MRI. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 186–193, 2016.
- Norbert Haala, Dieter Fritsch, Michael Peter, and A Khosravani. Pedestrian mobile mapping system for indoor environments based on MEMS IMU and range camera. *Archivum Fotogrametrii, Kartografii i Teledetekcji*, 22, 2011.
- Dirk Hahnel, Dirk Schulz, and Wolfram Burgard. Map building with mobile robots in populated environments. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 496–501. IEEE, 2002.
- Dirk Hahnel, Rudolph Triebel, Wolfram Burgard, and Sebastian Thrun. Map building with mobile robots in dynamic environments. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 1557–1563. IEEE, 2003.
- Alberto Hata and Denis Wolf. Road marking detection using LIDAR reflective intensity data and its application to vehicle localization. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 584–589. IEEE, 2014.
- Alberto Y Hata, Fernando S Osorio, and Denis F Wolf. Robust curb detection and vehicle localization in urban environments. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 1257–1262. IEEE, 2014.

- Marco Helbich, Chritoph Amelunxen, Pascal Neis, and Alexander Zipf. Comparative spatial analysis of positional accuracy of OpenStreetMap and proprietary geodata. *Proceedings of GI_Forum*, pages 24–33, 2012.
- Txomin Hermosilla, Luis A Ruiz, Jorge A Recio, and Javier Estornell. Evaluation of automatic building detection approaches combining high resolution images and lidar data. *Remote Sensing*, 3(6) :1188–1210, 2011.
- Guoquan Huang, Nikolas Trawny, Anastasios I Mourikis, and Stergios I Roumeliotis. On the consistency of multi-robot cooperative localization. In *Robotics : Science and Systems*, pages 65–72, 2009.
- Guoquan P Huang, Nikolas Trawny, Anastasios I Mourikis, and Stergios I Roumeliotis. Observability-based consistent EKF estimators for multi-robot cooperative localization. *Autonomous Robots*, 30(1) :99–122, 2011.
- IGN. Données | IGN - espace professionnel. <http://professionnels.ign.fr/donnees>. accédé le 01/09/2017.
- Jun-Hyuck Im, Sung-Hyuck Im, and Gyu-In Jee. Vertical corner feature based precise vehicle localization using 3D LIDAR in urban area. *Sensors*, 16(8) :1268, 2016.
- Shinya Iwashina, Atsushi Yamashita, and Toru Kaneko. 3-D map building in dynamic environments by a mobile robot equipped with two laser range finders. In *Proceedings of the 3rd Asia International Symposium on Mechatronics*, volume 3, pages 1–5, 2008.
- Luc Jaulin. A nonlinear set membership approach for the localization and map building of underwater robots. *IEEE Transactions on Robotics*, 25(1) :88–98, 2009.
- Edwin T Jaynes. Gibbs vs boltzmann entropies. *American Journal of Physics*, 33(5) : 391–398, 1965.
- Cansen Jiang, Danda Pani Paudel, Yohan Fougerolle, David Fofi, and Cédric Démonceaux. Static-map and dynamic object reconstruction in outdoor scenes using 3-D motion segmentation. *IEEE Robotics and Automation Letters*, 1(1) :324–331, 2016.
- Brian J Julian, Michael Angermann, Mac Schwager, and Daniela Rus. Distributed robotic sensor networks : An information-theoretic approach. *The International Journal of Robotics Research*, 31(10) :1134–1154, 2012.
- Simon J Julier and Jeffrey K Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *American Control Conference, 1997. Proceedings of the 1997*, volume 4, pages 2369–2373. IEEE, 1997.
- Simon J Julier and Jeffrey K Uhlmann. Simultaneous localisation and map building using split covariance intersection. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 3, pages 1257–1262. IEEE, 2001.
- Simon J Julier and Jeffrey K Uhlmann. Using covariance intersection for SLAM. *Robotics and Autonomous Systems*, 55(1) :3–20, 2007.

- Mikael Kais, Stéphane Dauvillier, Arnaud De La Fortelle, Ichiro Masaki, and Christian Laugier. Towards outdoor localization using GIS, vision system and stochastic error propagation. In *ICARA 2004-2nd International Conference on Autonomous Robots and Agents.*, pages 198–205, 2004.
- Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D) :35–45, 1960.
- Dominik Kellner, Michael Barjenbruch, Jens Klappstein, Jürgen Dickmann, and Klaus Dietmayer. Instantaneous ego-motion estimation using multiple Doppler RADARs. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1592–1597. IEEE, 2014.
- Carsten Keßler. Collecting a ground truth dataset for OpenStreetMap. *Mooney and Rehrl (2013)*, pages 55–60, 2013.
- Michel Kieffer, Luc Jaulin, Éric Walter, and Dominique Meizel. Robust autonomous robot localization using interval analysis. *Reliable computing*, 6(3) :337–362, 2000.
- Christopher Korpela, Kenneth Chaney, and Pareshkumar Brahmhatt. Applied robotics for installation and base operations for industrial hygiene. In *Technologies for Practical Robot Applications (TePRA), 2015 IEEE International Conference on*, pages 1–6. IEEE, 2015.
- Jean Laneurit, Roland Chapuis, and Frédéric Chausse. Accurate vehicle positioning on a numerical map. *International Journal of Control, Automation, and Systems*, 3(1) : 15–31, 2005.
- Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.
- Jesse Levinson and Sebastian Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4372–4378. IEEE, 2010.
- Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. Map-based precision vehicle localization in urban environments. In *Robotics : Science and Systems*, volume 4, page 1, 2007.
- Hao Li and Fawzi Nashashibi. Cooperative multi-vehicle localization using split covariance intersection filter. *IEEE Intelligent transportation systems magazine*, 5(2) :33–44, 2013.
- M-O Löwner, Andreas Sasse, and Peter Hecker. Needs and potential of 3D city information and sensor fusion technologies for vehicle positioning in urban environments. *Developments in 3D Geo-Information Sciences*, pages 143–156, 2010.
- Wenjie Lu, Emmanuel Seigne, F Sergio A Rodriguez, and Roger Reynaud. Lane marking based vehicle localization using particle filter and multi-kernel estimation. In *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*, pages 601–606. IEEE, 2014.
- David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

- Raj Madhavan and Hugh F Durrant-Whyte. Natural landmark-based autonomous vehicle navigation. *Robotics and Autonomous Systems*, 46(2) :79–95, 2004.
- Mairie de Paris. Open Data Paris. <http://opendata.paris.fr/>. accédé le 01/09/2017.
- András L Majdik, Damiano Verda, Yves Albers-Schoenberg, and Davide Scaramuzza. Micro air vehicle localization and position tracking from textured 3D cadastral models. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 920–927. IEEE, 2014.
- Guido Manfredi, Sandra Devin, Michel Devy, and Daniel Sidobre. Autonomous apartment exploration, modelling and segmentation for service robotics. *IFAC-PapersOnLine*, 49(15) :120–125, 2016.
- Andrew Mastin, Jeremy Kepner, and John Fisher. Automatic registration of LIDAR and optical images of urban scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2639–2646. IEEE, 2009.
- Takafumi Matsumaru, Hiroshi Yamamori, and Takumi Fujita. Dynamic remodeling of environmental map using range data for remote operation of mobile robot. *Journal of Robotics and Mechatronics*, 21(3) :332, 2009.
- Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, et al. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5) :832–841, 2012.
- Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 2125–2133, 2015.
- Peter Mooney, Pdraig Corcoran, and Adam C Winstanley. Towards quality metrics for OpenStreetMap. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 514–517. ACM, 2010.
- Kevin Patrick Murphy. Dynamic bayesian networks : representation, inference and learning, 2002.
- Sergiu Nedevschi, Voichita Popescu, Radu Danescu, Tiberiu Marita, and Florin Oniga. Accurate ego-vehicle global localization at intersections through alignment of visual data with digital map. *Intelligent Transportation Systems, IEEE Transactions on*, 14(2) :673–687, 2013.
- José Neira and Juan D Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on robotics and automation*, 17(6) :890–897, 2001.
- Esha D Nerurkar and Stergios Roumeliotis. Power-SLAM : a linear-complexity, any-time algorithm for SLAM. *The International Journal of Robotics Research*, page 0278364910390539, 2011.

- Esha D Nerurkar, Stergios I Roumeliotis, and Agostino Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 1402–1409. IEEE, 2009.
- Hoang V Nguyen, Emmanuel Müller, Jilles Vreeken, Pavel Efros, and Klemens Böhm. Multivariate maximal correlation analysis. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 775–783, 2014.
- Xiaoji Niu, Tong Yu, Jian Tang, and Le Chang. An online solution of LiDAR scan matching aided inertial navigation system for indoor mobile mapping. *Mobile Information Systems*, 2017(Article ID 4802159), July 2017.
- Lee-Ling Ong, Matthew Ridley, Ben Upcroft, Suresh Kumar, Tim Bailey, Salah Sukkarieh, and Hugh Durrant-Whyte. A comparison of probabilistic representations for decentralised data fusion. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on*, pages 187–192. IEEE, 2005.
- Lee-Ling Ong, Ben Upcroft, Tim Bailey, Matthew Ridley, Salah Sukkarieh, and Hugh Durrant-Whyte. A decentralised particle filtering algorithm for multi-target tracking across multiple flight vehicles. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4539–4544. IEEE, 2006a.
- Lee-ling Ong, Ben Upcroft, Matthew Ridley, Tim Bailey, Salah Sukkarieh, and Hugh Durrant-Whyte. Consistent methods for decentralised data fusion using particle filters. In *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, pages 85–91. IEEE, 2006b.
- Lee-Ling Ong, Tim Bailey, Hugh Durrant-Whyte, and Ben Upcroft. Decentralised particle filtering for multiple target tracking in wireless sensor networks. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE, 2008.
- OpenStreetMap contributors. yevaud.openstreetmap.org. <https://hardware.openstreetmap.org/servers/yevaud.openstreetmap.org/#filesystems>, a. accédé le 02/09/2017.
- OpenStreetMap contributors. Munin : : openstreetmap : : yevaud.openstreetmap. <https://munin.openstreetmap.org/openstreetmap/yevaud.openstreetmap/index.html#disk>, b. accédé le 02/09/2017.
- OpenStreetMap contributors. Planet dump retrieved from <https://planet.osm.org>. <https://www.openstreetmap.org>, 2017.
- M Oussalah and D Wilson. *Robot Collaboration for Simultaneous Map Building and Localization*. INTECH Open Access Publisher, 2007.
- Martin Parsley and Simon J Julier. SLAM with a heterogeneous prior map. In *In SEAS-DTC Conference*. Citeseer, 2009.
- Karl Pearson. On the criterion that a given system of deviations from the probable in the case of correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, 50 :157–175, 1900.

- Sébastien Peyraud, David Bétaille, Stéphane Renault, Miguel Ortiz, Florian Mougel, Dominique Meizel, and François Peyret. About non-line-of-sight satellite detection and exclusion in a 3D map-aided localization algorithm. *Sensors*, 13(1) :829–847, 2013.
- Oliver Pink. Visual map matching and localization using a global feature map. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–7. IEEE, 2008.
- Oliver Pink, Frank Moosmann, and Alexander Bachmann. Visual features for vehicle localization and ego-motion estimation. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 254–260. IEEE, 2009.
- Sam Pullen, Todd Walter, and Per Enge. System overview, recent developments, and future outlook for WAAS and LAAS. In *GPS Symposium, 2002 Proceedings of*, pages 45–56, Tokyo, Japan, Nov 2002. GPS Society/Japan Institute of Navigation.
- Yongliang Qiao, Cindy Cappelle, and Yassine Ruichek. Image based place recognition and LIDAR validation for vehicle localization. In *Mexican International Conference on Artificial Intelligence*, pages 304–315. Springer, 2014.
- Xiaozhi Qu, Bahman Soheilian, and Nicolas Paparoditis. Vehicle localization using mono-camera and geo-referenced traffic signs. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 605–610. IEEE, 2015.
- Nathan Rackliffe, Holly A Yanco, and Jennifer Casper. Using geographic information systems (GIS) for UAV landings and UGV navigation. In *Technologies for Practical Robot Applications (TePRA), 2011 IEEE Conference on*, pages 145–150. IEEE, 2011.
- Robert Racz, Christian Schott, and Samuel Huber. Electronic compass sensor. In *Sensors, 2004. Proceedings of IEEE*, pages 1446–1449. IEEE, 2004.
- Maria-Joao Rendas and Jose Manuel Leitao. Rumor-robust distributed data fusion. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*, pages 230–235. IEEE, 2010.
- Alexis Roche, Grégoire Malandain, Xavier Pennec, and Nicholas Ayache. The correlation ratio as a new similarity measure for multimodal image registration. *Medical Image Computing and Computer-Assisted Intervention–MICCAI'98*, pages 1115–1124, 1998.
- Diego Rodriguez-Losada, Fernando Matia, and Agustin Jimenez. Local maps fusion for real time multirobot indoor simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1308–1313. IEEE, 2004.
- VS Roshni and K Revathy. Using mutual information and cross correlation as metrics for registration of images. *Journal of Theoretical & Applied Information Technology*, 4(6), 2008.
- Franz Rottensteiner, John Trinder, Simon Clode, and Kurt Kubik. Using the Dempster–Shafer method for the fusion of LIDAR data and multi-spectral images for building detection. *Information fusion*, 6(4) :283–300, 2005.

- Stergios I Roumeliotis and George A Bekey. Bayesian estimation and Kalman filtering : A unified framework for mobile robot localization. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 3, pages 2985–2992. IEEE, 2000.
- Eric Royer, Maxime Lhuillier, Michel Dhome, and Thierry Chateau. Localization in urban environments : monocular vision compared to a differential GPS sensor. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 114–121. IEEE, 2005.
- Pietro Salvatori, Alessandro Neri, Cosimo Stallo, Veronica Palma, Andrea Coluccia, and Francesco Rispoli. Augmentation and integrity monitoring network and EGNOS performance comparison for train positioning. In *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pages 186–190. IEEE, 2014.
- Markus Schreiber, Carsten Knöppel, and Uwe Franke. Laneloc : Lane marking based localization using highly accurate maps. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 449–454. IEEE, 2013.
- Turgay Senlet and Ahmed Elgammal. A framework for global vehicle localization using stereo images and satellite and road maps. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2034–2041. IEEE, 2011.
- Claude E Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 :379–423, 1948.
- Claude E Shannon, Warren Weaver, and Arthur W Burks. The mathematical theory of communication. 1951.
- John Shore and Rodney Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on information theory*, 26(1) :26–37, 1980.
- Cherif Smaili, Maan E El Najjar, and François Charpillet. Multi-sensor fusion method using dynamic bayesian network for precise vehicle localization and road matching. In *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, volume 1, pages 146–151. IEEE, 2007.
- Bahman Soheilian, Olivier Tournaire, Nicolas Papanoditis, Bruno Vallet, and Jean-Pierre Papelard. Generation of an integrated 3D city model with visual landmarks for autonomous navigation in dense urban areas. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 304–309. IEEE, 2013.
- Hauke Strasdat, JMM Montiel, and Andrew J Davison. Scale drift-aware large scale monocular SLAM. *Robotics : Science and Systems VI*, 2, 2010.
- Murali Subbarao. Parallel depth recovery by changing camera parameters. In *ICCV*, pages 149–155, 1988.
- Niko Sünderhauf and Peter Protzel. Switchable constraints for robust pose graph SLAM. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1879–1884. IEEE, 2012.

- Niko Sunderhauf, Sven Lange, and Peter Protzel. Using the unscented kalman filter in mono-SLAM with inverse depth parametrization for autonomous airship control. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6. IEEE, 2007.
- Zui Tao, Ph Bonnifait, Vincent Fremont, and Javier Ibanez-Guzman. Lane marking aided vehicle localization. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 1509–1515. IEEE, 2013.
- Sebastian Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2) :111–127, 2003.
- Sebastian Thrun and Michael Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6) :403–429, 2006.
- Sebastian Thrun et al. Robotic mapping : A survey. *Exploring artificial intelligence in the new millennium*, 1 :1–35, 2002.
- Nicholas Timme, Wesley Alford, Benjamin Flecker, and John M Beggs. Multivariate information measures : an experimentalist’s perspective. *arXiv preprint arXiv :1111.6857*, 2011.
- David Törnqvist, Thomas B Schön, Rickard Karlsson, and Fredrik Gustafsson. Particle filter SLAM with high dimensional vehicle model. *Journal of Intelligent & Robotic Systems*, 55(4) :249–266, 2009.
- University of Pennsylvania. International GIS data : Europe. <http://guides.library.upenn.edu/c.php?g=475518&p=3254771>. accédé le 01/09/2017.
- Paul Viola and William M Wells. Alignment by maximization of mutual information. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 16–23. IEEE, 1995.
- Andrew Viquerat, Lachlan Blackhall, Alistair Reid, Salah Sukkariéh, and Graham Brooker. Reactive collision avoidance for unmanned aerial vehicles using doppler RADAR. In *Field and Service Robotics*, pages 245–254. Springer, 2008.
- Damien Vivet, Paul Checchin, Roland Chapuis, Patrice Faure, Raphaël Rouveure, and Marie-Odile Monod. A mobile ground-based RADAR sensor for detection and tracking of moving objects. *EURASIP Journal on Advances in Signal Processing*, 2012(1) :45, 2012.
- Erik Ward and John Folkesson. Vehicle localization with low cost radar sensors. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 864–870. IEEE, 2016.
- Lewis Warren. A new interpretation of the Shannon entropy measure. Technical report, Defence Science and Technology Group Edinburgh (Australia), 2015.
- Jochen Welle, Dirk Schulz, Thomas Bachran, and Armin B Cremers. Optimization techniques for LASER-based 3D particle filter SLAM. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3525–3530. IEEE, 2010.

- David Wong, Daisuke Deguchi, Ichiro Ide, and Hiroshi Murase. Vision-based vehicle localization using a visual street map with embedded SURF scale. In *ECCV Workshops (1)*, pages 167–179, 2014.
- Feihu Zhang, Christian Buckl, and Alois Knoll. Multiple vehicle cooperative localization with spatial registration based on a probability hypothesis density filter. *Sensors*, 14 (1) :995–1009, 2014.
- Sheng Zhao and Jay A Farrell. 2D LIDAR aided INS for vehicle positioning in urban environments. In *Control Applications (CCA), 2013 IEEE International Conference on*, pages 376–381. IEEE, 2013.